

Building Contour Extraction from Fused LiDAR and Photogrammetric Point Clouds Using PointNet++

Jianghong Zhao, Mengtian Cao, JiaYang

School of Geomatics and Urban Spatial Information, Beijing University of Civil Engineering and Architecture, Beijing 102616, China - zhaojiangh@bucea.edu.cn, cmt05142024@163.com, yangjia@huuc.edu.cn

Keywords: LiDAR Point Cloud, Photogrammetric Point Cloud, Building Contour Extraction, Multimodal Data Fusion, PointNet++.

Abstract

Accurate building contour extraction is critical for urban modeling but remains challenging due to limitations in single-source point clouds. LiDAR data suffers from sparsity and sensitivity to surface reflectance, while photogrammetric point clouds exhibit noise under occlusion and lighting variations. To overcome these constraints, we propose an end-to-end framework combining multimodal 3D fusion and deep geometric co-optimization. First, LiDAR and photogrammetric point clouds are fused through ICP registration, avoiding 2D-3D misalignment. Building points are then segmented using PointNet++. A novel Z-axis threshold projection is applied during projection, eliminating rooftop interference by constraining projections to structural walls. Initial contours extracted via Alpha-shapes undergo adaptive regularization: 1) Douglas-Peucker simplification, 2) angle-constrained vector optimization rectifying non-orthogonal corners. Validated on Ming and Qing heritage structures, our method achieves 3.7% area error (vs. 17.8% for CloudCompare) and 2.6% perimeter error. This represents the first unified pipeline combining 3D-3D data fusion with deep learning and geometric regularization, offering a promising approach for automated building modeling in complex urban and heritage environments.

1. Introduction

3D reconstruction of urban buildings is a central task in digital city development, with building contour extraction being a key component. As urbanization accelerates, the demand for high-precision 3D models in urban management, planning, and design continues to grow. Consequently, point cloud contour extraction is increasingly recognized as a fundamental 3D modeling technique.

Light Detection and Ranging (LiDAR) offers high accuracy, active acquisition, and vegetation penetration capabilities, making it a valuable source of 3D spatial data. However, its relatively low point cloud density and sensitivity to weather conditions and surface reflectance can limit its effectiveness. In contrast, photogrammetric point clouds provide high-density data with rich texture and geometric details, making them well-suited for detailed urban modeling, though their quality is influenced by lighting, image clarity, and occlusion. Relying on a single data source often fails to meet the requirements for high-precision and full-coverage building outline extraction in complex urban environments. Fusing LiDAR with photogrammetric point clouds allows each to compensate for the other's limitations, enhancing spatial consistency and detail representation. This integration provides reliable data support for the accurate depiction of building structures. Compared to fusing 2D imagery with point clouds, the integration of two 3D data sources ensures better spatial alignment and is more suitable for geometric analysis. As a result, this approach offers distinct advantages for extracting accurate and complete 3D building outlines and has become a key research focus in urban modeling.

Moreover, advances in computer vision and AI, particularly deep learning, have significantly improved point cloud processing efficiency and accuracy. These techniques enable

more intelligent and automated extraction of building contours, further advancing the field.

This paper proposes a building contour extraction method fusing LiDAR and photogrammetric point clouds. By integrating deep learning and contour optimization strategies, the proposed method achieves high-precision contour extraction. Its effectiveness and practicality are demonstrated through validation on real-world data.

2. Related work

Building outline extraction is a key technology in fields such as Geographic Information Systems (GIS), urban planning, and disaster monitoring. With the rapid development of remote sensing and computer vision technologies, significant advancements have been made in data acquisition and processing (Li et al., 2015; Wang and Tan, 2020; Wang and Kim, 2019). The current methods for building outline extraction can be broadly categorized into three types based on different data sources: high-resolution remote sensing imagery, point cloud data, and multi-source remote sensing data fusion.

2.1 Building Outline Extraction from High-Resolution Remote Sensing Imagery

The first category involves extracting building outlines from high-resolution remote sensing imagery. For example, integrating vector data can optimize outlines, better reflecting building details and improving match with reality (Tang et al., 2023). However, remote sensing imagery can sometimes suffer from issues such as shadows, occlusions, and poor contrast, as well as limitations in estimating accurate height information. Additionally, with the continuous evolution of computer vision technologies, many researchers have used Convolutional Neural Networks (CNNs) for automatic building outline extraction. For instance, the PRCUnet deep learning model, based on the U-Net

architecture, has been developed for building extraction and outline optimization from high-resolution remote sensing imagery (Xu et al., 2021). The Attention-based Feature Pyramid U-Net (AFP-Net) focuses on different building structures in high-resolution imagery, enabling efficient extraction of building outlines (Yu et al., 2022).

