The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-1/W5-2025
13th International Conference on Mobile Mapping Technology (MMT 2025)
"Mobile Mapping for Autonomous Systems and Spatial Intelligence", 20–22 June 2025, Xiamen, China

Patch-based Graph Cut Optimization for 3D Line Segment Extraction of Building Structures from Outdoor Point Cloud Data

Ruoming Zhai[1], Peng Wan[1], Xianquan Han[1], Jianzhou Li[1], Yifeng He[2], Bangning Ding[1]

[1] Changjiang River Scientific Research Institute, Wuhan, China – rmzhai@mail.crsri.cn, wanpeng@mail.crsri.cn, hanxq@mail.crsri.cn, lijianzhouljz@163.com, bnding@ mail.crsri.cn

[2] School of Geodesy and Geomatics, Wuhan University, Wuhan, China – heyifeng@whu.edu.cn

Keywords: Point cloud, line extraction, building structures, plane segmentation, graph-cut.

Abstract:
The extraction of architectural structural line features can simplify the 3D spatial representation of monitored objects, reduce the storage burden of massive point clouds, and provide crucial geometric parameters for subsequent 3D modelling. To address the problem of line segment discontinuity caused by point cloud quality, this paper proposes a line segment extraction algorithm based on planar graph-cut segmentation. Specifically, a region growing and merging algorithm is employed to segment the point cloud into multiple planar patches. Subsequently, 3D-2D projection is performed to generate grayscale images, from which line segment parameters are identified and extracted using a graph-cut segmentation algorithm combined with vectorization techniques. Then, depending on spatial 3D-2D back-projection, all extracted 3D line segments are optimized by the structural regularization to further mitigates the discontinuity among collinear line segments and provide high-quality linear features for subsequent 3D reconstruction.

1. Introduction

Leveraging its non-contact measurement, high speed, and high precision, three-dimensional (3D) laser scanning technology enables the efficient acquisition of large-scale, high-precision point cloud data of urban buildings, which provides essential support for digital twin applications, such as digital city development (Kase et al., 2025), cultural heritage preservation (Ursini et al., 2022), and smart infrastructure management (Xu et al., 2021). Meanwhile, building geometry representations using massive point clouds require more data storage and affect the efficiency of subsequent geometric modelling. Therefore, extracting 3D line features can accurately capture the main structural contours of buildings, simplify the digital representation, and reduce data redundancy, providing precise wireframe models for large-scale 3D reconstruction. Line feature extraction can be categorized into point-based and patch-based methods, depending on the geometry representation types of point clouds.

Point-based methods derive the geometric representation of each point within its neighbourhood, which includes eigenvalue-based features (Chen et al., 2023), spatial gradients (Xia et al., 2017; Chen et al., 2021), centroid offsets (Xin et al., 2024), and high-dimensional feature vectors derived from deep learning (Jiang et al., 2023; Zhang et al., 2023). Through adjusting threshold conditions based on variations in geometric representations, edge points are further filtered for line feature representations. However, their performance is highly sensitive to the quality of the point cloud data, and the extracted line features are typically represented as discrete edge points, lacking geometrically parameterized representations.

On the other hand, plane methods involve performing plane segmentation followed by the extraction of plane edge lines, which can be categorized into projection slicing-based, plane intersection-based, and patch-based methods. The projection slicing-based method (Zhao et al., 2022; Tian et al., 2022) slices the point cloud along different axes and aggregates all boundary points, obtaining the final building contour lines through 3D line segment fitting. The plane intersection-based method (Cui et al., 2019; Han et al., 2021; Chen et al., 2024) selects the optimal set of intersection segments as the final line segments by employing a global energy optimization algorithm according to the plane intersection relationships. However, when the number of extracted planes increases significantly, the energy optimization algorithm may suffer from low computational efficiency and may not yield satisfactory results for irregular geometric structures. The patch-based method clusters points that satisfy distance threshold conditions within the same plane to generate patches. It generally employs plane segmentation algorithms such as Random Sample Consensus (RANSAC), supervoxel (Mi et al., 2021), and region growing (Hu et al., 2022) to extract multiple patches, followed by line segment fitting through the identification of patch contours. Some studies prefer adopting projection-based methods, where points within the patches are projected onto a 2D plane for 2D line segment extraction using methods such as α-shape, LSD, and CannyLines, followed by 3D-2D reprojection to obtain 3D line segments. Due to the influence of point cloud noise, incompleteness, and uneven density, directly fitting 3D line segments in 3D space can lead to issues such as discontinuities in the line segments, unclosed intersections, and angular deviations, making it difficult to accurately represent the true architectural contours. Due to the influence of point cloud noise, incompleteness, and uneven density, these methods often encounter issues such as discontinuous line segments, unclosed intersections, and angular deviations, which hinder the accurate representation of architectural contours, especially in large-scale outdoor building scenes.

