Computers, Materials & Continua CMC, vol.65, no.1, pp.579-596, 2020

Automatic Terrain Debris Recognition Network Based on 3D Remote Sensing Data

Xu Han[1, #], Huijun Yang[1, 4, *], Qiufeng Shen[1, #], Jiangtao Yang[2], Huihui Liang[1], Cancan Bao[1] and Shuang Cang[3]

Abstract: Although predecessors have made great contributions to the semantic segmentation of 3D indoor scenes, there still exist some challenges in the debris recognition of terrain data. Compared with hundreds of thousands of indoor point clouds, the amount of terrain point cloud is up to millions. Apart from that, terrain point cloud data obtained from remote sensing is measured in meters, but the indoor scene is measured in centimeters. In this case, the terrain debris obtained from remote sensing mapping only have dozens of points, which means that sufficient training information cannot be obtained only through the convolution of points. In this paper, we build multi-attribute descriptors containing geometric information and color information to better describe the information in low-precision terrain debris. Therefore, our process is aimed at the multi-attribute descriptors of each point rather than the point. On this basis, an unsupervised classification algorithm is proposed to divide the point cloud into several terrain areas, and regard each area as a graph vertex named super point to form the graph structure, thus effectively reducing the number of the terrain point cloud from millions to hundreds. Then we proposed a graph convolution network by employing PointNet for graph embedding and recurrent gated graph convolutional network for classification. Our experiments show that the terrain point cloud can reduce the amount of data from millions to hundreds through the super point graph based on multi-attribute descriptor and our accuracy reached 91.74% and the IoU reached 94.08%, both of which were significantly better than the current methods such as SEGCloud (Acc: 88.63%, IoU: 89.29%) and PointCNN (Acc: 86.35, IoU: 87.26).

Keywords: Semantic segmentation, low-precision point cloud, large-scale terrain, debris recognition.

[1] College of Information Engineering, Northwest A&F University, Yangling, 712100, China.

[2] College of Water Resources and Architectural Engineering, Northwest A&F University, Yangling, 712100, China.

[3] School of Science and Technology, Nottingham Trent University, Nottingham, NG118NS, UK.

[4] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, 712100, China.

[*] Corresponding Author: Huijun Yang. Email: yhj740225@nwafu.edu.cn.

[#] These authors contribute equally to the article.

Received: 01 May 2020; Accepted: 25 May 2020.

1 Introduction

With the rapid development of the engineering construction industry, the demands of the topographic surveys and geological exploration are constantly increasing. Therefore, how to carry out efficient and accurate surveys of terrain and geomorphology has become a common focus in the civil and military fields. On the other hand, low-price and high-automation in the process of terrain acquisition make drone scanning become an important way to get 3D terrain data in recent years. However, the Digital Surface Model (DSM) obtained by remote sensing consists of surface buildings, bridges, and trees other than the ground. In 3D terrain modeling, terrain debris not only affects the accuracy of the survey, but also increases the additional redundant data processing in the terrain survey. Therefore, how to remove debris from the terrain point cloud becomes an inevitable challenge in terrain surveying. Traditional methods of removing terrain debris include point cloud layering and segmentation [Su, Bethel and Hu (2016)], point cloud uniformity based on vegetation point removal [Yang, Yang, Li et al. (2015)] and filtering processing [Yan, Liu, Tan et al. (2017)]. Although these methods can remove debris from the terrain point cloud, however, during the removal process, some useful data in the terrain will be eliminated, which will lead to the loss of terrain feature information. At the same time, these methods will also cause point cloud data sparsification and even hollowing out in lush vegetation, and the process of removing debris needs a lot of manual intervention.

In recent years, point cloud semantic segmentation has made a great progress in single target recognition and indoor or outdoor small scene segmentation such as the semantic segmentation method of indoor scenes based on RGBD descriptors [Deng, Todorovic and Latecki (2015); Yang, He, Jiang et al. (2013)] and point cloud 3D object recognition and model segmentation methods [Xie, Wang, Zhang et al. (2017)]. Apart from that, semantic segmentation also exists in the field of object tracking [Zhang, Jin, Sun et al. (2018)]. However, these semantic segmentation methods cannot cope with large-scale laser point clouds for broad outdoor terrain. Apart from that, some methods specifically for outdoor scenes are used in recent years such as three-dimensional outdoor scene semantic segmentation based on prior knowledge of higher-order categories [Tang, Zhou, Yu et al. (2016); Li, Li, Ren et al. (2020)] and SPG based on local reclassification strategy [Ladrieu and Simonovsky (2018)]. However, these methods directly perform feature extraction and semantic recognition on the point cloud. The extracted features cannot represent the original data well and are not comprehensive enough. In conclusion, the current point cloud semantic segmentation methods in the large-scale and low-precision outdoor terrain point cloud segmentation have faced two problems: one is the easy loss of outdoor space information and the blur of the generated object boundary; the other is the low recognition accuracy and semantic details missing.

Aiming at the problems of large number of point clouds and sparse distribution of debris which exist in segmentation methods in terrain point cloud debris removal, that are caused by the massive amount of point cloud of terrain remote sensing and the limited number of point cloud in sparse terrain debris, a multi-attribute descriptors based recurrent gated graph convolution network (RGGCNet) is proposed for remote sensing terrain point clouds. Inspired by Xiang et al. [Xiang, Shen, Qin et al. (2019)], RGGCNet

can reduce the computational complexity and adapt to large-scale and low-precision terrain point cloud by using graph structure on the premise of a better description of features. The proposed RGGCNet has important significance in mineral census and exploration, road construction, conservation of water and soil, topographic survey, and national defense construction. The main contributions of our work are:

• We did not need convolution to obtain the feature information contained in the point cloud, but use the multi-attribute descriptor to describe the terrain information contained in low-precision point by reconstructing the geometric information and color information of terrain points.

• The graph structure is taken to reduce the amount of terrain point cloud data to be processed by deep learning without losing the main fine details.

• We used PointNet for graph embedding and recurrent gated graph convolution for classification. For the latter, we used a new input gate form of Long Short Term Memory (LSTM).

• We produced a new public Low precision mountain terrain data set (LPMT) to obtain terrain remote sensing in real scenes. In particular, we improve mean per-class intersection over union (mIoU) by 6.82 points compared with PoinCNN and 4.79 points compared with SEGCloud.