2.2 Building Outline Extraction from Point Cloud Data

The second category involves building outline extraction based on point cloud data. Various methods are employed for extracting building point clouds. Some use filtering techniques that leverage the differences in features such as height and density between buildings and their surrounding environment to filter out non-building point clouds. For example, LiDAR data is used to acquire side-view point clouds of buildings, and a method based on point cloud density differences after projection is proposed for building outline extraction. Statistical Outlier Removal (SOR) filtering and adjacent-angle criteria are used to remove non-building points (Chen et al., 2023). Solely image-based 3D change detection is challenging, while traditional point cloud methods suffer from low automation and accuracy. As a result, some researchers have introduced deep learning-based point cloud semantic segmentation methods, such as RandLA-Net, to improve the accuracy and automation of change detection (Meng et al., 2022). After obtaining building point clouds, the Alpha-shapes algorithm is often used to extract building outlines. This method is robust and easy to implement, but the extracted outlines are susceptible to noise. To overcome these drawbacks, researchers have continuously improved the process, proposing modifications to the Alpha-shapes algorithm and using techniques such as the Douglas-Peucker algorithm and orthogonal optimization to correct irregularities, inaccuracies, and lack of smoothness in the building outlines (Saalfeld, 1999; Gardiner et al., 2018).

2.3 Building Outline Extraction via Multi-Source Remote Sensing Data Fusion

The third category is characterized by the fusion of multi-source remote sensing data to achieve complementary advantages. For example, building extraction can be performed by combining point clouds and imagery. This method utilizes attention mechanisms to drive the joint extraction of buildings, leveraging channel attention mechanisms to enhance effective semantic features in the channels and spatial attention mechanisms to enhance effective semantic features in spatial locations, thereby improving the precision of building extraction (Di, 2022). By extracting building outlines from airborne point clouds and aerial imagery, fitting line segments, and determining building corner points, precise registration and fusion of the two data types can be achieved. Furthermore, spectral information is used to cluster and separate land cover, combined with elevation data to accurately extract building outlines (Cheng et al., 2016). Some researchers have proposed an extraction method based on the fusion of LiDAR point clouds and orthophoto imagery using an improved genetic algorithm (IGA), proving its applicability in practical production (Lai et al., 2024). Addressing issues such as point cloud sparsity, hyperspectral variability, urban object diversity, environmental complexity, and data inconsistency, methods have been proposed for automated building detection and regularization using multi-source data, extracting and normalizing building outlines by combining point cloud and orthophoto features (Gilani et al., 2016).

Despite growing interest, most fusion methods face limitations: low geometric resolution, poor scalability, or reliance on heuristic rules. Some methods fuse 2D imagery and 3D point clouds, which introduces challenges in co-registration and inconsistent semantic labeling. Others are constrained by the sparsity or noise inherent in a single data modality. Moreover, many approaches lack robustness when applied to complex roof structures or heritage architecture, which are prevalent in our study area. These limitations highlight the need for a more unified, 3D-based, deep learning-enabled fusion strategy. Therefore, this study takes into account the respective strengths and limitations of photogrammetric and LiDAR point clouds, aiming to integrate them into a more comprehensive and accurate multi-source dataset. Through data fusion, the shortcomings of individual sources can be mitigated, enabling complementary advantages. Furthermore, by incorporating deep learning techniques, the overall quality and reliability of the fused data can be significantly enhanced.

3. Data Preparation

3.1 Study Area

Most existing studies, both domestic and international, have focused on building outline extraction in urban areas. Considering the vast geographical diversity and architectural variety across China, this study adopts a unique perspective by selecting a rural area as the research focus, with the main structures being ancient buildings. This choice not only highlights the distinctiveness of the dataset but also emphasizes the cultural significance of ancient architecture as heritage, making the study both meaningful and practically representative.

The study area is located in Lidukou Village, Jia County, Pingdingshan City, Henan Province, China. It covers approximately 0.6 square kilometers, with a length of about 780 meters and a width of approximately 770 meters, as shown in Figure 1. The buildings in this area are primarily traditional structures, including those from the Ming Dynasty, the Republic of China period, and the Qing Dynasty. The village features a variety of elements such as buildings, rivers, vegetation, and transportation routes.

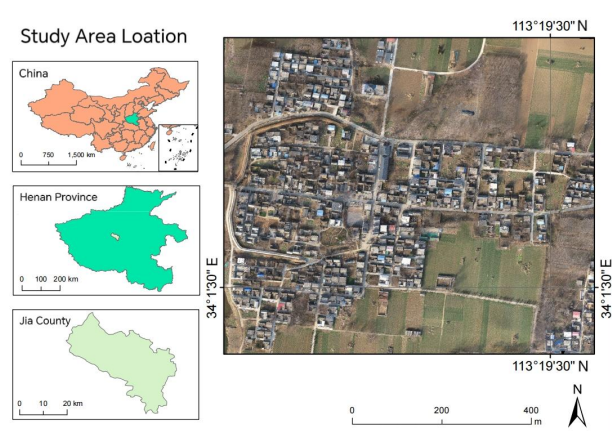


Figure 1. Remote sensing image of the study area

3.2 Multimodal Data Acquisition

3.2.1 Photogrammetric Point Cloud Acquisition: Unmanned aerial vehicle (UAV) aerial photography technology was first employed to collect high-resolution images, with

ground control points (GCPs) deployed throughout the study area in combination with a global positioning system. A DJI Phantom 4 RTK drone was used for image acquisition, with flight parameters listed in Table 1. A total of 1,677 images were captured using real-time kinematic (RTK) positioning.

