

Semantic-aware Multi-Scale Simplification of Urban-Scale 3D Real-Scene Mesh Models

Wang Xia¹, Yong Luo², Jiang Fan³, Shaoyi Wang⁴, Xinyi Liu³, Yongjun Zhang³, Liang Fei¹, Wei Wang⁴, Bin Zhang¹,
Jinming Zhang⁵, Zeshuang Zheng²

¹ China Railway Siyuan Survey and Design Group Co., Ltd., Wuhan, Hubei, China - 1548963280@qq.com,
feiliang3828912@126.com, 007981@crfsdi.com

² Guangdong Provincial Key Laboratory of Smart and Safety City Planning and Monitoring, Guangzhou, Guangdong,
China - (53016822, 51803728)@qq.com

³ School of Remote Sensing and Information Engineering, Wuhan University, Hubei, China - (fanjiang, liuxy0319,
zhangyj)@whu.edu.cn

⁴ Tianjin Key Laboratory of Spatio-Temporal Information Engineering and Technology, Tianjin, China -
wshaoy100@126.com, wangwei_sgg@163.com

⁵ Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China - nicnyzjm@whu.edu.cn

Keywords: 3D real scene data; Planar feature extraction; Semantic segmentation; 3D model simplification.

Abstract

Recent advances in measurement technologies have significantly improved the accuracy of multi-scale 3D reconstruction, yet the resulting large-scale data with inherent redundancy pose challenges for storage and real-time rendering. This paper proposes a systematic framework for efficient lightweight processing of 3D real-scene mesh model, integrating planar feature extraction, point cloud classification, and semantics-driven simplification. The key scientific contributions include: (1) A preprocessing process for the reality 3D model is added for the plane segmentation algorithm; (2) A training-free point cloud classification method employing 9 complementary geometric-semantic features and probabilistic smoothing to achieve computationally efficient classification without the need for deep learning or annotated data; and (3) An innovative semantic-driven simplification strategy that dynamically adjusts processing priorities based on feature importance. Experimental results demonstrate the framework's effectiveness in preserving critical architectural features (e.g., façades and roofs) while aggressively compressing less significant elements (e.g., terrain and clutter), achieving balanced data reduction and information retention. At equivalent simplification ratios, our algorithm achieves a **23%** improvement in model accuracy compared to the baseline method, with a **31%** accuracy enhancement specifically for critical geometric features. When maintaining equivalent accuracy levels, the proposed method reduces face count by **23%** relative to the baseline approach. The proposed methods advance 3D urban modeling by addressing both technical and practical challenges in large-scale scene processing.

1. Introduction

1.1 Motivation

The proliferation of low-altitude flight applications demands high-fidelity 3D environmental models enabled by recent advances in measurement technologies. While these advances have significantly improved the accuracy of multi-scale real-scene 3D reconstruction, they concurrently generate massive data volumes with inherent redundancy. This poses critical bottlenecks for low-altitude platforms constrained by limited onboard storage and real-time rendering requirements.

Existing mesh simplification techniques, including widely adopted QEM edge-collapse algorithms, prove inadequate for 3D real-scene data. Such data exhibits semantic-critical asymmetry: navigationally vital features require maximum retention, while non-essential terrain clutter permits aggressive compression. Current geometry-driven methods cause façade loss and roof distortions that directly jeopardize flight safety. Two domain-specific gaps persist:

1. **Failure to Address Spatial Heterogeneity:** Uniform simplification degrades navigationally dense urban features while under-utilizing compression potential in open areas, violating low-altitude platforms' need for adaptive resource allocation.

2. **Neglect of Feature-Specific Requirements:** Mission-critical vertical navigation features receive equal treatment to non-essential ground clutter, disregarding operational priorities for flight path planning and emergency response.

This work bridges these gaps through a semantics-driven simplification framework that dynamically prioritizes aeronautically significant features. By aligning compression intensity with navigational importance, our approach enables storage-efficient 3D models that maintain flight-critical fidelity—directly supporting 3D real scene of China goals for low-altitude digital infrastructure.

1.2 Related Work

Mesh simplification methods can be broadly categorized into static and dynamic approaches. Foundational work in static simplification includes: vertex clustering method (Rossignac, Borrel, 1993) pioneered a universal framework for spatial partitioning and vertex merging, though lacking in feature preservation, excels in efficiency; Surface region merging technique (Kalvin, Taylor, 1996) introduced the first rigorous error-bound control, preserving topological integrity through triangular face aggregation; Vertex decimation algorithm (Schroeder et al., 1992) established local geometric error metrics; Quadric Error Metric (QEM) (Garland, Heckbert, 1997) revolutionized the edge collapse optimization process, becoming the algorithmic cornerstone for industrial-grade simplification; Wavelet-based multiresolution analysis

(Lounsbery et al., 1997) provided a theoretically rigorous hierarchical representation for regular grids.