2 Related work

2.1 Feature selection

Feature extraction is a critical step in terrain processing such as topographic survey, terrain segmentation, and terrain reconstruction. The traditional terrain processing is mainly based on the ridgeline, ditch, trench edge line, contour line and tableland line [Koka, Anada, Goto et al. (2017); Ma, Li, Yang et al. (2015)] as well as 2D vector graphics [Peng, Lin, Zhang et al. (2019); Peng, Long and Lin (2019)]. With the lucubrating and extensive application of terrain, pixel-level based low-layer visualization features such as color information, texture features, and shape features of terrain point clouds have been gradually proposed [Yang (2011)]. Compared with terrain feature lines, these features have the advantages of invariance, less computation, and visualized expression, which has made great progress in the processing of image-based terrain features. Color histogram construction based on HSV, RGB and other color spaces has good discrimination ability for sky, grassland, land, and other categories in the scene [Shaik, Ganesan, Kalist et al. (2015)], but give poor segmentation effect for similar colors or clutters. In recent years, local image features such as edges, corners, lines, curves, and areas with special attributes have been widely used in multiple fields [Gharbi, Chen, Barron et al. (2017); Daniilidis, Deville, Durandcartagena et al. (2017)]. These image features are abundant in numbers, but low correlation between each other. At the same time [Gao and Yuan (2019)] found that texture information such as Gabor, LBP, and gradient co-occurrence matrix can be used to distinguish regions with different textures, which is effective in discrimination of thickness and density. However, they are helpless in distinguishing the fine difference between the texture information. Therefore, the image-based terrain feature extraction methods have the limitation on expressing terrain feature information and fail to make good use of the terrain point cloud context information.

The graphics-based terrain feature method has appeared in recent years. Topographic features, ridge points, and valley points [Kim (2013)] were used to repair point cloud holes in terrain areas. However, ridge points and valley points are extracted by GIS slope analysis and reclassification methods, which make it still not universal for the large-scale terrain point clouds. A terrain feature information based airborne LiDAR point cloud thinning algorithm [Qian, Zhang and Wang (2017)] mainly uses six types of terrain feature as local point density, elevation standard deviation, slope, gaussian curvature, average curvature, roughness degree, which was proved good thinning effect in high-precision terrain point clouds, with not being tested in low-precision point clouds. Octree filtering based ground and feature point extraction [Wang, Liu, Guo et al. (2017)] is not suitable for the description of complex terrain scenes for less enough semantic expressions. Feature line and feature Surface-based point cloud registration algorithms [Ponerleau, Colas and Siegwart (2015)], such as ground, horizontal roof edge elevation, and control point elevation, have a better effect in large-curvature terrain. In summary, due to limitations in processing range, the traditional terrain feature extraction methods cannot be used to extract point cloud features of low-precision and complex scenes.

Therefore, it's necessary for terrain debris recognition to obtain enough information in limited debris point cloud of terrain remote sensing.

2.2 3D semantic segmentation

Recent 3D terrain processing methods mainly focused on terrain modeling and visualization. For example, based on the vegetation distribution database, Liu et al. [Liu, Li and Jiang (2015)] proposed a method for automatic identification of vegetation-based remote sensing information, which realized the extraction and classification management of vegetation data in the sample area. Besides, Li [Li (2013)] proposed a GPU based three-dimensional seabed terrain visualization, where real terrain data is taken for DEM modeling by taking advantage of CG High-Level Shading Language in illumination, texture rendering, and environment mapping. These methods have greatly enhanced the expressive ability of graphics in terrain, can more effectively represent terrain features, and expand the application of graphics in various scenes. But research on terrain scene understanding and semantic processing has still been in the fledging period.

As an important research branch in the information science, scene understanding has important significance in the fields of data learning and mining, biometric cognition, scene recognition, and statistical modeling. Semantic segmentation has proven to be an important and effective method for scene understanding in graphics [Zhang, Liang, Yang et al. (2018)] as well as video [Song, Yang, Xie et al. (2017)]. However, the traditional 3D point cloud semantic segmentation method will cause a large amount of 3D information loss, because 3D point cloud needs to be projected onto a 2D image before inputting a convolutional neural network such as Snapnet [Bouch, Guerry, Saux et al. (2018)] and DeePr3SS [Lawin, Danelljan, Tosteberg et al. (2017)]. With the development of artificial intelligence technology, in order to retain the structural information of the 3D point cloud, the team of Yang et al. [Yang, Maturana and Scherer (2016)] respectively proposed a supervised 3DCNN based Voxel grid network. These networks can effectively retain the structural information in the point cloud, but not applicable for the

high-computational and information loss in the color and intensity of the point cloud. PointNet [Qi, Su, Mo et al. (2017)], SEGCLoud [Tchapmi, Choy, Armeni et al. (2017)] and PointCNN [Li, Bu, Sun et al. (2018)] network make full use of the multimodal information of the point cloud and reduce the complexity, but they cannot solve the problems of lack of clear structure and uneven training samples during the large-scale point clouds training.

How to introduce efficient scene recognition and classification network based on terrain simplification into large-scale terrain remote sensing recognition is very necessary for engineering construction.

3 Method

A multi-attribute descriptors based recurrent gated graph convolution network (RGGCNet) is proposed to recognize the debris in terrain remote sensing. At first, a multi-attribute descriptor is designed by restructuring the geometric and color features of the terrain point cloud to better describe the information in the limited debris point cloud. Then, by employing multi-attribute descriptors in unsupervised classification, we aggregate point clouds into several terrain areas and regard each terrain as super point. Next, by defining an oriented attribute graph, we combine the super points into a graph structure for reducing the number of terrain point clouds. Then, we use PointNet to embed the data of the edge of the super graph structure. At last, a LSTM based recurrent graph convolution is proposed to classify graph's nodes.

