2024 IEEE 18th International Conference on Control & Automation (ICCA)
June 18-21, 2024. Reykjavík, Iceland

UMeshSegNet: Semantic Segmentation of 3D Mesh Generated from UAV Photogrammetry *

Xinyi Liu, Zihang Liu, Yongjun Zhang, Zhi Gao and Yuhui Tan

***Abstract*— 3D mesh generated from UAV photogrammetry can depicts the urban scene realistically. Most of the studies on semantic segmentation of 3D mesh based on deep learning convert mesh data into point cloud or 2D image, resulting in original information lost and poor segmentation effect. To address the problem, a semantic segmentation convolutional neural network UMeshSegNet is designed in this paper based on MeshCNN, which directly processes the mesh data. The network combines geometric, elevation and texture features, and attention mechanism is also introduced to enhance the sensitivity to the feature. Experiments and analyses are conducted on public dataset SUM and our own Wuhan test data, and the experimental results indicate that UMeshSegNet can effectively segment mesh data with significantly higher semantic segmentation accuracy than previous deep learning methods.**

I. Introduction

3D Real Scene is an important component of Digital City, and currently, the construction of 3D Real Scene of China is rapidly progressing. Compared to traditional photogrammetry techniques, 3D mesh models generated from UAV photogrammetry have the advantages of accurate morphology and realistic texture, making them an important source of data for city-level realistic 3D construction. However, the model constructed through UAV oblique photogrammetry technology is in an inseparable state and lacks semantic information on specific features, which limits its further application. To better utilize 3D mesh models and serve urban level realistic 3D construction, it is necessary to obtain semantic information on specific features. Deep learning has achieved good results in the semantic segmentation of images and point clouds, and has achieved significant results.

Due to the current mainstream deep learning methods mainly targeting data such as images and point clouds, rather than mesh data of oblique photography models, most classic deep learning models cannot be directly applied to oblique photography models. Most existing semantic segmentation methods for oblique photography models convert them into images or point clouds for processing. Image-based methods [1] obtain semantic results on multi-view images of mesh models and map the results back to the mesh model. These methods can utilize mature two-dimensional image processing techniques, but are susceptible to interference from occluded objects. Point cloud based methods directly or indirectly convert mesh data into point cloud data. Tutzauer et al. [2] calculated multi-scale geometric and spectral features on a triangular surface, created feature points at the center of gravity of the triangular surface, constructed feature point clouds, and inputted 1DCNN for semantic segmentation. Laupheimer et al. [3] fused LiDAR point cloud features with mesh features, extended the feature vectors of mesh data, and implemented semantic segmentation of mesh data using PointNet++. Point cloud based methods can directly utilize existing neural network models for point clouds, but cannot effectively utilize the geometric and texture information of mesh data itself.

Compared to the point clouds, mesh data can adaptively change density according to expression needs, storing geometric information of the same resolution in less space. At the same time, mesh data contains topological information and high-resolution texture information, making direct semantic segmentation based on mesh data have great potential. In recent years, some deep learning methods for mesh data have been proposed [4-6]. Among them, Hanoka et al. [7] analogized CNN models on images, defined grid convolution and pooling with neighboring edges as the neighborhood and central edges as the core, and proposed a neural network called MeshCNN that can be directly used for mesh data, making it possible to directly apply deep learning methods to the semantic segmentation task of mesh models.

In this paper, the state-of-the-art neural network MeshCNN is improved for mesh data by incorporating texture and elevation features, and introducing attention mechanisms to enhance feature perception. A convolutional neural network, UMeshSegNet, is proposed for semantic segmentation of mesh models. To address the partitioning problem of large-scale mesh data, Breadth First Search (BFS) algorithm is used to process mesh data. Experiments were conducted using the public dataset SUM and data from a certain area in Wuhan. The results showed that the semantic segmentation results of UMeshSegNet are superior to other classical 3D deep learning methods, and all the used features and attention mechanisms could improve the model performance. The research provides new ideas and solutions for semantic segmentation of mesh data generated from UAV photogrammetry based on deep learning.