Core breakthroughs in dynamic simplification focus on: Progressive Meshes (PM) (Hoppe, 2023) enabled lossless continuous LOD transitions via edge collapse/vertex split operation sequences, establishing the paradigm for dynamic methods; View-dependent simplification (Xia, Varshney, 1996) pioneered real-time control mechanisms incorporating view-frustum constraints and screen-space error metrics; Edge collapse sequence approach (Ronfard, Rossignac, 1996) generated continuous simplification models based on geometric deviation ordering; Octree-based adaptive spatial partitioning (Pan et al., 2001) combined with QEM significantly enhanced the engineering feasibility of vertex clustering by preserving geometric features while maintaining controllable simplification errors; View-dependent refinement (Hoppe, 1997) further integrated normal constraints and geometric error optimization into the PM framework. Out-of-core simplification algorithm (Lindström, 2000) addresses real-time processing of massive-scale meshes, while simplification envelopes (Cohen et al., 1996) ensure globally controlled geometric deviations through error-bound constraints. These contributions have laid the theoretical and technical foundations for the application of mesh simplification in storage, transmission, and real-time rendering.

However, traditional simplification algorithms fail to adequately address the spatial heterogeneity of geographic entities and variations in their semantic significance. This results in two persistent challenges: **1)** difficulty in processing unevenly distributed geographic data, and **2)** inability to dynamically adapt simplification strategies according to feature categories. These limitations impede end-to-end lifecycle management of 3D real scene data.

In this paper, by analyzing the problems arising from existing simplified ways of processing real-scene 3D data, a set of systematic processing frameworks ranging from planar feature extraction, point cloud classification to semantics-driven lightweighting are proposed. After an incremental research, efficient lightweighting of 3D real scene data is achieved. We validated the proposed method using multi-scale mesh data from a Chinese city. Quantitative evaluation via average distance and standard deviation metrics between simplified and original models confirms the method's efficacy in preserving critical structural characteristics during simplification. This capability directly supports applications of multidimensional urban models in low-altitude flight operations and urban management scenarios where retention of key architectural features is essential.

2. Methodology

This paper presents a comprehensive technical framework for processing and analyzing 3D real-scene data, systematically addressing feature extraction, semantic classification, and semantically-guided simplification. Through progressive methodology, efficient processing and optimization of real-scene 3D mesh models have been achieved. Figure 1 shows the pipeline of our method.

In low-altitude flight applications, building roofs and façades in 3D real-scene models significantly impact aircraft takeoff/landing and flight navigation. Variable vegetation, vehicles, and other low-lying clutter exhibit lesser influence on UAV operations, while terrain features demonstrate minimal impact. Balancing processing efficiency with these considerations, our algorithm simplifies data into four

fundamental categories during processing: building roofs, building façades, clutter, and terrain. This classification scheme remains adjustable for complex application scenarios or mission-specific requirements. Furthermore, as the algorithm processes real-scene 3D data—where geometric primitives maintain uniform scale characteristics—identical processing parameters deliver consistent performance across varying scene sizes.

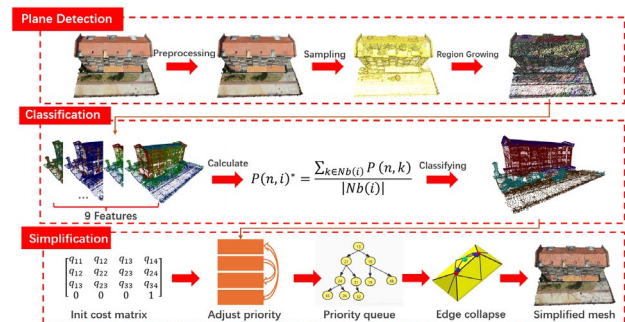


Figure 1. The pipeline of our method.

2.1 Plane segmentation based on Region growing

For existing multi-scale 3D real-scene geographic scene models constructed using conventional photogrammetry and emerging computer vision methods, preprocessing is essential before planar feature extraction. The preprocessing workflow consists of: triangulation, removal of duplicate geometric structures, and centroid normalization. Triangulation standardizes diverse data sources by converting all surface patches into triangles, ensuring compatibility across heterogeneous datasets for subsequent processing. Removal of duplicate geometric structures reduces data redundancy by scanning all points in the 3D file, prioritizing elimination of duplicate points and proximity-filtered points, followed by topology-preserving adjustments to edges/faces containing deleted vertices. Centroid normalization processes all vertices' 3D coordinates relative to the spatial centroid to enhance computational efficiency and mitigate precision errors caused by large geospatial coordinate values.

Through the aforementioned process, vertex extraction from the processed model yields input vertex data. For planar feature extraction, each vertex requires surface fitting within a defined neighborhood range. The gradient direction of the fitted surface determines the vertex's normal vector.

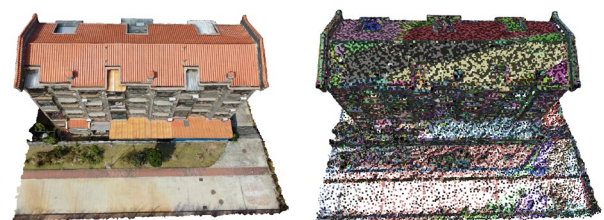


Figure 2. The left image displays the input 3D real-scene model; the right image shows vertex data after planar extraction, where points belonging to the same region share identical color.