Three GCPs and two check points were evenly distributed for accurate georeferencing. The measurements were conducted using a Qianxun RS6 GNSS receiver, which achieved a horizontal accuracy of $\pm(8+1 \times 10^{-6} D)$ mm and a vertical accuracy of $\pm(15+1 \times 10^{-6} D)$ mm under CORS mode.

Flight Altitude (m)	Forward Overlap (%)	Side Overlap (%)	Gimbal Angle(°)
90	70	80	-60°

Table 1. Flight Parameters of the Phantom 4 RTK

The dense point cloud reconstructed using photogrammetry was evaluated for accuracy through a comparative analysis of checkpoint coordinates. The comparison between the model coordinates of all ground control points (GCPs) and checkpoints and their corresponding ground truth values shows that both the horizontal position errors (X/Y directions) and elevation errors (Z direction) are within ± 3 cm. This level of error strongly indicates that the reconstructed model possesses high geometric accuracy, meeting the precision requirements for building contour extraction.

3.2.2 LiDAR Point Cloud Acquisition: The second part involves the collection of LiDAR point cloud data using an M300 equipped with a DJI Zenmuse L1 sensor for flight-line scanning. A total of two flight missions were conducted, with the specific flight parameters listed in Table 2. The collected data were imported into DJI Terra for point cloud reconstruction, with the point cloud density set to 100% and an effective point cloud range of 300 meters.

Flight Altitude (m)	Flight Speed (m/s)	Pulse Repetition Rate (kHz)	Scan Rate (kHz)
80	6m/s	240	720

Table 2. Flight Parameters of the Matrice 300 (M300)

3.2.3 Data Fusion: In this study, the Iterative Closest Point (ICP) algorithm embedded in CloudCompare was used to register the photogrammetric point cloud with the LiDAR point cloud, thereby integrating them into multimodal data. This algorithm is a classical method for point cloud registration, capable of spatially aligning two point clouds. The registration process consists of two stages: an initial coarse registration and a fine registration. The final output is a transformation matrix that ensures consistency between coordinate systems.

4. Methodology

The objective of this study is to fuse LiDAR point clouds and photogrammetric point clouds to form multimodal data, and to perform building point cloud classification using the deep learning neural network PointNet++. During planar projection, a Z-axis threshold method reduces errors from point cloud misclassification and complex rooftops. Subsequently, the Alpha-shapes algorithm extracts initial contours, followed by Douglas-Peucker key point extraction. Finally, the contour lines

formed by these key points are regularized to accurately reflect the true geometric shapes of the buildings within a certain precision. The experimental procedure is illustrated in figure 2.

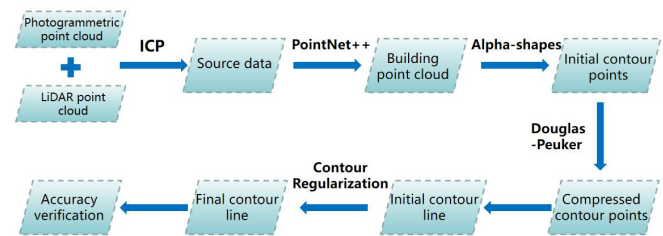


Figure 2. Experimental Procedure

4.1 Building Point Cloud Extraction Based on PointNet++

4.1.1 PointNet++ Network Architecture Design:

PointNet++ is a deep learning algorithm for point cloud processing. A schematic diagram of the PointNet++ network is shown in Figure 3. It extracts local features through hierarchical sampling and multilayer perceptrons (MLPs), and then progressively aggregates these features using set abstraction operations to generate a global feature representation of the point cloud. This approach significantly enhances the network's ability to understand 3D spatial structures, enabling PointNet++ to perform effectively in tasks such as 3D shape analysis, object classification, and semantic segmentation.

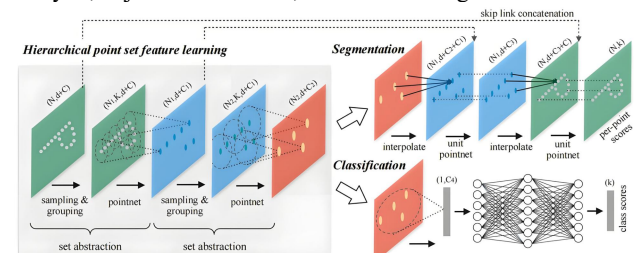


Figure 3. Architecture of the PointNet++ network

A multi-scale grouping strategy (Figure 4) enriches feature extraction with three scale groups.

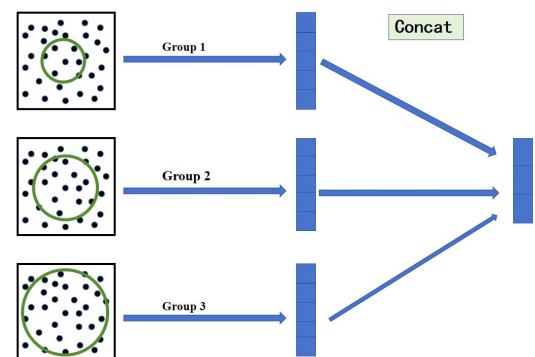


Figure 4. Multi-Scale Grouping

This approach enhances the richness of point cloud features. In this network architecture, the point cloud data is first divided into multiple sets of points based on predefined scale parameters. These scale parameters, which typically include both smaller local scales and larger global scales, can be adjusted according to specific requirements. For each scale

group, a PointNet layer is used for feature extraction. This layer is capable of handling the unordered nature of point sets and extracting local features for each point. Next, features from different scales are fused together. The fusion can be performed through simple concatenation or more advanced methods such as attention mechanisms. The fused features are then passed as input to the next layer, and the above process is repeated until the desired network depth is reached.