II. Methods

A. Convolution and Pooling

In a triangular mesh, the shared edge between two triangular faces has four fixed neighboring edges, which can form a fixed convolutional receptive field. Therefore, edges can be treated as "pixels", and each edge can be assigned a feature vector to define convolution on an edge basis. As shown in Fig. 1, given the eigenvector e of a shared edge, the

* Research supported by National Natural Science Foundation of China Under Grant 42201474. (*Corresponding author*: Yongjun Zhang)

Xinyi Liu, Zihang Liu, Zhi Gao and Yuhui Tan are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, Hubei, China {email: liuxy0319, liuzh2022, gaozhinus, tyh2019 @whu.edu.cn}.

Yongjun Zhang is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (phone: +86-027-68771101; email: zhangyj@whu.edu.cn)

set of eigenvectors of its neighboring edges is fixed. In counterclockwise order, the adjacent edge feature vectors of $\boldsymbol{e}$ have two arrangements: $(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d})$ and $(\boldsymbol{c},\boldsymbol{d},\boldsymbol{a},\boldsymbol{b})$. Due to the need for convolution operations to ensure that each feature vector in the receptive field has a definite order, symmetry transformation is used to eliminate the influence of order. The transformation formula is shown in the formula (1). After eliminating the influence of order, conventional convolution operations can be performed on the edges. The convolution formula is shown in the formula (2), where $\boldsymbol{e}$ and $\boldsymbol{e}^j$ represent the feature vectors of the shared edge and the adjacent edge of the shared edge, respectively. k_0 and k_j are convolution kernel parameters. Since convolution is only calculated within local neighborhoods, traversing all edges in any order is feasible.

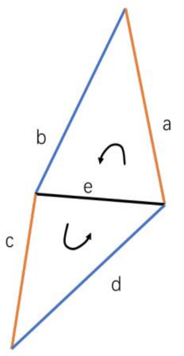

Figure 1. Edge of Mesh and Neighbor Edges

$$\left(\boldsymbol{e}^1,\boldsymbol{e}^2,\boldsymbol{e}^3,\boldsymbol{e}^4\right)=\left(\left|\boldsymbol{a}-\boldsymbol{c}\right|,\boldsymbol{a}+\boldsymbol{c},\left|\boldsymbol{b}-\boldsymbol{d}\right|,\boldsymbol{b}+\boldsymbol{d}\right) \tag{1}$$

$$e\cdot k_0+\sum_{j=1}^{4}k_j\cdot\boldsymbol{e}^j \tag{2}$$

The pooling operation on two-dimensional images can be abstracted into three steps: dividing data into specific pooling regions, merging features in pooling regions, and updating the adjacency relationships of merged features. Due to the irregularity of mesh data, pooling operations on two-dimensional images cannot be directly transferred to mesh data. Therefore, by analogy with these three steps, a pooling method for mesh data is proposed. Firstly, define each edge and its 1-ring neighborhood as a pooling region. As shown in Fig. 1, for the shared edge feature vector *e*, the set composed of its feature vectors is used as the pooling region. Different pooling orders produce different results, therefore, based on the importance extent of each edge, the pooling order is determined by the L2 norm of the eigenvalues. To control the degree of pooling, it is also necessary to define the target number of edges *n* for pooling. During the pooling process, it will collapse in sequence until there are *n* remaining edges. During the collapse process, the central edge is folded, and the remaining four adjacent edges are merged into two edges. As shown in the formula (3), the eigenvector of the collapsed new edge $(\boldsymbol{p},\boldsymbol{q})$ is obtained by taking the average of the eigenvectors $(\boldsymbol{a},\boldsymbol{b},\boldsymbol{e})$ and $(\boldsymbol{c},\boldsymbol{d},\boldsymbol{e})$ of the original triangle edge.

$$\boldsymbol{p}_i=avg\left(\boldsymbol{a}_i,\boldsymbol{b}_i,\boldsymbol{e}_i\right) \tag{3}$$

$$\boldsymbol{q}_i=avg\left(\boldsymbol{c}_i,\boldsymbol{d}_i,\boldsymbol{e}_i\right) \tag{4}$$

Unpooling process can restore collapsed edge structures during the pooling process, which is the reverse process of pooling. The UMeshSegNet proposed in this paper adopts an encoding-decoding structure, where each pooling layer corresponds to an unpooling layer. In the decoder module, the pooled feature map will be aligned with the corresponding pre-pooled feature map, and will be stacked and added into the convolutional layer. To achieve the above operations, it is necessary to record changes in the mesh structure during pooling, and to recover collapsed edge structures by inputing the recorded structural changes during unpooling. After performing the pooling operation, except for maintaining the original features of the central edge, the features of other adjacent edges are consistent with those of the collapsed edges. The specific process is shown in Fig. 2.

