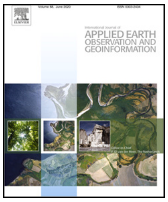
Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

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



3D building model generation from MLS point cloud and 3D mesh using multi-source data fusion

Weiquan Liu [a], Yu Zang [a,*], Zhangyue Xiong [a,b], Xuesheng Bian [c], Chenglu Wen [a], Xiaolei Lu [d], Cheng Wang [a], José Marcato Junior [e], Wesley Nunes Gonçalves [e], Jonathan Li [f]

[a] *Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China*
[b] *Institute of Information Technology, China Railway Nanning Group Co.,Ltd., Nanning 530000, China*
[c] *School of Information Engineering, Yancheng Institute of Technology, Yancheng 224051, China*
[d] *College of Foreign Languages and Cultures, Xiamen University, Xiamen 361005, China*
[e] *Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Campo Grande, MS 79070-900, Brazil*
[f] *Departments of Geography and Environmental Management and Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada*

ARTICLE INFO

Keywords:
3D building model generation
MLS point cloud
3D mesh
Multi-source data fusion

ABSTRACT

The high-precision generation of 3D building models is a controversial research topic in the field of smart cities. However, due to the limitations of single-source data, existing methods cannot simultaneously balance the local accuracy, overall integrity, and data scale of the building model. In this paper, we propose a novel 3D building model generation method based on multi-source 3D data fusion of 3D point cloud data and 3D mesh data with deep learning method. First, A Multi-Source 3D data Quality Evaluation Network (MS3DQE-Net) is proposed for evaluating the quality of 3D meshes and 3D point clouds. Then, the evaluation results are utilized to guide 3D building model generation. The MS3DQE-Net uses 3D meshes and 3D point clouds as inputs and fuses the learned features to obtain a more complete shape description. To train MS3DQE-Net, a multi-source 3D dataset is constructed, which collected from a real scene based on mobile laser scanning (MLS) 3D point clouds and 3D mesh data, including pairs of matching 3D meshes and 3D point clouds of the 3D building model. Specifically, to our knowledge, we are the first researchers to propose such multi-source 3D dataset. The experimental results show that MS3DQE-Net achieves a state-of-the-art performance in multi-source 3D data quality evaluation. We demonstrate the large-scale and high-precision, 3D building model generation approach on a campus.

1. Introduction

With the development of remote sensing and urban intelligence, urban data visualization is the key to convenient urban management. The construction of 3D urban models is an essential approach to data visualization, and 3D building models are widely utilized in all aspects of urban management. However, due to the complexity of a scene and the limitations of existing methods, obtaining high-precision and large-scale, 3D building models remains challenging.

The data captured by different sensors represent data from different sources, which are referred to as multi-source data or cross-domain data (Liu et al., 2021). Specifically, the fusion of multi-source data aggregates the characteristics of different sensors and the respective advantages of multi-source data, which enrich target information such as multisensing, multiperspective, and multidimensional information. In this paper, multi-source 3D data fusion is introduced into 3D building model generation.

Currently, there are three mainstream methods for obtaining large-scale, urban, 3D model data. The first method is the 3D model generated by the Structure from Motion (SfM) (Méndez-Barroso et al., 2018; Moran et al., 2021) or Multi-View Stereo (MVS) (Seitz et al., 2006; Chen et al., 2021c) algorithm through the aerial image sequence captured by oblique, unmanned aerial vehicle (UAV) photography, and multiple cameras onboard manned aerial platforms (Remondino et al., 2011; Nex and Remondino, 2014; Mohsan et al., 2022). The second method is the 3D point cloud obtained by the mobile laser scanning (MLS) system (Liu et al., 2020a; Su et al., 2022). The third method is the 3D point cloud obtained by the Airborne Laser Scanner (ALS) system (Roshandel et al., 2021, 2022).

Specifically, the 3D model generated by the SfM or MVS algorithm has high integrality, but the local accuracy is limited. The 3D point cloud obtained by MLS has high accuracy, but it is easily occluded,

[*] Corresponding author.
E-mail address: zangyu7@126.com (Y. Zang).

https://doi.org/10.1016/j.jag.2022.103171
Received 12 July 2022; Received in revised form 18 December 2022; Accepted 23 December 2022
Available online 4 January 2023

resulting in missing parts of a building. The ALS system can quickly obtain a large range of urban 3D point cloud. However, ALS system obtain the 3D data on the top of the target very well, but it is easily affected by occlusion, and it is difficult to obtain the 3D point cloud data of the target facade. Thus, the 3D model generated from single source data cannot simultaneously balance the local accuracy, overall integrity, and scope of a scene. In conclusion, the SfM/MVS data and point cloud data are complementary in space and accuracy and can be employed as basic 3D data for generating large-scale and high-precision, 3D building models.

3D models are usually expressed as mesh data, which comprise a collection of vertices, edges and faces of objects in 3D space (Yu et al., 2022). Thus, 3D mesh is suitable for use as the mathematical representation of urban scenes (Rocchini et al., 2001; Neto et al., 2021). The methods for obtaining 3D building models can be divided into two categories: the first category consists of manual modeling by modeling software, and the second category comprises a 3D model generation algorithm. Although the manual modeling method artificially controls the details of the 3D model, the modeling cost is very high since the process is time-consuming and labor-intensive, which is not suitable for the acquisition of 3D urban models. Currently, the mainstream method for obtaining 3D building models involves the use of 3D model generation algorithms, which are divided into traditional methods (Méndez-Barroso et al., 2018; Moran et al., 2021; Seitz et al., 2006; Chen et al., 2021c) and deep learning methods (Nan and Wonka, 2017; Ladicky et al., 2017; Wang et al., 2018). However, existing 3D model generation methods cannot simultaneously take into account the local accuracy, overall integrity and data scale of the 3D models. Details are presented as follows:

(1) Although the overall high integrality of a 3D model is ensured, the local high precision cannot be guaranteed. By using the SfM or MVS algorithm, the aerial image sequence captured by UAV tilt photography generates a large-scale, urban, 3D building model with complete scene coverage in a short time. However, aerial images are easily affected by the weather and distribution of buildings, which may cause uneven light distribution and occlusion of buildings. As a result, the local accuracy of the 3D building models generated by the SFM or MVS algorithm is low, and the scene details are absent. For example, the building structures at the eaves occlusion are abnormal, and the walls have holes.
(2) Although the local high precision of the 3D model is ensured, the 3D reconstruction of the large-scale scenes cannot be guaranteed. 3D model generation methods based on deep learning improve the problem of insufficient details, but due to the large consumption of computing power, the scale of generating 3D models is limited, and only smaller objects or scenes can be generated.
(3) Although the local high precision of the model is ensured, the high integrality of the 3D model cannot be guaranteed. 3D model generation methods based on point cloud meshing rely on the accuracy of 3D point clouds. However, the working area of the laser scanning system is constantly blocked by trees, pedestrians, vehicles, etc., making the captured point cloud data incomplete. This disadvantage results in the absence of the local area from the 3D model.

In this paper, in view of the limitations of the 3D model generated by single-source data and the existing 3D model generation methods, we focus on large-scale and high-precision, 3D building model generation. Moreover, deep learning has promoted the development of many computer vision tasks (Chen et al., 2021b; Liu et al., 2022; Chen et al., 2021a, 2022; Wu et al., 2022). Thus, a deep learning method is proposed for generating 3D building models based on the fusion of multi-source 3D data. The overall process of this method is shown in Fig. 1, which consists of three steps: 3D point cloud registration, 3D model quality evaluation, and 3D model generation based on multi-source data fusion.

The main contributions of this paper are summarized as follows:

(1) A Multi-Source 3D Data Quality Evaluation Network, MS3DQE-Net, is innovatively proposed for evaluating the data quality of the paired matching of 3D point clouds and 3D meshes. Experimental results show that MS3DQE-Net achieves a state-of-the-art performance in multi-source 3D data quality evaluation.
(2) Based on the evaluation results of the paired matching of 3D point clouds and 3D meshes by MS3DQE-Net, we propose an adaptive, 3D building model generation method based on multi-source data fusion. We use the MS3DQE-Net quality evaluation results for guidance, adaptively adjusting the fusion scheme in different areas of the 3D building model so that the generated 3D model balances the accuracy and completeness.
(3) To the best of our knowledge, we are the first researchers to propose a multi-source dataset, the paired matching of 3D mesh patch and 3D point cloud patch dataset of buildings, based on mobile laser scanning (MLS) 3D point cloud and 3D mesh data.

The organization of this paper is as follow: First, we introduce the background in Section 1; Second, the related work is described in Section 2; Third, we construct a multi-source 3D dataset of paired matching 3D mesh and 3D point cloud patches in Section 3; Fourth, we propose the MS3DQE-Net in Section 4; Fifth, we describe the 3D building model generation based on multi-source fusion in Section 5; Sixth, Section 6 shows the experimental results. Finally, the conclusion (Section 7) summarizes the works of this paper.

2. Related work

2.1. Deep learning with 3D data

2.1.1. Deep learning with 3D point clouds

Before deep learning with 3D point clouds, traditional, handcrafted, 3D point cloud feature descriptors rely on manual design, focusing on constructing the local geometric features of a point cloud, such as Fast Point Feature Histograms (FPFH) (Rusu et al., 2009), Signature of Histograms of OrienTations (SHOT) (Tombari et al., 2010), and Rotational Projection Statistics (ROPS) (Guo et al., 2013). Recently, many cloud feature descriptors based on deep learning strategies have been proposed with better overall performance (Guo et al., 2020).

The disorder, discreteness and uneven distribution of point clouds pose a challenge to deep neural networks that directly process points for feature extraction. Qi et al. (2017a) designed a neural network named PointNet that directly uses point cloud data as input, using each point as the smallest input unit. Subsequently, Qi et al. proposed PointNet++ (Qi et al., 2017b), which aims to overcome the limitation that PointNet cannot capture the local features of a point cloud. Inspired by PointNet, a series of works that directly input point clouds into deep neural networks for feature extraction, such as PointCNN (Li et al., 2018a), PointConv (Wu et al., 2019), PointASNL (Yan et al., 2020), SCF-Net (Fan et al., 2021) have been proposed.

2.1.2. Deep learning with 3D mesh data

Most of the current processing of 3D mesh data is based on traditional, handcrafted feature descriptors. Lien et al. (Lien and Kajiya, 1984) calculated moments of each tetrahedron in the mesh data. For 2D/3D objects in mesh representation, Zhang et al. (Zhang and Chen, 2001) developed several functions applied to each triangle and added all the resulting values. To solve the unstructured problem of mesh data, Hubeli et al. (Hubeli and Gross, 2001) extended the features of faces to a multiresolution setting. Kazhdan et al. (2003) proposed using the Spherical Harmonic descriptor (SPH) as a rotation invariant

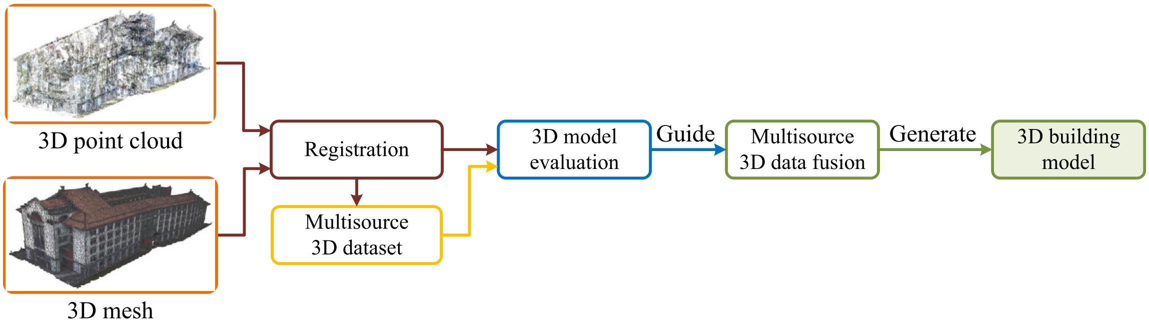


Fig. 1. Overview of 3D building model generation based on multi-source 3D data fusion.

representation for mesh data. Kokkinos et al. (2012) introduced the intrinsic shape context (ISC) descriptor to develop a generalization to faces, which solves the problem of orientational ambiguity.