To address these issues, this study proposes a 3D line extraction method for point clouds based on patch segmentation and graph-cut partitioning, which extracts 3D line features by leveraging projected points for 2D line detection and back-projects them to reconstruct building contours in large-scale outdoor scenes.

2. Methodology

2.1 Technical framework

This study consists of three key components:
(1) The geometric features of point clouds are computed using the PCA algorithm to provide clustering constraints for region growing and merging in patch segmentation.
(2) For each extracted patch, a 3D-to-2D projection is performed to generate an image, followed by graph-cut segmentation for 2D line extraction, and then a 2D-to-3D back-projection to obtain the initial patch line sets. (3) The adjacent patch constraint and structural constraint are applied to refine patch lines, eliminating discontinuities, unclosed intersections, and angular deviations, and ultimately producing the final 3D line segments depicting building contours.

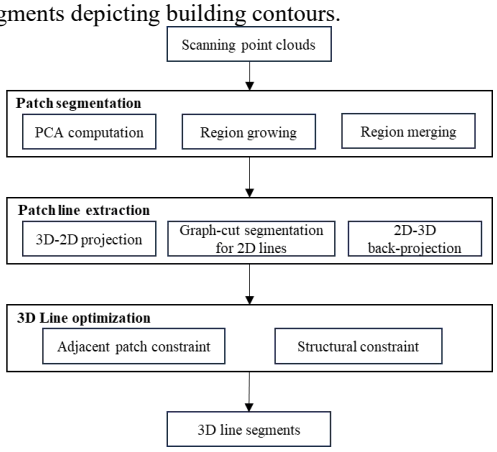


Figure 1. The workflow of 3D line extraction.

2.2 Patch segmentation

Region Growing. To facilitate geometric clustering of point clouds, KD-tree indexing is applied for neighbourhood search, followed by covariance matrix computation to obtain eigenvalues λ_1, λ_2, λ_3 ($\lambda_1 > \lambda_2 > \lambda_3$)and their corresponding eigenvectors. The normal vector n is determined by the eigenvector associated with λ_3, and curvature is computed as:

$$C = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{1}$$

Then, A distance scale parameter d_s is defined as the distance to the third nearest neighbour, ensures generalization to point cloud scenes of different scales. Region growing begins by sorting points according to curvature C, selecting seed points from 90% of the lowest-curvature points, and expanding clusters based on three constraints:

$$\cos^{-1}\left(n_s^i \cdot n_j^i\right) < \theta_n \tag{2}$$

$$\left| n_s^i \cdot \overline{P_s^i P_j^i} \right| < d_{ortho} \tag{3}$$

$$\left|\overline{P_s^i P_j^i}\right| < d_{proj} \tag{4}$$

where θ_n denotes the angle between the normal vectors of p_s^0 and p_j^0, serving to constrain the angular deviation between the seed point and candidate points, d_{ortho} defines the perpendicular distance from the candidate point to the neighborhood plane along the seed point's normal direction, and d_{proj} is set to $50d_s^i$ adaptively determining the distance threshold between the seed point and candidate points, with d_s^i denoting the distance between a point and its third nearest neighbor.

Points satisfying all conditions are added to a temporary cluster. This process iterates until no new points meet the criteria.

Region Merging. Since initial clustering often results in over-segmentation, region merging is conducted to form larger planar patches. Geometric properties are computed for each cluster, and adjacency is determined based on overlapping points. Regions are merged if they satisfy:

$$\arccos\left(n_i \cdot n_j\right) < \theta_{p\_n} \tag{5}$$

$$\left|\left(p_i - p_j\right) \cdot n_i\right| < d_{p\_ortho} \tag{6}$$

where θ_{p_n} and d_{p_ortho} are the angle and distance thresholds among cluster i and j. Clusters meeting both constraints are merged and, if a merged region contains more than 100 points, it is added to the final patch set, with the process iterating until all candidate regions are processed.