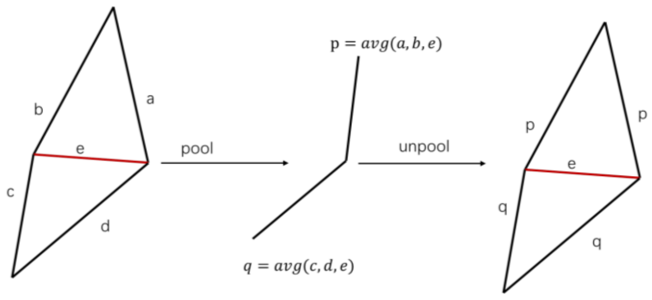

Figure 2. Pool and Unpool of Mesh

B. Network Structure

The UMeshSegNet network is designed based on the encoding decoding structure of the U-Net network, and introduces an attention mechanism module. The network structure is shown in Fig. 3. Each layer of the encoder consists of a DownConv convolution module and an SE (Squeeze and Excitation) attention mechanism module, which gradually extracts high-level semantic features through four rounds of pooling. Each layer of the decoder consists of an UpConv deconvolution module, which restores the original input size through four rounds of unpooling. The input of each layer of the decoder includes the unpooling result of the upper output feature map and the corresponding encoder output feature map. Finally, the feature map restored to its original input size will be subjected to a 1×1 convolution module and a sigmoid activation function to obtain semantic segmentation results. The DownConv convolution module and UpConv deconvolution module consist of basic MeshConv operations, where the DownConv module's MeshConv operation increases the number of output feature channels, while the UpConv module's MeshConv operation reduces the number of output feature channels. The operation process of MeshConv is shown in Fig. 4.

UMeshSegNet uses SE attention mechanism [8] to model the importance of different features for semantic segmentation tasks. Attention mechanism considers the importance of different data regions in learning. The SE attention mechanism assigns weight parameters to each input feature, emphasizing important features and suppressing secondary features, thereby improving the accuracy of the results. SE attention multiplies the learned channel weight vector with the feature map to adjust the model's feature perception ability.

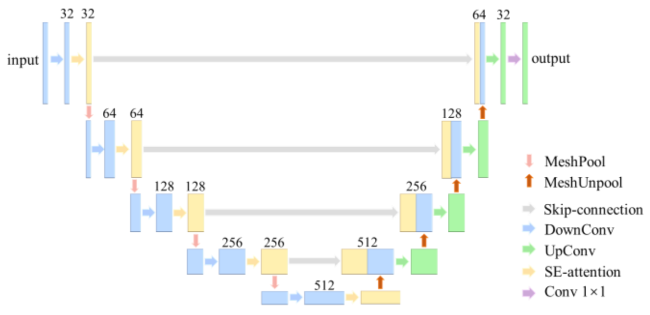

Figure 3. Network Structure

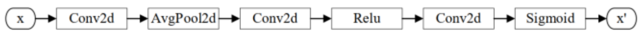

Figure 4. MeshConv Operation Procedure

The SE attention module can be divided into two parts: Squeeze and Excitation, as shown in Fig. 5. In which $\boldsymbol{x} \in R^{H \times W \times C}$ is the input feature, $\tilde{\boldsymbol{x}} \in R^{H \times W \times C}$ is the output feature, H and W are the height and width of the feature, and C is the number of channels for the feature. The compression part compresses the input features of $H \times W \times C$ dimension into $1 \times 1 \times C$ through global average pooling. Global pooling can fully consider all samples within each channel. The incentive part consists of two fully connected layers and a Sigmoid activation function alternately connected. Two fully connected layers are used to compress and recover the number of feature channels, and reduce redundancy through this process. Normalize the weight vector using the sigmoid activation function. Finally, the input features of $H \times W \times C$ dimension are multiplied by the weight vector of $1 \times 1 \times C$ dimension to obtain the output features $\tilde{\boldsymbol{x}}$.