4.1.2 PointNet++ Model Parameter Settings: PointNet++ performance is highly sensitive to network parameter configuration. Therefore, careful tuning is required during experiments to achieve optimal segmentation or classification results. In this study, the model pointnet2_sem_seg_msg is adopted, and the specific parameter settings are shown in Table 3.

Batch Size	Block Size	Number Points	Learning Rate	Stride	Optimizer
64	1*1	4096	0.001	0.6	Adam

Table 3. Parameters of the PointNet++ network

4.1.3 Evaluation Metrics: To evaluate the performance of the PointNet++ network model during the experiment, Intersection over Union (IoU) is used to calculate the ratio of the intersection to the union between the predicted segmentation region and the ground truth. In addition, Point Accuracy (Point Acc) is also used as an evaluation metric. In this experiment, the evaluation of point cloud semantic segmentation accuracy categorizes the segmentation results into four cases: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Here, “positive” and “negative” indicate whether each point belongs to a certain class, while “true” and “false” refer to whether the predicted class of the point matches the ground truth label. Therefore, a higher proportion of true results and fewer misclassifications (false positives and false negatives) generally indicate better segmentation accuracy of the network model. The specific formula for calculating the Intersection over Union (IoU) is shown in Equation 1.

$$IOU = \frac{TP}{TP + FP + FN} \quad (1)$$

Among the two accuracy evaluation metrics mentioned above, the IoU value is used as the primary evaluation metric for the experiment, while the Point Accuracy (Point Acc) metric is presented as a reference to indicate the overall accuracy.

4.1.4 Data Augmentation: Deep learning models typically require large datasets to achieve good generalization. To address data scarcity, we employ six data augmentation strategies in PointNet++ to expand the original point cloud dataset and enhance model robustness under various scenarios, as illustrated in Figure 5. These include: (1) random shuffling of point clouds and their labels; (2) small random rotations of points and normals along the XYZ axes (Fig. 5a); (3) random scaling (Fig. 5b); (4) random translation by adding a displacement vector to each point (Fig. 5c); (5) adding noise for local perturbation; and (6) random point dropout based on a predefined dropout rate during each training iteration (Fig. 5d).

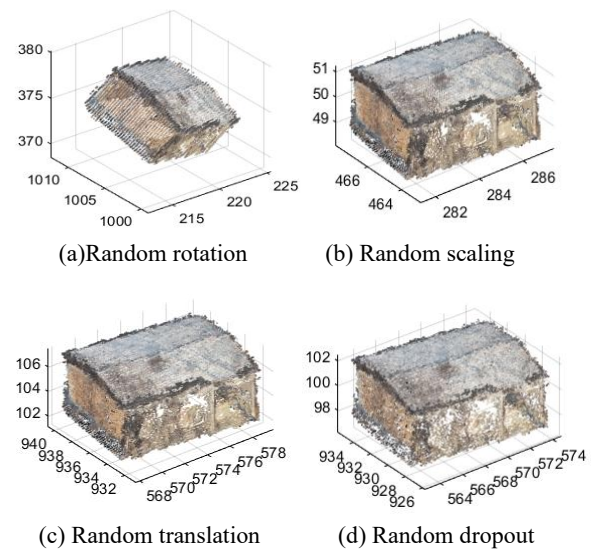


Figure 5. Illustration of data augmentation methods: (a) Random rotation, (b) Random scaling, (c) Random translation, (d) Random dropout .

4.2 Building Contour Extraction

4.2.1 Z-Axis Threshold Method: The building point cloud data obtained through PointNet++ is three-dimensional. To more accurately extract the edge points of each building wall, it is necessary to project the data onto the XOY plane to obtain the building's planar projection points. However, due to the complexity of building roof structures and the randomness of point cloud acquisition, directly ignoring the Z-axis for projecting the roof could result in discrepancies between the obtained area and the actual situation, leading to lower accuracy.

Therefore, a Z-axis threshold is set during the projection. This threshold is established above the ground but below the roof height range, ensuring that the projection plane is within the main structure of the building, rather than directly using the roof as the projection reference. The Z-axis threshold is set above a certain height from the ground to avoid mistakenly identifying non-building points on the ground as building points when using PointNet++ to extract the building point cloud. Such missegmentation could result in the projection including non-building ground points, thereby affecting the accuracy of the analysis. At the same time, the threshold is set below the roof height to account for the possibility that some buildings' roof structures may extend beyond the main body. If the roof is used as the projection reference, the projected area could exceed the actual area.