Subsequent region growing plane detection proceeds as follows:

- (1) **Neighborhood identification:** Using spherical neighborhood search with radius r , the algorithm identifies proximal points around each vertex to detect potential coplanar clusters;
- (2) **Dual-Threshold plane validation:** Least-squares plane fitting evaluates candidate points against dual thresholds:

maximum distance-to-plane tolerance and maximum normal deviation angle;

- (3) **Region growing:** Points satisfying both constraints aggregate into planar regions;
- (4) **Region assignment:** Each vertex associates with an optimal planar region. For visualization, distinct colors denote regions (Figure 2).

The algorithm outputs are vertex-to-plane region mapping, and a plane set which establishes the data base for subsequent classification.

2.2 Point cloud classification

Given that each vertex/face in the 3D real-scene data carries explicit semantic labels, this section defines four semantic classes: terrain, façade, roof, and clutter. In this part, the calculation methods for various feature metrics used in classification and their impact on classification will be elaborated, followed by an analysis of how to construct the discriminant energy equation for classification based on these feature metrics. Classification is ultimately achieved through probability maximization to enable rapid categorization of point cloud data. This classification framework requires no supervision or learning while yielding relatively accurate results. For point cloud feature metrics, considerations primarily encompass three aspects: local neighborhood features, geometric features, and elevation distribution features. The involved features include: distance to plane, vertical dispersion, elevation, vertical range, verticality, region size, curvature, planarity, anisotropy. The definitions for each feature and the necessary computational procedures for certain feature will be detailed individually below:

- (1) **Distance to plane**, a feature for point cloud classification, represents the perpendicular distance between a point and its locally fitted plane. This feature characterizes the planarity of a point's region by analyzing its neighborhood, fitting a local plane, and computing the geometric distance from the point to this plane.
- (2) **Vertical Dispersion** quantifies the discrete degree of elevation values within a point's neighborhood. It reflects vertical undulation characteristics through statistical elevation differences among neighboring points. Higher values indicate more drastic elevation changes, implying greater likelihood of being a non-ground point. Computed as the proportion of unoccupied intervals in uniformly partitioned elevation ranges:

$$\text{Vertical Dispersion} = 1 - \frac{N_{occ}}{M} \quad (1)$$

where N_{occ} = occupied interval count ;
 M = total intervals.

M represents total intervals derived from:

$$M = \left\lfloor \frac{z_{max} - z_{min}}{\Delta z} \right\rfloor + 1 \quad (2)$$

where Δz = grid resolution;
 z_{max} = the maximum elevation;
 z_{min} = the minimal elevation.

- (3) **Elevation** corresponds directly to the Z-coordinate in 3D real-scene data, requiring no computation. Primarily distinguishes roof planes from ground planes.
- (4) **Vertical range** represents the elevation span within a local grid region ,which means difference between maximum z_{max} and minimum z_{min} . This feature

quantifies vertical height variations and is effective for distinguishing buildings with significant elevation changes from flat areas:

$$\text{Verticalrange} = z_{max} - z_{min} \quad (3)$$

- (5) **Verticality** quantifies surface orientation by measuring the deviation between a local normal vector and vertical direction. This metric differentiates critical vertical structures like façades from horizontal surfaces such as ground planes:

$$\text{Verticality} = 1 - |n_g \cdot n_v| \quad (4)$$

where n_g = ground normal vector;
 n_v = vertex normal vector.

- (6) **Region Size** is the count of vertices (x) within a planar region, obtained through planar detection mapping. Requires normalization due to its large numerical range:

$$\text{RegionSize} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (5)$$

where x_{min} = minimal vertex number;
 x_{max} = maximal vertex number.

- (7) **Curvature** estimates vertex deviation using normal vectors of one-ring adjacent faces. It detects ground clutter and is computed from Planar region normal n_f which is obtained from planar segmentation $Ax + By + Cz + D = 0$:

$$\text{Curvature} = \frac{\sum_{i=1}^m (1 - n_f \cdot n_{t_i})}{m} \quad (6)$$

where m = one-ring face count;
 n_{t_i} = normal of one-ring adjacent faces.