Compared with the traditional processing method for 3D meshes, 3D mesh processing based on deep learning is just emerging. Feng et al. (2019) proposed MeshNet to learn 3D shape representation from 3D mesh data to solve the complexity and irregularity problem of mesh and to adequately conduct 3D shape representation. Li et al. (2020) presented a deep normal filtering network, DNF-Net, for mesh denoising. DNF-Net receives patches of facet normals as inputs and directly outputs the corresponding denoised facet normals. Zhang et al. (2021) proposed MeshingNet3D for the generation of high-quality tetrahedral meshes by using artificial neural networks.

2.2. Multi-source 3D data fusion

Bódis-Szomorú et al. (2016) proposed the fusion of airborne laser scanner (ALS) point clouds and MLS point clouds for modeling and provided a solution to the problem of excessively smooth model faces caused by using only ALS point clouds. Yang et al. (2017) investigated a workflow that utilizes factor graph Simultaneous Localization and Mapping (SLAM), dense 3D reconstruction and Iterative Closest Point (ICP) to efficiently generate LiDAR and camera point clouds and then coregistered them in a navigation frame to provide a consistent and more detailed reconstruction of the environment. Brell et al. (2019) presented the application of fusing the first pulse return information from ALS data at a subdecimeter spatial resolution with the lower-spatial-resolution hyperspectral information available from the HyperSpectral Imager (HSI) into a hyperspectral point cloud.

3. Multi-source 3D dataset acquisition

To the best of our knowledge, there is currently no public, large-scale, paired matching of a 3D point cloud patch and a 3D mesh patch for 3D building models. In this paper, the Zhangzhou Port campus at Xiamen University, Fujian Province, China, which covers an area of approximately 2 square kilometers, is selected as a real experimental scene. We propose a strategy to generate the multi-source 3D dataset of a building, which comprise the paired matching of 3D point cloud and 3D mesh patches.

As shown in Fig. 2, first, we use the RIEGL VMX-450 Mobile Laser Scanner (MLS) system to capture the LiDAR point cloud and street-view images of the campus. Second, we use a UAV equipped with a multiangle lens to capture the aerial image sequence of the campus and generate a 3D mesh model by using ContextCapture modeling software. Last, data preprocessing is performed on the original MLS point cloud and 3D mesh model, and only the data of the building area are retained.

Specifically, the height of UAV flight depends on the height of the building, the flight height is controlled at about 80–110 m, and the Ground Sampling Distance (GSD) is about 2–3 cm per pixel. The number of 3D mesh faces in 1 m^2 varies depending on the structure of the building. The number of points in the obtained 3D point cloud within 1 m^2 is about 4000 points, which will attenuate with distance.

Algorithm 1: Partition of the 3D mesh model based on the mesh face structure.

```
Input: The face set $F$ and the discrete vertex set $P$ of 3D mesh
  model, the size $grid_{size}$ of the 3D grid
Output: The 3D mesh patch set $C_{obj}$
  Determining the largest vertex coordinate $v_m$ and the smallest
    vertex coordinate $v_s$ in $F$, calculating the number of partitioned
    3D grids $num_{box} = (v_m - v_s)/grid_{size}$, and initializing the list $C_{obj}$
    with a size of $num_{box}$ to store the partitioned 3D mesh patches;
  Traversing $F$ and performing the following operations on each
    mesh face $f_i$:
    (a) Calculating the center point $v_c$ of $f_i$;
    (b) Calculating $I_c = [(v_c - v_s)/grid_{size}]$, and $C_{obj}[I_c]$ was added to
    $Index(f_i)$, denoting that $f_i$ belongs to the 3D mesh patch $C_{obj}[I_c]$;
  Traversing $P$ and performing the following operations on each
    discrete vertex $p_i$:
    Calculating $I_p = [(p_i - v_s)/gridsize]$, and $C_{obj}[I_p]$ was added to
    $Index(p_i)$, denoting that $p_i$ belongs to the 3D mesh patch $C_{obj}[I_p]$;
  Output 3D mesh patch set $C_{obj}$.
```

3.1. Multi-source point cloud registration

In this paper, we use the SAC-IA algorithm (Rusu et al., 2009) for coarse registration and then use the Scale-ICP algorithm (Ying et al., 2009) for fine registration. In coarse registration, the number of iterations is set to 1000, and the distance threshold between two point sets is set to 10 cm. In fine registration, the number of iterations is set to 1000, and the distance threshold between two point sets is set to 1 mm. The registration results are shown in Fig. 3.

3.2. Generation of paired matching of 3D point cloud patch and 3D mesh patch dataset

The matching of 3D point cloud and 3D mesh patches are generated based on the 3D mesh model and MLS 3D point cloud, which is shown in Fig. 4, including 3 steps: partition, outlier removal and labeling.

3.2.1. Partition of 3D data

The partition of the 3D mesh model is based on a fixed-size 3D grid. The partition algorithm (Algorithm 1) treats the entire 3D mesh model as a large cube and divides it into small cubes of the same size according to the 3D grid. The data contained in each small cube are treated as a small 3D mesh patch.

After completing the partition of the 3D mesh model, the corresponding MLS point cloud needs to be partitioned as a point cloud patch with the same size as the 3D mesh patch. The idea of the partition algorithm of the MLS point cloud is similar to that of the 3D mesh model. The partition algorithm (Algorithm 2) of the MLS point cloud

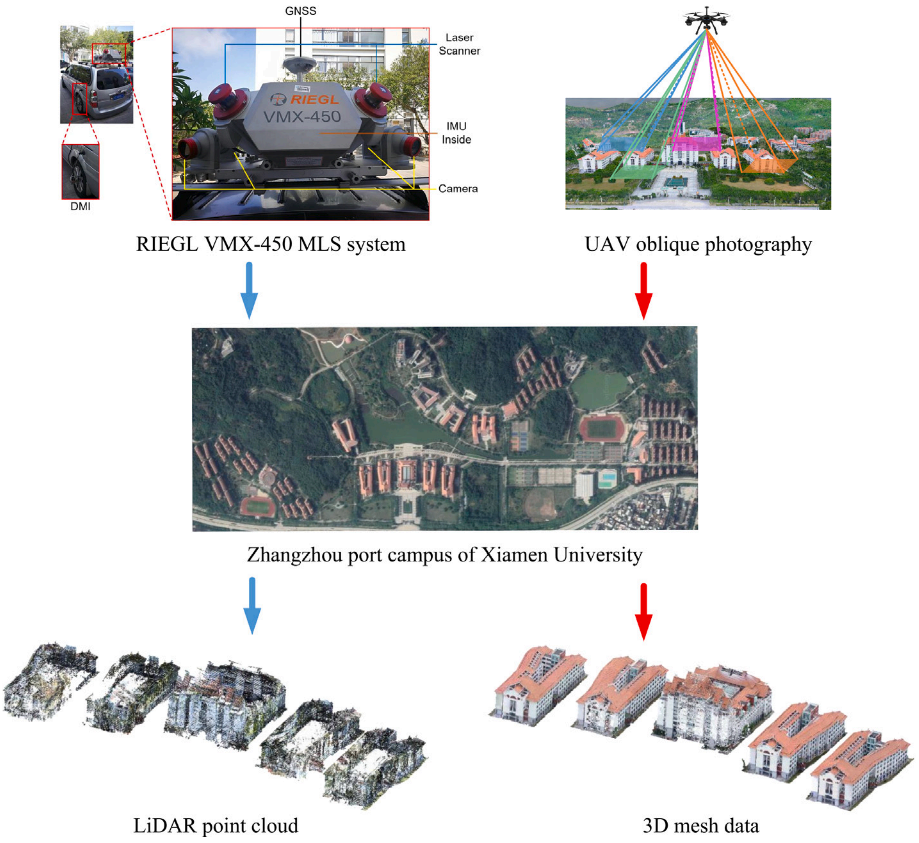


Fig. 2. Schematic of obtaining MLS point clouds and 3D mesh data.

Algorithm 2: Partition of the MLS point cloud based on the discrete points.

```
Input: Point cloud set $P$, and the size $grid_{size}$ of the 3D grid
Output: Point cloud patch set $C_{las}$
Determining the largest vertex coordinate $v_m$ and the smallest vertex coordinate $v_s$ in $F$, calculating the number of partitioned 3D grids $num_{box} = (v_m - v_s)/grid_{size}$, and initializing the list $C_{las}$ with a size of $num_{box}$ to store the partitioned 3D mesh patches;
Traversing $P$ and performing the following operations on each mesh face $p_i$:
    Calculating $I_p = [(p_i - v_s)/grid_{size}]$, and $C_{las}[I_p]$ was added to $Index(p_i)$, denoting that $p_i$ belongs to the point cloud patch $C_{las}[I_p]$;
Output point cloud patch set $C_{las}$.
```

considers the set of points that fall into the same small cube as a point cloud patch.

In this paper, we set $grid_{size}$ of the 3D grid to a cuboid with length, width and height of 23 cm, 10 cm and 7 cm, respectively. The partition results of the 3D mesh model patch and MLS point cloud patch are shown in Fig. 5.

3.2.2. Outlier removal of 3D data volume

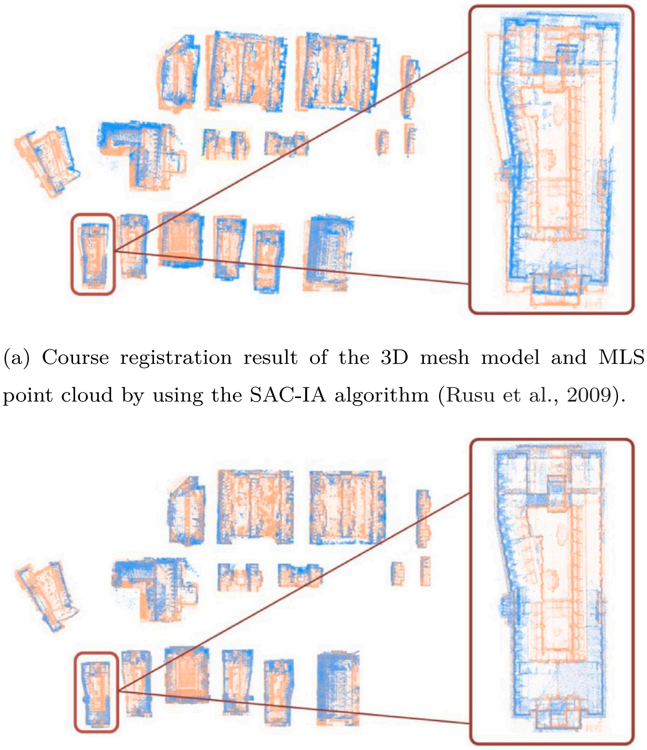
The shapes of the 3D mesh and 3D point cloud patches are irregular, so the number of faces or points contained between each 3D patch differs. According to the number of points or faces, the 3D data patches are distributed and counted. The 3D data patches with a lower proportion in the whole are outlier data. Note that the outlier 3D data patches are not representative, and they will affect the effect of the experiments, so it is necessary to remove the outlier 3D data patches. We use graphical analysis to remove outliers. In detail, we model the distribution of discrete faces or points into a histogram and fit a normal distribution curve on the histogram.

Fig. 6 presents an example of removing outlier data from 3D mesh patches. Fig. 6(a) shows that the proportion of faces with many or fewer faces overall is small. Therefore, they belong to outlier data and should be removed. After removing the outlier data, the distribution of the number of faces in each 3D mesh patch is shown in Fig. 6(b), which shows that removing the outliers makes the distribution more concentrated. Intuitively, the distribution of the number of faces in the 3D mesh patch dataset tends to be concentrated after outlier data are removed.

3.2.3. 3D data labeling

After removing the outlier data, the paired matching of the 3D mesh and 3D point cloud patches need to be labeled. Specifically, the purpose of the 3D model quality evaluation method proposed in this paper is to assess the quality of each area of the 3D model. Therefore, the paired matching of the 3D mesh and 3D point cloud patches need to be labeled as good or bad. The quality of the paired matching 3D mesh and 3D point cloud patches is labeled by 2 teachers and 8 students. They observe the building structure on the spot, compare with the photos of the building, then discuss together to form a consensus, and finally complete the labeling. The detailed criteria are as follows: (1) If there are holes in the building facade of the paired matching 3D mesh and 3D point cloud patches, it is judged as bad; (2) If the structure of paired matching 3D mesh and 3D point cloud patch is inconsistent with the structure of the building (protruding or recessed), it is judged as bad.