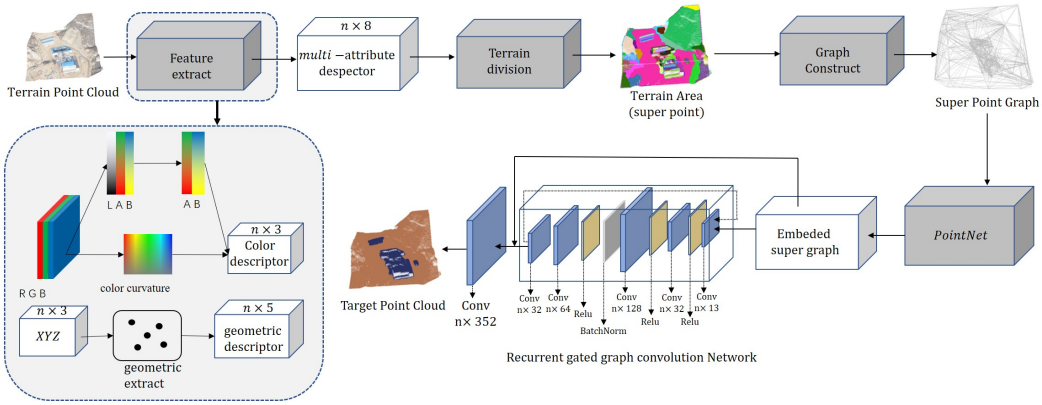


Figure 1: Struct of RGGCNet

3.1 Multiple attribute descriptor

Since each point in the low-precision point cloud contains less available information, we extend the available extra dimension to describe the local information at each point in the input terrain point clouds D. For each point p_i, we construct its multi-attribute descriptor ρ_i in Eq. (1):

$$\rho_i = (GE_i(c_i, l_i, s_i, v_i, e_i), CR_i(a_i, b_i, r_i)) \quad (1)$$

where GE_i is geometric descriptor, consisting of five defined geometric dimensions. And CR_i is color descriptor, consisting of three color dimensions.

First, we design a five dimensions geometric descriptor GE_i in Eq. (2), where five characteristics-curvature c_i Eq. (2-1), linear l_i Eq. (2-2), planar s_i Eq. (2-3), spatial v_i Eq. (2-4) and verticality e_i Eq. (2-5) are employed to describe the spatial terrain.

$$c_i = \frac{\delta_3}{\delta_1+\delta_2+\delta_3} \tag{2-1}$$

$$l_i = \frac{\delta_1-\delta_2}{\delta_1} \tag{2-2}$$

$$s_i = \frac{\delta_2-\delta_3}{\delta_1} \tag{2-3}$$

$$v_i = \frac{\delta_3}{\delta_1} \tag{2-4}$$

$$e_i = \frac{\tau_3^2}{\sqrt{\tau_1^2+\tau_2^2+\tau_3^2}} \tag{2-5}$$

where $\delta_1,\ \delta_2,\ \delta_3(\delta_1 > \delta_2 > \delta_3)$ are three eigenvalues of the covariance matrix of positions of the point cloud neighborhoods. τ_1, τ_2, τ_3 are the three eigenvectors associated with $\delta_1,\ \delta_2,\ \delta_3$ respectively.

Further, to accurately describe the difference of color gradation in each point, two color components a_i Eq. (3-3) and b_i Eq. (3-4) ranged from 0 to 1 in the LAB color coordinate system are employed in Eq. (3-1) and Eq. (3-2) to deal with the terrain point cloud for its wider color gamut.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3-1}$$

$$\alpha(t) = \begin{cases} t^{1/3} & t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & otherwise \end{cases} \tag{3-2}$$

$$a_i = 5[f(X/X_n) - f(Y/X_n)] \tag{3-3}$$

$$b_i = 2[f(Y/Y_n) - f(Z/Z_n)] \tag{3-4}$$

Apart from that, we also define color curvature r_i for better describing the color differences between each point in Eq. (3-5):

$$r_i = \frac{\theta_3}{\theta_1+\theta_2+\theta_3} \tag{3-5}$$

where $\theta_1,\ \theta_2,\ \theta_3(\theta_1 > \theta_2 > \theta_3)$ are three eigenvalues of the covariance matrix of RGB color of the point cloud neighborhoods, with RGB value ranging in [0,1].

3.2 Terrain division

By taking advantage of minimum global energy proposed by Landrieu et al. [Landrieu and Simonovsky (2018)], the generalized minimization division problem of the terrain is converted to a minimized optimization problem. However, due to non-convex and non-continuous of this minimization function, it cannot quickly solve the problem of feature partitioning in large-scale point cloud scenarios such as terrain.

In this section, an unsupervised classification is adapted to break down the terrain points into several minimum divisions by introducing a cut-pursuit algorithm [Raguet and Landrieu (2017)] to the problem of global energy. Thus, we improved the global energy algorithm by defining the minimum division as the solutions of the normally connected components in the following equation:

$$arg \min_{g \in R^{\rho_i}} \sum_{i \in C} ||g_i - \rho_i|| + \mu \sum_{(i,j) \in E_{nn}} w_{i,j} \cdot [g_i - g_j \neq 0] \quad (4)$$

where $[\cdot]$ is the Iverson bracket, if the condition is satisfied in the Iverson bracket, it means 1, otherwise, it means 0. $w \in R_+^{|E|}$ express a decreasing linear relation with the edge length. μ is a factor, the regularization strength which determines the coarseness of the resulting division.

As a result, the solution of Eq. (4), constantly connected components $P = \{P_1, P_2, P_3, \cdots, P_k\}$, define our geometrically simple elements named super point.

3.3 Graph construction

Based on the above terrain super point P, the super point graph is defined as an oriented attribute graph $G = (P, E, \gamma)$ with three components P-the super point, E-the super edge, and γ-the edge feature of E.

Super edge E is defined based on the Voronoi adjacency graph $G_{vor} = (P, E_{vor})$ in Eq. (5):

$$E = \{(M, N) \in P^2 | \exists (i, j) \in E_{vor} \cap (M \times N)\} \quad (5)$$

If two super points M and N can be connected by at least one edge, then these two points will constitute an adjacency relationship E in Eq. (5).

The edge feature offset-γ is calculated from two super points M and N which is joined by edge E. The seven components of edge features γ describe the different super edge characteristics in Tab. 1:

The spatial features φ_{m} in Tab. 1 is defined in Eq. (6):

$$\varphi(M, N) = \{(L_i - L_j) | (i, j) \in E_{vor} \cap (M \times N)\} \quad (6)$$

where L_i means the 3D position of the super point P_i.