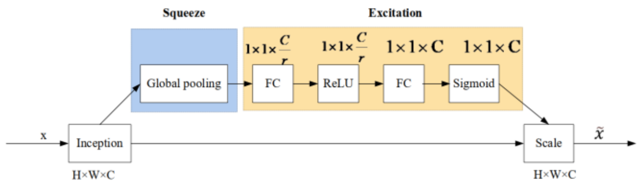

Figure 5. Network Structure of Attention Mechanism

C. Feature calculation

This section introduces the calculation method for the initial eigenvectors of each edge. The original MeshCNN only uses local geometric features to calculate the initial feature vector. As Rouhani et al. [10] and Verdie et al. [11] mentioned, texture features play an important role in semantic segmentation of mesh data generated from UAV photogrammetry. The elevation characteristics are also crucial for distinguishing buildings, vegetation, and ground. Therefore, UMeshSegNet extends the input features by using local geometric features, texture features, and elevation features related to edges as initial feature vectors.

Geometric features: The geometric features of mesh edges include dihedral angles of two adjacent faces on each edge θ_{AB}; The opposite vertex angle of two adjacent triangles γ_A and γ_B; Height h_A and h_B; the ratio of height to the corresponding bottom edge length, $ratioA$ and $ratioB$. The geometric features are independent of the absolute position of edges in the mesh, and the position information of edges is only used for pooling operations throughout the entire network calculation process. As shown in Fig. 6, CD is the shared edge of triangle ACD and BCD, $\boldsymbol{m}$ and $\boldsymbol{n}$ is the normal vector of the triangle ACD and BCD, respectively. The calculation method for their geometric features is:

$$cos\theta_{AB} = -\frac{\boldsymbol{m} \cdot \boldsymbol{n}}{|\boldsymbol{m}| \cdot |\boldsymbol{n}|} \tag{5}$$

$$cos\gamma_A = -\frac{|AC|^2 + |AD|^2 - |CD|^2}{2|AC||AD|} \tag{6}$$

$$cos\gamma_B = \frac{|BC|^2 + |BD|^2 - |CD|^2}{2|BC||BD|} \tag{7}$$

$$ratioA = \frac{h_A}{|CD|} \tag{8}$$

$$ratioB = \frac{h_B}{|CD|} \tag{9}$$

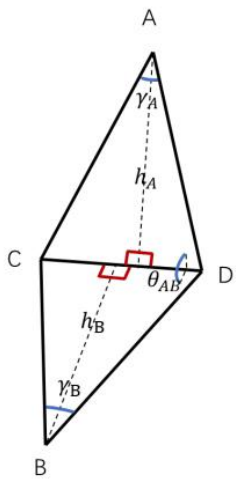

Figure 6. Geometrical Features of Mesh Edge

Texture features: We extract the RGB color values of mesh textures as texture features. Obtain the RGB color values of the texture corresponding to the two vertices of each edge in the mesh, and calculate their mean as the RGB color value for that edge. a, b represent the vertices of edge e, and the color is (R_a, G_a, B_a) and (R_b, G_b, B_b), then the color value of edge e is:

$$R_e = \frac{R_a + R_b}{2} \tag{10}$$

$$G_e = \frac{G_a + G_b}{2} \tag{11}$$

$$B_e = \frac{B_a + B_b}{2} \tag{12}$$

Elevation features: We extract the mean value of Z-coordinate of each edge and two vertices in the mesh as elevation features. For possible terrain undulations, we use relative elevation as the input feature, and the relative elevation of each edge can be calculated:

$$Z_e = \frac{Z - Z_{\min}}{Z_{\max} - Z_{\min}} \tag{13}$$

where Z represents the absolute elevation of the edge, $Z_{\min}$ represents the minimum elevation of the data block where the edge is located, and $Z_{\max}$ represents the maximum elevation of the data block where the edge is located. Z_e is the relative elevation of each edge within this data block. The relative elevation value of each edge is limited within $(0,1)$.

