

Article

Automatic Reconstruction of 3D Building Models from ALS Point Clouds Based on Façade Geometry

Tingting Zhao ^{1,2}, Tao Xiong ^{1,*} , Muzi Li ³ and Zhilin Li ¹

¹ Faculty of Geosciences and Engineering, Southwest Jiaotong University, Chengdu 611756, China

² Surveying and Mapping Technology Service Center, Sichuan Bureau of Surveying, Mapping and Geoinformation, Chengdu 611756, China

³ Faculty of Geography Resource Sciences, Sichuan Normal University, Chengdu 610068, China

* Correspondence: xiongtao@my.swjtu.edu.cn

Abstract

Three-dimensional (3D) building models are essential for urban planning, spatial analysis, and virtual simulations. However, most reconstruction methods based on Airborne LiDAR Scanning (ALS) rely primarily on rooftop information, often resulting in distorted footprints and the omission of façade semantics such as windows and doors. To address these limitations, this study proposes an automatic 3D building reconstruction method driven by façade geometry. The proposed method introduces three key contributions: (1) a façade-guided footprint generation strategy that eliminates geometric distortions associated with roof projection methods; (2) robust detection and reconstruction of façade openings, enabling reliable identification of windows and doors even under sparse ALS conditions; and (3) an integrated volumetric modeling pipeline that produces watertight models with embedded façade details, ensuring both structural accuracy and semantic completeness. Experimental results show that the proposed method achieves geometric deviations at the decimeter level and feature recognition accuracy exceeding 97%. On average, the reconstruction time of a single building is 91 s, demonstrating reliable reconstruction accuracy and satisfactory computational performance. These findings highlight the potential of the method as a robust and scalable solution for large-scale ALS-based urban modeling, offering substantial improvements in both structural precision and semantic richness compared with conventional roof-based approaches.

Keywords: Airborne LiDAR Scanning (ALS); 3D building models; façade geometry; automatic reconstruction; window and door detection



Academic Editors: Wolfgang Kainz and Sisi Zlatanova

Received: 13 September 2025

Revised: 12 November 2025

Accepted: 20 November 2025

Published: 25 November 2025

Citation: Zhao, T.; Xiong, T.; Li, M.; Li, Z. Automatic Reconstruction of 3D Building Models from ALS Point Clouds Based on Façade Geometry. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 462. <https://doi.org/10.3390/ijgi14120462>

Copyright: © 2025 by the authors. Published by MDPI on behalf of the International Society for Photogrammetry and Remote Sensing. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Three-dimensional (3D) building models play a vital role in a wide range of urban applications, including digital twins, urban planning, route navigation, energy analysis, and virtual reality visualization. For example, urban planning and heat island analyses typically rely on Level of Detail (LoD) building models, such as LoD2, to obtain detailed roof and wall geometries [1]. And energy analysis and road navigation require clearly defined façade elements (e.g., windows and doors) to enhance recognition efficiency [1]. Accurate reconstruction of geometric structures and comprehensive representation of semantic components are essential for faithfully restoring spatial structures. In addition, these factors ensure the reliability and effectiveness of subsequent analyses and applications. Among the various approaches for acquiring 3D spatial data, Airborne LiDAR Scanning

(ALS) has become a mainstream method for large-scale building modeling. This is due to its high resolution and wide area coverage within a single acquisition [1,2].

However, most existing ALS-based reconstruction methods primarily focus on roof structures [3–5]. A typical approach is to project rooftop boundaries vertically to estimate building footprints [6,7]. Although effective for regular building forms, these methods often fail when dealing with overhanging eaves or complex roof geometries. This failure leads to footprint expansion, positional shifts, and reduced geometric accuracy [8]. To mitigate these issues, several studies have incorporated façade points. Projection-based linear fitting improves footprint delineation but struggles to generate closed or curved outlines [8,9]. Learning-based approaches using annotated façade datasets have also been explored [10]. However, their performance depends heavily on annotation quality, and their generalization capability remains limited.

At the semantic level, façade openings such as windows and doors are critical for both functional description and model recognizability [11,12]. Fusion methods that integrate street-view imagery [13] or external contour data [14] can enhance detection accuracy. However, they require reliable external data, precise cross-modal registration, and extensive preprocessing, limiting scalability for large-scale automatic reconstruction.

Despite these challenges, ALS façade point clouds still preserve valuable information: building walls typically exhibit planarity and vertical continuity, whereas windows and doors appear as local density variations [15–18]. Recent high-quality datasets, such as those from Dublin City [19], demonstrate the feasibility of capturing façade openings. They also support their reliable identification and contour reconstruction [20,21].

In this study, we propose an automatic 3D building reconstruction method from ALS point clouds. It directly leverages façade geometry to reconstruct accurate building footprints and semantic elements such as windows and doors. The method also integrates rooftop structures to generate complete models. Compared with roof-based methods such as City3D [3], the proposed method preserves detailed roof reconstruction. It also overcomes the geometric distortions and semantic deficiencies of those methods. It enhances the structural accuracy and semantic richness of building models and improves the refinement and applicability of automated large-scale 3D reconstruction.

2. Analysis of Existing Methods and a Strategy

2.1. Analysis of Existing Methods

Among the existing ALS-based 3D building modeling methods, roof reconstruction remains the most common approach [3,22,23]. A typical technique is to project roof boundaries vertically onto the ground to estimate building footprints [6,7]. However, when buildings have overhanging eaves or complex roof forms, such methods often cause footprint expansion and displacement, thereby reducing geometric accuracy. Moreover, they overlook façade semantics such as doors and windows, resulting in models with limited semantic detail.

To improve footprint accuracy, some studies have incorporated façade point clouds. Ref. [9] used mobile laser scanning data to project façade points onto the ground and fit linear boundaries. However, linear fitting cannot guarantee the closed reconstruction of curved or complex outlines. Similarly, Ref. [8] projected ALS wall points and applied line fitting, which failed to generate closed footprints, making them unsuitable for volumetric modeling. Ref. [10] proposed a prediction method based on manually annotated façade contour datasets. However, it is highly dependent on annotation quality and shows limited generalization to diverse façade morphologies.

At the semantic level, windows and doors are key façade components that serve functional roles such as lighting, ventilation, and access [11]. They also play an important

role in object recognition [12]. Existing studies have commonly employed street-view images [13] or external contour data [14] for window and door detection. However, street-view images are limited by occlusions and spatial coverage and require precise registration with ALS point clouds, which substantially increases processing complexity. External contour data, often derived from manual annotations or heterogeneous databases, vary across cities and building types, thereby constraining their universality and applicability.

It is noteworthy that although the façade regions of ALS point clouds are sampled less densely than roofs, they still retain critical geometric and semantic information. Façade walls typically exhibit strong planarity and vertical continuity, whereas openings manifest as distinct variations in point density. These characteristics provide a basis for directly identifying and extracting window and door boundaries from ALS façade point clouds.

2.2. A Strategy for 3D Building Model Reconstruction Based on Façade Geometry

Based on the above analysis, this study proposes a 3D building modeling strategy that relies solely on ALS point clouds while ensuring both geometric accuracy and semantic detail. The proposed strategy exploits the geometric continuity and density variations of façade point clouds to support reconstruction from structural outlines to fine semantic elements.

The first step involves separating roofs from façades to obtain façade point clouds. Because roof projections are easily distorted by overhanging eaves and complex geometries, they often introduce geometric biases. In contrast, façade point clouds, although sparse, provide stable vertical structures through normal analysis and plane fitting. Their vertical projection enables the accurate extraction of ground-attached footprints, fundamentally avoiding the errors introduced by roof-based methods.