Table 1: Hyperedge features between two super points M and N. Where we define $|M|$ as the number of points comprised in a super point M, shape features $length(M) = \lambda_1$, surface $(M) = \lambda_1 \cdot \lambda_2$, $volume(M) = \lambda_1 \cdot \lambda_2 \cdot \lambda_3$, $\lambda_1, \lambda_2, \lambda_3$ is the eigenvalues of the covariance of the points positions comprised in each super point, sorted by decreasing value

Edge feature (γ)	Size	Description
Mean offset	3	$\mathrm{mean}_{\mathrm{m} \in \varphi(M,\mathrm{N})} \varphi_{\mathrm{m}}$
Offset deviation	3	$\mathrm{Std}_{\mathrm{m} \in \varphi(M,\mathrm{N})} \varphi_{\mathrm{m}}$
Centroid ratio	3	$\mathrm{mean}_{\mathrm{i} \in \mathrm{M}} O_{\mathrm{i}} - \mathrm{mean}_{\mathrm{j} \in \mathrm{N}} O_{\mathrm{j}}$
Length ratio	1	log length(M)/length(N)
Surface ratio	1	log surface(M)/surface(N)
Volume ratio	1	log volume(M)/volume(N)
Point count ratio	1	log |M|/|N|

3.4 PointNet embedding

In order to reduce the number of low-precision terrain point clouds, the original point clouds are abstracted into a super point graph. Different from point cloud-based data, multiple feature fusion descriptors based super-point data will cause the general feature generation networks to fail in extracting features directly. Accordingly, by introducing the PointNet, we embed each

input super point P_i into a vector of fixed dimension size d_z to form an embedded super point z_i as in Eq. (7). Therefore, the processing object in general feature extraction network is replaced with a standard vector for the better feature extraction of super point.

$$z_i = I(U_i, d_z) \tag{7}$$

where, I is the embedding operation, d_z is the vector dimension of the super point embedding. The embedding operation I is designed to facilitate the feature extraction of subsequent layers and improve the operation efficiency of network, based on the aligning of input super point data by spatial transformation network T-net [Qi, Su, Mo et al. (2017)], a multi-layer perceptron MLP is adopted to achieve feature enhancement, and maximum pooling is employed to aggregate super point features to achieve super point embedding in the PointNet network. At last, an embedded super point z_i is formed for charactering the super point.

In order to make PointNet better learn the spatial distribution of different scales, each super point P_i is scaled into a unit sphere U_i, each point p_j is represented by their normalized position $normal(p_j)$ and multi-attribute descriptor ρ_j in Eq. (8), before embedding.

$$U_i = \{normal(p_j), \rho_j | \, p_j \in P_i\} \tag{8}$$

3.5 Recurrent gated graph convolution

3.5.1 Graph convolution neural network

Aiming at the problem that the general network needs too many parameters and consumes a large amount of memory in processing low-precision, large-scale data sets, edge conditional convolutional ECC [Li, Bu, Sun et al. (2018)] is introduced to solve these challenges. The convolution weights are dynamically generated by taking advantages of ability of multi-layer perceptron in continuously processing super edge feature γ_{ji} in Eq. (9), which further reduces the number of parameters and saves memory space.

$$m_i^{(t)} = mean_{j|(j,i)\in\varepsilon} \Theta(\gamma_{ji}; W_e) \cdot h_j^{(t)} \tag{9}$$

where, Θ is a multilayer perceptron, W_e is an edge weight vector generated by regression on a specific edge vector; m_i is a cyclic convolution weight obtained by performing vector multiplication on each element. By combing ECC, the graph convolution neural network generates the convolution weights m_i through the multi-layer convolution of Eq. (10), according to the embedded z_i of super point P_i and the connection between super point with its neighbors in super point graph, Feature extraction is performed on the super-point data to obtain the feature vector $f_{(i,j)}$ of the super point data in Eq. (10).

$$f_{(i,j)} = Relu(m_i \cdot [z_i] + b_i) \tag{10}$$

where, $Relu$ is the activation function layer.

3.5.2 Gated recurrent

By taking advantage of LSTM forgetting layer, the LSTM gated neural unit is improved to solve the problem of gradients disappearance and gradients explosion, which control the retention or forgetting of input point cloud feature, based on the feature vector $f_{(i,j)}$ and the input message x_t At each LSTM iteration, the hidden status h_t and the incoming message x_t are obtained as inputs, and used to calculate its new hidden status h_{t+1}. At

last, the final feature graph is obtained after multiple iterations of the LSTM, shown in Fig. 2. A specific implementation process is designed as following:

First step, the feature vector $f_{(i,j)}$ of each super point P_i is stored as a hidden status in the LSTM, and then the initial hidden status h_0 is initialized by the super point descriptor z_i embedded by PointNet network. By taking the initial cell status C_0, as input information, the judgment information x_0 is generated from the initial hidden status h_0, which generated by the data variable filling process.

In second step, the information that the cell status needs to be discarded is determined by the forgetting layer of LSTM. According to the information of h_{t-1} and x_t, a vector between 0 and 1 in Eq. (11) is calculated to control the retention or forgetting of the information in cell status C_{t-1} in the fourth step.

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + k_f\right) \tag{11}$$

where, h_{t-1} is the hidden status at the previous moment, and x_t is the judgment information at the current moment.

In the third step, the information which is added to the cell status is determined by the input gate through LSTM. Taking the hidden status h_{t-1} and the judgment information x_t as input, a cell status update control factor is generated by the input gated neural network unit as in Eq. (12); a candidate cell information is further optimized by taking advantage of the candidate information layer of the cell, as in Eq. (13).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_i] + k_i) \tag{12}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + k_c) \tag{13}$$

where, i_t is a cell update control factor, $\tilde{C}_t$ is candidate cell information, and tanh is activation function layer.

In the fourth step, the old cell information is updated to the new cell information through the output gate shown in Eq. (14).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{14}$$

where, f_t is the forgetting layer, determines which part of the cell information C_{t-1} should be forgotten. i_t is the cell update factors, that determines the candidate information for the cell information C_{t-1} to be updated, and obtains the new cell information C_t .

Finally, after the cell status updated, the judgment condition o_t is calculated through the fully connected layer by inputting h_{t-1} and x_t in Eq. (15), to determine status characteristics of the cell should be output. Accordingly, the current iterative output h_t is obtained by multiplying the judgment condition o_t with the new cell status C_t which goes through the activation function layer, shown in Eq. (16).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + k_o) \tag{15}$$

$$h_t = o_t \cdot tanh(C_t) \tag{16}$$

where, h_t is the output of the current iteration, that is, the input status of the next iteration process.