Considering these factors, this experiment proposes setting a Z-axis threshold, ensuring that the projection range lies within the main structure of the building. This ensures that the projection result more closely aligns with the actual geometry and layout of the building, and the experiment demonstrates that this method not only effectively eliminates the interference of non-building points but also avoids the influence of complex roof structures on the projection surface, providing strong support for subsequent building structure analysis.

4.2.2 Contour Extraction Based on the Alpha-Shapes Algorithm: This experiment adopts the Alpha-Shapes algorithm to extract building contour lines. The Alpha-Shapes

algorithm is a geometric method used to extract boundary outlines from scattered point sets. Its essence lies in the "empty circle" principle to determine whether the connection between two points constitutes a boundary segment.

Consider a point set P containing n points, these n points can potentially form $n \times (n-1)$ directed line segments. The next step is to determine which of these segments constitute the boundary of the point set.

For any two points P_1 and P_2 within the point set P , a circle with radius α is drawn through them. If there are no other points inside this circle, then P_1 and P_2 are considered boundary points, and the line segment P_1P_2 is identified as a boundary edge. Given the coordinates of point P_1 as (x_1, y_1) and point P_2 as (x_2, y_2) , the center P_3 of the circle passing through these two points can be computed using Equation 2. Once the center is obtained, whether other points fall within the circle is determined by comparing the distance from each point to the circle's center with the radius α . Traverse all possible points and retain the boundary segments that meet the conditions. All retained boundary segments together form the final Alpha-Shapes contour.

$$\begin{cases} x_3 = x_1 + \frac{1}{2}(x_2 - x_1) + L(y_2 - y_1) \\ y_3 = y_1 + \frac{1}{2}(y_2 - y_1) + L(x_1 - x_2) \end{cases} \quad (2)$$

$$L = \frac{\alpha^2}{\sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2} - \frac{1}{4}}$$

where x_1, y_1 = Coordinates of P_1
 x_2, y_2 = Coordinates of P_2
 x_3, y_3 = Coordinates of P_3
 L = Factor for computing the circle center coordinates
 α = Parameter used to control the radius of the circle

4.3 Contour Line Regularization

Due to the inherent randomness of point cloud data acquisition and the uneven reflection and scattering of laser beams, the initial contour lines directly formed by connecting edge points often do not perfectly match the true outlines of buildings. Typically, these initial contours are rough, with numerous inflection points, and exhibit a "zigzag" shape (as shown in Fig. 6). Such contours cannot be directly applied to 3D building reconstruction or architectural mapping. Therefore, regularization smooths contours to better reflect true boundaries.

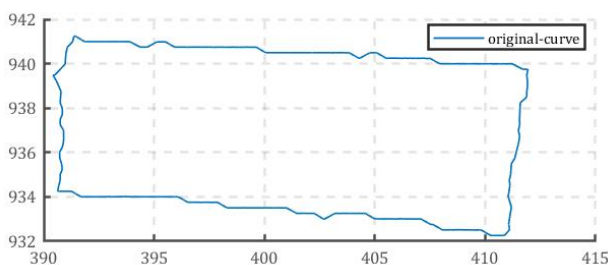


Figure 6. Initial Contour Line

4.3.1 Douglas-Peucker Keypoint Detection: The Douglas-Peucker algorithm is used to compress large amounts of

redundant graphical data points and extract only the essential ones. As a classic algorithm for line feature simplification, it simplifies curves into polylines by sampling, while preserving the skeletal structure of the original geometric shape to a certain extent.

First, the starting and ending points of the curve to be processed are connected to form a straight line. The perpendicular distances from all other points on the curve to this line are calculated, and the maximum distance value D_{max} is identified. A simplification threshold ε is set. If $D_{max} < \varepsilon$, all intermediate points on the curve are discarded, and the straight line segment is used as an approximation of the curve. If $D_{max} \geq \varepsilon$, the point with the maximum distance is used as a boundary to divide the curve into two parts. The above process is then recursively applied to each part until all points have been processed. When all curve points are processed, the segmented points are connected in sequence to form a polyline, which serves as an approximation of the original curve. The result of key point extraction using the Douglas-Peucker algorithm is shown in Figure 7.

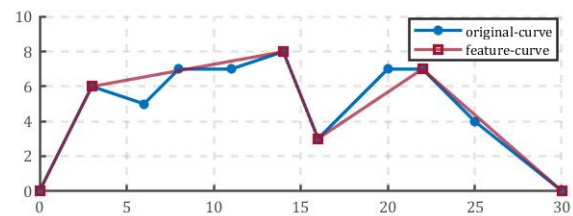


Figure 7. Key Point Extraction Using the Douglas-Peucker Algorithm

4.3.2 Optimization of Contour Key Points: To address the issue of adjacent edges not being perpendicular at building corners, key point optimization is required. The principle is illustrated in Figure 8, where the blue lines represent the original contour and the red lines represent the boundary after key point optimization. First, the longest edge direction is selected as the reference for optimization. Then, the angle between each pair of vectors is calculated. If the dot product of two vectors is zero, it indicates a right angle. If the dot product is non-zero, it is checked against a threshold. If the threshold is exceeded, the position of the key point needs to be adjusted to change the direction and magnitude of the vector such that the dot product becomes zero, thereby achieving perpendicularity.