III. EXPERIMENTAL RESULTS

A. Dataset and Preprocessing

The experiment in this paper was conducted on the semantic segmentation dataset SUM [12] of urban area mesh models and mesh data in Wuhan, China. SUM data covers four areas of the Finnish capital Helsinki, which includes six categories of land features: buildings, vegetation, ground, vehicles, water bodies, and ships. The SUM dataset divides the raw data into 64 pieces of data with a size of approximately 250m × 250m, including 40 pieces of training data, 12 pieces of validation data, and 12 pieces of testing data. The data distribution and labels are shown in Fig. 7. Wuhan data includes ground, buildings, vegetation, water bodies, et al., as shown in Fig. 8.

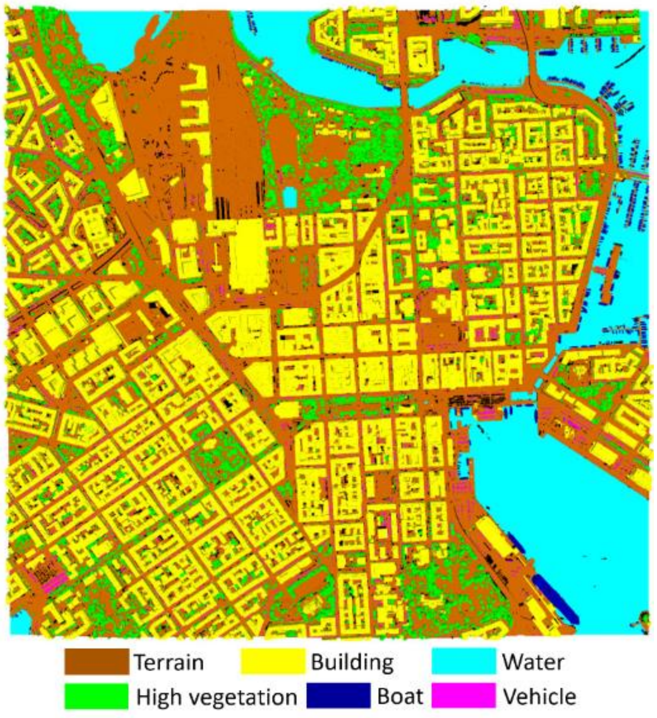

Figure 7. Distribution of SUM Dataset

In the process of deep learning training, large sample data cannot be directly input into the network, thus a blocking strategy is often adopted for large sample data. To ensure training effectiveness, the partitioned data should be kept of the same size as much as possible. Two-dimensional images can be divided into data of the same size according to a unified size, while oblique photography model data is irregular, and there may be significant differences in the number of edges between data divided according to a unified size. In response to the above issues, this paper uses a Breadth First Search (BFS) based method to divide the mesh model data into sub blocks with the same number of edges, while ensuring the topological continuity of each sub block.