4 Experiments

In this paper, we evaluated the proposed RGGCnet model on Low precision mountain terrain data set (LPMT) in three parts. First, we show LPMT dataset collection and production (Section 4.1); Second, the data set attribute preprocessing and super point graph construction is shown on LPMT (Section 4.2); Third, we demonstrate the training, testing and evaluation of the RGGCNet on LPMT (Section 4.3). In Section 4.3, a comparison between our model and the state of art is evaluated on three metrics: Average per-class accuracy (Acc) [Xu, Schwing and Urtasun (2014)], Overall Accuracy (OA) [Masocha, Dube and Mpofu (2018)] and Intersection over Union (IoU) [Taran, Gordienko and Kochura (2018)]. At last, ablation studies are employed to verify the critical role of multi-attribute descriptors and LSTM in terrain data processing (Section 4.4).

4.1 Collection and production of LPMT dataset

The source data of LPMT comes from Jiulong mountain of Gansu-China, which is scanned by low-precision image-type unmanned aerial vehicles (UAVS) with an area of 8.75 square kilometers in interval of 0.5 meter. The resulting files are JPG format and Digital Surface Model (DSM) files. In light of the demands of sematic segmentation network for input dataset, we convert terrain files into point cloud in PCD formats, where the LPMT composed of more than 20 million points, including hills, basins, ridges and other terrain, as well as houses, roads and other sundries. Finally, the large-scale point cloud is cut into fourteen parts as training set (Tab. 2) with hundreds of thousands of points and seven files as test set (Tab. 3) with points ranging from four million to six million. Where, Name means different subset of terrain point cloud scenes; Size shows the memory space occupied by each subset; Points represents the number of points in each subset of point cloud scene; debris number means the number of debris in tags of this scene; Terrain's number means the number of terrain in tags of this scene.

Table 2: Training dataset

Name	Size	Points	Debris number	Terrain's number
Dataset1	14.6 M	169695	8	5
Dataset2	14.4 M	170497	4	5
Dataset3	16.3 M	178531	2	6
Dataset4	5.77 M	66732	2	5
Dataset5	42.0 M	205185	6	8
Dataset6	19.8 M	256162	5	6
Dataset7	47.3 M	589270	6	9
Dataset8	28.9 M	312000	4	5
Dataset9	4.78 M	60480	1	4
Dataset10	13.0 M	152641	4	6
Dataset11	26.4 M	310734	9	6
Dataset12	39.8 M	420000	9	9
Dataset13	41.4 M	415338	4	15
Dataset14	24.4 M	249804	2	7

Table 3: Test dataset

Name	Size	Points	Debris number	Terrain's number
test_1	621 MB	7387808	21	19
test_2	339 MB	3811151	6	7
test_3	274 MB	4349427	26	11
test_4	313 MB	4834444	15	7
test_5	258 MB	3711438	21	8
test_6	223 MB	3183450	10	5
test_7	172 MB	2473143	13	3

4.2 Construction of terrain super point graph

4.2.1 Label mapping

Before processing the input point cloud, the extra dimension is precalculated to determine the corresponding relationship between the label point and the point cloud scene, named the label mapping. For each labeled point in the scene, the label mapping of this point is set as 1, and label mapping of the part dataset is shown in Tab. 4. Where Point serial number 3556 means the 3556th point in dataset1, where the debris is 1, so the points belong to the debris.

Table 4: Label mapping (dataset1 in LPMT)

Point serial number	debris	Terrain
3556	1	0
3557	1	0
3558	1	0
3559	1	0
3660	1	0
3661	1	0
3662	0	1
3663	0	1
3664	0	1
3665	0	1

4.2.2 Multi-attribute descriptors generation

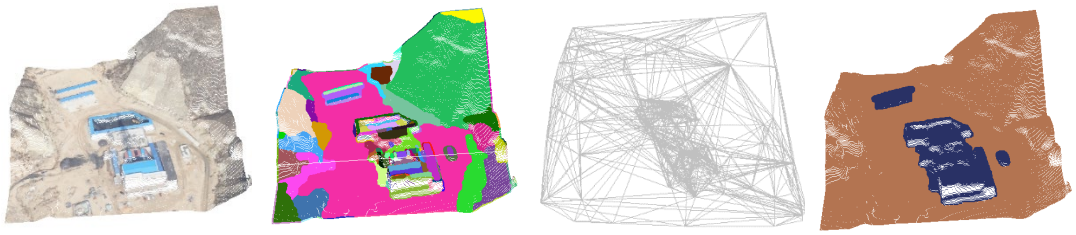
For each point in the LPMT, we first calculate and extract its multi-attribute descriptor according to the reconstruction algorithm proposed in Section 3.1, which is used for accurate terrain division in the next step. Tab. 5 gives part of the calculation results of the multi-attribute descriptors of dataset1 in LMPT. By extracting multi-attribute descriptors, we can mine a lot of information hidden in the coefficient point cloud to better describe the terrain point cloud.

Table 5: Multi-attribute descriptors (dataset1 in LMPT)

Geometric descriptor					Color descriptor		
Curvature	**Linear**	**Planar**	**Spatial**	**Verticality**	**a**	**b**	**Color curvature**
0.006669	0.303082	0.685526	0.011393	0.021503	0.109440	0.027812	0.065637
0.007448	0.282405	0.704706	0.012889	0.020396	0.115613	0.029463	0.063258
0.009399	0.303651	0.680254	0.016096	0.014376	0.113472	0.028886	0.051297
0.010012	0.369031	0.614474	0.016495	0.014483	0.105312	0.026715	0.059000
0.011141	0.407960	0.574103	0.017937	0.022980	0.106195	0.026949	0.029233
0.010321	0.443247	0.540518	0.016235	0.032737	0.108486	0.027558	0.044755
0.009075	0.443174	0.542568	0.014258	0.041665	0.105312	0.026715	0.022629
0.008897	0.443102	0.0676313	0.013975	0.050702	0.109923	0.027941	0.017384
0.009111	0.442910	0.0601516	0.014317	0.057450	0.108486	0.027558	0.054411

4.2.3 Terrain division and super graph composition

Based on the multi-attribute descriptors in Section 4.2.2, we divide the terrain point cloud into serval simple shapes in geometric sense according to the method in Section 3.2. In this way, the terrain point cloud is divided into several simple geometric divisions which named super point (Fig. 2(b)). Then, by simulating method of Section 3.3 on each adjacent super point, the graph structure is formed by connecting the super point with rich attributes edges (Fig. 2(c)). The comparison of the number of points before and after super point composition is shown in the Tab. 6. By dividing the point cloud into super points, we reduce the amount of data in the subsequent semantic segmentation process, thus simplifying the calculation process.