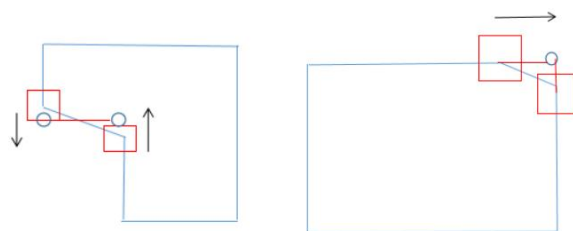


Figure 8. Key Point Optimization Principle

5. Experiments and Results

The experimental setup included a computer configured with an NVIDIA GRID RTX8000-24Q GPU and an Intel(R) Core(TM) i5-8300H CPU operating at 2.30 GHz. Software: A virtual deep learning environment on Windows OS. The Python version used was 3.7, with PyTorch 1.7.0 (GPU version) and CUDA 10.1. All other libraries were compatible with this setup.

In this study, ContextCapture Center was used to reconstruct point clouds from aerial photogrammetric data and to generate accuracy reports. For LiDAR data, DJI Terra was employed to process the raw data into point clouds. Point cloud tiling and labeling were subsequently performed using CloudCompare, a software platform that supports a variety of 3D point cloud processing algorithms.

5.1 Point Cloud Segmentation Based on PointNet++

5.1.1 Dataset Preparation and Labeling : Since the data acquisition platforms have an extended capture range, the merged point cloud obtained after registration is first trimmed to remove excess points outside the boundaries of the study area. Due to limitations in computational power, the entire point cloud cannot be directly used for training and testing. Therefore, the point cloud is divided into smaller, similarly sized blocks and split into training and testing datasets. In this experiment, the ratio of the training dataset to the testing dataset is set at 2:1. The partial area of the training dataset is shown in Figure 9, and the partial area of the testing dataset is shown in Figure 10.



Figure 9. Sample area from the training dataset

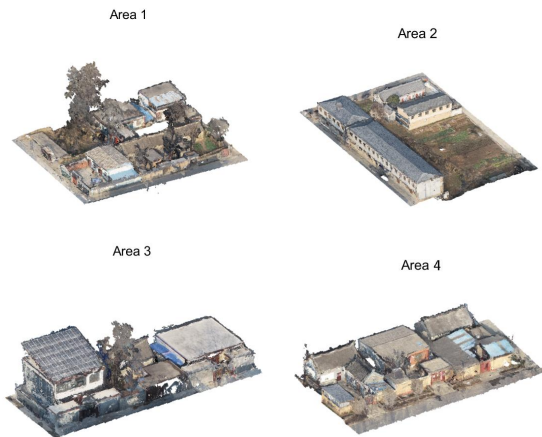


Figure 10. Sample area from the testing dataset

Each partitioned point cloud region was imported into CloudCompare for label creation. Within each region, building points and non-building points were assigned scalar values of 0 and 1, respectively. After labeling all regions, the data were saved separately according to the designated training and testing

areas, serving as input for subsequent supervised learning using the PointNet++ deep learning model.

5.1.2 PointNet++ Segmentation Results and Analysis: The segmentation results obtained after training and testing with the collected multi-source fused dataset using PointNet++ are shown in Figure 11.

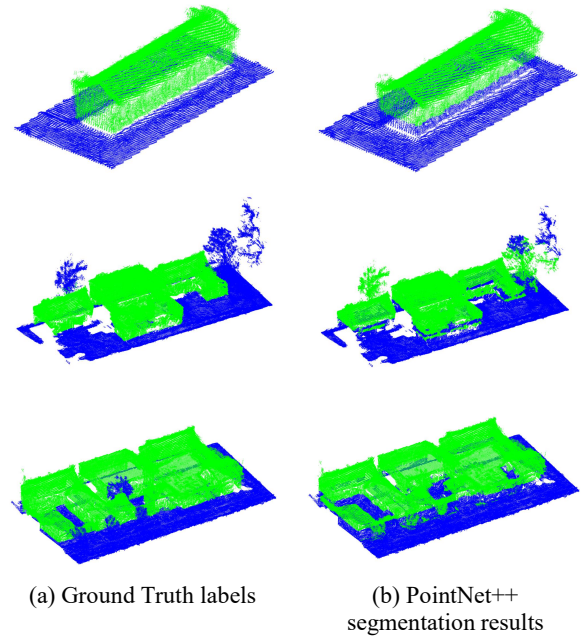


Figure 11. Comparison of Point Cloud Semantic Segmentation Results

The point cloud semantic segmentation accuracy of the model is shown in Table 4. Although there is a certain degree of confusion in the experimental area, the Intersection over Union (IoU) values remain above 0.8. This indicates that the accuracy generally meets the requirements. When evaluating accuracy, we should not only consider the IoU values but also assess them in relation to the specific requirements of the task. In this study, the primary objective is to extract point cloud data of buildings. Based on the segmentation results, this goal has been effectively achieved. Therefore, it can be concluded that the experimental results meet the expected requirements.