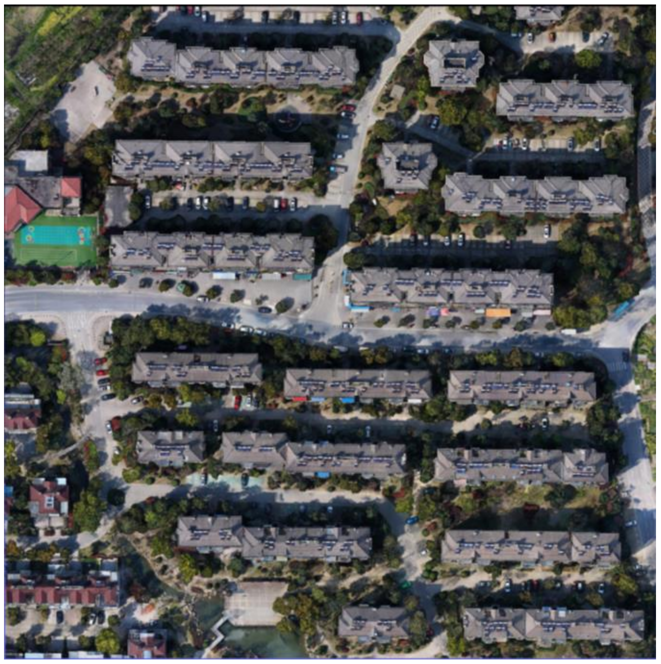

Figure 8. Distribution of Wuhan Dataset

During the iteration process, a complete face is added to each sub block to ensure the correctness of the topological relationship. The principle of searching for adjacent faces is that if the number of vertices shared by two faces is greater than or equal to 1, they are considered adjacent. During the iteration process, for each output sub block, all triangular faces belonging to that sub block will be marked as visited in the original data. When all faces are marked as visited, the iteration stops.

The training set in the SUM dataset is divided into 9654 sub blocks and the test set into 2109 sub blocks using this partitioning method. Each sub block contains 6000 edges, and due to the randomness of seed face selection, there may be a slight overlap between the sub blocks.

B. Parameter Settings and Evaluation Criterion

Precision is used to evaluate the semantic segmentation results of each category. Evaluate the overall results of semantic segmentation using *F1* value, overall accuracy (*OA*), and intersection over union (*IoU*). The definitions of accuracy, recall, *F1* value, *OA*, and *IoU* are as follows:

$$precision = \frac{TP}{TP + FP} \quad (14)$$

$$recall = \frac{TP}{TP + FN} \quad (15)$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (16)$$

$$OA = \frac{TP + TN}{TP + FP + TN + FN} \quad (17)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (18)$$

We train the UMeshSegNet on NVIDIA GeForce GTX 3090 on AMD Ryzen 9 5950X 16 Core Processor 3.40GHz using AdamW as the optimizer, with a batch size of 8. The loss function is cross entropy and the initial learning rate is set to 0.001.

C. Analysis of Comparative Experimental Results

To verify the effectiveness of the proposed model, this paper compared four classic 3D data semantic segmentation networks: PointNet [13], PointNet++ [14], SPG [15], and RandLA Net [16]. The specific results are shown in Table 1. Compared to the classical methods mentioned above, this model achieved the highest mIoU, OA, and F1 values, and performed the best in vegetation and vessel categories. The performance in ground, building, and vehicle categories also differed slightly from the optimal method. Fig. 9 shows the results of partial semantic segmentation. From the first line, it can be seen that for the category of buildings, roofs can basically be correctly distinguished, but some facades are mistakenly divided into vehicles and floors, which are mostly located at the junction of facades and floors. Although elevation and texture features help distinguish between buildings, ground, and vegetation, the influence of features is weakened due to the tendency of the intersection between building facades and ground to collapse more edges during pooling. From the second line, it can be seen that the method proposed in this article has a good segmentation effect on vegetation, but there are some cases of misclassification between low and low vegetation and the ground. In addition, the method proposed in this article can segment most vehicles, but the edges of the vehicles are easily misclassified as the ground, and some vehicles are completely misclassified as the ground.

TABLE I. COMPARATION OF SEGMENTATION RESULT(%)

Criterion	PointNet	RandLANet	SPG	PointNet++	UMeshSeg Net
Ground	56.3	38.9	56.4	**68.0**	67.3
Vegetation	14.9	59.6	61.8	73.1	**84.2**
Building	66.7	81.5	**87.4**	84.2	86.7
Water	**83.8**	27.7	36.5	69.9	50.7
Vehicle	0.0	22.0	34.4	0.5	34.2
Boat	0.0	2.1	6.2	1.6	**20.1**
mIoU	36.9	38.6	47.1	49.5	**57.3**
OA	71.4	74.9	79.0	85.5	**87.6**
F1	44.6	49.9	49.9	57.1	**69.5**