For window and door extraction, although ALS data lack texture information, openings manifest as sparse point distributions compared with surrounding walls. By leveraging this density difference, façade points are projected onto 2D images. Alpha Shape construction combined with geometry–semantic detection methods is applied to extract opening boundaries, thereby delineating windows and doors.

Finally, the footprints, façade openings, and roof structures are integrated into a complete building model. Within a model-driven framework, polygonal surface assembly and Boolean operations are employed to embed the extracted semantic features into volumetric models. This process achieves a unified representation of geometry and semantics.

In summary, this study proposes an automatic reconstruction strategy based on façade geometry to achieve fine-grained 3D building modeling from ALS point clouds. The main steps are

- (1) Building point-cloud segmentation and footprint construction;
- (2) Window and door contour extraction based on façade geometry;
- (3) Construction of a 3D building model with enhanced façade details.

This strategy mitigates the geometric distortions of roof projection methods, avoids reliance on external data, and provides a feasible solution for large-scale automated ALS-based building modeling. The corresponding flowchart is shown in Figure 1.

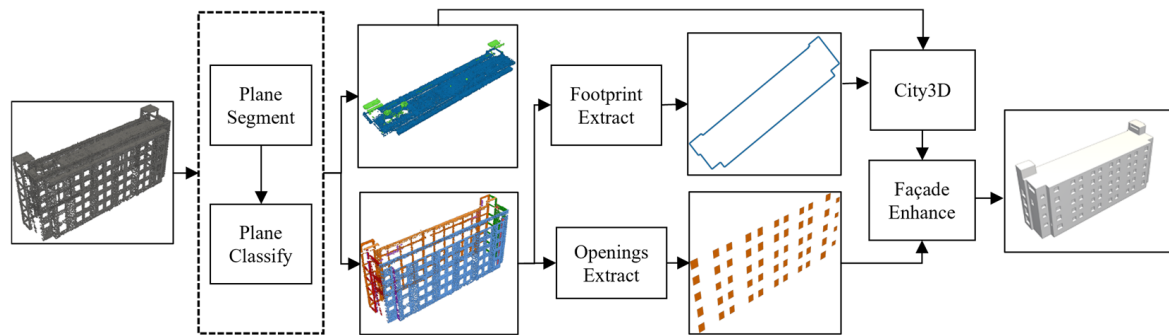


Figure 1. Flowchart of the proposed method.

3. Building Point Cloud Segmentation and Construction of Façade-Based Footprints

In this study, distinguishing roofs from façades and constructing accurate ground-attached footprints were critical for ensuring geometric precision. These steps also facilitated the subsequent recognition of façade semantic components. This section focuses on two main tasks: (1) the automatic extraction of roofs and façades through plane segmentation and classification, and (2) the construction of true building footprints from projected façade point clouds.

3.1. Segmentation and Classification of Building Planes and Façades

Existing plane segmentation methods can generally be divided into two categories: (1) Data-driven methods, which rely on spatial proximity and topological similarity to cluster or segment point clouds [24,25]: These methods require high-quality data and are suitable for low-noise point clouds. and (2) Model-driven methods, which fit geometric models to points and exhibit greater robustness to noise and non-uniform distributions [26,27]. Random Sample Consensus (RANSAC) is a representative model-driven approach [27]. By iteratively sampling and estimating models, the maximum inlier set is identified to fit planes. Compared with other model-driven techniques, such as the Hough transform, RANSAC demonstrates higher stability and noise resistance when handling sparse or noisy ALS point clouds [22]. It has therefore been widely applied in building point cloud processing [3,5,28]. Therefore, RANSAC was adopted in this study for ALS building plane segmentation.

The segmentation and classification of roof and façade regions are conducted in two steps:

(1) Plane segmentation was performed using the RANSAC algorithm to extract the dominant planes from the building point clouds. The segmentation process relied on a distance threshold that determines whether a point is an inlier in a candidate plane. The distance threshold is determined with reference to the inherent accuracy of the input point cloud, which is typically chosen to be of the order of measurement precision. Points falling within the expected positional error are considered inliers of a candidate plane [29]. To reduce over-segmentation, adjacent small planes with similar orientations and close proximity were merged following criteria such as a mutual distance smaller than 0.025 m [30]. The plane parameters were refined using Principal Component Analysis (PCA) after merging [31]. The plane segmentation results are shown in Figure 2a.

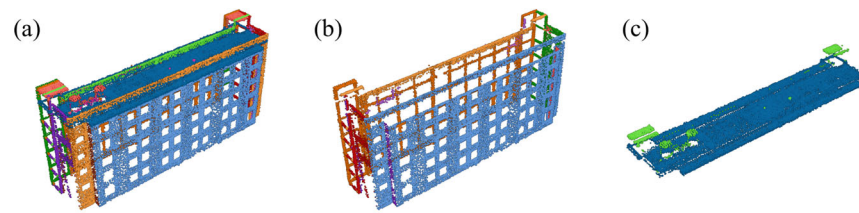


Figure 2. Results of segmentation and classification, where each color represents a planar instance: (a) segmented planes; (b) extracted façade planes; and (c) extracted roof planes.

(2) Plane classification: The normal vector is the most important feature for classifying roof and wall planes. Planes with normal approximately perpendicular to the z -axis were classified as façades, whereas the others were categorized as roofs. The final extracted façade and roof planes are shown in Figure 2b and Figure 2c, respectively.

3.2. Automated Footprint Construction Based on Façade Point Cloud Projection

Based on the extracted façade point clouds described in Section 3.1, an orthogonal projection was applied to obtain 2D point sets. These 2D point sets were subsequently used to generate closed building footprints. Existing approaches can be divided into two categories: (1) structure detection methods, which employ hypothesis–selection strategies to detect line segments in projected points and assemble them into closed contours [8,9]. These methods are effective for regular buildings but cannot handle curved or irregular outlines; and (2) edge-tracking methods, which sequentially trace the boundaries of projected points and can adapt to arbitrary shapes [6,32]. Among the latter, the Alpha Shape algorithm is widely used [5] and was adopted in this study.

In this study, the automated footprint construction process consisted of three steps:

(1) Projection and point simplification: Façade points are orthogonally projected onto the ground to form 2D point sets. To reduce redundancy while preserving spatial distribution, the point sets were simplified using a regular grid. Each grid cell retains only one representative point (e.g., a centroid or random point). A simplified two-dimensional (2D) scatterplot is shown in Figure 3a.

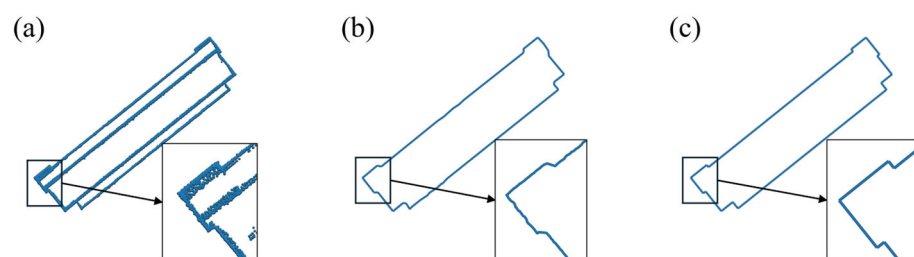


Figure 3. Building footprint extraction: (a) 2D scatter plot of projected façade points; (b) footprint generated using Alpha Shape; (c) simplified footprint using the Ramer–Douglas–Peucker algorithm.