- (8) **Planarity** indicates whether a neighborhood around a vertex exhibits distinct planar characteristics. For each vertex, the neighborhood point set is determined using K-Nearest Neighbors (K-NN). Then the covariance matrix M_k of the point set is computed. Finally, eigen decomposition of M_k yields three eigenvalues ($\lambda_0 \leq \lambda_1 \leq \lambda_2$). Planarity is then calculated as:

$$\text{Planarity} = \frac{\lambda_2 - \lambda_0}{\lambda_0} \quad (7)$$

- (9) **Anisotropy** indicates whether points in a vertex's neighborhood are uniformly distributed. Higher values correspond to more uniform distributions, while lower values indicate greater disorder. Anisotropy likewise utilizes the eigenvalues of covariance matrix M_k , defined as:

$$\text{Anisotropy} = \frac{\lambda_1 - \lambda_0}{\lambda_0} \quad (8)$$

To classify vertices in the triangular mesh using the aforementioned 9 features, this paper constructs an energy equation that transforms the classification problem into a probability maximization problem. To prevent scale differences among features from biasing results, each feature requires an

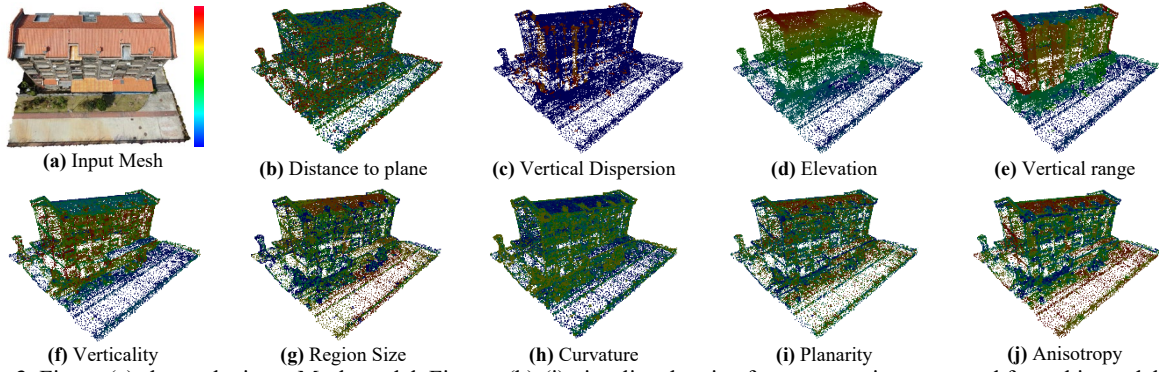


Figure 3. Figure (a) shows the input Mesh model. Figures (b)-(j) visualize the nine features metrics computed from this model. In the visualizations, higher feature values are represented by colors closer to red, while lower values correspond to colors closer to blue.

evaluation function $F_j(x_i)$ defined and computed as follows:

$$F_j(x_i) = \begin{cases} 1 - \max\left(\min\left(1, \frac{f_j(i)}{w_j}\right), 0\right), & \text{if } f_j \text{ favors } x_i \\ 0.5, & \text{if } f_j \text{ is neutral for } x_i \\ \max\left(\min\left(1, \frac{f_j(i)}{w_j}\right), 0\right), & \text{if } f_j \text{ penalizes } x_i \end{cases} \quad (9)$$

where $x = (x_t)_{t=1..N}$ is a latent classification result;
 N = number of input items;
 x_t = category of item i ;
 $f_j(t)$ = raw value of the j -th feature at the i -th item;
 w_j = weight of feature j ;
 $F_j(x_i)$ = normalized value of feature j at item i .

Three label influence types affect computation:

FAVORING: Label prioritized when feature values are high;
NEUTRAL: Label unaffected by feature values;
PENALIZING: Label prioritized when feature values are low.

On this defined basis, the discriminant energy function $E_z(x_i)$, $i = 1, 2, 3, 4$ for four categories is defined and computed as follows:

$$E_z(x_i^n) = \sum_{j=1}^9 F_j(x_i^n) \quad (10)$$

where x^n = the n -th point of the original mesh model.

Since points of the same type are often clustered in reality, neighborhood influences must be considered during classification. Therefore, discriminant energy requires smoothing within the points neighborhood range.

$$E_z(x_i)^* = \frac{\sum_{k \in Nb(n)} E_z(x_i^k)}{|Nb(n)|} \quad (11)$$

where $Nb(n)$ = point set of points in the neighborhood of the n -th point

The total energy function $E(x^n)$ is computed as:

$$E(x^n) = \sum_{i=1}^4 E_z(x_i^n)^* \quad (12)$$

This paper transforms the minimization of the above energy into a probability maximization problem. After inputting point cloud data, an initial probability distribution is first generated using exponential decay of weighted feature sums. For the n -th point in the cloud, the weighted feature sum $S(n, i)$ for each category is calculated using:

$$S(n, i) = \sum_{j=1}^9 w_j \cdot F_j(x_i^n) \quad (13)$$

where w_j = weight of the j -th feature.

Applying exponential decay transformation yields the initial probability $P(n, i)$. This transformation converts weighted sums to positive values, with smaller sums indicating better feature matching and thus producing larger results.

$$P(n, i) = e^{-S(n, i)} \quad (14)$$

Using this probability estimation method, E_z is replaced by the exponentially transformed probability P . For each point, its label is adjusted based on classification results within its neighborhood—specifically through classification probability adjustment. After processing, each point's classification result incorporates probabilities $P(n, i)^*$ that fully consider neighboring points:

$$P(n, i)^* = \frac{\sum_{k \in Nb(i)} P(n, k)}{|Nb(i)|} \quad (15)$$

The category L_n of the n -th point is the category corresponding to the maximum P :

$$L_n = \arg \max_i P(n, i)^* \quad (16)$$

2.3 Semantic-aware simplification algorithm

In this section, we implement an adaptive 3D model simplification strategy based on feature categories through the coupling of vertices with semantic information based on the QEM algorithm. Ultimately, it is possible to implement both conservative simplification of key elements such as façade and roof, as well as aggressive compression of secondary elements such as terrain and clutter, so that the data can be simplified while retaining the primary information.