(a) (b) (c) (d)

Figure 2: Terrain division and composition (dataset1). (a) Description of raw point cloud; (b) Description of aggregate division; (c) Description of super point graph. (d) prediction of terrain point

Table 6: The number of super points in Test dataset

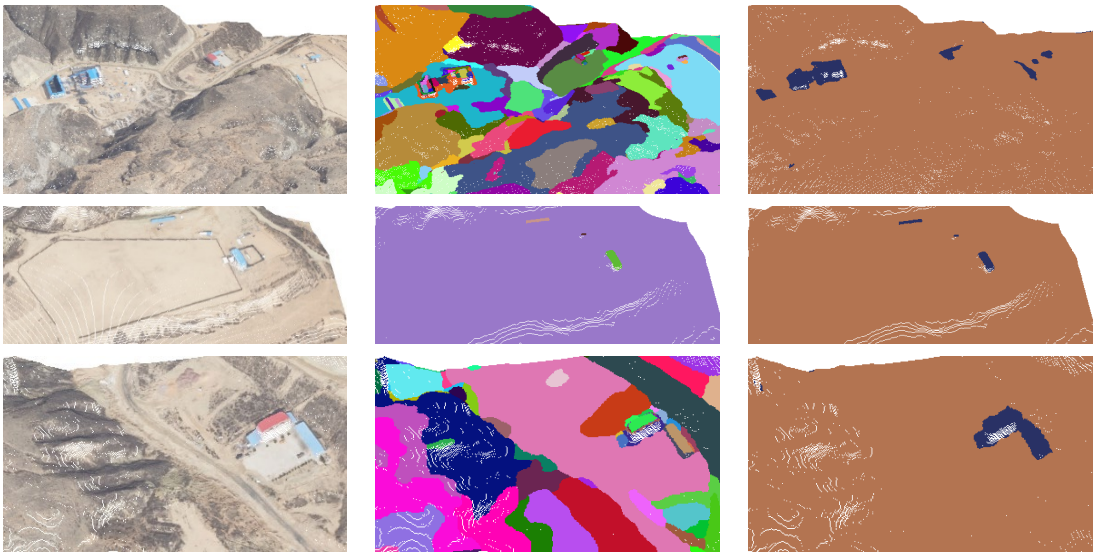
Name	Points	Super Points
test_1	7387808	543
test_2	3811151	294
test_3	4349427	392
test_4	4834444	414
test_5	3711438	307
test_6	3183450	245
test_7	2473143	213

4.3 Training and test

In this section, RGGCNet is trained and tested on LPMT, and evaluated by comparing it with SEGCloud, SPG, PointNet, PointCNN and 3D-BoNet [Yang, Wang, Clark et al. (2019)] in Acc, OA and IoU (Tab. 6). In the process of training and testing, we adopted the micro-averaging method mentioned by Engelmann et al. [Engelmann, Kontogianni and Hermans (2017); Qi, Su, Mo et al. (2017)] to perform 6-fold cross validation on the LMPT. In our 6-fold cross validation, the original data were divided into 6 groups, where each subset data was used as a verification set and the remaining 5 subsets as the training set. Therefore, 6 models were obtained, and the average classification accuracy of the final verification dataset of the 6 models is taken as the performance index of the network structure. In this sense, each sample data is used as both training data and test data, which can effectively avoid the occurrence of over-learning and under-learning. By taking the 6-fold cross-validation, our training process is carried out on the training set of LPMT (Tab. 2) and the obtained prediction accuracy is defined as the training accuracy. Then, we verify the trained model on the test dataset (Tab. 3) of LPMT. The quantitative results obtained in this paper are shown in Tab. 7, while the qualitative results are shown in Fig. 3.

Table 7: The comparison of RGGCnet with SEGCloud, SPG, PointNet and PointCNN (test dataset of LMPT)

Method	Terrain	Debris	Acc	mIoU	OA
Ours	99.22	88.95	91.74	94.08	99.26
SPG	98.86	85.20	90.76	92.03	98.93
SEGCloud	97.25	83.34	88.63	89.29	96.32
PointNet	96.45	86.68	87.32	85.56	96.45
PointCNN	97.34	84.34	86.35	87.26	95.46
3D-BoNet	98.34	84.16	89.98	93.64	94.65



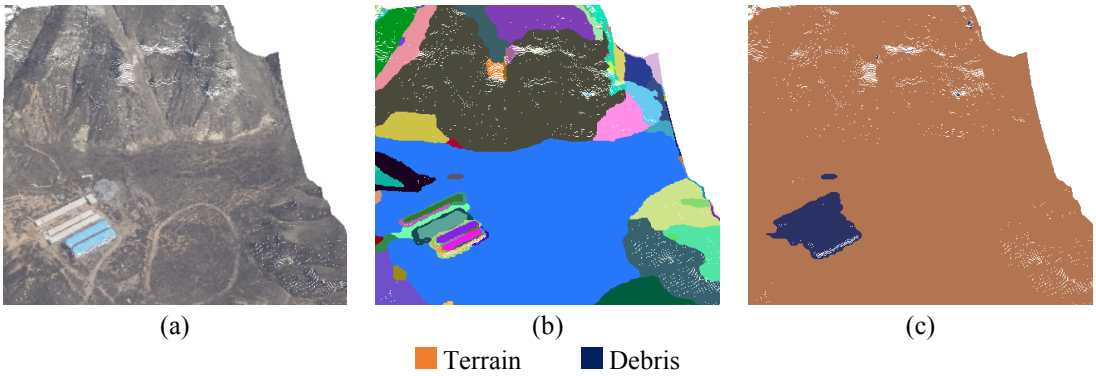
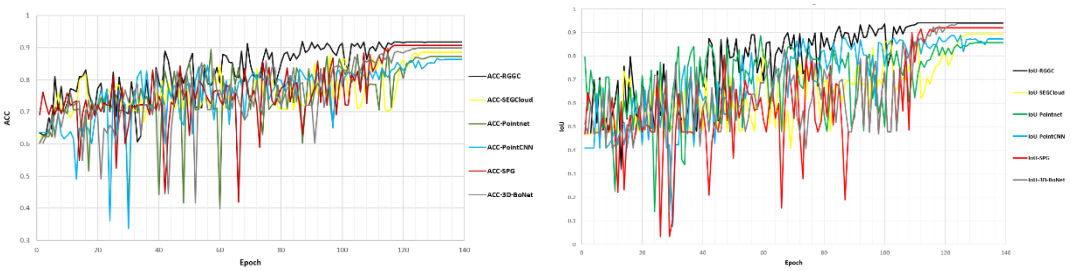