(2) Binary-search-based contour extraction: Delaunay triangulation is first performed on the projected points. The Alpha Shape algorithm was then applied, with α serving as the key parameter that balances detail preservation and geometric closure. To avoid manual tuning, a binary search is used to identify the critical α within a predefined range $(\alpha_{min}, \alpha_{max})$, with a tolerance α_{ϵ} . Specifically, the optimal α is defined as the smallest value that yields a single closed footprint, while $\alpha - \alpha_{\epsilon}$ still results in multiple polygons. To ensure rational parameterization, α_{min} and α_{ϵ} are set to the minimum point pair distance in the point cloud, making the method sensitive to fine-scale details. In contrast, α_{max} is set to a scale larger than the radius of bounding sphere of point cloud, ensuring that the search

range encompasses the correct solution. The binary search converges to the final α and the corresponding footprint (Figure 3b). The detailed process is formalized in Algorithm 1.

Algorithm 1: Binary-search-based contour extraction

Input: Projected 2D point set P

Output: Optimal alpha value α_{opt} , footprint polygon FP

```

1  Compute the nearest-neighbor distances  $d_i$  of all points in  $P$ 
2  Set  $\alpha_{min} \leftarrow \min(d_i)$  # Sensitive to fine-scale details
3  Set  $\alpha_e \leftarrow \alpha_{min}$  # Convergence tolerance
4  Compute the radius  $R$  of the bounding sphere of  $P$ 
5  Set  $\alpha_{max} \leftarrow 2R$  # Ensure search range covers solution
6  Initialize  $\alpha_{low} \leftarrow \alpha_{min}, \alpha_{high} \leftarrow \alpha_{max}$ 
7  while  $|\alpha_{high} - \alpha_{low}| > \alpha_e$  do
8       $\alpha \leftarrow (\alpha_{low} + \alpha_{high})/2$ 
9      Construct  $\alpha$ -shape  $S(\alpha)$  from  $P$ 
10     if  $S(\alpha)$  contains multiple disjoint polygons then
11          $\alpha_{low} \leftarrow \alpha$ 
12     else
13          $\alpha_{high} \leftarrow \alpha$ 
14     end if
15 end while
16  $\alpha_{opt} \leftarrow \alpha_{high}$ 
17  $FP \leftarrow S(\alpha_{opt})$ 
18 return  $\alpha_{opt}, FP$ 

```

(3) Contour simplification: The raw footprint contours extracted from the Alpha Shape algorithm often contain excessive local irregularities. The Douglas–Peucker (DP) algorithm was applied to simplify these contours by iteratively approximating curved segments with straight lines. The process is controlled by a threshold defined as the maximum allowable perpendicular distance between a point and its simplified segment [33]. The threshold was selected based on point cloud density and building scale. In this study, experimental validation showed that a threshold value between 0.1 m and 0.5 m effectively preserved structural fidelity. At the same time, it removed noise-induced fluctuations [6]. For footprints primarily composed of orthogonal edges, a quadratic programming (QP) method was further introduced to globally adjust vertex positions and enforce right-angle relationships between adjacent segments [34]. This resulted in a geometrically consistent footprint (Figure 3c).

4. Window and Door Contour Extraction Based on Façade Geometry

This section focuses on extracting window and door contours from ALS building façade point clouds. In ALS data, openings typically appear as local holes or sparse regions on façades, and their accurate extraction is crucial for enhancing the semantic richness of building models. Existing approaches can be categorized into two groups: geometry/rule-driven methods and learning-driven methods. The former relies on geometric features such as curvature variations, normal vector distributions, or linear edges to identify openings [35,36]. Although effective for dense and complete façade point clouds obtained from TLS or MMS. However, they often fail on sparse ALS data due to discontinuous hole boundaries and noise interference [37]. The latter employs deep neural networks to directly learn geometric and semantic features for window detection and segmentation [28,38]. The YOLO family of models [39] represents a typical learning-driven approach. With

advantages such as end-to-end training, rapid detection, and high robustness [40], YOLO can directly identify the bounding boxes of windows and doors in façade projection images. This effectively addresses the issue of missing boundaries in ALS data. Leveraging this capability, YOLO was applied in this study to detect the positions and scales of windows and doors in images. These detections were then back-projected onto 3D space to generate contours. The main steps are as follows:

(1) Façade projection and image generation: The façade point clouds extracted in Section 3.1 are projected into 2D using Equation (1). A 2D image is generated based on the range and resolution of the point set, and the pixel coordinates of each point are computed using Equation (2). To mitigate sparsity issues in ALS, each point is dilated during image generation (Figure 4a), enhancing the contrast between openings and walls, and improving detectability.

$$P' = R \cdot (P - P_c) \quad (1)$$

$$p = \lfloor (P' - P'_{min}) / r \rfloor \quad (2)$$

where R is the transformation matrix composed of three basis vectors obtained by PCA. P' is the projected point. P is the original point. P_c is the centroid of the façade points. p is the pixel coordinate. P'_{min} is the minimum projected coordinate. r is the image resolution. And $\lfloor \cdot \rfloor$ denotes the floor operation.

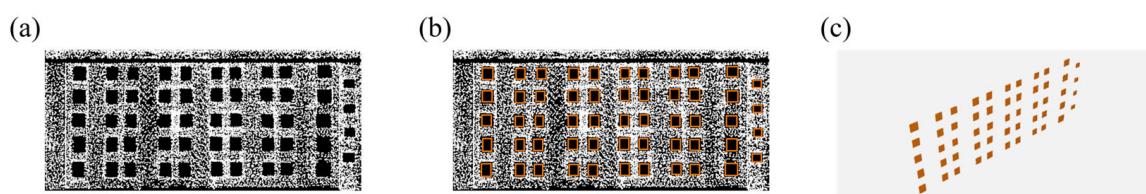


Figure 4. Façade window/door openings extraction via (a) planar image sampling; (b) opening object detection; and (c) inverse projection to global coordinates. White dots represent projected points, orange boxes represent detected openings, and orange blocks represent inverse-projected openings.

(2) Model training: The window and door targets were manually annotated on the projected images to generate a bounding-box detection dataset. Standard data augmentation (random scaling, rotation, and brightness/contrast adjustment) was applied to enrich the sample distribution, as referenced in [41]. The YOLOv11 framework [40] was adopted in this study, offering improvements in both inference speed and detection accuracy over its predecessor. Following reference [39], the number of epochs was set to 200 to prevent premature termination before reaching the optimal model. The remaining key parameters, such as input resolution, early stop strategy and learning rate, were kept at the default values of YOLOv11 framework.

(3) Opening object detection: Using the trained model, window and door bounding boxes were predicted for each façade image. Non-Maximum Suppression (NMS) [42] is applied to remove redundant boxes with lower confidence scores. In general object detection tasks, the Intersection over Union (IoU) threshold in NMS is commonly set within 0.3–0.5 to suppress overlapping boxes of the same object [43]. Since doors and windows are typically arranged in regular patterns and are located close to one another. Using a larger threshold facilitates the retention of adjacent, slightly overlapping targets. Conversely, a smaller threshold enforces a stricter separation between nearby detection boxes, thereby ensuring the semantic consistency of the detection results. Following existing research [44], the IoU threshold of NMS was set to 0.1 in this study. The final bounding boxes were considered as window/door contours (Figure 4b).