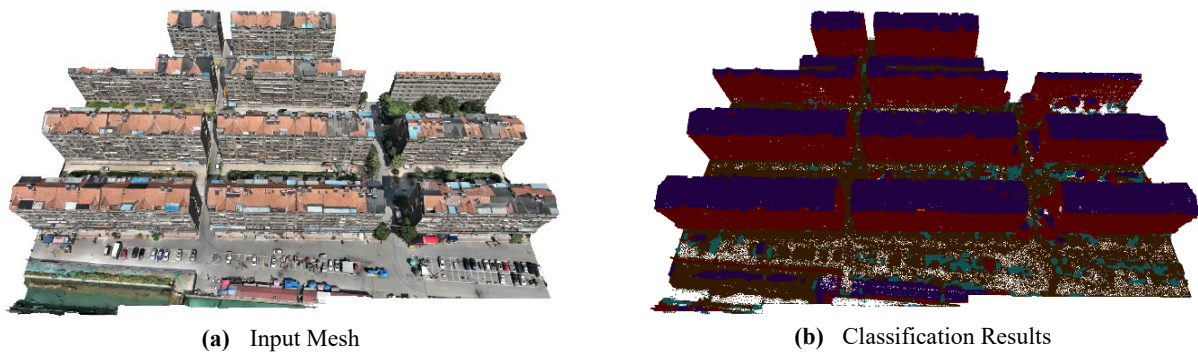


Figure 4. The left figure displays a 3D model of a residential building complex. The right figure shows the classification result, where vertices are categorized into four classes: Terrain (brown), Roofs (blue), Building façades (red), and Clutter (green).

For each vertex pair processed by the QEM algorithm, the category of the first vertex is used as its category, and the simplification priority is adjusted according to the four categories obtained from the point cloud classification.

The algorithm initiates by computing quadric error matrices Q_i for all vertices during initialization, followed by generating all collapsible vertex pairs (v_1, v_2) . For each pair, the optimal contraction target vertex $\bar{v}^T$ is calculated to minimize the post-collapse error $\bar{v}^T(Q_1 + Q_2)\bar{v}$ where Q_1 and Q_2 are quadric matrices of v_1 and v_2 , with this error defining the collapse cost. Semantic-aware priority adjustment assigns each vertex pair a category. Terrain receives elevated priority due to its low informational value, while building façades/roofs receive suppressed priority given their geometric significance. However, when normal deviation exceeds the coplanarity threshold, priority increases to eliminate redundant features while preserving non-coplanar geometry through normal-consistency checks. Clutter priority is moderately raised to simplify non-critical elements like vegetation or vehicles without structural compromise, whereas default categories retain original priorities. Vertex pairs are subsequently inserted into a min-heap priority queue ordered by collapse cost, triggering an iterative collapse process that sequentially extracts the top pair (v_1, v_2) , collapses it into $\bar{v}^T$, and updates costs for $\bar{v}^T$ -adjacent pairs until heap exhaustion, ultimately outputting the simplified mesh M .

3. Experimental Results

3.1 Experimental data and platform

The experiments were conducted on a workstation equipped with an Intel® Core™ i7-13700KF CPU @ 3.40 GHz and 64GB RAM. The test data consisted of OBJ models generated through 3D reconstruction of oblique photogrammetry data. The proposed method was implemented using The Computational Geometry Algorithms Library (The CGAL Project, 2024) and Visualization and Computer Graphics Library (Cignoni et al., 2023).

3.2 Parameter Settings

For the 9 features described in Section 2.2, positive and negative labels are assigned to their evaluation functions based on their influence on the four target categories: terrain, building façades/roofs, clutter, and default. The specific label assignments are shown in Table 1.

The weights assigned to each feature are shown in Table 2. Notably, since the planar region size feature exhibits significantly

larger magnitude, its weight is set to 0.01 times the maximum value to ensure it falls within a reasonable range after normalization.

FEATURES	Terrain	Façade	Roof	Clutter
Distance to plane	N	N	N	F
Vertical Dispersion	N	N	N	F
Elevation	P	F	F	P
Vertical range	P	F	P	P
Verticality	P	F	P	N
Region Size	F	F	F	P
Curvature	P	P	P	F
Planarity	F	F	F	P
Anisotropy	F	F	F	P

Table 1. Feature Label Settings for Each Category (P: Penalizing, N: Neutral, F: Favoring)

Features	Weights
Distance to plane	5.0
Vertical Dispersion	5.0
Elevation	5.0
Vertical range	2.0
Verticality	5.0
Region Size	0.01*max
Curvature	0.5
Planarity	2.5
Anisotropy	2.5

Table 2. Weights of Features.