Finally, we construct 1000 pairs for matching the 3D mesh patch dataset and 3D point cloud patch dataset, of which 575 pairs of samples are labeled as good and 425 pairs of samples are labeled as bad.



(a) Course registration result of the 3D mesh model and MLS point cloud by using the SAC-IA algorithm (Rusu et al., 2009).

(b) Fine registration result of the 3D mesh model and MLS point cloud by using the Scale-ICP algorithm (Ying et al., 2009).

Fig. 3. Registration result of the 3D mesh model and MLS point cloud. The orange points represent the 3D mesh model vertex set, and the blue points represent the MLS point cloud. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Method

In this section, a Multi-Source 3D data Quality Evaluation Network (MS3DQE-Net) is proposed to evaluate the paired matching of 3D mesh and 3D point cloud patches. Note that the quality of each local area of the 3D building model can be evaluated as good or bad. Therefore, we convert the quality evaluation problem of the multi-source 3D data fusion of the 3D building model to a classification problem.

4.1. 3D data preprocessing

4.1.1. 3D data outlier removal and normalization

First, the outliers of faces or points in the 3D data are removed to reduce noise (as mentioned in Section 3.2.2). Second, the faces or points in the 3D data are moved to the center point and normalized. After normalization, the coordinates of the 3D data are moved to the center point and limited to the range of $(-1, 1)$.

4.1.2. 3D data augmentation

To improve the robustness and performance of MS3DQE-Net, 3D mesh and 3D point cloud are performed for data augmentation. Specifically, random noise following a normal distribution is added to the 3D point cloud (for the 3D mesh model, the vertex set of the 3D mesh model is approximately the same as the 3D point cloud). Selecting the 3D point cloud as an example, Fig. 7 shows the noise and 3D point cloud data augmented by noise. Figs 7(b), and 7(c) shows the results of 3D point cloud data augmentation, where purple represents the original 3D point cloud, and orange denotes the noise point cloud.

4.1.3. 3D data sampling

To ensure the same size of the 3D data patches, i.e., inputs of MS3DQE-Net, the number of faces or points of the input 3D data patches need to be fixed. When the number of faces or points is less than the fixed value, we randomly select faces or points in the current 3D data patch to fill up to the preset value. When the number of faces or points is greater than the set maximum number, we use the Farthest Point Sampling (FPS) method (Eldar et al., 1997) to downsample the 3D data patch to the fixed value. Fig. 8 shows the comparison before and after downsampling the 3D point cloud using the FPS method. The point density of the 3D point cloud is reduced after downsampling, but the overall structure has not changed.

4.2. 3D data characteristic analysis

In this paper, the features of the 3D mesh and 3D point cloud are fused to improve the accuracy of MS3DQE-Net in the quality evaluation of the generated 3D building models. When constructing the geometric structure relationship of 3D data, it is necessary to construct the neighborhood information of the faces or points of the 3D data. For 3D mesh data, the connection relationship between two faces is regular, that is, two faces sharing one edge are adjacent. The face of the 3D mesh and its neighborhood are shown in Fig. 9. The blue triangle represents the current face, and the red triangles represent the neighborhood faces. For 3D point clouds with fuzzy definitions of neighbor relations, the K-nearest neighbor algorithm (Altman, 1992) is employed to calculate the neighborhood.

4.3. MS3DQE-Net

The overall structure of MS3DQE-Net is shown in Fig. 10; it uses the paired matching of 3D mesh patches and 3D point cloud patches as inputs. First, MS3DQE-Net consists of two branches, a 3D point cloud feature extraction branch and a 3D mesh feature extraction branch, which are used to extract the features of 3D mesh patches and 3D point cloud patches, respectively. Second, the features extracted by the two branches are coupled by a feature aggregation module. Last, the complete feature pool is fed into the multilayer perception (MLP) layer to output the quality evaluation results of the model. Specifically, the processing of 3D point clouds refers to the idea of symmetric functions in PointNet (Qi et al., 2017a) and MeshNet (Feng et al., 2019). The function of each branch and module in the proposed MS3DQE-Net is shown in Table 1. For 3D mesh data, face-by-face processing combined with symmetric functions is utilized to solve the problem of face disorder. The detailed information of the 3D mesh data and 3D point cloud data input to MS3DQE-Net is shown in Table 2.

4.3.1. 3D point cloud feature extraction branch

Compared with the mesh data obtained by tilt modeling, the 3D LiDAR point cloud has higher accuracy, and its structure is closer to a real object or scene, so we use 3D LiDAR point cloud features to guide the classification of 3D data. For the point set, we use the shared MLP to extract high-dimensional features and then use max pooling as the symmetry function to solve the disorder problem. The spatial characteristics of the LiDAR point cloud are generated. The structure of the 3D point cloud feature extraction branch is shown in Fig. 11. The output dimension of MLP is set to $(32, 64)$, and the output of the 3D point cloud feature is Fea_{point}.

4.3.2. 3D mesh feature extraction branch

The structure of the 3D mesh feature extraction branch is shown in Fig. 12. Face features are employed to describe the structural features of the 3D mesh. The face in the 3D mesh contains its own shape and local structure information, so the face feature extraction subbranch can be subdivided into two subbranches: face-shape feature extraction and surface-local structure feature extraction. On the other hand, the

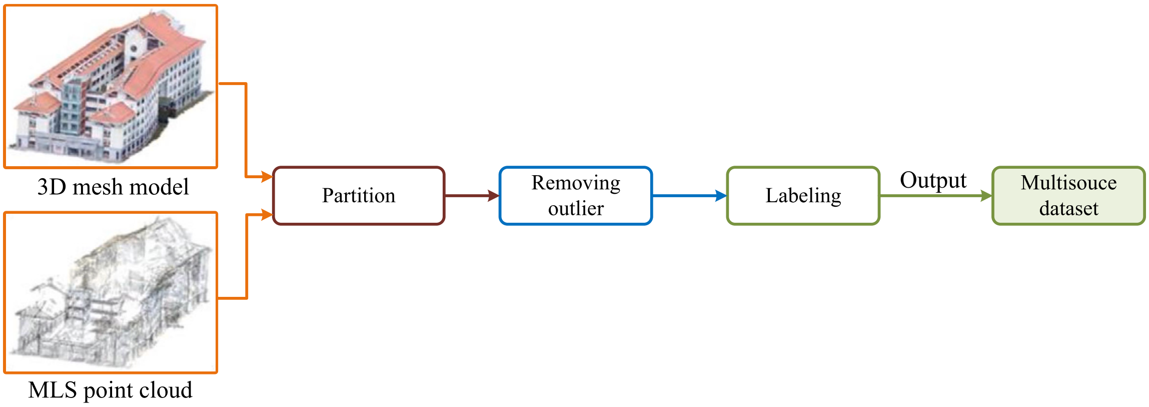
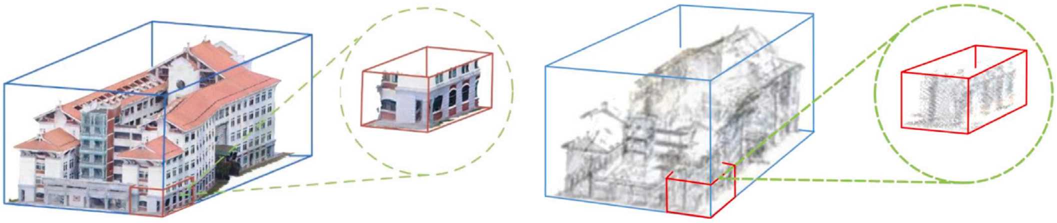


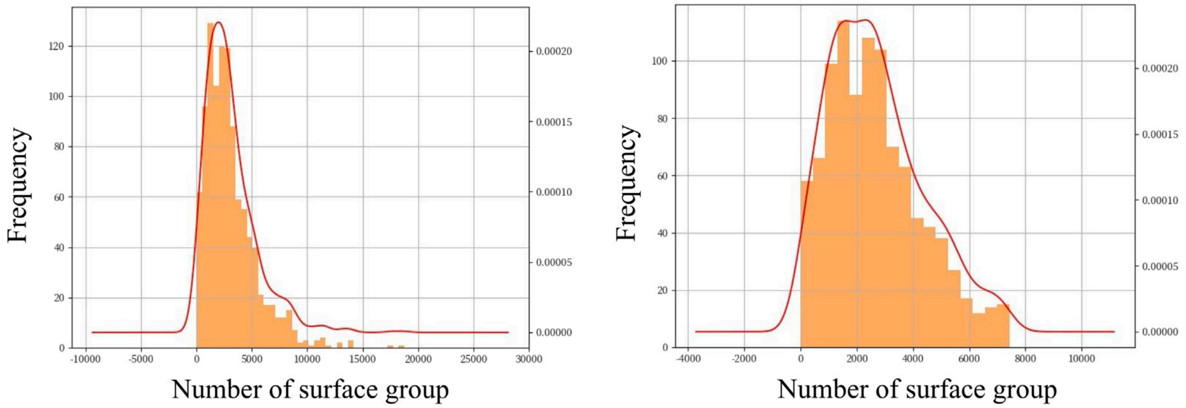
Fig. 4. Pipeline for generating paired matching of 3D point cloud patch and 3D mesh patch dataset.



(a) Partition of the 3D mesh model

(b) Partition of the MLS point cloud

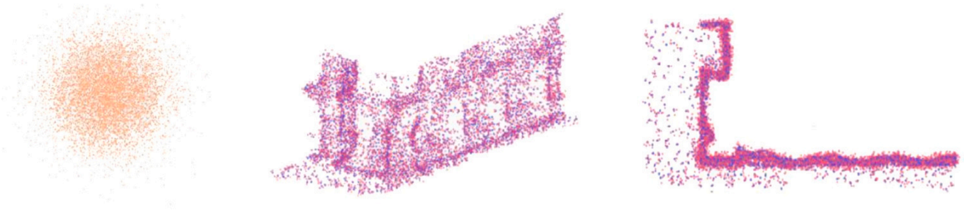
Fig. 5. Partition of the corresponding 3D mesh model and MLS point cloud.



(a) Original distribution

(b) Distribution after de-outlier

Fig. 6. Example of removing outlier data from 3D mesh patches in a 3D mesh model.

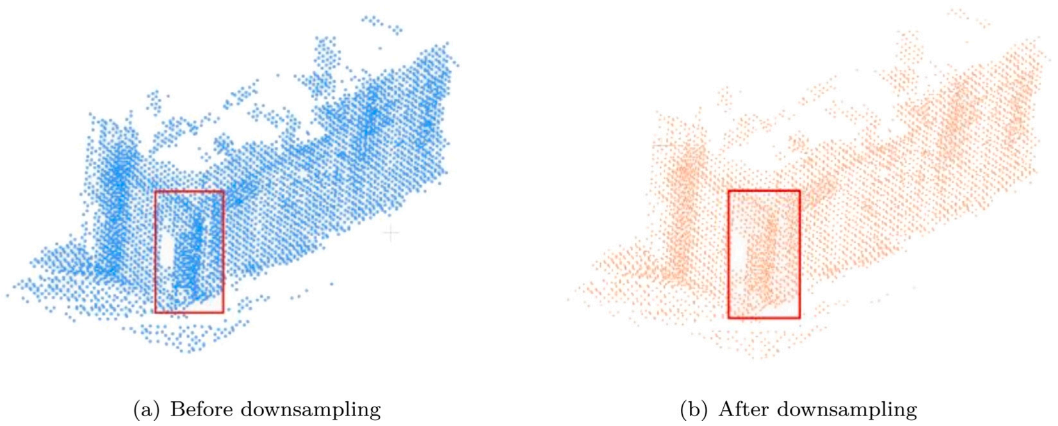


(a) Noise

(b) 3D point cloud data augmentation

(c) top view of (b)

Fig. 7. Example of 3D point cloud data augmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Before downsampling (b) After downsampling

Fig. 8. Comparison before and after downsampling the 3D point cloud.