(4) Contour back-projection: The pixel coordinates of the four bounding box corners are computed. Then, Equation (3) was applied to back-project the contours from the 2D image space to 3D global coordinates, producing the final results (Figure 4c). The complete procedure is formalized in Algorithm 2.

$$P = R^T \cdot (p * r + P'_{min}) + P_c \quad (3)$$

Algorithm 2: Window and door contour extraction based on façade geometry

Input: Façade point cloud P ; trained YOLO model M ; projection parameters

(R, P_c, P'_{min}, r)

Output: Set of 3D window and door contours C_{3D}

```
//Step 1: Façade projection and image generation
1  For each point  $P_i \in P$  do
2      Compute projected coordinate:  $P'_i = R \cdot (P_i - P_c)$ 
3      Compute pixel coordinate:  $p_i = floor\left(\frac{(P'_i - P'_{min})}{r}\right)$ 
4      Dilate pixel  $p_i$  to enhance local density
5  end for
6  Generate façade image  $I$  from projected pixels
//Step 2: Object detection using YOLO
7   $D \leftarrow M(I)$ 
   //Detect bounding boxes for windows and doors
8  Apply Non-Maximum Suppression to  $D$  with IoU < 0.1
9   $C_{2D} \leftarrow$  Remaining bounding boxes in  $D$ 
//Step 3: 2D–3D back-projection
10 Initialize  $C_{3D} \leftarrow \emptyset$ 
11 For each box  $b \in C_{2D}$  do
12     For each corner  $p$  of  $b$  do
13         Compute 3D coordinate:  $P = R^T \cdot (p \cdot r + P'_{min}) + P_c$ 
14     end for
15     Add quadrilateral contour  $\{P\}$  to  $C_{3D}$ 
16 end for
17 return  $C_{3D}$ 
```

5. Reconstruction of 3D Building Volumetric Framework and Enhancement of Façade Details

This section integrates the building footprints derived in Section 3, the window and door contours extracted in Section 4, and the roof planes segmented in Section 2 into a unified modeling process. This process generates watertight 3D building models that enable component embedding and facilitate subsequent analysis. The workflow comprises two main stages: (1) volumetric framework reconstruction and (2) façade detail enhancement through the integration of window and door contours.

5.1. Reconstruction of 3D Building Volumetric Framework

Current ALS-based building reconstruction methods are generally divided into two categories: data- and model-driven approaches [3,45,46]. Data-driven approaches focus on reconstructing building surfaces by exploiting the geometric and topological relations between points [46]. In contrast, model-driven approaches generate models by identifying structural primitives from point clouds [45]. Given the availability of building footprints, window/door contours, and roof segmentation results, a model-driven

strategy was preferable in this study. By constraining the reconstruction with footprints and planes, wall and roof primitives were generated and selected to form closed polygonal meshes. Compared to interpolation- or fitting-based surface methods, the model-driven strategy offers several advantages: (1) it explicitly preserves the topological relationship between footprints and walls, (2) it supports interpretable energy-term optimization for roof details, and (3) it facilitates the subsequent Boolean embedding of windows and watertight control [43].

City3D is a representative model-driven software tool for ALS-based building reconstruction [3]. Its implementation process involves generating candidate vertical walls and roof patches from the given footprints and roof point clouds. Consistent and smooth patch combinations are then selected through roof-oriented energy optimization to produce a polygonal volumetric framework. This tool effectively restores roof details while maintaining planar wall geometry. However, direct patch assembly may result in cracks or holes along roof–wall intersections, compromising watertightness. For postprocessing, edge-stitching and hole-filling modules from the Computational Geometry Algorithms Library (CGAL) were employed to mitigate this issue [34]. The former merges adjacent edges and vertices based on geometric consistency, while the latter triangulates local gaps to fill residual holes, thereby producing closed and connected meshes. The workflow consisted of the following steps:

(1) Framework generation: The input building footprints and roof point clouds are processed in City3D to generate a set of candidate walls and roof patches. Patch selection is formulated as an optimization problem in which an energy function evaluates the quality of different combinations. Energy consists of three major terms: (i) a data fidelity term, that measures how well each patch explains the corresponding point cloud, and ensures that the selected patches are geometrically consistent with the observations; (ii) a model complexity term, that penalizes overly fragmented or redundant patches, encouraging compact and parsimonious structures; and (iii) a roof preference term, that biases the solution toward patches with geometric characteristics typical of roof surfaces, such as orientation and elevation. By jointly minimizing these terms, the algorithm selects a subset of patches, which are then merged into an initial polyhedral framework.

(2) Watertight repair: Edge stitching was applied to remove cracks and gaps, holes were detected and filled, normal orientations were unified, and nonmanifold elements were removed. These operations resulted in watertight building models suitable for component embedding. The results are shown in Figure 5.

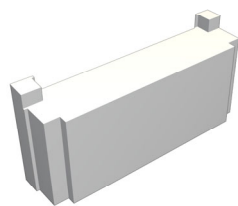


Figure 5. Constructed 3D building volumetric framework.

5.2. Integration of Façade Windows and Doors into the Volumetric Model

This subsection embeds window and door structures into the volumetric framework, thereby enriching the façade details and unifying the geometric and semantic representations. A common approach is solid embedding based on 3D Boolean operations [47,48]. By computing the intersection boundaries between the openings and the framework, the two models were split and recombined according to Boolean logic. This process resulted in complete building models with detailed openings. In this study, window and door contours were extruded into solids using mesh extrusion and Boolean modules in CGAL [34] and

then embedded into the volumetric framework. Considering the noise and positional errors in the ALS data, the extracted window contours may deviate from the wall surfaces. This can cause misalignment and poor embedding results. To address this, we propose a ray-projection-based preprocessing strategy. In this strategy, window and door contours are projected along their normal vectors onto wall surfaces to ensure precise alignment. The process consists of the following steps.

(1) Projection of contours: Window and door contours were projected onto the building wall surfaces, and invalid contours that could not be fully projected were removed (Figure 6a).

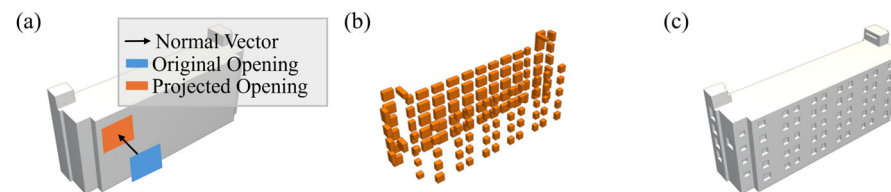


Figure 6. Enhanced building model with integrated façade details: (a) Projecting windows/doors onto wall surfaces; (b) Extruding windows/doors into solids; (c) Embedding window/door solids into the wall.

(2) Solid construction: Following building design specifications (e.g., Masonry Structure Design Code GB 50003-2011), contours are extruded along their normal in both directions to a specified thickness, e.g., 0.24 m [49]. This produces 3D solids consistent with wall thickness constraints (Figure 6b).

(3) Boolean embedding: Boolean difference or union operations are applied depending on the opening type. Concave windows/doors are carved out using difference, whereas protruding windows are fused using union. This results in a building model with embedded façade details (Figure 6c).

6. Experimental Evaluation

6.1. Experimental Results

To evaluate the effectiveness of the proposed method, ten buildings with different structural types in an urban area of Chengdu City were selected as test data. These buildings represent a wide range of architectural diversity. They include common shapes such as rectangular and U-shaped layouts, flat and pitched roof forms, and varying densities and regularities of opening layouts. The detailed information for these buildings is provided in Table 1. The ALS point cloud had an average density of 132 points/m².

Table 1. Details of ten buildings.

Building	Type	Area (m ²)	Height (m)	Vertices Count	Planes Count	Footprint Shape	Roof Type	Opening Layout
Building 1	Residential	516.59	21.81	68,530	7	Rectangle	Flat	DR ¹
Building 2	Public	962.31	24.64	75,821	13	U-shape	Flat	MR ²
Building 3	Public	1029.59	20.29	77,031	9	Rectangle	Flat	MR ²
Building 4	Public	1051.90	41.15	147,648	9	Rectangle	Flat	DR ¹
Building 5	Commercial	1084.43	32.07	101,514	9	Rectangle	Flat	DR ¹
Building 6	Residential	180.09	15.23	37,554	8	Rectangle	Pitched	MI ³
Building 7	Commercial	633.97	19.67	98,873	12	L-Shape	Pitched	MR ²
Building 8	Commercial	542.23	20.57	115,959	15	Concave	Pitched	DR ¹
Building 9	Commercial	425.39	20.39	72,235	8	L-shape	Pitched	MR ²
Building 10	Residential	185.19	14.62	10,920	7	Rectangle	Pitched	SR ⁴

¹ DR—Dense—Regular; ² MR—Moderate—Regular; ³ MI—Moderate—Irregular; ⁴ SR—Sparse—Regular, indicating different façade opening density and regularity.