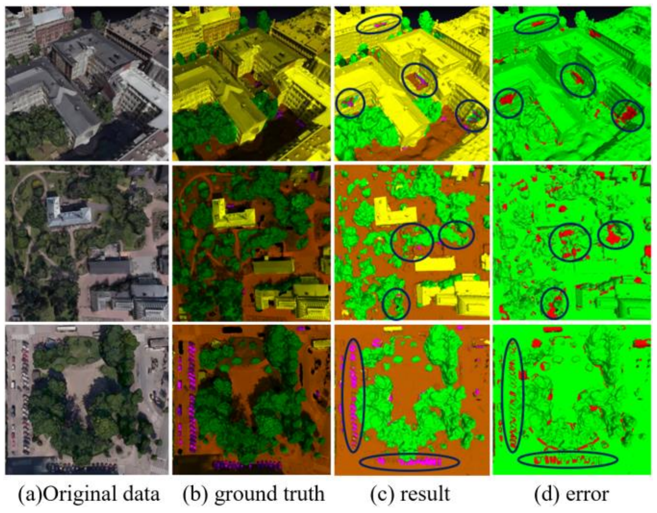

Figure 9. Segmentic Segmentation Results of SUM Dataset

D. Ablation Study

Ablation experiments are conducted on the SUM dataset, using the complete UMeshSegNet model as a benchmark. The elevation feature Z, texture feature RGB, geometric feature (dihedral angle θ_{AB}, inner angle γ, ratio of height and bottom *ratio*), and SE attention mechanism modules were removed, respectively. The models without Z, RGB, θ, γ, *ratio*, and SE were obtained respectively, and compared with the complete model to test the impact of each input feature and SE attention mechanism on the final result. Table 2 shows the specific results of the ablation experiment.

TABLE II. COMPARATION OF SEGMENTATION RESULT(%)

Method	Z	RGB	θ_{AB}	γ	*ratio*	SE	OA	F1	ΔOA	ΔF1
All	√	√	√	√	√	√	**87.6**	**69.5**	0	0
No Z	×	√	√	√	√	√	79.9	57.9	-7.7	-11.6
No RGB	√	×	√	√	√	√	78.2	51.4	**-9.4**	**-18.1**
No θ	√	√	×	√	√	√	82.6	62.9	-5.0	-6.6
No γ	√	√	√	×	√	√	84.0	66.9	-3.6	-2.6
No ratio	√	√	√	√	×	√	83.3	65.7	-4.3	-3.8
No SE	√	√	√	√	√	×	84.4	61.6	-3.2	-7.9

The complete UMeshSegNet model performs the best. All input features and attention mechanisms contribute to improving model performance, with the removal of any term leading to a decrease in OA within the range of [3.2%, 9.4%] and F1 values within the range of [2.6%, 18.1%]. Among them, the model without RGB with removed texture features shows the greatest decrease in OA and F1 values, with an F1 value reduced by 18.1% compared to the complete model, indicating that texture features are the most important in improving model performance. This is related to the ability of texture features to distinguish between vegetation and non-vegetation. Elevation features are also important in models, which is related to their ability to distinguish between ground and non-ground features. The OA and F1 values of the model without SE with removed attention mechanism module and the model without θ, without γ, and without *ratio* with geometric features removed have both decreased, but the difference is not significant. The impact of attention mechanism on each category is relatively average.

E. Generalization Study

To test the generalization ability of the UMeshSegNet model, we use the model trained on the SUM dataset to perform semantic segmentation on Wuhan data. The results are shown in Fig. 10. It can be seen that the buildings, vegetation, and ground in the area have been roughly segmented correctly, but there are still some errors. For building categories, there are differences between the building types in Wuhan data and the SUM dataset, with some building facades being misclassified into ground and vegetation. For vegetation categories, some of the vegetation in the Wuhan data is misclassified into buildings, and the misclassified areas are mostly located on the shaded side of the vegetation and at the junction of the vegetation and the ground. This may be caused by the poor texture quality of these parts. The segmentation accuracy of water bodies and vehicles in Wuhan data is relatively low, with most water bodies mistakenly divided into ground and buildings, and some vehicles divided into ground. Due to objective differences in data, the model may make some errors when applied, but it retains the segmentation performance on the source data. Among them, categories with a large amount of training data such as buildings and vegetation have better segmentation results, while categories with less training data such as water bodies and vehicles have lower segmentation accuracy.