2.3 Patch line extraction

2.3.1 3D-2D Projection

In the 3D-2D projection for image generation, points within a patch are projected onto a fitted plane, thereby defining a corresponding plane coordinate system for contour extraction and 2D line segment fitting. The plane equation is established based on a point $P_0\left(x_0, y_0, z_0\right)$ and the normal vector $n_0\left(n_x, n_y, n_z\right)$ within the patch as follows:

$$\frac{x - x_0}{n_x} = \frac{y - y_0}{n_y} = \frac{z - z_0}{n_z} \tag{7}$$

Assuming a point P_i within the patch is projected along the normal vector n_0 onto the plane, the unit vector u_x between the projected point and the centroid is treated as the x-axis, while the cross product of u_x and n_0 defines the y-axis, forming the plane's coordinate system. The projected coordinates of points on the fitted plane are formulated as follows:

$$\begin{cases} x_i = \left(\overrightarrow{P_0 P_i'} - \overrightarrow{P_i P_i'}\right) \cdot u_x \\ y_i = \left(\overrightarrow{P_0 P_i'} - \overrightarrow{P_i P_i'}\right) \cdot u_y \\ \overrightarrow{P_i P_i'} = \left(\overrightarrow{P_0 P_i'} \cdot n_0\right) \cdot n_0 \end{cases} \tag{8}$$

where $\left(x_i, y_i\right)$ is pixel coordinates for each point and P_i' is the projected point of P_0.

2.3.2 Graph-cut method for 2D line segment extraction

According to the image coordinate system established above, the projected grayscale image of the patch can be obtained, where pixels corresponding to the projected point cloud are set to 255, and all other regions are set to 0.

In the graph-cut segmentation algorithm, the image is discretized into an undirected graph $G = (V, E)$, where each node represents a pixel in the set of vertices V, and the edge set E defines the connectivity between nodes. Connectivity is constructed using the 8-neighborhood relationship, with edge weights representing the connectivity between adjacent pixels. These weights are formulated as follows:

$$w(u,v) = \frac{\left\|(x_u, y_u) - (x_v, y_v)\right\|}{\sigma} \tag{10}$$

where (x_u, y_u) and (x_v, y_v) represent the coordinates of pixels u and v, respectively, and σ is a smoothing factor that controls the sensitivity of edge weights to distance. This weight function ensures that pixels with greater spatial distances exhibit lower connectivity, effectively reducing boundary redundancy.

Then, the cut-pursuit approach formulates the minimization optimization function for region segmentation over the constructed weighted graph:

$$F(x) = \sum_{v \in V} f\left(y_v, x_v\right) + \sum_{(u,v) \in E} w(u,v)\left\|x_u - x_v\right\|_0 \tag{11}$$

where the loss function $f\left(y_v, x_v\right)$ measures the differences between the pixel label x_v and the ground true label y_v. The l_0 pseudo-norm $\left\|x_u - x_v\right\|_0$ represents the label differences between adjacent pixels. Finally, the cut-pursuit method iteratively solves the energy minimization problem to partition the image graph into optimal connected components. Once the image has been accurately segmented, the widely used Potrace algorithm is employed to obtain compact and closed boundary lines, as shown in Figure 2.

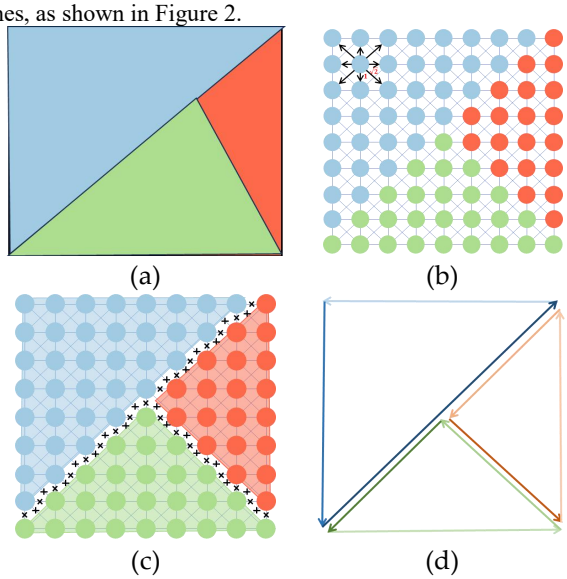


Figure 2. Graph-cut method for vectorized lines generation.