Experiments were conducted on a workstation equipped with an Intel Xeon W-2123 3.60 GHz CPU, 32 GB DDR4 memory, and an NVIDIA Quadro P4000 8 GB GPU, running Windows 10 x64 22H2, CUDA 11.8, Python 3.10.15, and Pytorch 2.5.1. Feature extraction and model reconstruction were performed using the PCL 1.15.0, YOLO v11, City3D (GitHub, commit C9299EF, <https://github.com/tudelft3d/City3D> accessed on 19 November 2025), and CGAL 6.0.1 software.

The parameters used in the experiment are as follows. The YOLO training dataset was manually annotated and augmented, comprising 768 images and 14,506 bounding boxes. It was divided into training and validation sets in an 8:2 ratio. The number of training epochs was set to 200, and all other parameters were kept at the default YOLO settings. The NMS IoU threshold was set to 0.1. The plane segmentation and contour simplification thresholds were both set to 0.2 m, determined based on the 95th percentile of the k-nearest neighbor local plane fitting error. The value of k was set to 30 as suggested by [50]. For reconstruction, the default parameter settings of City3D were adopted, and the extrusion depth of openings was set to 0.24 m according to GB 50003-2011 [49]. Examples of reconstructed 3D building models are shown in Figure 7.

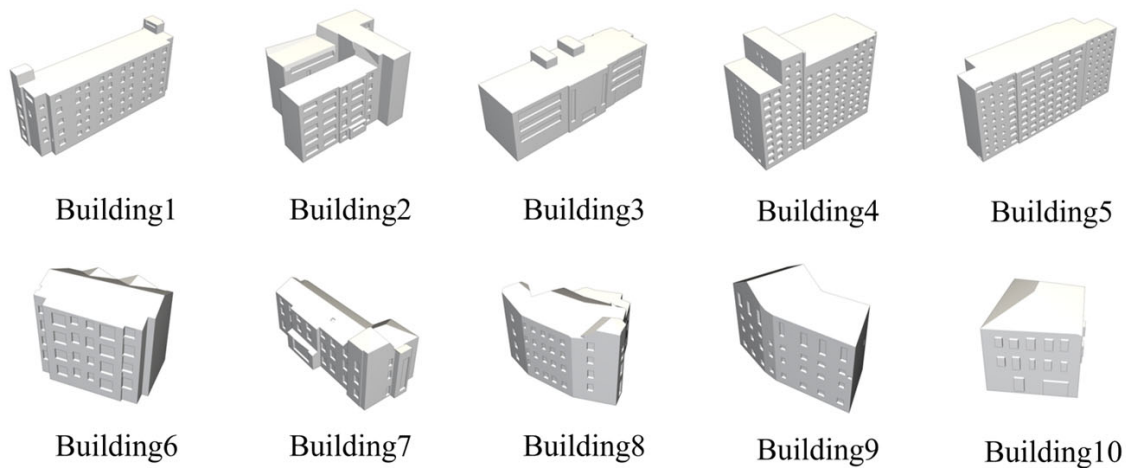


Figure 7. Reconstructed models of buildings using the proposed method.

The runtimes for the feature extraction stage (segmentation, footprint generation, and window/door contour construction), the modeling stage (volumetric reconstruction and façade enhancement), and the overall process were recorded. The results show that the total modeling time averaged 91.16 s per building, with feature extraction requiring approximately 2 s and modeling approximately 89 s. These findings indicate that the proposed method achieves high feature extraction efficiency, with the total modeling time remaining within the order of minutes.

6.2. Evaluation of Reconstruction Results

Manually constructed models (Figure 8) were used as references for the quality evaluation. A visual comparison showed that the reconstructed models closely matched the reference models in terms of overall geometry.

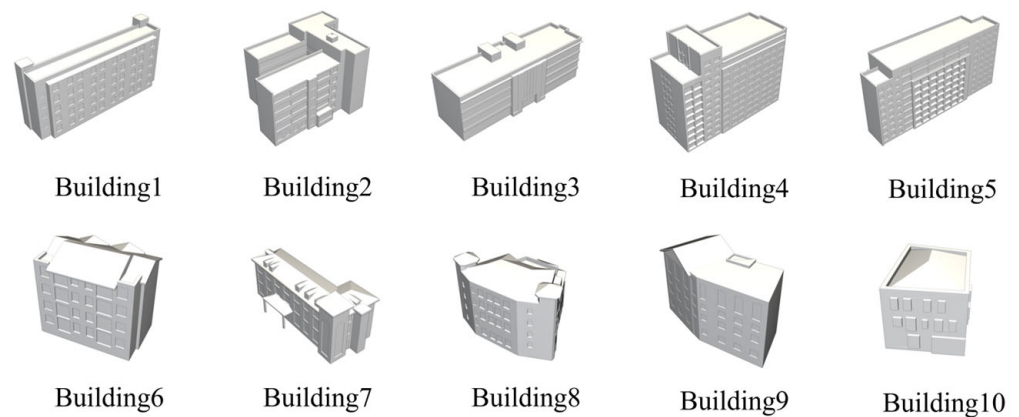


Figure 8. Manually constructed reference models.

A quantitative evaluation was conducted from two aspects: overall structural accuracy and façade window/door recognition.

(1) Overall structure quality

The point-to-point distance errors between the reconstructed and reference models were computed. The mean absolute error (MAE) and root mean square error (RMSE) were calculated for each building and averaged across all cases. The results are shown in Figure 9, and the spatial error distribution is shown in Figure 10.

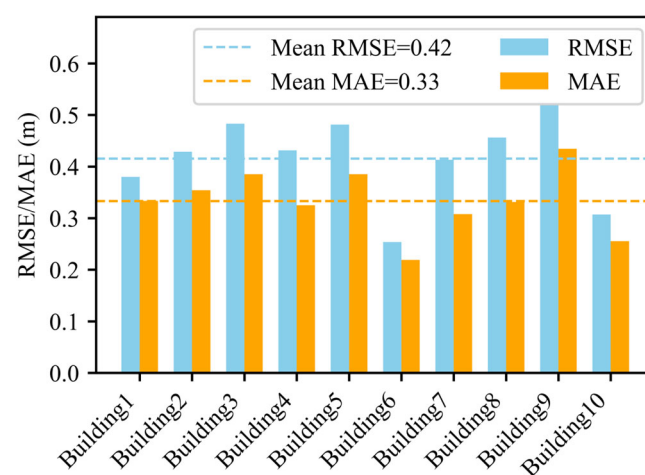


Figure 9. RMSE and MAE results for 10 reconstructed models.

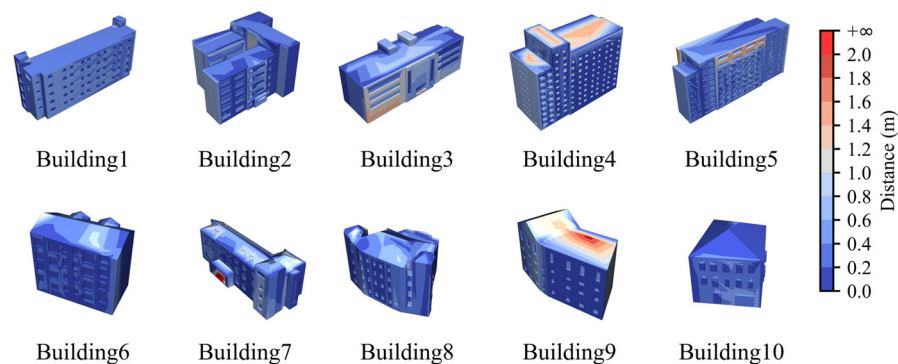


Figure 10. Distance error between reconstructed and reference models.