3.3 Metrics

This study employs the mean distance from the simplified model to the original model as the quantitative evaluation metric. This metric quantifies the overall geometric deviation introduced by simplification into a single scalar value. A smaller mean distance indicates that the vertices of the simplified model collectively adhere closer to the original surface, preserving more geometric details and achieving higher geometric fidelity. The calculation of meanDistance is as follows:

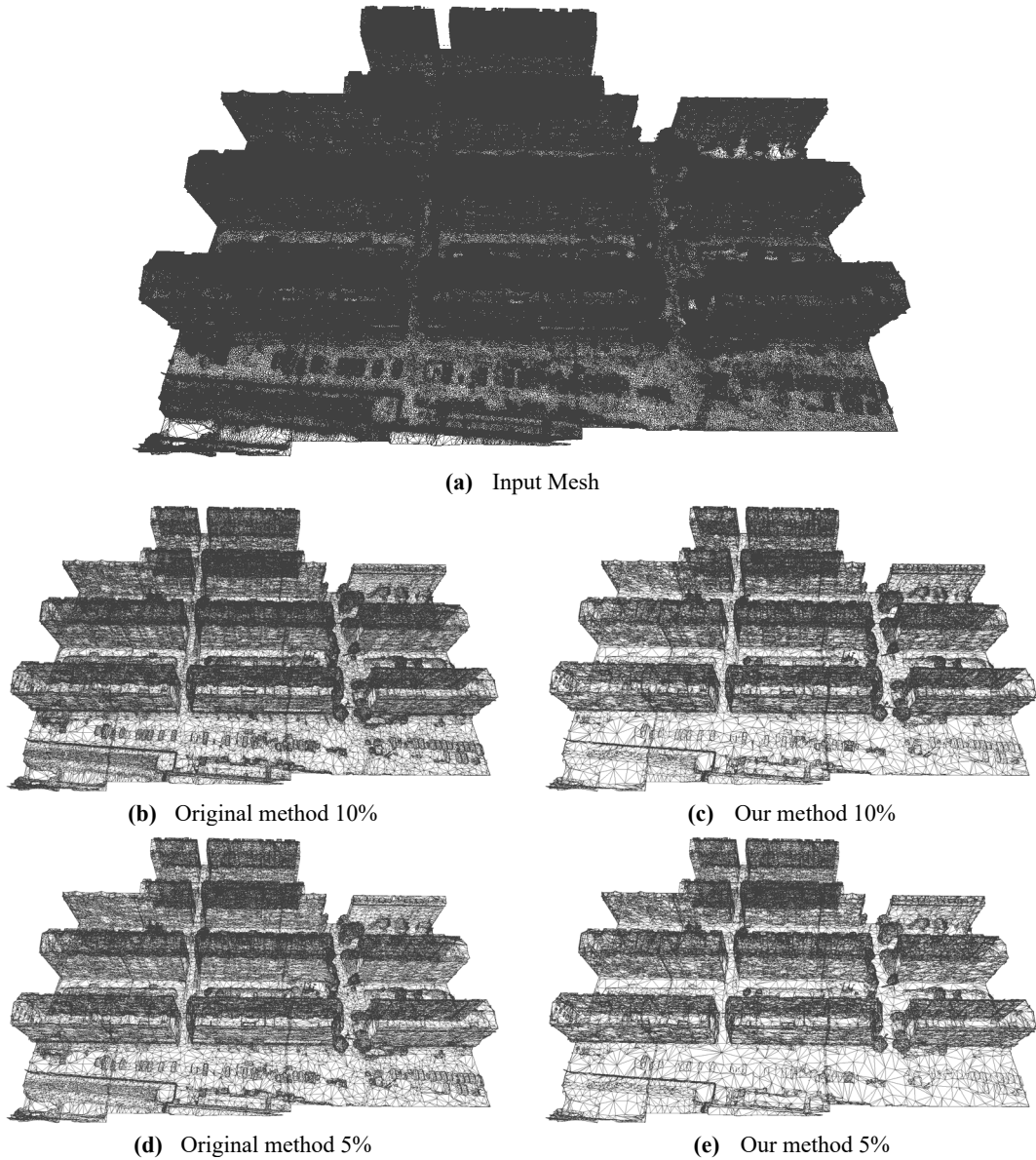


Figure 5. (a) Input 3D model; (b)(d) Results of the baseline method simplified to 10% and 5% ratios; (c)(e) Results of our method simplified to 10% and 5% ratios. Comparative analysis demonstrates that our method preserves critical geometric features more effectively than the uniform simplification strategy of the baseline approach.

$$\text{meanDistance} = \frac{1}{N} \sum_{i=1}^N \min_{y \in \text{RefMesh}} |x_i - y| \quad (17)$$

where

x_i = i -th vertex in the simplified model;
RefMesh = reference model;
 $\min_{y \in \text{RefMesh}}$ calculates the Euclidean distance to the closest point on RefMesh from x_i ;
 N = total vertices number on the simplified model.