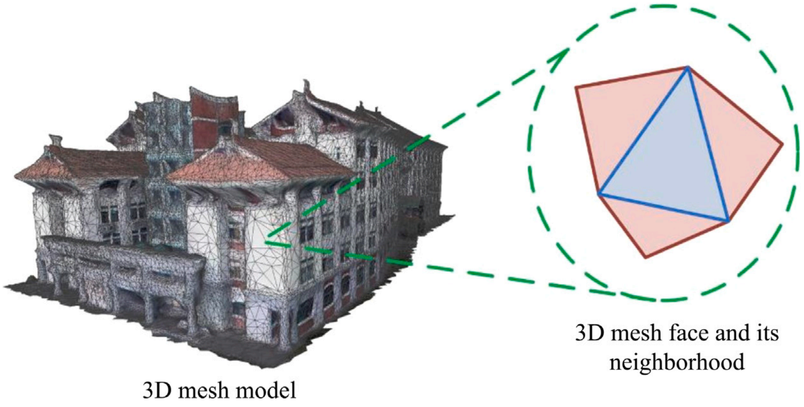


Fig. 9. Face of the 3D mesh model and its neighborhood faces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
The function of each branch and module in the proposed MS3DQE-Net.

Branch/Module			Function
3D point cloud feature extraction branch			Extracting the features of 3D point cloud patches
3D mesh feature extraction branch	Face feature extraction subbranch	Face-shape feature extraction subbranch	Capturing the geometric structure feature of each 3D mesh face
		Face-local structure feature extraction subbranch	Capturing the local neighborhood structure feature centered on each 3D mesh face
	Face center point global feature extraction subbranch		Describing the overall spatial features of the 3D mesh
	Face center point local feature extraction subbranch		Describing the detailed local geometric spatial features of the 3D mesh
Feature aggregation module	Point feature aggregation		Combining all the above extracted features
	Mesh feature aggregation		

Table 2
Detailed information of the 3D mesh face and 3D point cloud input to MS3DQE-Net.

Type	Input	Data dimension
3D mesh (face)	$center$	3
	$normal$	3
	$corner$	18
	$I_{neighbor\_face}$	3
	$I_{neighbor\_center}$	3
3D point cloud	$point$	3

center point feature expresses the position information of the face in space, where it is subdivided into global features and local geometric features.

Next, as shown in Fig. 12, we decompose the 3D mesh feature extraction branch into (i) the face feature extraction subbranch; (ii) the face center point global feature extraction subbranch; and (iii) the face center point local feature extraction subbranch and describe them in detail.

(i) Face feature extraction subbranch

Fig. 13 shows the structure of the face feature extraction subbranch, which combines the face-shape feature extraction subbranch and face-local structure feature extraction subbranch.

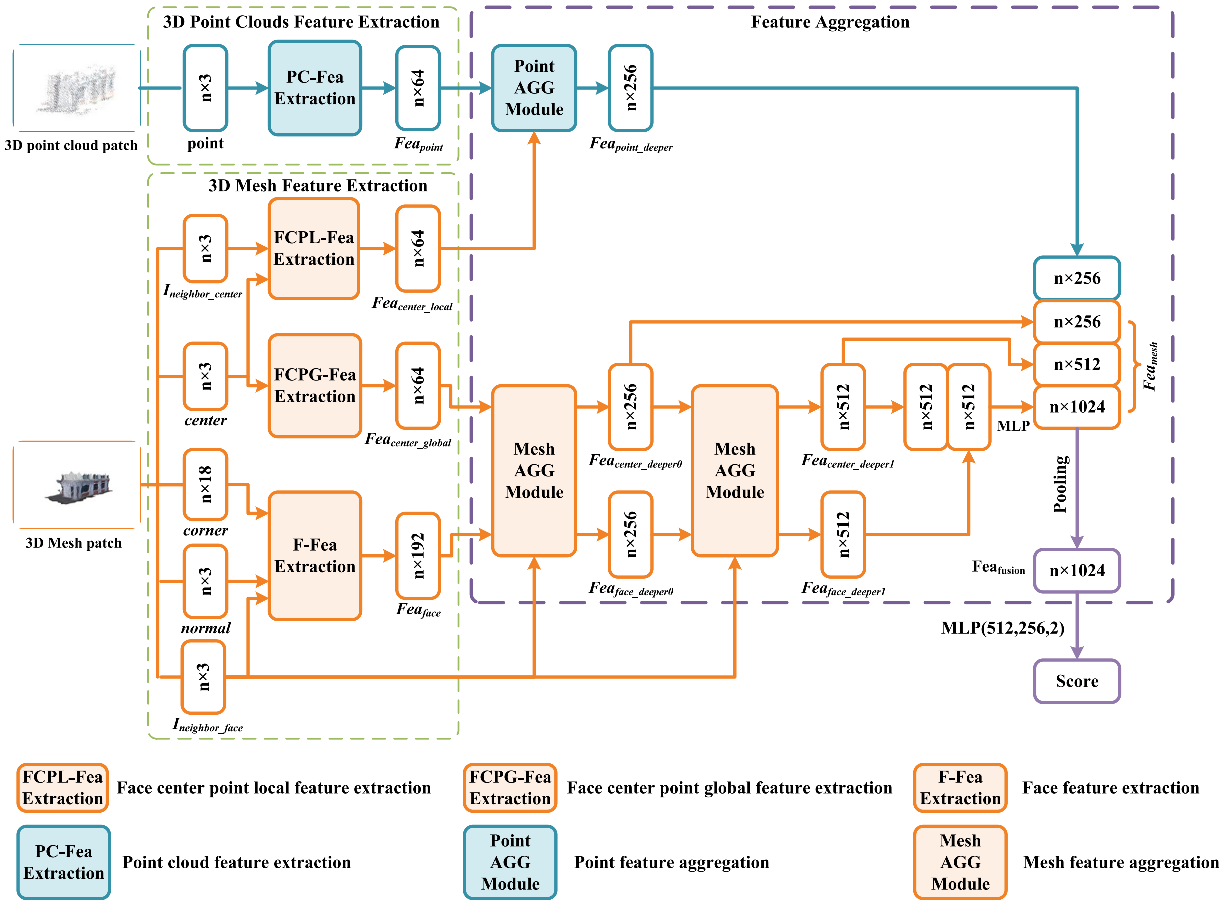


Fig. 10. The architecture of the proposed MS3DQE-Net.

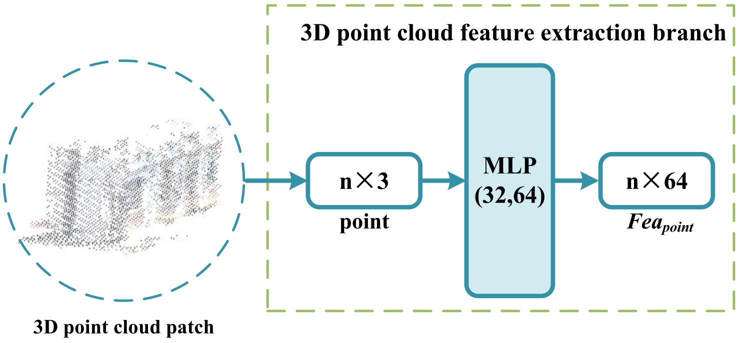


Fig. 11. Architecture of the 3D point cloud feature extraction branch.

① *Face-shape feature extraction subbranch*

The face-shape feature extraction subbranch aims to capture the geometric structure feature of each 3D mesh face. The entrance to this subbranch is a triangular mesh composed of 3 corners of each face, as they clearly express their shape characteristics. The processed data of this branch are triangular mesh data, and the three corners of each face can clearly express its shape characteristics, so the branch takes the three corners of the face as input. In this paper, we use two edge vectors to represent a corner, which can be represented by 6-dimensional parameters.

For each corner, we use MLP to obtain its feature in a higher dimensional space. To fuse the features of the three corners, a symmetric operation is used to eliminate their disorder. We use average pooling as the symmetric operation, which stabilizes the extracted corner fusion feature regardless of the order of corner input. The output features of the three corners are defined as follows:

$$g(h(corner_A) + h(corner_B) + h(corner_C)) \tag{1}$$

where h is the MLP(32,32), each layer of which performs a one-dimensional convolution operation, performs batch normalization, and then uses ReLU as the activation function. g is the average symmetrical function defined as follows:

$$g\left(h_1, h_2, \dots, h_n\right) = \frac{1}{N}\sum\left(h_1, h_2, \dots, h_n\right) \tag{2}$$

Through an MLP(64, 64), a 64-dimensional face shape feature Fea_{face_shape} is obtained.

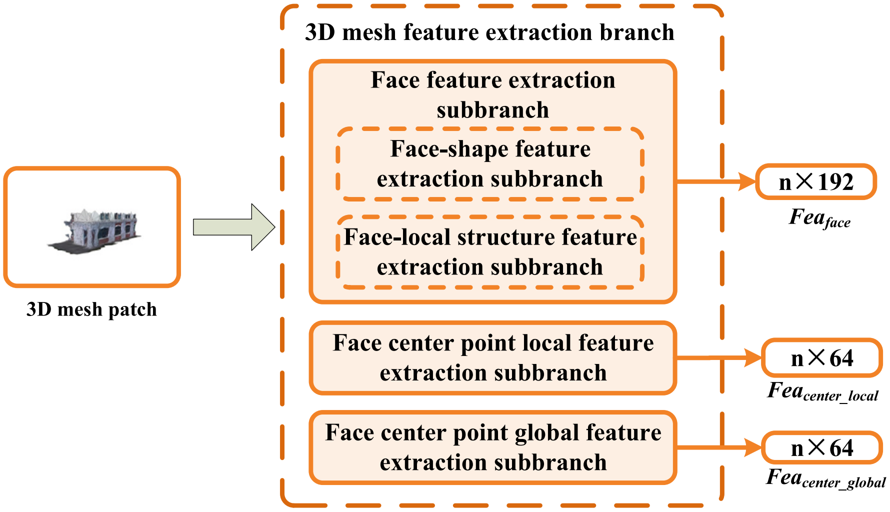

3D mesh feature extraction branch
Face feature extraction subbranch
Face-shape feature extraction subbranch
Face-local structure feature extraction subbranch
Face center point local feature extraction subbranch
Face center point global feature extraction subbranch
3D mesh patch
n×192
Fea_face
n×64
Fea_center_local
n×64
Fea_center_global

Fig. 12. Architecture of the 3D mesh feature extraction branch.

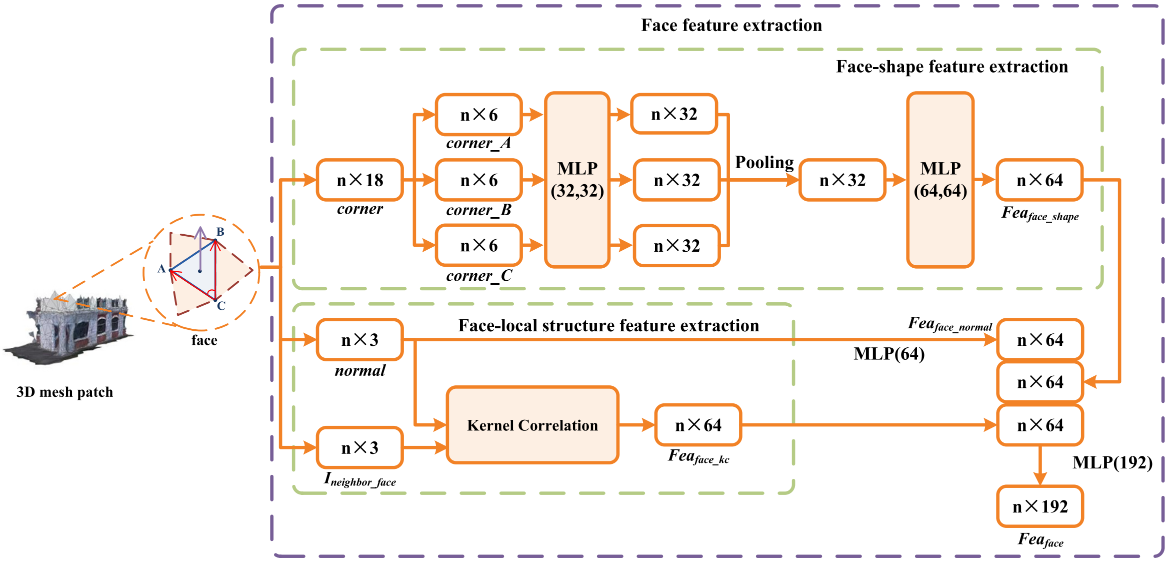


Fig. 13. Architecture of the face feature extraction subbranch.