The average MAE was 0.33 m, while the average RMSE was 0.42 m. MAE reflects global offset, and values at the 0.3 m level indicate strong agreement with reference models.

The RMSE, which is more sensitive to local errors, was slightly higher, indicating larger deviations in complex regions, such as multi-ridge roofs or irregular façades. Most buildings exhibited similar MAE and RMSE values with stable error distributions. Only a few buildings (e.g., Buildings 3, 5, and 9) showed higher RMSE, suggesting local deviations. As illustrated in Figure 9, most building surfaces appear blue, indicating vertex deviations below 0.6 m. Errors mainly arose from complex roof reconstruction, while footprints and façade details maintained high accuracy and stability.

To evaluate the accuracy and efficiency of the proposed method, the precision and runtime of City3D were evaluated using ten buildings as test cases. The average runtime per building was 81.26 s, with a mean absolute error (MAE) of 0.36 m and a root mean square error (RMSE) of 0.46 m. These results indicate that the proposed method achieves a comparable level of computational performance to City3D, with an average difference of only about 10 s. At the same time, it provides higher geometric accuracy and richer semantic representation. In addition, the modular design of the proposed workflow enables direct parallelization, enhancing its scalability for larger datasets and city-scale reconstruction tasks.

(2) façade window/door recognition analysis

For façade details, the window and door recognition performance was evaluated using *Precision*, *Recall*, *F1 Score*, and mean IoU (*mIoU*) [51].

Precision: ratio of correctly detected windows/doors among all detections.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

where *TP* is the number of correctly detected windows/doors and *FP* is the number of incorrectly detected non-window objects.

Recall: ratio of correctly detected windows/doors among all ground-truth openings.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

where *FN* is the number of real windows/doors that were missed.

F1: harmonic mean of *Precision* and *Recall*.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (6)$$

mIoU: geometric overlap between the predicted and ground truth bounding boxes.

$$mIoU = \frac{1}{N} \sum_{i=1}^N \frac{A_i \cap B_i}{A_i \cup B_i} \quad (7)$$

where A_i is the area of the i -th predicted bounding box, B_i is the area of the corresponding ground-truth bounding box, $A_i \cap B_i$ is the intersection area, $A_i \cup B_i$ is the union area, and N is the total number of evaluated instances.

The results (Figure 11) show excellent performance with an average *Precision*, *Recall*, and *F1* of 0.97, 0.98, and 0.98, respectively. A standard deviation of 0.03 indicates both high accuracy and strong stability. This means that false positives and negatives were rare, and nearly all windows and doors were correctly identified.

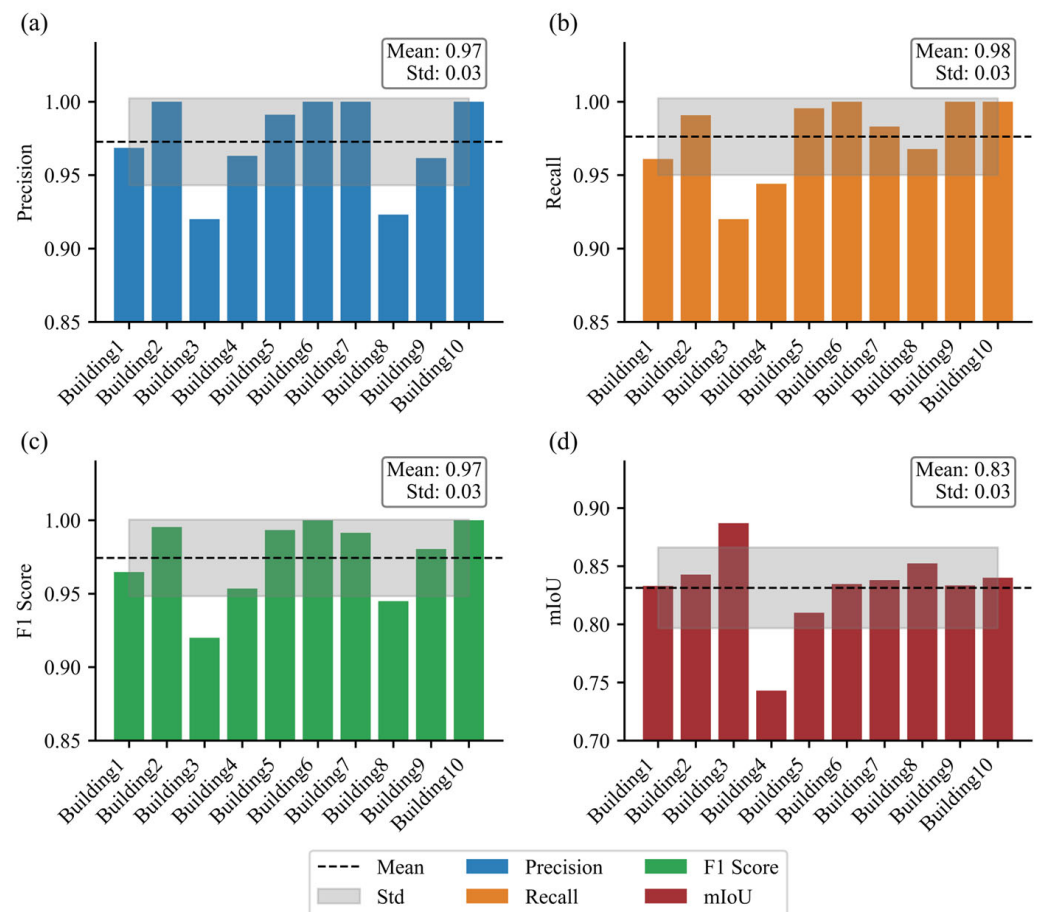


Figure 11. Results of four metrics for 10 constructed building modes: (a) Precision; (b) Recall; (c) F1; (d) mIoU.

The average *mIoU* of 0.83 was at a high, indicating good spatial consistency between the predicted and reference contours. Even under sparse point clouds or incomplete RANSAC segmentation, the minimum *mIoU* remained at 0.74, confirming the robust localization performance. A standard deviation of 0.03 further demonstrates the reliability of the method across different building types and data conditions.

The evaluation results indicate that the proposed method achieves strong geometric consistency with manual models, with global deviations within 0.3–0.4 m. It also provides better accuracy in façade component detection while maintaining good localization precision.

7. Conclusions

This study addresses the limitations of existing ALS-based 3D building modeling methods, including the insufficient use of façade information, distortions in footprint reconstruction, and the absence of façade details such as windows and doors. We propose an automated 3D reconstruction method based on façade geometry, establishing a complete workflow from data preprocessing to model generation: (1) roof-façade separation using RANSAC-based plane segmentation and classification, followed by footprint extraction via façade projection and Alpha Shape with binary search; (2) robust detection and back-projection reconstruction of windows and doors in façade images using disk rendering combined with the YOLO deep learning detector; and (3) volumetric framework reconstruction and watertight repair based on the City3D framework and CGAL, with subsequent

Boolean embedding of windows and doors to generate closed models enriched with façade semantics.

The main contributions of this work are as follows:

(1) A roof–façade joint reconstruction strategy was proposed to address geometric deviations in wall structures commonly caused by traditional roof-only methods. Compared with City3D, the proposed method achieved higher reconstruction accuracy, with an RMSE of 0.42 m compared to 0.46 m for City3D.