2.4 3D Line optimization

Due to the errors caused by 2D line segment fitting in the image, the same line segment extracted from the contours of different patches may result in two non-overlapping 3D line segments

when back-projected into 3D space. Therefore, this study leveraging intersecting line constraints from adjacent patches, aiming to further optimize the extracted line segments using the planar characteristics of the patches.

As illustrated in Figure 3, for each intersection line, adjacent patches are identified, and their line segment sets are searched for those with similar direction and proximity, which are then adjusted to align their direction vectors with the intersection line, optimizing the segments while preserving the planar properties of the patches.

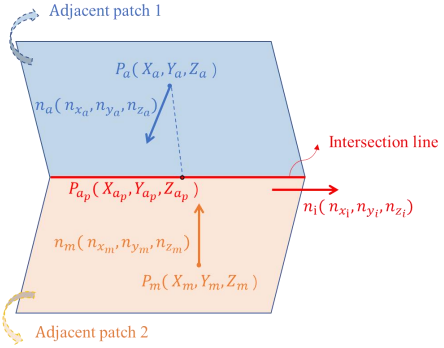


Figure 3. Intersection line spatial relationship.

Specifically, Equation (12) represents the vector angle between the direction vector of the intersection line and the direction vector of a line segment within the corresponding patch, while Equation (13) defines the perpendicular distance between the projections of the two endpoints of the line segment onto the intersection line.

$$\cos^{-1}\left(v_{L_I}\cdot v_{L_R}\right)<\theta_L \tag{12}$$

$$\begin{cases}\overrightarrow{P_{L_R^0}P_{L_I}}-\left|v_{L_I}\cdot\overrightarrow{P_{L_R^0}P_{L_I}}\right|\cdot v_{L_I}<d_I\\ \overrightarrow{P_{L_R^1}P_{L_I}}-\left|v_{L_I}\cdot\overrightarrow{P_{L_R^1}P_{L_I}}\right|\cdot v_{L_I}<d_I\end{cases} \tag{13}$$

Additionally, to address discontinuity and non-collinearity in the extraction of certain non-intersecting line segments, the structural collinear optimization is applied to merge collinear segments and reduce unnecessary redundancy, and it considers three spatial relationships for collinear improvement, as shown in Figure 3.

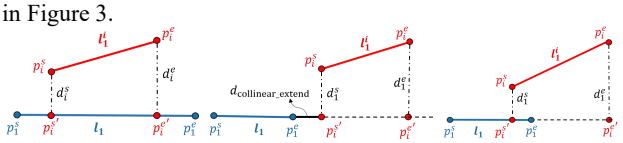
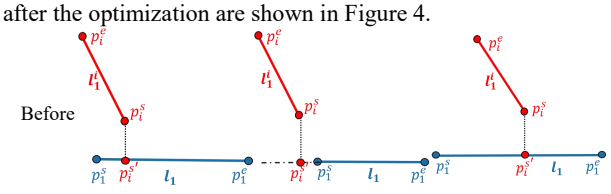


Figure 4. The structural collinear optimization results.

Then, it is followed by the structural perpendicular optimization considering three spatial relationships, which connects unclosed line segments and corrects orthogonal angles by extending segments or adjusting their orientations. The results before and after the optimization are shown in Figure 4.



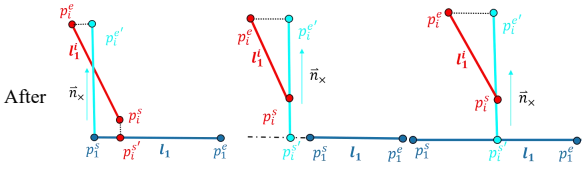
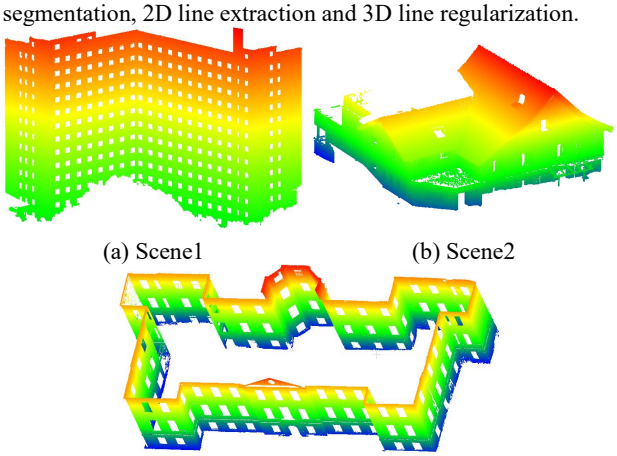


Figure 5. The structural perpendicular optimization results.