② *Face-local structure feature extraction subbranch*

To more comprehensively represent the feature of each face in the 3D mesh data, in addition to capturing the internal structural features of each face (e.g., face-shape feature), it is also necessary to extract the external structural features. Inspired by KC-Net (Shen et al., 2018), the designed face-local structure feature extraction subbranch aims to capture the local neighborhood structure feature centered on each face.

As shown in Fig. 13, first, the face-shape feature Fea_{face_shape} and the face-local structure feature Fea_{face_kc} are combined by a cascade strategy to obtain the face feature, which includes the internal and external structure features. Second, this face feature and the normal feature Fea_{face_normal} are concurrently cascaded. Through this operation, the internal shape feature, external local structure feature and normal vector feature of the face are coupled, and last, this coupled feature is fed into the MLP to obtain the face feature Fea_{face}.

(ii) Face center point global feature extraction subbranch

The global features of the face center points of the 3D mesh data are used to describe the overall spatial features of the 3D mesh data. Generally, the points in the 3D mesh data refer to the vertices of each face, which express the position of each face in the 3D space. To reduce the number of network calculations, we use the face center point instead of the three vertices of the face as input. The calculation of the center point (x_c, y_c, z_c) of vertices A, B and C is presented as follows:

$$\begin{cases} x_c = 1/3\left(x_A + x_B + x_C\right) \\ y_c = 1/3\left(y_A + y_B + y_C\right) \\ z_c = 1/3\left(z_A + z_B + z_C\right) \end{cases} \tag{3}$$

Considering the face center point of the 3D mesh as a 3D point cloud and using PointNet to extract the global features, the architecture of the face center point global feature extraction subbranch is shown in Fig. 14. In detail, the face center point is fed into the MLP(64, 64), followed by pooling, and the 64-dimensional global feature Fea_{center_global} is output.

(iii) Face center point local feature extraction subbranch

The face center point local feature of the 3D mesh is employed to describe the detailed local geometric spatial features of the 3D mesh. We regard the face center point set as a 3D point cloud and apply the idea of kernel correlation (Shen et al., 2018), which is capturing

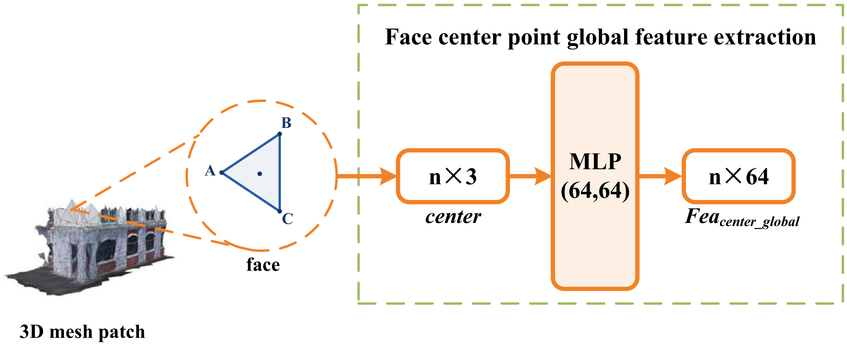


Fig. 14. Architecture of the face center point global feature extraction subbranch.

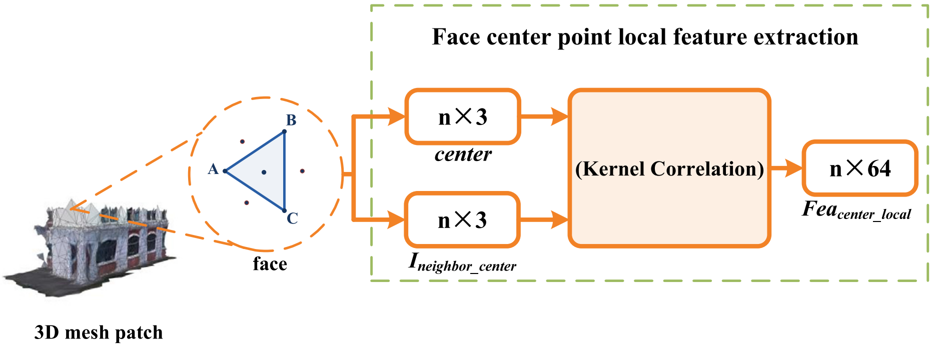


Fig. 15. Architecture of the face center point local feature extraction subbranch.

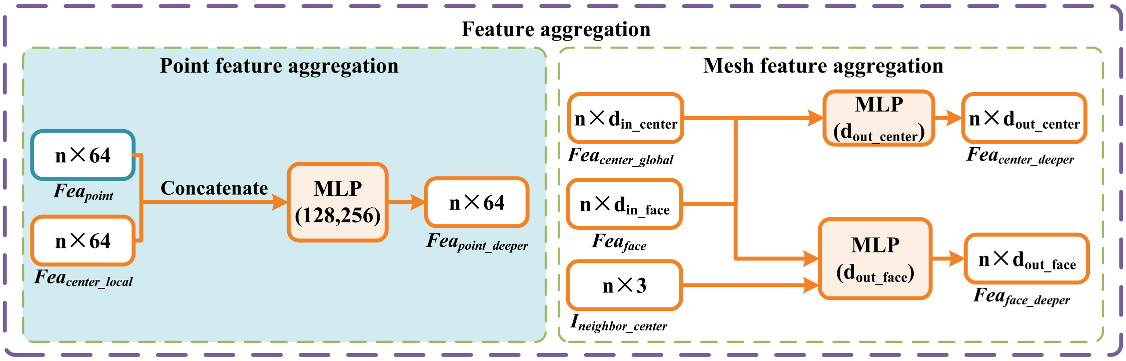


Fig. 16. Architecture of the Feature aggregation module.

local structures with nearest neighbors, to capture the local geometric features. The structure of the face center point local feature extraction subbranch is shown in Fig. 15.

We regard the K points closest to the face center point in Euclidean space as its neighbors. We use the K-nearest neighbor algorithm to calculate the neighbor points and the KD tree to establish a K-nearest neighbor search. The inputs of this subbranch are the *center* point and its neighbor index $I_{neighbor\_center}$. Similar to the face-local structure feature extraction subbranch, this subbranch makes the kernel a set of points in a spherical coordinate system and replaces the source in the kernel correlation function with a collection of face center points and neighbor points. The kernel function was also the Gaussian kernel function. The final output of this subbranch is a 64-dimensional feature Fea_{center_local}.

4.3.3. Feature aggregation

We propose a feature aggregation module that combines the above extracted face feature Fea_{face}, face center point global feature Fea_{center_global}, face center point local feature Fea_{center_local} and 3D point cloud feature Fea_{point} and that performs a complete description of the multi-source 3D data for the final classification task. Fig. 16 shows the structure of the feature aggregation module, which is divided into two modules: (i) point feature aggregation and (ii) mesh feature aggregation.

(i) Point feature aggregation

The point feature aggregation module uses the face center point local feature Fea_{center_local} and 3D point cloud feature Fea_{point} as inputs, cascades and feeds them into the MLP(128, 256) layer, and then outputs a 64-dimensional Fea_{point_deeper}.

(ii) Mesh feature aggregation

The inputs of mesh feature aggregation module are face feature Fea_{face}, face center point global feature Fea_{center_global} and face neighbor index list $I_{neighbor\_face}$. The face center point global feature

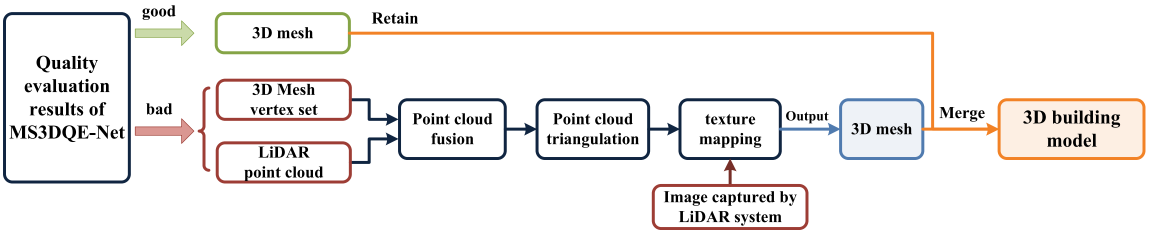


Fig. 17. Pipeline of the proposed 3D building model generation method based on multi-source 3D data fusion.

Fea_{center_global} and face feature Fea_{face} are cascaded and fed into the MLP to obtain a deeper face central point feature Fea_{center_deeper}. Fea_{center_deeper} contains richer spatial and structural features. On the other hand, the face feature Fea_{face} and face neighbor index list $I_{neighbor\_face}$ are defined as inputs. In the local neighborhood of each face, we cascade the central face feature with the neighbor face feature, perform maximum pooling, and then feed it into the MLP. This module further fuses the face features of the local neighborhood to obtain a deeper face feature Fea_{face_deeper}.

As shown in Fig. 10, after the point and mesh feature aggregation module operations, the strategy of final classification result of 3D mesh and 3D point cloud is as follows: (1) the output features Fea_{center_deeper} and Fea_{face_deeper} are cascaded and fed into an MLP to obtain a 1024-dimensional feature. (2) The 1024-dimensional features are cascaded with the multidimensional $Fea_{center\_deeper0}$ and $Fea_{center\_deeper1}$ to obtain mesh feature Fea_{mesh}. (3) The mesh feature Fea_{mesh} and point feature Fea_{point_deeper} are cascaded and performed to obtain a 1024-dimensional fusion feature Fea_{fusion}. The fusion feature Fea_{fusion} is a complete description of multi-source 3D data. (4) The Fea_{fusion} is fed into the MLP(512, 256, 2) to obtain the classification result.

MS3DQE-Net updates the network parameters by minimizing the cross-entropy (De Boer et al., 2005) between the predicted classification result and the true labeled value and obtains the final model quality evaluation model through multiple iterations.

4.4. Training strategy

MS3DQE-Net was implemented using Python 3.6 and PyTorch 1.7.0 on the Ubuntu 18.01 operating system with CUDA 11.0 and cuDNN 8.0. To accelerate the training, an Nvidia RTX 3090 GPU, whose training time is usually approximately 4 h, was employed. The cross-entropy is used as the loss function; the stochastic gradient descent (SGD) algorithm with momentum is used as the optimizer; and the momentum is set. The optimization algorithm restrains the loss oscillation and accelerates the convergence of the network. L2 regularization is utilized to prevent the model from overfitting, and the MultiStepLR method is employed to adjust the learning rate. The initial learning rate was set to 0.001; the momentum factor was set to 0.9; the training batch size was 8; and the network was trained for 100 epochs.

5. 3D building model generation based on multi-source fusion

In the step of 3D building model generation, we propose considering the characteristics of different source 3D data. We propose a novel, 3D building model generation method that is based on multi-source data fusion and that adopts an adaptive fusion strategy according to the quality evaluation results of MS3DQE-Net. In this way, different regions adaptively adjust the fusion strategy. The specific measures are presented as follows: the strategy of retaining the original 3D mesh is adopted for the areas with excellent evaluation, and the 3D mesh reconstruction strategy is adopted for the areas with poor evaluation. The reconstructed 3D mesh is merged with the retained mesh of the high-quality area, and a high-precision, 3D model of the building is obtained. The pipeline of the proposed 3D building model generation method based on multi-source 3D data fusion is shown in Fig. 17.