(2) Robust detection and reconstruction of façade openings (windows and doors) were achieved under sparse ALS conditions. The accuracy metrics for opening detection, *Precision*, *Recall*, and *F1* were all above 97%, and *mIoU* reached 83%. These results indicate high reliability in window and door representation and improved support for building recognition.

(3) An integrated framework was developed to efficiently generate watertight, structurally consistent, and semantically detailed building models suitable for large-scale urban reconstruction. The average reconstruction time was 91 s per building, slightly higher than that of City3D (81 s per building). However, the proposed method demonstrated superior geometric and semantic performance, achieving a balanced trade-off between efficiency and quality.

Despite its effectiveness, the proposed method has two main limitations. First, its data adaptability is limited by the dependence on planar segmentation and the sparsity of ALS point clouds, which constrains its applicability to buildings with curved façades. This also makes it difficult to accurately distinguish irregular façade elements and their open or closed states from sparse point clouds. Second, the current framework does not yet incorporate standardized data models such as CityGML or IFC. This restricts its interoperability and integration with other urban information systems.

Future work will focus on enhancing both the applicability and interoperability of the proposed framework. Specifically, future research will improve support for complex structures and more detailed semantics. This will be achieved by refining surface-fitting strategies and integrating high-precision point cloud data. Additionally, efforts will be made to adopt CityGML-based standardized model representations to strengthen interoperability and scalability. This will enable seamless integration across data, models, and applications.

Author Contributions: Conceptualization, Tingting Zhao and Tao Xiong; methodology, Tingting Zhao and Tao Xiong; software, Tingting Zhao and Tao Xiong; validation, Tingting Zhao, Tao Xiong and Muzi Li; formal analysis, Tingting Zhao and Tao Xiong; investigation, Tingting Zhao; resources, Tingting Zhao; data curation, Tingting Zhao; writing—original draft preparation, Tingting Zhao; writing—review and editing, Tao Xiong and Zhilin Li; visualization, Tingting Zhao; supervision, Tao Xiong; funding acquisition, Muzi Li and Zhilin Li. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Sichuan Science and Technology Program (2025ZNSFSC0325); the Sichuan Society of Surveying and Mapping Geoinformation Project (CCX202502; CCX202505); and the Task-based Research Project of the Department of Natural Resources of Sichuan Province (No. ZDKJ-2025-004).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request. No publicly archived datasets were generated or analyzed during the study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Beil, C.; Kolbe, T.H. Applications for semantic 3D streetspace models and their requirements—A review and look at the road ahead. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 363. [\[CrossRef\]](#)
2. Mathur, P.; Sharma, C.; Azeemuddin, S. Autonomous inspection of high-rise buildings for façade detection and 3D modeling using UAVs. *IEEE Access* **2024**, *12*, 18251–18258. [\[CrossRef\]](#)
3. Huang, J.; Stoter, J.; Peters, R.; Nan, L. City3D: Large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sens.* **2022**, *14*, 2254. [\[CrossRef\]](#)
4. Kong, G.; Zhang, C.; Fan, H. Large-scale 3-D building reconstruction in LoD2 from ALS point clouds. *IEEE Geosci. Remote Sens. Lett.* **2025**, *22*, 6500305. [\[CrossRef\]](#)
5. Kong, G.; Fan, H. Automatic generation of 3-D roof training dataset for building roof segmentation from ALS point clouds. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5708812. [\[CrossRef\]](#)
6. Li, X.; Qiu, F.; Shi, F.; Tang, Y. A recursive hull and signal-based building footprint generation from airborne LiDAR data. *Remote Sens.* **2022**, *14*, 5892. [\[CrossRef\]](#)
7. Widyaningrum, E.; Gorte, B.; Lindenbergh, R. Automatic building outline extraction from ALS point clouds by ordered points aided hough transform. *Remote Sens.* **2019**, *11*, 1727. [\[CrossRef\]](#)
8. Nurunnabi, A.; Teferle, N.; Balado, J.; Chen, M.; Poux, F.; Sun, C. Robust techniques for building footprint extraction in aerial laser scanning 3d point clouds. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Proceedings of the ISPRS TC III and TC IV Urban Geoinformatics Conference, Beijing, China, 1–4 November 2022*; Jiang, J., Li, S., Zlatanova, S., Eds.; Copernicus GmbH: Göttingen, Germany, 2022; Volume XLVIII–3/W2, pp. 43–50. [\[CrossRef\]](#)
9. Xia, S.; Wang, R. Semiautomatic construction of 2-D façade footprints from mobile LiDAR data. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4005–4020. [\[CrossRef\]](#)
10. Dai, H.; Xu, J.; Hu, X.; Shu, Z.; Ma, W.; Zhao, Z. Deep projective prediction of building façade footprints from ALS point cloud. *Int. J. Appl. Earth Obs. Geoinf.* **2025**, *139*, 104448. [\[CrossRef\]](#)
11. Bakar, Z.A.; Mohemad, R.; Ahmad, A.; Deris, M.M. A comparative study for outlier detection techniques in data mining. In *Proceedings of the 2006 IEEE Conference on Cybernetics and Intelligent Systems, Bangkok, Thailand, 7–9 June 2006*; pp. 1–6. [\[CrossRef\]](#)
12. Tarkhan, N.; Szcześniak, J.T.; Reinhart, C. Façade feature extraction for urban performance assessments: Evaluating algorithm applicability across diverse building morphologies. *Sustain. Cities Soc.* **2024**, *105*, 105280. [\[CrossRef\]](#)
13. Ogawa, Y.; Oki, T.; Chen, S.; Sekimoto, Y. Joining street-view images and building footprint GIS data. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Searching and Mining Large Collections of Geospatial Data, Beijing, China, 2 November 2021*; ACM: New York, NY, USA, 2021; pp. 18–24. [\[CrossRef\]](#)
14. Ledoux, H.; Biljecki, F.; Dukai, B.; Kumar, K.; Peters, R.; Stoter, J.; Commandeur, T. 3dfier: Automatic reconstruction of 3D city models. *J. Open Source Softw.* **2021**, *6*, 2866. [\[CrossRef\]](#)
15. Chen, D.; Zhang, L.; Mathiopoulos, P.T.; Huang, X. A methodology for automated segmentation and reconstruction of urban 3-D buildings from ALS point clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4199–4217. [\[CrossRef\]](#)
16. Hakula, A.; Ruoppa, L.; Lehtomäki, M.; Yu, X.; Kukko, A.; Kaartinen, H.; Taher, J.; Matikainen, L.; Hyypä, E.; Luoma, V.; et al. Individual tree segmentation and species classification using high-density close-range multispectral laser scanning data. *Remote Sens.* **2023**, *9*, 100039. [\[CrossRef\]](#)
17. Hong, Y.; Liu, S.; Li, Z.-P.; Huang, X.; Jiang, P.; Xu, Y.; Wu, C.; Zhou, H.; Zhang, Y.-C.; Ren, H.-L.; et al. Airborne single-photon LiDAR towards a small-sized and low-power payload. *Optica* **2024**, *11*, 612–618. [\[CrossRef\]](#)
18. Walicka, A.; Pfeifer, N. Semantic segmentation of buildings using multisource ALS data. In *Recent Advances in 3D Geoinformation Science*; Kolbe, T.H., Donaubauer, A., Beil, C., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 381–390. [\[CrossRef\]](#)
19. Zolanvari, S.M.I.; Ruano, S.; Rana, A.; Cummins, A.; da Silva, R.E.; Rahbar, M.; Smolic, A. DublinCity: Annotated LiDAR point cloud and its applications. *arXiv* **2019**, arXiv:1909.03613. [\[CrossRef\]](#)
20. da Silva Ruiz, P.R.; Almeida, C.M.; Schimalski, M.B.; Liesenberg, V.; Mitishita, E.A. Multi-approach integration of ALS and TLS point clouds for a 3-D building modeling at LoD3. *Int. J. Archit. Comput.* **2023**, *21*, 652–678. [\[CrossRef\]](#)
21. Zhang, J.; Xia, X.; Liu, R.; Li, N. Enhancing human indoor cognitive map development and wayfinding performance with immersive augmented reality-based navigation systems. *Adv. Eng. Inform.* **2021**, *50*, 101432. [\[CrossRef\]](#)
22. Tarsha Kurdi, F.; Landes, T.; Grussenmeyer, P.; Koehl, M. Model-driven and data-driven approaches using LIDAR data: Analysis and comparison. In *Proceedings of the ISPRS Workshop, Photogrammetric Image Analysis (PIA07), Munich, Germany, 19–21 September 2007*; p. 87.
23. Zhao, R.; Xiong, S.; Liu, Y.; Men, C.; Tian, Z. 3D reconstruction of building surface from airborne LiDAR point clouds based on improved structural constraints. *Int. J. Remote Sens.* **2024**, *45*, 4500–4526. [\[CrossRef\]](#)