Additionally, the root mean square (RMS) is computed to assess the reasonableness of data fluctuations, calculated as:

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^N D_i}{N}} \quad (18)$$

where

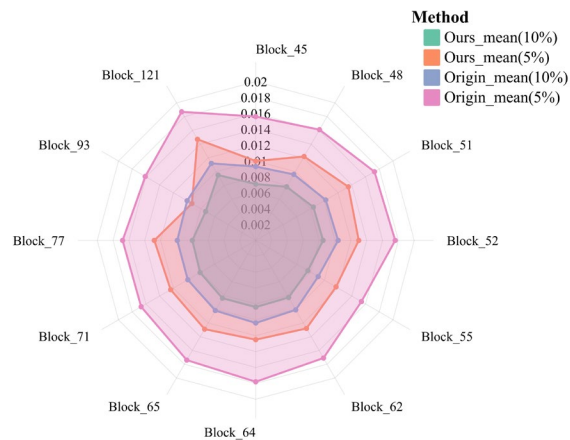
D_i = the distance from x_i to RefMesh.

3.4 Results

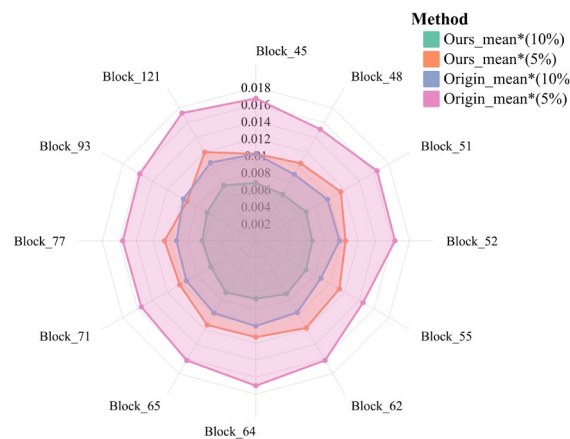
To comprehensively evaluate our algorithm's improvement over the baseline, we tested simplification on 3D real-scene models at 5% and 10% face retention ratios. The simplification results are shown in Figure 5.

For the simplification results, the mean distance of the simplified model relative to the original model and the mean distance of key components relative to their original counterparts were calculated separately. Aggregated data are presented in Figure 6 and Figure 7. In these figures, the proposed algorithm progressively reduces the mean distance compared to the baseline method at equivalent simplification ratios, improving the accuracy of simplified models while showing no significant increase in RMS values. When maintaining equivalent model accuracy (Figure 8), the proposed algorithm reduces face count by approximately 23% relative to the baseline method.

Our method facilitates conservative simplification of key elements alongside aggressive compression of secondary elements like terrain and clutter, preserving primary information during data reduction. For roofs and building façades, the simplification algorithm in this paper is able to retain the details of the model while simplifying the planar areas in it, based on the semantic information and the planar extraction results, compared to the original simplification algorithm. For the ground and clutter, the approach in this paper adopts a more aggressive simplification strategy. The results are shown in Figure 9.



(a) All points



(b) Points of important parts

Figure 6. Mean distance from simplified mesh to reference mesh.

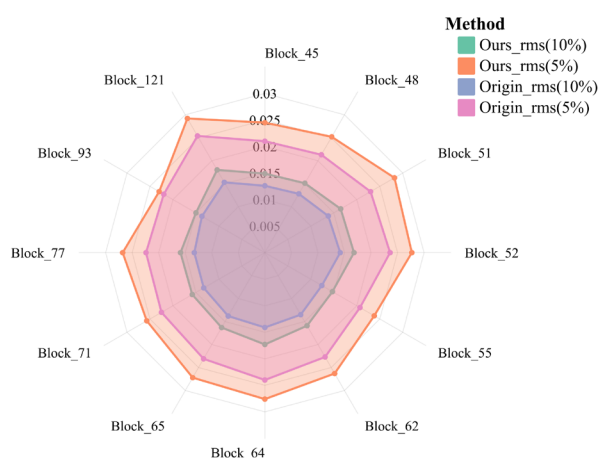


Figure 7. Root mean square of simplified mesh.

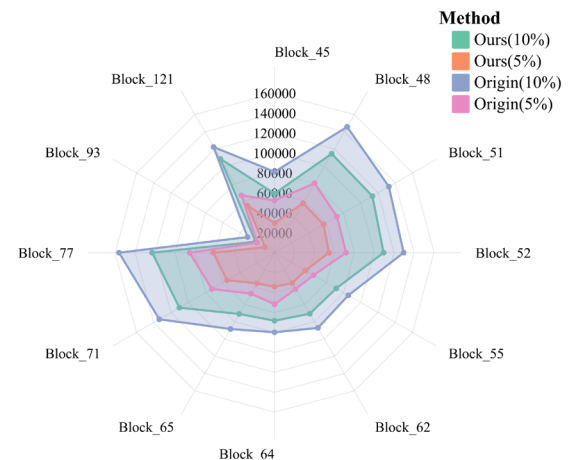


Figure 8. Face Count at Equivalent Geometric Accuracy. Achieved by the baseline algorithm when matching the accuracy thresholds of our method at 10% and 5% simplification ratios.