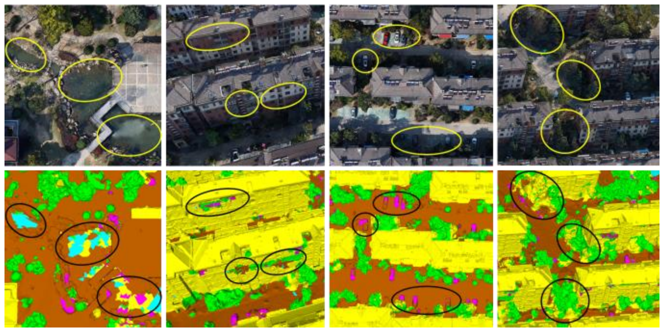

Figure 10. Segmentic Segmentation Results of Wuhan Data

IV. CONCLUSION

A deep learning network called UMeshSegNet is proposed in this paper, which is designed for semantic segmentation of 3D mesh generated from UAV photogrammetry and to address the problem of semantic segmentation methods that convert mesh models into images or point clouds for processing, resulting in loss of geometric and texture information. The proposed model incorporates geometric, texture, and elevation information from 3D mesh data and combines attention mechanisms to improve network performance. Experimental results on the public dataset SUM have shown that the semantic segmentation results of the UMeshSegNet are superior to other classical deep learning networks. The ablation experimental results indicate that all input features and attention mechanisms contribute to the improvement of model performance. Compared with other classical deep learning networks, UMeshSegNet has significant advantages in semantic segmentation performance, but there are still cases of building misclassification and low accuracy in some categories. In the future, the settings of edges number during partitioning and the combination with the point cloud based deep learning methods will be further researched to improve the network performance.

REFERENCES

[1] J. Chen, Y. Xu, S. Lu, et al. 3D Instance Segmentation of MVS Buildings. *IEEE Transactions on Geoscience and Remote Sensing*, 2022, 60, pp. 1-14.

[2] P. Tutzauer, D. Laupheimer, N. Haala. Semantic Urban Mesh Enhancement Utilizing A Hybrid Model. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019, IV-2/W7, pp. 175-182.

[3] D. Laupheimer, N. Haala. Juggling with representations: On the information transfer between imagery, point clouds, and meshes for multi-modal semantics. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021, 176, pp. 55-68.

[4] Y. Zheng, J. Zhao, Y. Chen, et al. 3D Mesh Model Classification with a Capsule Network. *Algorithms*, 2021, 14(3), pp. 99.

[5] D. George, X. Xie, G. K. Tam. 3D mesh segmentation via multi-branch 1D convolutional neural networks. *Graphical Models*, 2018, 96, pp. 1-10.

[6] L. Gao, J. Yang, T. Wu, et al. SDM-NET: Deep Generative Network for Structured Deformable Mesh. *arXiv*, 2019.

[7] R. Hanocka, A. Hertz, N. Fish, et al. MeshCNN: A Network with an Edge. *ACM Transactions on Graphics*, 2019, 38(4), pp. 1-12.

[8] V. Mnih, N. Heess, A. Graves, et al. Recurrent Models of Visual Attention. *//Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, 2, pp. 2204-2212.

[9] J. Hu, L. Shen, S. Albanie, et al. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 42(8), pp. 2011-2023.

[10] M. Rouhani, F. Lafarge, P. Alliez. Semantic segmentation of 3D textured meshes for urban scene analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017, 123, pp. 124-139.

[11] Y. Verdie, F. Lafarge, P. Alliez. LOD Generation for Urban Scenes. *ACM Transactions on Graphics*, 2015, 34(3), pp. 1-14.

[12] W. Gao, L. Nan, B. Boom, et al. SUM: A benchmark dataset of Semantic Urban Meshes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021, 179, pp. 108-120.

[13] R. Q. Charles, H. Su, M. Kaichun, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, 2017, pp. 77-85.

[14] R. Q. Charles, L. Yi, H. Su, et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv 1706.02413*, 2017.

[15] L. Landrieu, M. Simonovsky. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. *//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, 2018, pp. 4558-4567.

[16] Q. Hu, B. Yang, L. Xie, et al. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11105-11114.