24. Che, E.; Olsen, M.J. Multi-scan segmentation of terrestrial laser scanning data based on normal variation analysis. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 233–248. [[CrossRef](#)]
25. Sampath, A.; Shan, J. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1554–1567. [[CrossRef](#)]
26. Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A. The 3D hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Res.* **2011**, *2*, 3. [[CrossRef](#)]
27. Tran, T.-T.; Cao, V.-T.; Laurendeau, D. Extraction of reliable primitives from unorganized point clouds. *3D Res.* **2015**, *6*, 44. [[CrossRef](#)]
28. Yang, L.; Li, Y.; Li, X.; Meng, Z.; Luo, H. Efficient plane extraction using normal estimation and RANSAC from 3D point cloud. *Comput. Stand. Interfaces* **2022**, *82*, 103608. [[CrossRef](#)]
29. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the Weighted RANSAC Approaches for building roof plane segmentation from LiDAR point clouds. *Remote Sens.* **2016**, *8*, 5. [[CrossRef](#)]
30. Su, Z.; Gao, Z.; Zhou, G.; Li, S.; Song, L.; Lu, X.; Kang, N. Building plane segmentation based on point clouds. *Remote Sens.* **2022**, *14*, 95. [[CrossRef](#)]
31. Yeon, S.; Jun, C.; Choi, H.; Kang, J.; Yun, Y.; Doh, N.L. Robust-PCA-based hierarchical plane extraction for application to geometric 3D indoor mapping. *Ind. Robot. Int. J. Robot. Res. Appl.* **2014**, *41*, 203–212. [[CrossRef](#)]
32. Cao, R.; Zhang, Y.; Liu, X.; Zhao, Z. 3D building roof reconstruction from airborne LiDAR point clouds: A framework based on a spatial database. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1359–1380. [[CrossRef](#)]
33. Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartogr. Int. J. Geogr. Inf. Geovis.* **1973**, *10*, 112–122. [[CrossRef](#)]
34. Coeurjolly, D.; Lachaud, J.-O.; Katrioplas, K.; Lorient, S.; Pađen, I.; Rouxel-Labbé, M.; Saeed, H.; Tournois, J.; Yaz, I.O. CGAL 6.0.1—Polygon Mesh Processing: User Manual. 2025. Available online: https://doc.cgal.org/latest/Polygon_mesh_processing/index.html#PSRepairing (accessed on 2 August 2025).
35. Yang, B. Developing a mobile mapping system for 3D gis and smart city planning. *Sustainability* **2019**, *11*, 3713. [[CrossRef](#)]
36. Zheng, Y.; Peter, M.; Zhong, R.; Oude Elberink, S.; Zhou, Q. Space subdivision in indoor mobile laser scanning point clouds based on scanline analysis. *Sensors* **2018**, *18*, 1838. [[CrossRef](#)]
37. Mwangangi, K.K.; Mc'Okeyo, P.O.; Oude Elberink, S.J.; Nex, F. Exploring the potentials of UAV photogrammetric point clouds in façade detection and 3D reconstruction of buildings. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2022**, *XLIII–B2*, 433–440. [[CrossRef](#)]
38. Cheng, B.; Chen, S.; Fan, L.; Li, Y.; Cai, Y.; Liu, Z. Windows and doors extraction from point cloud data combining semantic features and material characteristics. *Buildings* **2023**, *13*, 507. [[CrossRef](#)]
39. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. *arXiv* **2016**, arXiv:1506.02640. [[CrossRef](#)]
40. Sharma, A.; Kumar, V.; Longchamps, L. Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and faster R-CNN models for detection of multiple weed species. *Smart Agric. Technol.* **2024**, *9*, 100648. [[CrossRef](#)]
41. He, L.; Zhou, Y.; Liu, L.; Cao, W.; Ma, J. Research on object detection and recognition in remote sensing images based on YOLOv11. *Sci. Rep.* **2025**, *15*, 14032. [[CrossRef](#)]
42. Gong, M.; Wang, D.; Zhao, X.; Guo, H.; Luo, D.; Song, M. A review of non-maximum suppression algorithms for deep learning target detection. In Proceedings of the Seventh Symposium on Novel Photoelectronic Detection Technology and Applications, Kunming, China, 5–7 November 2020; Su, J., Chu, J., Jiang, H., Yu, Q., Eds.; SPIE: Bellingham, WA, USA, 2021; p. 1176332. [[CrossRef](#)]
43. Yan, J.; Wang, H.; Yan, M.; Diao, W.; Sun, X.; Li, H. IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery. *Remote Sens.* **2019**, *11*, 286. [[CrossRef](#)]
44. Qin, J.; Sun, R.; Zhou, K.; Xu, Y.; Lin, B.; Yang, L.; Chen, Z.; Wen, L.; Wu, C. Lidar-based 3D obstacle detection using focal voxel R-CNN for farmland environment. *Agronomy* **2023**, *13*, 650. [[CrossRef](#)]
45. Nan, L.; Wonka, P. PolyFit: Polygonal surface reconstruction from point clouds. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017; pp. 2372–2380. [[CrossRef](#)]
46. Zang, Y.; Mi, W.; Xiao, X.; Guan, H.; Chen, J.; Li, D. Compound 3D building modeling with structure-aware partition and primitive assembly from airborne laser scanning point clouds. *Int. J. Digit. Earth* **2024**, *17*, 2375112. [[CrossRef](#)]
47. Pereira, A.M.B.; de Arruda, M.C.; Miranda, A.C.d.O.; Lira, W.W.M.; Martha, L.F. Boolean operations on multi-region solids for mesh generation. *Eng. Comput.* **2012**, *28*, 225–239. [[CrossRef](#)]

-
48. Wysocki, O.; Hoegner, L.; Stilla, U. MLS2LoD3: Refining low LoDs building models with MLS point clouds to reconstruct semantic LoD3 building models. In *Recent Advances in 3D Geoinformation Science*; Kolbe, T.H., Donaubauer, A., Beil, C., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 367–380. [[CrossRef](#)]
 49. Cai, Y.; Shi, J.L.; Yang, W.C.; Lv, X.Y.; Li, D.J. Research of masonry shear strength under shear-compression action. *Adv. Mater. Res.* **2015**, *1065–1069*, 1309–1318. [[CrossRef](#)]
 50. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Dolha, M.; Beetz, M. Towards 3D point cloud based object maps for household environments. *Robot. Auton. Syst.* **2008**, *56*, 927–941. [[CrossRef](#)]
 51. Wu, Q.; Song, Z.; Chen, H.; Lu, Y.; Zhou, L. A Highway pavement crack identification method based on an improved U-Net model. *Appl. Sci.* **2023**, *13*, 7227. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.