Figure 9. The figure shows the simplification results of the original method and our method for four classes at 5% and 10% simplification rates. From left to right are roof, building façade, ground, clutter

4. Discussion and Conclusion

In this study, we propose an integrated approach to enhance lightweight processing of 3D mesh models by combining semantic information with QEM method. Our work extends the QEM algorithm—the most widely adopted mesh simplification method—through a semantics-guided priority adjustment mechanism. First, a region growing-based plane detection algorithm establishes a plane detection pipeline for triangular mesh data. Through input data triangulation, redundant vertex removal, and coordinate normalization, data quality and standardization are improved. On this basis, region growing via least-squares plane fitting extracts planar features using threshold conditions. Subsequently, a plane segmentation-based point cloud classification method is proposed. By integrating planar features with geometric attributes, nine geometric-semantic features are designed. Combined with a probability smoothing strategy, rapid classification without model training is achieved. Finally, semantic information is innovatively integrated into traditional edge collapse simplification, proposing semantics-driven simplification. By combining semantic information with the simplification workflow, dynamic semantics-aware priority adjustment enables conservative simplification for critical elements and aggressive compression for secondary elements thus preserving essential information during simplification.

Experiments demonstrate our method achieves higher model accuracy at equivalent simplification rates. However, region growing-based plane detection suffers from local over-segmentation due to random seed distribution. Feature weighting in classification remains empirical. Future work will implement multi-source region growing to enhance robustness and develop automatic optimization and multi-scale features.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 42201474, 42192581), the State Key Laboratory of Micro-Spacecraft Rapid Design and Intelligent Cluster (Grant No. MS01240125), Hubei Provincial Natural Science Foundation of China (2024AFD8), Tianjin Key Laboratory of Spatio-Temporal Information Engineering and Technology (Grant No. 20241210), Scientific Research Project of China Railway Construction Corporation Limited (2024-W31) and Guangdong Provincial Key Laboratory of Smart and Safety City Planning and Monitoring (Grant No. GPKLIUSMSCP-2024-KF-01).

References

- Cignoni, P. et al., 2023. The Visualization and Computer Graphics Library (VCG). Version 2023. Visual Computing Lab of the Italian National Research Council. vcglib.net. (1 July 2024)
- Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., et al. 1996: Simplification envelopes, *International Conference on Computer Graphics and Interactive Techniques*, 119-128, doi.org/10.1145/237170.237220.
- Garland, M., Heckbert, P. S., 1997: Surface simplification using quadric error metrics, *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, 209-216, doi.org/10.1145/258734.258849.
- Hoppe, H., 2023: Progressive Meshes, *Seminal Graphics Papers: Pushing the Boundaries*, 111–120, doi.org/10.1145/3596711.3596725.
- Hoppe, H., 1997: View-dependent refinement of progressive meshes, *International Conference on Computer Graphics and Interactive Techniques*, 189-198, doi.org/10.1145/258734.258843.
- Kalvin, A. D., Taylor, R. H., 1996 : Superfaces: polygonal mesh simplification with bounded error, *IEEE Computer Graphics and Applications*, vol. 16, no. 3, 64–77, doi.org/10.1109/38.491187.
- Lounsbery, M., DeRose, T. D., Warren, J. 1997: Multiresolution analysis for surfaces of arbitrary topological type, *ACM Transactions on Graphics*, vol. 16, no. 1, 34–73, doi.org/10.1145/237748.237750.
- Lindström, P. ,2000: Out-of-core simplification of large polygonal models,” *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 259-262, doi.org/10.1145/344779.344912.
- Pan, Z., Zhou, K., Shi, J., 2001: A new mesh simplification algorithm based on triangle collapses, *Journal of computer science and technology*, vol. 16, no. 1, 57-63, doi.org/10.1007/bf02948853.
- Ronfard, R., Rossignac, J., 1996: Full-range approximation of triangulated polyhedra.,” *Computer Graphics Forum*, vol. 15, no. 3, 67–76, doi.org/10.1111/1467-8659.1530067.
- Rossignac, J., Borrel, P., 1993: Multi-resolution 3D approximations for rendering complex scenes, *Modeling in computer graphics: methods and applications*, 455–465. doi.org/10.1007/978-3-642-78114-8_29.
- Schroeder, W. W., Zarge, J. A., Lorensen, W. E., 1992: Decimation of triangle meshes, *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 65-70, doi.org/10.1145/133994.134010.
- The CGAL Project, 2024. The Computational Geometry Algorithms Library (CGAL). Version 6.0.1. CGAL Editorial Board. cgal.org. (1 July 2024)
- Xia, J. C., Varshney, A., 1996: Dynamic view-dependent simplification for polygonal models, *Proceedings of Seventh Annual IEEE Visualization'96*, 327-334, doi.org/10.1109/visual.1996.568126.