Area	IoU
Area_1	0.8890
Area_2	0.8846
Area_3	0.8109
Area_4	0.8123
Area_5	0.8222
Area_6	0.9391

Table 4. Accuracy Metrics for PointNet++ Segmentation

5.2 Building Contour Extraction Results

5.2.1 Extraction of Initial Contour Lines : The building point cloud extracted through the PointNet++ network is projected onto the XOY plane. This step aims to eliminate the height information along the Z-axis, thereby focusing on the

geometric shape of the building on the horizontal plane. The Alpha-shapes algorithm is then applied to process the projected point cloud to accurately extract the boundary points of the building, with the results shown in Figure 12.

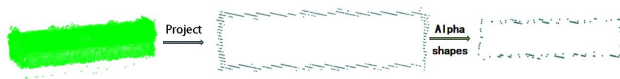


Figure 12. Contour Extraction Results

5.2.2 Contour Regularization: The result of extracting key points from the initial building contour using the Douglas-Peucker algorithm is shown in Figure 13. The blue points represent the initial boundary points obtained through the Alpha-shapes algorithm, which closely follow the actual building outlines and depict the geometric characteristics of the structures. The red points are the selected key points retained after filtering, providing a simplified yet representative outline of the building's geometry. As shown in the figure, the number of points is significantly reduced, and the spatial distribution of the key points is more reasonable, effectively capturing the main corners and changes in the building shape.

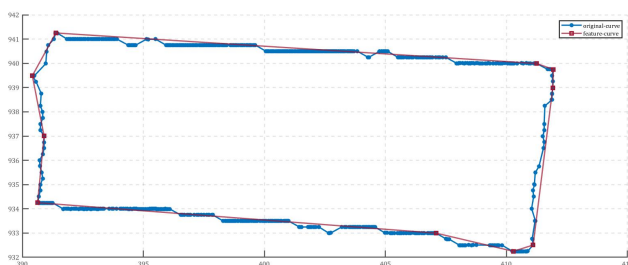


Figure 13. Key Point Extraction Result

Saving and connecting the red points in Figure 13 to form a line still cannot serve as the final building contour. As shown in the left image of Figure 14, connecting these points results in an irregular shape. Therefore, further regularization is needed to transform it into the shape shown in the right image, which presents a rectangular form with all angles being right angles.

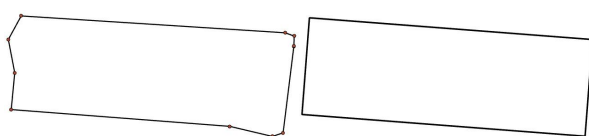


Figure 14. Building contour before (left) and after (right) regularization.

5.3 Accuracy Verification and Analysis

To evaluate the feasibility and accuracy of the proposed method, building area and perimeter were selected as key metrics and compared with ground truth measurements and the building footprints fitted by CloudCompare from point cloud data.

As illustrated in Figure 15 (left), the contours extracted by CloudCompare approximate the overall building geometry at a macro level. However, noticeable discrepancies are observed in local details, resulting in reduced geometric fidelity. In contrast, the contours extracted by our proposed method (Figure 15, right) closely match the actual building shapes, demonstrating higher consistency in both visual structure and boundary alignment. These results suggest that although CloudCompare provides a coarse approximation, it lacks the precision required for

accurate contour delineation, especially in detailed structural modeling.

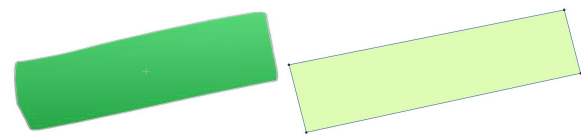


Figure 15. CloudCompare Fitting Results vs. Actual Geometry

The quantitative results presented in Tables 5 – 7 further confirm the superiority of our method. In terms of area extraction (Table 6), CloudCompare yields an average relative error of 17.8%, while our method achieves a significantly lower error of 3.7%, representing a 79.2% improvement. In complex cases such as Building 4, where rooftop structures cause CloudCompare to overestimate area by up to 40.7%, our method effectively reduces this error to 11.1% through the application of Z-axis threshold projection and geometric regularization.

Regarding perimeter extraction (Table 7), our method maintains a high degree of geometric fidelity, with an average relative error of just 2.6%. For Buildings 2 and 5, the relative errors are as low as 0.8% and 2.0%, respectively. These results indicate that the proposed contour regularization strategy not only suppresses noise-induced distortions but also effectively preserves architectural orthogonality.

The marked improvement in accuracy is attributed to three key technical innovations:

- (1) 3D – 3D multimodal point cloud fusion based on ICP, which eliminates 2D/3D misalignment errors common in traditional approaches;
- (2) Z-thresholded deep segmentation using PointNet++, which filters out rooftop interference and improves structural segmentation;
- (3) Angle-constrained contour regularization, which ensures that final contours maintain orthogonal characteristics typical of building footprints.