5.1. Point cloud fusion

For the area where the 3D building model quality evaluation result is poor, it is necessary to fuse the higher-precision LiDAR point cloud (i.e., closer to the real scene) with the matching 3D mesh patch to obtain richer information in the current area. However, based on the pros and cons of the MLS 3D LiDAR point cloud and oblique photography modeling 3D mesh model, the following key integration factors need to be considered: (1) The 3D mesh model has high integrity, large scale, and comprehensive coverage of the scene. (2) Because the MLS system is restricted by height or road obstructions, the LiDAR point cloud in certain areas is partially missing (such as roofs and areas where vehicles cannot enter), but the accuracy of the point cloud obtained is extremely high. (3) For the same scene, when the LiDAR point cloud can be obtained, the 3D mesh data in the same area lack model details, and the data quality is low.

Therefore, we use a point cloud fusion strategy based on the graph cut algorithm (Bódis-Szomorú et al., 2016; Li et al., 2018b), which uses the vertex set of the 3D mesh model as the reference basic data and the LiDAR point cloud as the candidate data. In detail, for the same 3D data patch region, when the similarity between the vertices of the 3D mesh and the point of a LiDAR point cloud exceeds the threshold, the point of the LiDAR point cloud and its neighboring points are used to replace and supplement the current 3D mesh vertices. Both types of points are measured by the Euclidean distance between two points and the angle between the normals as the similarity.

5.2. Point cloud triangulation

After the vertex set of the 3D mesh data is fused with the 3D LiDAR point cloud to generate a new point cloud, it retains the integrity of the scene area while containing the model details. Based on this fused point cloud, a more refined 3D building model can be generated. The most direct way is to turn the point cloud data into mesh data, and a common way is to triangulate the point cloud. We use the point cloud triangulation algorithm based on Delaunay triangulation (Gopi et al., 2000).

5.3. Texture mapping

After triangulation of the fusion point cloud to generate a 3D mesh, the texture of the original 3D mesh data may have issues such as missing or distorted data, and it needs to be remapped with a high-quality new texture. During the LiDAR point cloud scan, the MLS system also captures high-quality images that have internal and external camera parameters. These MLS images can be employed as a new texture for the generated 3D mesh data. In this paper, we use the projection texture mapping algorithm (Segal et al., 1992) to map the MLS images to the generated 3D mesh. Note that the standard for manually selecting high-quality texture is that the camera lens is facing the building facade when the texture image is taken.

Table 3
Comparison of classification results between MS3DQE-Net and representative methods.

Method	Data	Acc (%)	mAP (%)	Parameter	Time (s)
PointNet	point clouds	83.7	80.3	3,528,790	0.167
PointNet++	point clouds	85.5	82.0	968,269	0.744
PointCNN	point clouds	86.7	82.9	207,226	3.210
MeshNet	mesh	87.1	83.7	4,254,631	0.433
DIFD-Net	point clouds, mesh	85.6	82.7	14,889,054	0.083
Y-Net	point clouds, mesh	86.7	84.1	14,882,750	0.070
2D3D-MVPNet	point clouds, mesh	87.8	85.3	10,240,740	0.012
MS3DQE-Net	**point clouds, mesh**	**88.7**	**86.1**	4,541,057	0.514

Table 4
Classification results of ablation experiment on MS3DQE-Net.

Network structure	(1)	(2)	(3)	(4)	(5)	(6)	**Full**
3D point cloud feature extraction branch		✓	✓	✓	✓	✓	✓
Face feature extraction subbranch	✓	✓		✓	✓	✓	✓
Face center point global feature extraction subbranch	✓		✓	✓	✓	✓	✓
Face center point local feature extraction subbranch		✓	✓	✓	✓		✓
Kernel correlation module	✓	✓	✓		✓	✓	✓
Point feature aggregation		✓	✓	✓		✓	✓
Acc (%)	83.1	83.6	82.9	86.1	86.7	87.9	**88.7**
Time (s)	0.411	0.509	0.421	0.488	0.496	0.441	0.514

6. Results

In this paper, the quality evaluation of the proposed MS3DQE-Net is considered a classification task, so the evaluation metrics used in this experiment are Accuracy (Acc) and mean Average Precision (mAP), in which the accuracy ratio is utilized to measure the accuracy of classification and the accuracy index is used to measure the multiclassification performance of the network.

As there is no publicly available multi-source 3D dataset that contains a 3D mesh and corresponding 3D point cloud for 3D data classification, the dataset used in this paper is the multi-source 3D dataset (paired matching of 3D mesh patches and 3D point cloud patches) presented in Section 3. We divide the entire dataset into training data and testing data in a 4:1 ratio, of which 800 pairs are used to train the network and 200 pairs are used to test the network.

6.1. Multi-source 3D data fusion quality evaluation comparisons

To verify the effectiveness of the method, we apply MS3DQE-Net to the university campus multi-source 3D dataset to perform the quality evaluation task. The quality evaluation is approximate to the classification task, which is compared with the representative methods PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), PointCNN (Li et al., 2018a), and MeshNet (Feng et al., 2019). We also compare the cross-domain data matching network, DIFD-Net (Liu et al., 2020b), Y-Net (Liu et al., 2021), 2D3D-MVPNet (Lai et al., 2022), with our method. We keep the framework of the cross-domain data matching network, change the input data to 3D mesh and 3D point cloud patches, keep the loss function unchanged, and concatenate the extracted features of the two branches and put them into the MLP(512, 256, 2) to obtain the classification result.

Table 3 shows the results of 3D data classification of the MS3DQE-Net proposed in this paper and other representative methods. From a data perspective, many methods for 3D data classification can be divided into point cloud-oriented methods and mesh-oriented methods. The results show that both the Acc and mAP of MS3DQE-Net classification are higher than those of other representative methods, indicating that feature fusion of multimodal 3D data can better describe 3D objects and thus improve the classification accuracy. It is also possible to observe that the mesh-based method, MeshNet, obtained better results than other point cloud-based methods. This finding corroborates therefore that the proposed method is capable of extracting relevant information from the mesh and point cloud. The time in Table 3 is the time cost for a pair of 3D data to propagate forward through the network.

Table 5
The influence of parameters σ on the classification results.

σ	Acc (%)	*max_point*	Acc (%)
0.001	88.0	**7500**	**88.7**
0.005	**88.7**	5500	87.6
0.01	87.9	3500	87.1
		1500	86.9

6.2. Ablation study

6.2.1. Influence of key modules in MS3DQE-Net

To verify the effectiveness of the key modules in the proposed MS3DQE-Net, an ablation experiment was conducted to compare the effect of data classification by changing or removing the network structure. The results of the experiments are shown in Table 4.

The results show that the Acc is lower in cases (1), (2) and (3) and higher in cases (6) and Full. Therefore, the modules in MS3DQE-Net that have the greatest influence on the classification effect are the LiDAR point cloud feature extraction branch, the central point global feature extraction branch, and the surface feature extraction branch. The analysis results can be inferred as follows: The feature fusion of homologous data is very beneficial to 3D data classification. Compared with point cloud data, the mesh has a stronger ability to describe 3D data, which helps to improve the classification effect.

6.2.2. The influence of hyperparameters

There are special hyperparameters in MS3DQE-Net: the maximum number of points, maximum number of faces, and parameter in the kernel function. This experiment explores the influence of these parameters on the results of the proposed method.

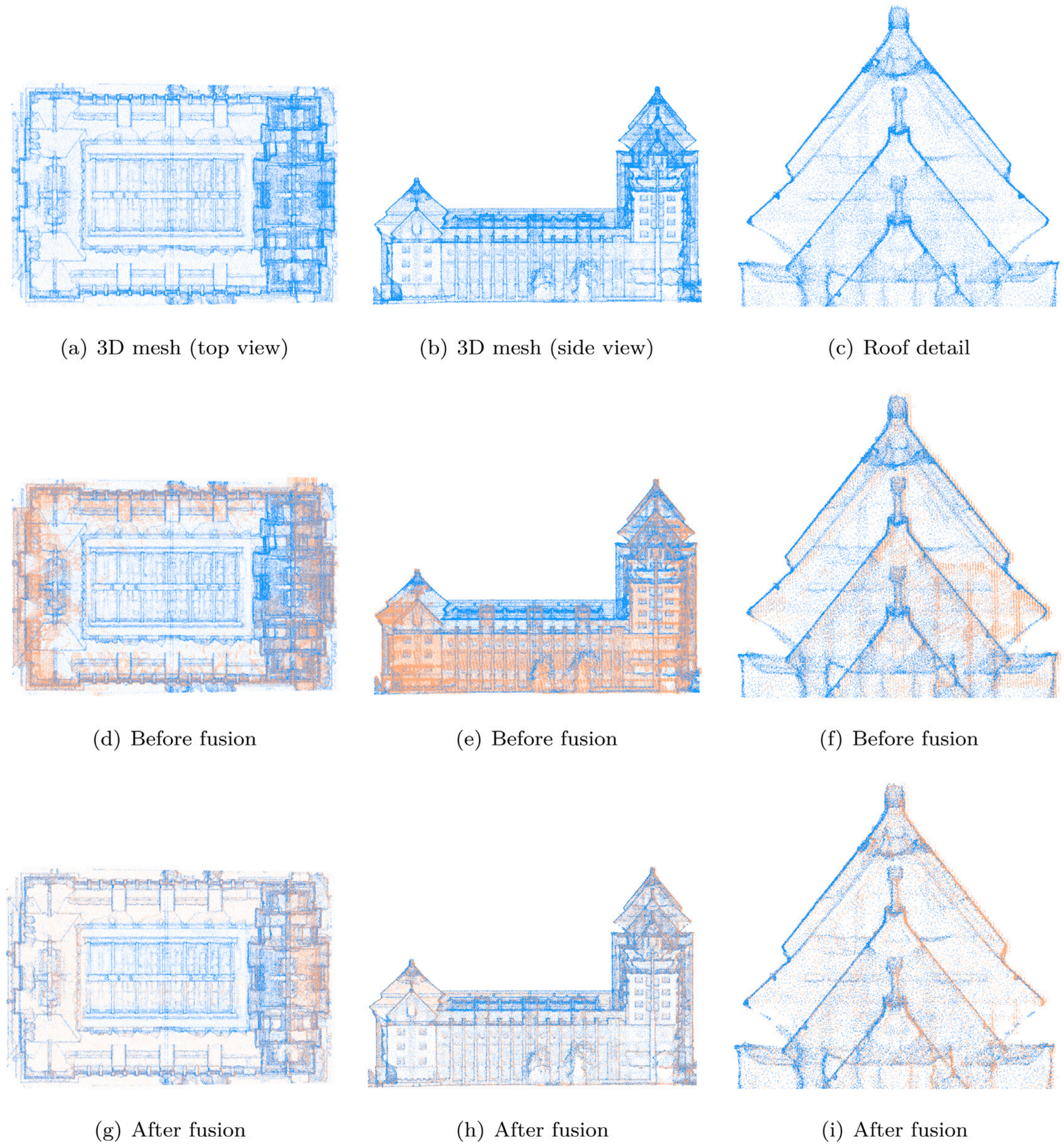
(i) Scope of influence of Gaussian kernels

As shown in Table 5, the parameter σ that controls the influence range of the Gaussian kernel was set to 0.005 based on the experience of Shen et al. (2018). Experimental results show that very large or small values worsen the classification.

(ii) Maximum number of points

To explore the influence of the number of points on the LiDAR point clouds, experiments were carried out, as shown in Table 5.

The results show that the accuracy of classification decreases with a decrease in the number of points, which indicates that MS3DQE-Net


(a) 3D mesh (top view)

(b) 3D mesh (side view)

(c) Roof detail

(d) Before fusion

(e) Before fusion

(f) Before fusion

(g) After fusion

(h) After fusion

(i) After fusion

Fig. 18. Results of point cloud fusion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Influence of parameters max_face on the effect of experiment.

max_face	Number of faces	Proportion (%)	Acc (%)
7500	(5000, 7500]	12	88.3
5000	(2500, 5000]	37	89.0
2500	(0, 2500]	**51**	**87.9**

is more sensitive to the reduction in the number of LiDAR point clouds. These results also indirectly indicate that the features of LiDAR point clouds have a great influence on classification and that the addition of multi-source data features can improve the classification accuracy.