3. Experiment

3.1 Dataset deployment

To validate the effectiveness of the proposed method for extracting 3D line features in outdoor building structures, we utilized ground-based LiDAR sensor, Leica P50, to scan structural facades of the building with multiple windows (Scene1), and selected two additional outdoor building scenes from Semantic3D (Scene2) and Oxford-Spires (Scene3) datasets, as shown in Figure 6. Comparative experiments were conducted to demonstrate the generalization of our proposed method in large-scale and complex building structures, focusing on patch segmentation, 2D line extraction and 3D line regularization.



(a) Scene1 (b) Scene2

(c) Scene3

Figure 6. Point cloud dataset for outdoor building scenes.

3.2 Result of patch segmentation

In the patch segmentation stage, the number and quality of segmented patches are determined by the angular threshold used in the region growing and merging processes. When the threshold is set too low, over-segmentation occurs during region growing, resulting in an excessive number of point cloud clusters. In the subsequent merging stage, clusters with only a small number of points are often regarded as noise and filtered out, which may lead to the loss of critical structural features. Conversely, when the threshold is too high, the loss of clusters is reduced; however, patches belonging to different planes may be erroneously merged, leading to missing line segments during subsequent 2D line extraction. To balance noise filtering and patch fragmentation, a range of angular thresholds—5°, 10°, 15°, and 20°—is tested for patch extraction. The corresponding results are presented in Table 1. In Figure 7, it can be observed that an angular threshold of 15° yields the most effective extraction of valid line segments.

Table 1. results of patch segmentation in different angular threshold constraints.

Angle (°)	Metrics (%)	Scene1	Scene2	Scene3
5	Drop rate	6.96	23.7	23.1
	Cluster amount	22	84	296

10	Drop rate	4.77	14.3	12.9
	Cluster amount	23	90	335
15	Drop rate	3.81	10.6	9.3
	Cluster amount	10	82	309
20	Drop rate	3.42	9.0	7.6
	Cluster amount	9	82	232

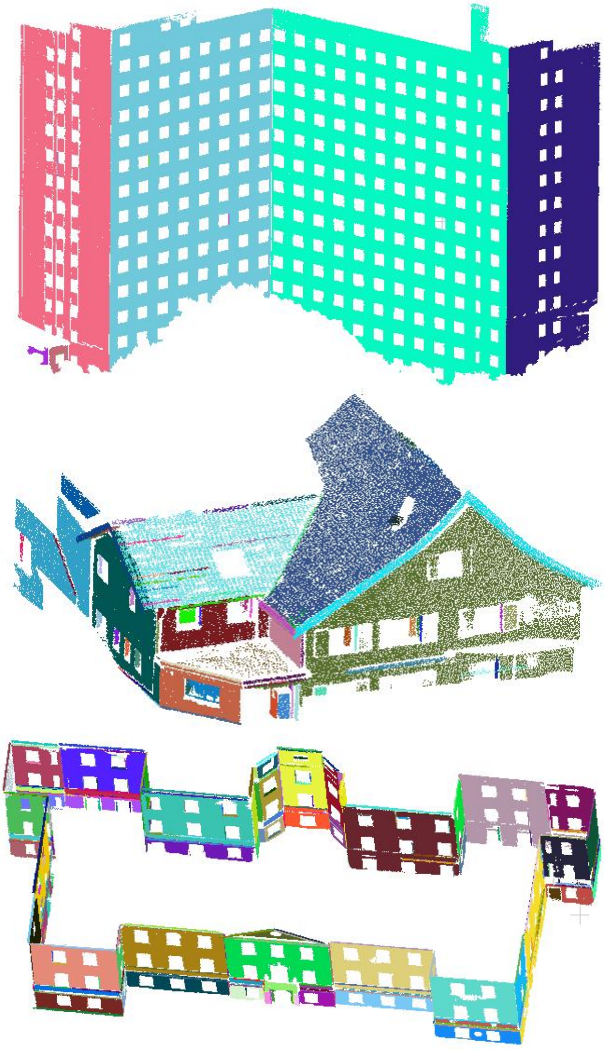


Figure 7. Visualization results of patch segmentation.