Figure 3: Raw point and prediction (Test dataset of LMPT) (a) Description of raw point cloud; (b)Terrain division; (c) Description of prediction



(a) (b)

Figure 4: Accuracy of IoU and Acc of each framework in test dataset of LPMT. (a) Description of Acc; (b) Description of IoU

5 Discussion

From Section 4.3 (Tab. 6) we can see that RGGCNet is much better than SEGCloud (3.11 Acc, 4.79 mIoU, 2.94 OA), PointNet (4.42 Acc, 8.52 mIoU, 2.81 OA), PointCNN (5.39 Acc, 6.82 mIoU, 3.80 OA) and 3D-BoNet (1.76 Acc, 0.44 mIoU,4.61 OA) for them only designed for the indoor scene. When dealing with the debris of terrain remote sensing data, these networks could not extract effective information only from limited number of point cloud of terrain debris. Therefore, in the accuracy of Debris prediction, RGGCNet leads SEGCloud by 5.25 percentage points, PointNet by 2.27 percentage points, and PointCNN by 4.61 percentage points. The advantage of RGGCNet over SPG is in the introduction of the LSTM gate structure (2.05 mIoU), which can effectively improve the problem of Gradient explosion and gradient disappearance during training process. And The comparison between RGGCNet and NoLSTM in Section 4.4 (Tab. 7) also proves this view.Apart from that, The result in Fig. 4 indicates that in the same test on LPMT, RGGCNnet only used 116 epoches to achieve convergence, 20 epochs less than other network structures, indicating that the training process is accelerated by reducing the number of point clouds (Tab. 6).

We also explore the advantages of several designed attributes by individually removing them from RGGCNet to compare the framework's performance of each attributes, and

the comparing result is shown in Tab. 8. In Color Descriptor Only, we removed the geometric descriptor from multi-attribute descriptors during the terrain division process; In Geometric Descriptor Only, we removed color descriptor from multi-attribute during the terrain division process; In NoLSTM, we remove the gate control LSTM from RGGCNet; In GRU, we changed the gate control unit from LSTM to GRU. It can be seen from the first three lines in Tab. 8, geometric descriptor accounts for about 10 mIoU points and color descriptor accounts for about 3 mIoU points, which indicates that both geometric descriptor and color descriptor are important information in the terrain point cloud. Next, without LSTM, RGGCNet's performance has declined significantly (5.35 mIoU). And Compared with LSTM, GRU decreases the performance by 2.12 mIoU. Therefore, we can see that LSTM can effectively improve the semantic segmentation accuracy of large-scale low-precision terrain point cloud while solving the gradient explosion phenomenon in the training process, and it has more advantages than GRU.

Table 8: Ablation study and comparison on the test set of LPMT

Model	Acc	mIoU
RGGCNet	91.74	94.08
Color Descriptor Only	81.32	83.86
Geometric Descriptor Only	88.56	91.14
NoLSTM	86.24	88.73
GRU	89.65	91.96

6 Conclusions

In this paper, a terrain faced 3D semantic segmentation framework for remote sensing is proposed by combing multi-attribute descriptors, graph structure and graph convolution to recognize terrain debris. The experiment results show that the proposed multi-attribute descriptor can commendably describe the limited point cloud of terrain debris, and the gated graph convolutional network with LSTM can effectively improve the semantic segmentation ability on graph structure which reduce the amount of terrain point cloud data from millions to hundreds. In conclusion, the proposed RGGCNet can effectively improve the semantic segmentation ability on remote sensing point cloud of terrain.

Availability of Data and Materials: Our project code-RGGCNet is available online at: https://github.com/XuHan-CN/RGGCNet. and our dataset-LPMT is available online at: http://118.126.108.211/LPMT/dataset/HTML/data.html.

Funding Statement: This research was funded by grant from the Key Research and Development Program of Shaanxi Province (2018NY-127, 2019ZDLNY07-02-01, 2020NY-205) and National Undergraduate Training Program for Innovation and entrepreneurship plan (S201910712240, X201910712080).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Boulch, A.; Guerry, J.; Saux, L.; Audebert, N. (2018): Snapnet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*, vol. 71, no. 1, pp. 189-198.

Daniilidis, A.; Deville, R.; Durandcartagena, E.; Rifford, L. (2017): Self-contracted curves in riemannian manifolds. *Journal of Mathematical Analysis and Applications*, vol. 457, no. 2, pp. 1333-1352.

Deng, Z.; Todorovic, S.; Latecki, L. J. (2015): Semantic segmentation of RGBD images with mutex constraints. *IEEE International Conference on Computer Vision*, pp. 1733-1741.

Engelmann, F.; Kontogianni, T.; Hermans, A. (2017): Exploring spatial context for 3D semantic segmentation of point clouds. *IEEE International Conference on Computer Vision Workshop*, pp. 716-724.

Gao, Y. T.; Yuan, Y. (2019): Real-time nature scene understanding based on superpixel segmentation. *Journal of Shenyang University*, vol. 31, no. 3, pp. 210-216.

Gharbi, M.; Chen, J.; Barron, J. T.; Hasinoff, S. W. (2017): Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 118.

Kim, S. K. (2013): Extraction of ridge and valley lines from unorganized posints. *Multimedia Tools and Applications*, vol. 63, no. 1, pp. 265-279.

Koka, S.; Anada, K.; Goto, T.; Noto, H. (2017): Ridge line detection of terrain maps represented by homogeneous triangular dissections. *IEEE/ACIS International Conference on Computer and Information Science*, pp. 879-883.

Landrieu, L.; Simonovsky, M. (2018): Large-scale point cloud semantic segmentation with super point graphs. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4558-4567.