Building	Ground Truth (m ²)	CloudCompare (m ²)	Proposed Method (m ²)
Building 1	67.6	80.4	65.4
Building 2	188.7	201.9	187.0
Building 3	53.8	63.1	51.8
Building 4	27.0	38	30.0
Building 5	51.1	63.3	49.0

Table 5. Building area extraction results and comparison

Building	CloudCompare Rel. Err. (%)	Proposed Method Rel. Err. (%)
Building 1	18.9%	3.3%
Building 2	7.0%	0.9%
Building 3	17.3%	3.7%

Building 4	40.7%	11.1%
Building 5	23.9%	4.1%
Average	17.8%	3.7%

Table 6. Relative area errors of proposed method vs. CloudCompare

Building	Ground Truth (m)	Proposed Method (m)	Proposed Method Rel. Err. (%)
Building 1	41.7	40.3	3.4%
Building 2	63.1	63.6	0.8%
Building 3	31.1	30.5	1.9%
Building 4	21.3	22.6	6.1%
Building 5	30.3	29.7	2.0%
Average			2.6%

Table 7. Perimeter accuracy of the proposed method

In summary, the proposed method clearly outperforms traditional heuristic-based approaches such as CloudCompare in both area and perimeter accuracy. By leveraging an end-to-end deep learning framework and fusing LiDAR with photogrammetric point clouds, the method achieves superior geometric precision and robustness. It is particularly effective for modeling complex urban and heritage environments with structural irregularities, occlusions, and noise. This makes it highly applicable to urban planning, architectural surveying, heritage documentation, and smart city development.

6. Conclusions

This study presents a novel end-to-end framework for building contour extraction, integrating LiDAR and photogrammetric point clouds via precise 3D-3D registration and deep geometric segmentation. By introducing a Z-axis threshold projection, the method effectively filters rooftop noise, focusing contour extraction on structural walls. The use of PointNet++ enables robust semantic segmentation across multimodal data, while contour regularization via Alpha-shapes, Douglas-Peucker simplification, and angle-constrained vector optimization ensures geometrically faithful and orthogonalized outlines. Experimental results on heritage structures demonstrate that our approach significantly outperforms conventional tools such as CloudCompare, reducing area and perimeter errors to 3.7% and 2.6%, respectively. This study presents a unified pipeline that combines multimodal 3D fusion, deep learning, and geometric regularization for building contour extraction. It offers a robust and scalable solution for complex urban and heritage modeling scenarios.

References

Chen, J., Liu, M.H., Zhao, Z.J., et al., 2023: Research on Building Contour Extraction Method Considering Point Cloud Density Differences. *Surv. Sci.*, 48(12), 188-200.

Cheng, X.J., Cheng, X.L., Hu, M.J., et al., 2016: Building Detection and Contour Extraction from Fusion of Airborne Imagery and LiDAR Data. *Chin. J. Lasers*, 43(05), 253-261.

Di, W.Q., 2022: A Method for Joint Building Extraction Based on Deep Learning with Point Cloud and Image Data. Master's Thesis, Guilin University of Technology, Guilin, China.

Gardiner, J.D., Behnsen, J., Brassey, C.A., 2018: Alpha Shapes: Determining 3D Shape Complexity across Morphologically Diverse Structures. *BMC Evol. Biol.*, 18, 1-16.

Gilani, S.A.N., Awrangjeb, M., Lu, G., 2016: An Automatic Building Extraction and Regularisation Technique Using LiDAR Point Cloud Data and Orthoimage. *Remote Sens.*, 8(3), 258.

Lai, X.D., Pian, W.R., Bo, L.M., et al., 2024: Building Extraction Method Based on LiDAR Point Cloud and Orthophoto Image Fusion Using an Improved Genetic Algorithm (IGA). *J. Infrared Millim. Waves*, 43(01), 116-125.

Li, R., Liu, B., Fang, C.F., 2015: Aerial Remote Sensing in the New Era. *Mod. Phys. Knowl.*, 27(02), 54-60.

Meng, C.T., Zhao, Y.D., Han, W.Q., et al., 2022: Urban Building Change Detection Using RandLA-Net Point Cloud Semantic Segmentation Based on Airborne LiDAR. *J. Nat. Resour. Remote Sens.*, 34(04), 113-121.

Saalfeld, A., 1999: Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm. *Cartogr. Geogr. Inf. Sci.*, 26(1), 7-18.

Tang, Q., Xu, S.H., Gao, X.J., et al., 2023: Building Contour Optimization Method Using High-Resolution Remote Sensing Imagery with Vector Data Fusion. *Surv. Sci.*, 48(12), 143-152.

Wang, Q., Kim, M.K., 2019: Applications of 3D Point Cloud Data in the Construction Industry: A Fifteen-Year Review from 2004 to 2018. *Adv. Eng. Inform.*, 39, 306-319.

Wang, Q., Tan, Y., Mei, Z., 2020: Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications. *Arch. Comput. Methods Eng.*, 27(2), 479-499.

Xu, J.W., Liu, W., Shan, H.Y., et al., 2021: Building Extraction from High-Resolution Remote Sensing Imagery Using the PRCUnet Deep Learning Model. *J. Geoinf. Sci.*, 23(10), 1838-1849.

Yu, M.Y., Chen, X.X., Zhang, W.Z., et al., 2022: Building Extraction from High-Resolution Remote Sensing Imagery Using Grid Attention Gate and Feature Pyramid Structure. *J. Geoinf. Sci.*, 24(09), 1785-1802.