(iii) Maximum number of faces

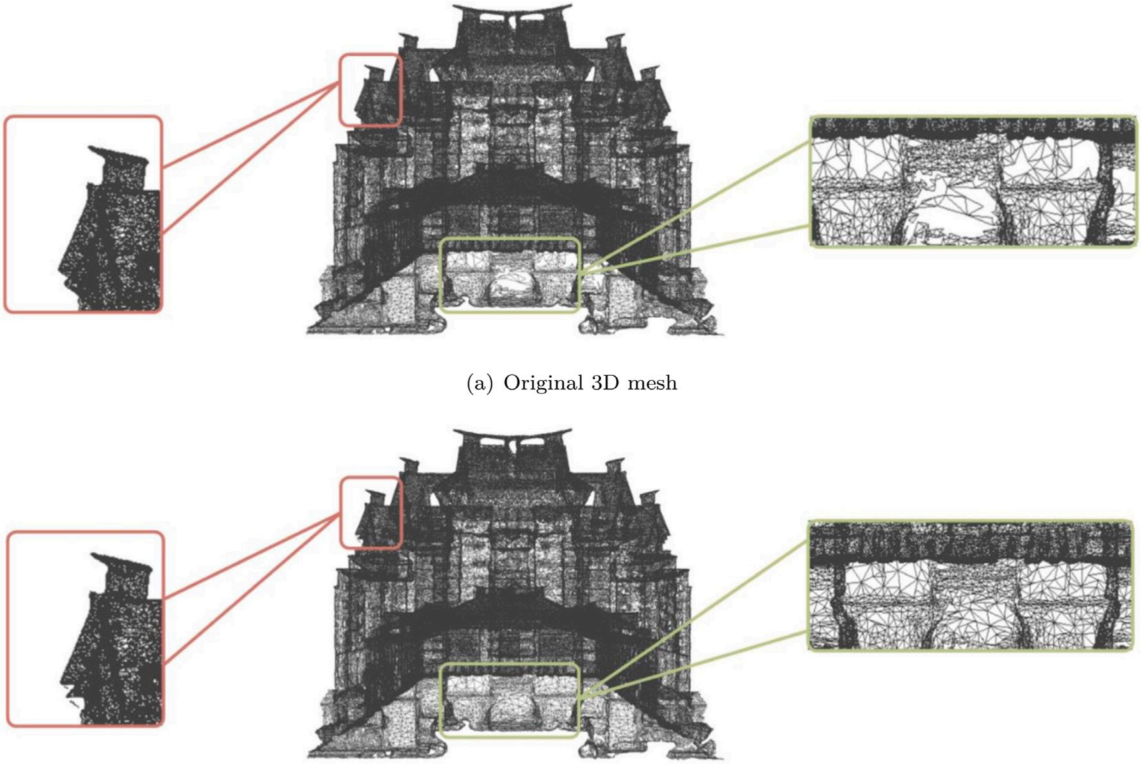
For the maximum number of faces, due to the different distribution of the number of faces in the mesh data, the mesh data are grouped according to an interval of 2500. The purpose of this experiment is to explore the influence of the number of faces on the classification effect. The experimental results are shown in Table 6.

According to the results, the classification accuracy of the group with the largest number of faces was the highest, while that of the group with the largest number of faces was the second highest, indicating that the correlation between the classification effect and the number of faces was lower. There was a small difference among the three groups of data, indicating that the sensitivity of the change in the number of opposite faces of the MS3DQE-Net model was lower.

6.3. Point cloud fusion

Fig. 18 shows the results of point cloud fusion based on the graph cut method, where the blue points represent the vertex set of the 3D mesh model, and the orange points denote the MLS LiDAR point cloud.

A comparison of the top view of the 3D building data, as shown in Figs. 18(a), 18(d) and 18(g), reveals that the points near the central area of the roof originate from the 3D mesh vertex set and that only the points on the outer edge of the roof are replaced by the points of the MLS LiDAR point cloud. The reason is that the scan height of the MLS system is limited and certain roads are blocked, which produces an incomplete scan area. Therefore, the idea of using the vertex set of



(a) Original 3D mesh

(b) Point cloud triangulation to new 3D mesh after fusion

Fig. 19. Triangulation of the fused point cloud.

the 3D mesh as the basic reference data is correct and can ensure the integrity of the target scene model.

A comparison of the side view of the 3D building data, as shown in Figs. 18(b), 18(e) and 18(h), indicates that part of the 3D mesh vertices in the building wall area and roof edge area are replaced by the LiDAR point cloud. This finding shows that the fusion method proposed in this paper effectively replaces the low-precision, 3D mesh vertices with a high-precision LiDAR point cloud and provides high-precision point cloud data for subsequent 3D building model generation.

A comparison of the roof details of the 3D building data, as shown in Figs. 18(c), 18(f) and 18(i), indicates that most of the points under the eaves are derived from LiDAR point clouds. The results show that the redundant building structure under the eaves is corrected to a certain extent after the fusion and that the model has more architectural details, but a few 3D mesh vertices still remain at the eaves occlusion. The analysis shows that the area where the residual points are located does not have a high similarity to the top LiDAR point cloud of the mesh, so the vertices of the 3D mesh are not removed.

6.4. Point cloud triangulation

The fused point cloud is meshed by the triangulation algorithm; the meshing result is shown in Fig. 19. The results show that the quality of the 3D mesh generated by the fusion point cloud is improved to a greater extent than that of the original 3D mesh. The holes on the building surface have been repaired; the quality improvement effect under the eaves is slightly lacking; and the remaining mesh vertices are meshed.

6.5. Texture mapping and model generation

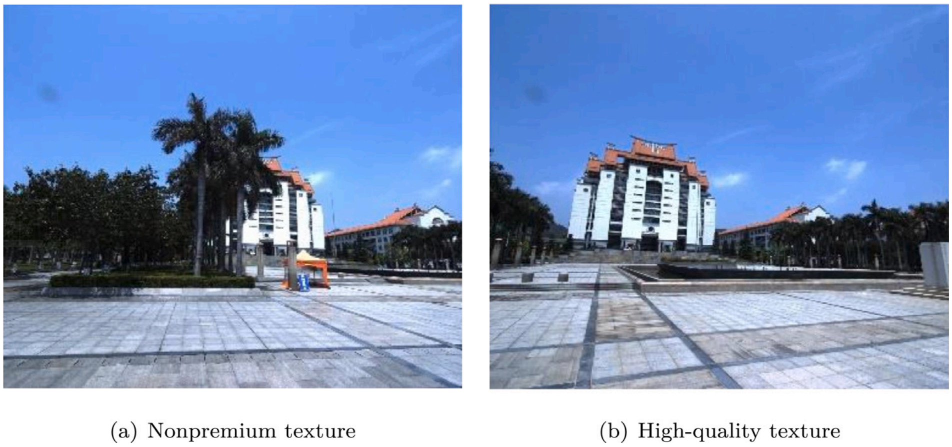
For texture mapping of the 3D mesh data generated by point cloud triangulation, it is first necessary to filter out high-quality texture images from the photos taken by the MLS system. The texture image candidates are shown in Fig. 20 (we use the manual method to select high-quality candidate texture images from MLS system, and the selected images are usually images where buildings are not occluded). The building in Fig. 20(a) is occluded, and the front of the building in Fig. 20(b) is unobstructed and has less distortion, which can be selected as a high-quality texture. The texture coordinates are then calculated for each vertex through the projection texture mapping method, and the selected texture image is mapped to the mesh to obtain a reconstructed 3D mesh. The reconstructed 3D mesh is merged with the 3D mesh of the reserved area, and finally, a high-precision, large-scale, building 3D model is generated.

Fig. 21 shows a comparison of the model generation results, which reveals that under the guidance of the quality score by our proposed MS3DQE-Net, the holes on the building surface of the model are repaired, the accuracy of the local area of the model is greatly improved, and the distorted building structure under the eaves is improved to a certain extent. This finding shows that the 3D model generation method based on multi-source data fusion ensures the high integrity and large scale of the model but also ensures the high integrity and large scale of the model. The local accuracy of the model is greatly improved, and the details of the model are increased.

Note that, first, there are holes on the face of the original 3D mesh model, due to a lack of triangular face structure and texture. Second, our point cloud fusion algorithm provides supplementary points for the holes, and the fusion point cloud meshing solves the problem of a lack of a triangular face structure. Third, the images obtained by the MLS system provide new textures for the generated 3D mesh, which compensates for the missing texture.

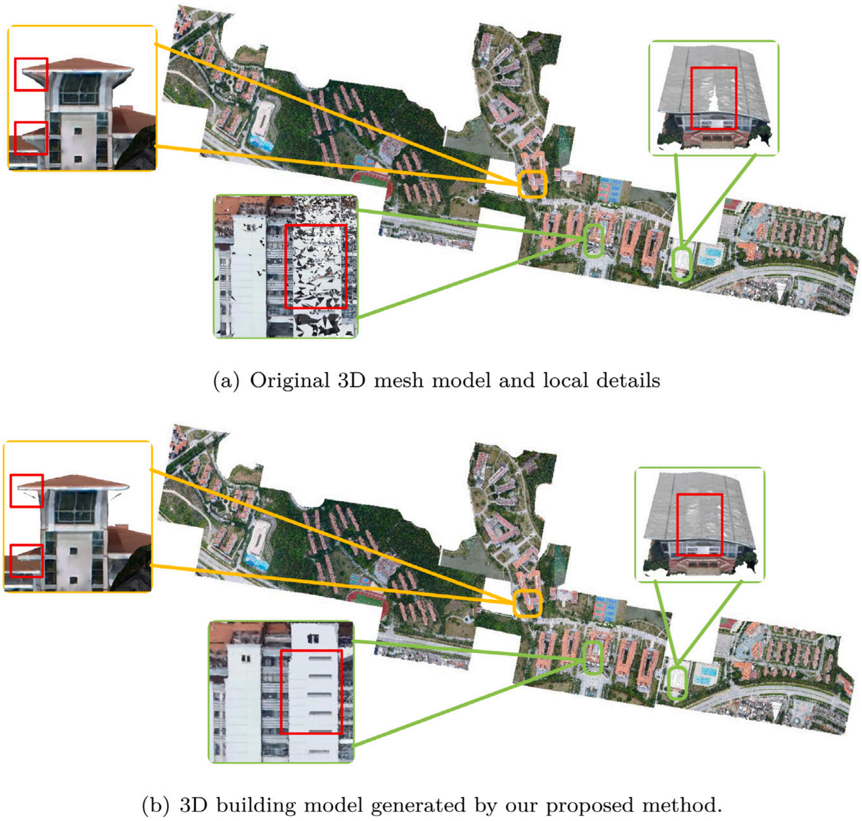
6.6. Discussion

3D mesh data are a representation that has richer information than 3D point cloud data. In addition to rich information such as points,


(a) Nonpremium texture

(b) High-quality texture

Fig. 20. Candidate texture image captured by the MLS system.


(a) Original 3D mesh model and local details

(b) 3D building model generated by our proposed method.

Fig. 21. Comparison between the original 3D model and the repaired 3D model.

faces, and textures, 3D mesh data also have surface neighborhood relationships. Our proposed MS3DQE-Net uses the face neighborhood structure to extract the face-local structure features so that the fusion features of the 3D mesh data have a better representation. Therefore, the classification performance of MS3DQE-Net is better than that of point cloud-based classification methods, e.g. PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), PointCNN (Li et al., 2018a), which in turn leads to better multi-source 3D data quality evaluation performance.

The proposed MS3DQE-Net integrates the 3D mesh face center point feature with the LiDAR point cloud features to form better point feature descriptions, which can better guide the classification, thereby improving the accuracy of the classification and the performance of multi-source 3D data quality evaluation.

In addition, the proposed quality evaluation network of multi-source 3D data and 3D building model generation methods have limitations. MS3DQE-Net only classifies 3D data into two categories, good and bad, in the model quality evaluation, and the multi-source 3D data with bad quality evaluation uniformly employs the same fusion scheme. However, areas classified as bad in the original 3D model of the scene may have different characteristics, and using the same fusion scheme may yield poor results. The selection of texture images for texture mapping still relies on manual work, and it is necessary to explore automated methods to automatically reduce the difference between old textures and new textures.