Lawin, F. J.; Danelljan, M.; Tosteberg, P.; Bhat, G.; Khan, F. S. (2017): Deep projective 3D semantic segmentation. *International Conference on Computer Analysis of Images and Patterns*, pp. 95-107.

Li, K. L. (2013): *Visualization of Seabed 3D Terrain Based on Computer Graphics Processor (Ph.D. Thesis)*. Harbin Institute of Technology, China.

Li, T.; Li, X.; Ren, Y. J.; Xia, J. Y. (2020): Ground nephogram recognition algorithm based on selective neural network ensemble. *Computers, Materials & Continua*, vol. 63, no. 2, pp. 621-631.

Li, Y.; Bu, R.; Sun, M.; Chen, B. (2018): PointCNN: convolution on x-transformed points. *Conference and Workshop on Neural Information Processing Systems*, pp. 828-838.

Liu, S. L.; Li, F. Y.; Jiang, R. Q. (2015): A method of loess landform automatic recognition based on slope spectrum. *Journal of Geo-Information Science*, vol .17, no. 10, pp. 1234-1242.

Ma, Q. Z.; Li, E.; Yang, G. D.; Liang, Z. (2015): Terrain simulation for crustal movement simulation system based on contour lines. *Chinese Control Conference*, pp. 8795-8880.

Masocha, M.; Dube, T.; Mpofu, N. T. (2018): Accuracy assessment of Modis active fire products in southern African savannah woodlands. *African Journal of Ecology*, vol. 56, no. 3, pp. 563-571.

Peng, F.; Lin, Z. X.; Zhang, X.; Long, M. (2019): Reversible data hiding in encrypted 2D vector graphics based on reversible mapping model for real numbers. *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2400-2411.

Pomerleau, F.; Colas, F.; Siegwart, R. (2015): A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1-104.

Qi, C. R; Su, H.; Mo, K.; Guibas, L. J. (2017): Pointnet: deep learning on point sets for 3D classification and segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 77-85.

Qian, J. J.; Zhang, C. S.; Wang, K. (2017): Research review on thinning algorithm of airborne lidar point cloud data. *Bulletin of Surveying and Mapping*, vol. S1, no. 1, pp. 33-35.

Raguet, H.; Landrieu, L. (2017): Cut-pursuit algorithm for regularizing nonsmooth functionals with graph total variation. *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1724-1766.

Shaik, K. B.; Ganesan, P.; Kalist, V.; Sathis, B. S.; Jenitha, J. M. M. (2015): Comparative study of skin color detection and segmentation in HSV and YCBCR color space. *Procedia Computer Science*, vol. 57, no. 1, pp. 41-48.

Song, Y.; Yang, G. B.; Xie, H. T.; Zhang, D. Y.; Sun, X. M. (2017): Residual domain dictionary learning for compressed sensing video recovery. *Multimedia Tools and Applications*, vol. 76, no. 7, pp. 10083-10096.

Su, Y. T.; Bethel, J.; Hu, S. W. (2016): Octree-based segmentation for terrestrial lidar point cloud data in industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 113, no. 1, pp. 59-74.

Tang, B. X.; Zhou, Y.; Yu, Y.; Du, S. D. (2016): Higher-order class-specific priors for semantic segmentation of 3D outdoor scenes. *IEEE Winter Conference on Applications of Computer Vision*, pp. 1-9.

Taran, V.; Gordienko, N.; Kochura, Y. (2018): Performance evaluation of deep learning networks for semantic segmentation of traffic stereo-pair images. *International Conference on Computer Systems and Technologies*, pp. 73-80.

Tchapmi, L. P.; Choy, C. B.; Armeni, I.; Gwak, J.; Savarese, S. (2017): Segcloud: semantic segmentation of 3D point clouds. *International Conference on 3D Vision*, pp. 537-547.

Wang, P. S.; Liu, Y.; Guo, Y. X.; Sun, C. (2017): O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1-11.

Xiang, L. Y.; Shen, X. B.; Qin, J. H.; Hao, W. (2019): Discrete multi-graph hashing for large-scale visual search. *Neural Processing Letters*, vol. 49, no. 3, pp. 1055-1069.

Xie, C. H.; Wang, J. Y.; Zhang, Z. S.; Zhou, Y. Y.; Xie, L. X. et al. (2017): Adversarial examples for semantic segmentation and object detection. *IEEE International Conference on Computer Vision*, pp. 1378-1387.

Xu, J.; Schwing, A. G.; Urtasun, R. (2014): Tell me what you see and i will show you where it is. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3190-3197.

Yan, L.; Liu, H.; Tan, J.; Li, Z.; Chen, C. (2017): A multi-constraint combined method for ground surface point filtering from mobile lidar point clouds. *Remote Sensing*, vol. 9, no. 9, pp. 958-977.

Yang, B.; Wang, J.; Clark, R.; Hu, Q.; Wang, S. et al. (2019): Learning object bounding boxes for 3D instance segmentation on point clouds. *Conference and Workshop on Neural Information Processing Systems*, pp. 6737-6746.

Yang, F. S.; Yang, Y. Q.; Li, H. E.; Cao, M. M. (2015): Removal efficiencies of vegetation-specific filter strips on nonpoint source pollutants. *Ecological Engineering*, vol. 82, no. 1, pp. 145-158.

Yang, H. J. (2011): A novel algorithm for segmenting fruit from unorganized point clouds. *International Conference on Virtual Reality Continuum and Its Applications in Industry*, pp. 379-382.

Yang, H. J.; He, D. J.; Jiang, S. H.; Wang, H. (2013): Research on simplification and extraction technology of apple point cloud. *Journal of Applied Sciences*, vol. 13, no. 18, pp. 3684-3690.

Yang, S.; Maturana, D.; Scherer, S. (2016): Real-time 3D scene layout from a single image using convolutional neural networks. *International Conference on Robotics and Automation*, pp. 2183-2189.

Zhang, D. Y.; Liang, Z. S.; Yang, G. B.; Li, Q. G.; Li, L. D. et al. (2018): A robust forgery detection algorithm for object removal by exemplar-based image inpainting. *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 11823-11842.

Zhang, J. M.; Jin, X. K.; Sun, J.; Wang, J.; Sangaiah, A. K. (2018): Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, https://doi.org/10.1007/s11042-018-6562-8.