7. Conclusion

In this paper, we propose a novel 3D building model generation method based on multi-source 3D data fusion. Our proposed MS3DQE-Net creatively integrates the characteristics of the 3D mesh and 3D LiDAR point cloud to evaluate the quality of the model, and the final output quality evaluation results guide the generation of the 3D building model. The MS3DQE-Net integrates the global and local features of the 3D mesh, which are fused with the features of the 3D point cloud, to obtain a better quality evaluation of the local multi-source 3D data fusion of the 3D building model. Experimental results show that MS3DQE-Net is more effective than certain representative methods. To train MS3DQE-Net, to the best of our knowledge, we are the first researchers to propose a multi-source 3D dataset (paired matching of 3D point cloud patch and 3D mesh patch dataset) for 3D data classification, which is collected from a real university campus scene and contains textured 3D mesh data and corresponding region 3D point cloud data. In future work, we plan to explore not only a larger, scene-scale, 3D building model generation based on the fusion of multi-source 3D data but also a more comprehensive, multi-source 3D data quality evaluation and fusion scheme.

CRediT authorship contribution statement

Weiquan Liu: Conceptualization, Methodology, Writing – original draft, Funding acquisition. **Yu Zang:** Conceptualization, Supervision, Writing – review & editing, Resources, Funding acquisition. **Zhangyue Xiong:** Conceptualization, Methodology, Software, Writing – original draft. **Xuesheng Bian:** Methodology, Software, Writing – original draft. **Chenglu Wen:** Methodology, Writing – review & editing, Funding acquisition. **Xiaolei Lu:** Conceptualization, Writing – review & editing. **Cheng Wang:** Conceptualization, Methodology, Resources, Supervision. **José Marcato Junior:** Methodology, Writing – review & editing. **Wesley Nunes Gonçalves:** Methodology, Writing – review & editing. **Jonathan Li:** Supervision, Resources.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61971363, 62171393, 41871380, 41971424); the China Postdoctoral Science Foundation (No. 2021M690094); the FuXiaQuan National Independent Innovation Demonstration Zone Collaborative Innovation Platform, China (No. 3502ZCQXT2021003); the National Key R and D Program of China (No. 2021YFF0704600); the China Fundamental Research Funds for the Central Universities (No. 20720210074); and the open fund of PDL, China (No. 20215250113).

References

Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Amer. Statist. 46 (3), 175–185.

Bódis-Szomorú, A., Riemenschneider, H., Van Gool, L., 2016. Efficient volumetric fusion of airborne and street-side data for urban reconstruction. In: Proceedings of the International Conference on Pattern Recognition. ICPR, pp. 3204–3209.

Brell, M., Segl, K., Guanter, L., Bookhagen, B., 2019. 3D hyperspectral point cloud generation: Fusing airborne laser scanning and hyperspectral imaging sensors for improved object-based information extraction. ISPRS J. Photogramm. Remote Sens. 149, 200–214.

Chen, Z., Deng, L., Luo, Y., Li, D., Junior, J.M., Gonçalves, W.N., Nurunnabi, A.A.M., Li, J., Wang, C., Li, D., 2022. Road extraction in remote sensing data: A survey. Int. J. Appl. Earth Obs. Geoinf. 112, 102833.

Chen, Z., Li, D., Fan, W., Guan, H., Wang, C., Li, J., 2021a. Self-attention in reconstruction bias U-Net for semantic segmentation of building rooftops in optical remote sensing images. Remote Sens. 13 (13), 2524.

Chen, Z., Wang, C., Li, J., Fan, W., Du, J., Zhong, B., 2021b. Adaboost-like end-to-end multiple lightweight U-Nets for road extraction from optical remote sensing images. Int. J. Appl. Earth Obs. Geoinf. 100, 102341.

Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H., 2021c. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 14124–14133.

De Boer, P.-T., Kroese, D.P., Mannor, S., Rubinstein, R.Y., 2005. A tutorial on the cross-entropy method. Ann. Oper. Res. 134 (1), 19–67.

Eldar, Y., Lindenbaum, M., Porat, M., Zeevi, Y.Y., 1997. The farthest point strategy for progressive image sampling. IEEE Trans. Image Process. 6 (9), 1305–1315.

Fan, S., Dong, Q., Zhu, F., Lv, Y., Ye, P., Wang, F.-Y., 2021. Scf-Net: Learning spatial contextual features for large-scale point cloud segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 14504–14513.

Feng, Y., Feng, Y., You, H., Zhao, X., Gao, Y., 2019. Meshnet: Mesh neural network for 3D shape representation. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. AAAI, pp. 8279–8286.

Gopi, M., Krishnan, S., Silva, C.T., 2000. Surface reconstruction based on lower dimensional localized delaunay triangulation. In: Computer Graphics Forum, Vol. 19. Wiley Online Library, pp. 467–478.

Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J., 2013. Rotational projection statistics for 3D local surface description and object recognition. Int. J. Comput. Vis. 105 (1), 63–86.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep learning for 3D point clouds: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (12), 4338–4364.

Hubeli, A., Gross, M., 2001. Multiresolution feature extraction for unstructured meshes. In: Proceedings of IEEE Visualization. pp. 287–294.

Kazhdan, M., Funkhouser, T., Rusinkiewicz, S., 2003. Rotation invariant spherical harmonic representation of 3 D shape descriptors. In: Symposium on Geometry Processing, Vol. 6. pp. 156–164.

Kokkinos, I., Bronstein, M.M., Litman, R., Bronstein, A.M., 2012. Intrinsic shape context descriptors for deformable shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 159–166.

Ladicky, L., Saurer, O., Jeong, S., Maninchedda, F., Pollefeys, M., 2017. From point clouds to mesh using regression. In: Proceedings of the IEEE International Conference on Computer Vision. ICCV, pp. 3893–3902.

Lai, B., Liu, W., Wang, C., Fan, X., Lin, Y., Bian, X., Wu, S., Cheng, M., Li, J., 2022. 2D3D-Mvpnet: Learning cross-domain feature descriptors for 2D-3D matching based on multi-view projections of point clouds. Appl. Intell. 1–16.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018a. PointCNN: Convolution on X-transformed points. Adv. Neural Inf. Process. Syst. 31, 820–830.

Li, X., Li, R., Zhu, L., Fu, C.-W., Heng, P.-A., 2020. DNF-Net: A deep normal filtering network for mesh denoising. IEEE Trans. Vis. Comput. Graphics 27 (10), 4060–4072.

Li, W., Wang, C., Zai, D., Huang, P., Liu, W., Wen, C., Li, J., 2018b. A volumetric fusing method for TLS and SFM point clouds. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (9), 3349–3357.

Lien, S.-l., Kajiya, J.T., 1984. A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra. IEEE Comput. Graph. Appl. 4 (10), 35–42.

Liu, W., Guo, H., Zhang, W., Zang, Y., Wang, C., Li, J., 2022. Toposeg: Topology-aware segmentation for point clouds. In: IJCAI. pp. 1201–1208.

Liu, W., Lai, B., Wang, C., Bian, X., Yang, W., Xia, Y., Lin, X., Lai, S.-H., Weng, D., Li, J., 2020a. Learning to match 2D images and 3D LiDAR point clouds for outdoor augmented reality. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops. VRW, IEEE, pp. 654–655.

Liu, W., Lai, B., Wang, C., Cai, G., Su, Y., Bian, X., Li, Y., Chen, S., Li, J., 2020b. Ground camera image and large-scale 3-D image-based point cloud registration based on learning domain invariant feature descriptors. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 997–1009.

Liu, W., Wang, C., Chen, S., Bian, X., Lai, B., Shen, X., Cheng, M., Lai, S.-H., Weng, D., Li, J., 2021. Y-Net: Learning domain robust feature representation for ground camera image and large-scale image-based point cloud registration. Inform. Sci. 581, 655–677.

Méndez-Barroso, L.A., Zárate-Valdez, J.L., Robles-Morúa, A., 2018. Estimation of hydromorphological attributes of a small forested catchment by applying the structure from motion (SFM) approach. Int. J. Appl. Earth Obs. Geoinf. 69, 186–197.

Mohsan, S.A.H., Khan, M.A., Noor, F., Ullah, I., Alsharif, M.H., 2022. Towards the unmanned aerial vehicles (UAVs): A comprehensive review. Drones 6 (6), 147.

Moran, D., Koslowsky, H., Kasten, Y., Maron, H., Galun, M., Basri, R., 2021. Deep permutation equivariant structure from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 5976–5986.

Nan, L., Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds. In: Proceedings of the IEEE International Conference on Computer Vision. ICCV, pp. 2353–2361.

Neto, J.A.B., Lima, J.L., Pereira, A.I., Costa, P., 2021. Low-cost 3D LiDAR-based scanning system for small objects. In: Proceedings of the IEEE International Conference on Industrial Technology, Vol. 1. ICIT, IEEE, pp. 907–912.

Nex, F., Remondino, F., 2014. UAV for 3D mapping applications: A review. Appl. Geomat. 6 (1), 1–15.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 30.

Remondino, F., Barazzetti, L., Nex, F., Scaioni, M., Sarazzi, D., 2011. UAV photogrammetry for mapping and 3D modeling–current status and future perspectives. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 38 (1).

Rocchini, C., Cignoni, P., Montani, C., Pingi, P., Scopigno, R., 2001. A low cost 3D scanner based on structured light. In: Computer Graphics Forum, Vol. 20. Wiley Online Library, pp. 299–308.

Roshandel, S., Liu, W., Wang, C., Li, J., 2021. 3D ocean water wave surface analysis on airborne LiDAR bathymetric point clouds. Remote Sens. 13 (19), 3918.

Roshandel, S., Liu, W., Wang, C., Li, J., 2022. Semantic segmentation of coastal zone on airborne LiDAR bathymetry point clouds. IEEE Geosci. Remote Sens. Lett. 19, 1–5.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: Proceedings of the IEEE International Conference on Robotics and Automation. ICRA, pp. 3212–3217.

Segal, M., Korobkin, C., Van Widenfelt, R., Foran, J., Haeberli, P., 1992. Fast shadows and lighting effects using texture mapping. In: Proceedings of the Conference on Computer Graphics and Interactive Techniques. pp. 249–252.

Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 519–528.

Shen, Y., Feng, C., Yang, Y., Tian, D., 2018. Mining point cloud local structures by kernel correlation and graph pooling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 4548–4557.

Su, Y., Liu, W., Yuan, Z., Cheng, M., Zhang, Z., Shen, X., Wang, C., 2022. Dla-Net: Learning dual local attention features for semantic segmentation of large-scale building facade point clouds. Pattern Recognit. 123, 108372.

Tombari, F., Salti, S., Stefano, L.Di., 2010. Unique signatures of histograms for local surface description. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 356–369.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.-G., 2018. Pixel2mesh: Generating 3D mesh models from single RGB images. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 52–67.

Wu, D., Liu, W., Fang, B., Chen, L., Zang, Y., Zhao, L., Wang, S., Wang, C., Marcato, J., Li, J., 2022. Intracity temperature estimation by physics informed neural network using modeled forcing meteorology and multispectral satellite imagery. IEEE Trans. Geosci. Remote Sens. 60, 1–15.

Wu, W., Qi, Z., Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3D point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 9621–9630.

Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S., 2020. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 5589–5598.

Yang, Y., Koppanyi, Z., Toth, C.K., 2017. Stereo image point cloud and LiDAR point cloud fusion for the 3D street mapping. In: Annual Conference of the American Society of Photogrammtery and Remote Sensing. IGTF 2017–Imaging & Geospatial Technology Forum 2017, Baltimore, Maryland, March, pp. 11–17.

Ying, S., Peng, J., Du, S., Qiao, H., 2009. A scale stretch method based on ICP for 3D data registration. IEEE Trans. Autom. Sci. Eng. 6 (3), 559–565.

Yu, D., He, L., Ye, F., Jiang, L., Zhang, C., Fang, Z., Liang, Z., 2022. Unsupervised ground filtering of airborne-based 3D meshes using a robust cloth simulation. Int. J. Appl. Earth Obs. Geoinf. 111, 102830.

Zhang, C., Chen, T., 2001. Efficient feature extraction for 2D/3D objects in mesh representation. In: Proceedings of the International Conference on Image Processing, Vol. 3. ICIP, pp. 935–938.

Zhang, Z., Jimack, P.K., Wang, H., 2021. Meshingnet3D: Efficient generation of adapted tetrahedral meshes for computational mechanics. Adv. Eng. Softw. 157, 103021.