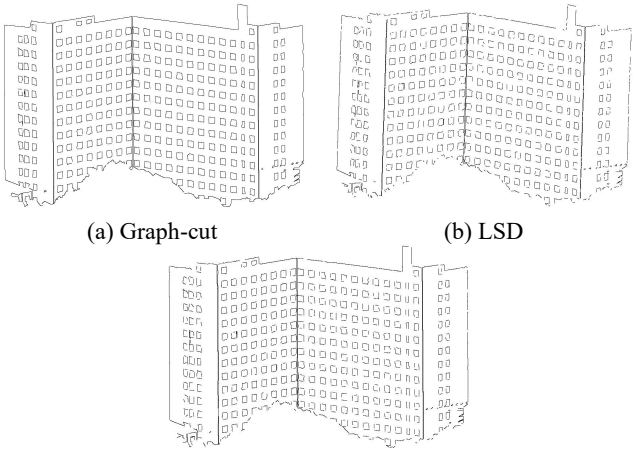
3.3 Result of 2D line extraction

Based on the extracted patches, we perform a 3D-2D spatial projection to establish a local planar coordinate system for each patch, facilitating 2D line segment detection. Three algorithms, namely Graph-cuts, LSD, and FLD, are employed to evaluate the effectiveness of different 2D line detection and extraction methods. The detected 2D line segments are then re-projected into 3D space to generate a set of 3D line segments.

Table 2. Amount of 3D line segments in different 2D line detection algorithms.

Number of 3D line segments	Graph-cut	LSD	FLD
Scene1	1505	1687	1552
Scene2	542	542	427
Scene3	2393	2738	2121

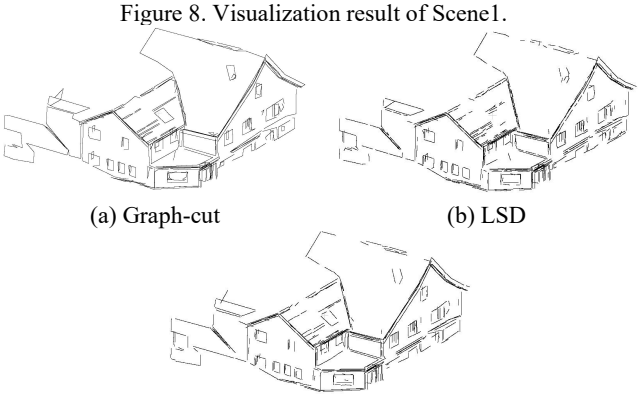
As illustrated in Table 2 and Figure 8, both LSD and FLD extract many line segments, yet their completeness is relatively low, primarily due to the over-segmentation of individual line structures into multiple shorter segments. In contrast, the Graph-cuts-based method used in this study extracts fewer line segments but achieves higher precision, indicating an improvement in the continuity of 2D line detection. However, there remains some discontinuities, particularly among vertical line segments, resulting in incomplete closures. Moreover, sparse point cloud distributions along linear patterns tend to induce the extraction of redundant or spurious line segments. The proposed method emphasizes the extraction of closed line structures and demonstrates greater robustness to density variations caused by linear scanners when compared to alternative methods.



(a) Graph-cut (b) LSD

(c) FLD

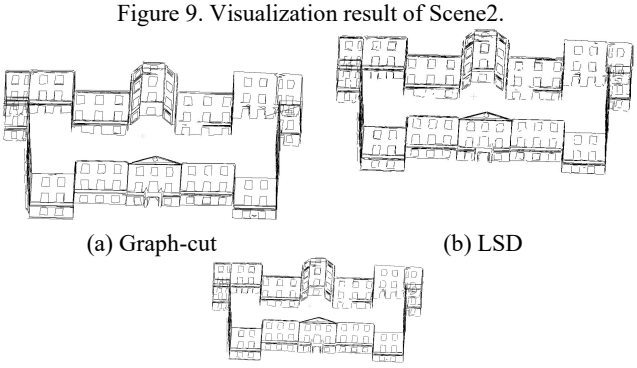
Figure 8. Visualization result of Scene1.



(a) Graph-cut (b) LSD

(c) FLD

Figure 9. Visualization result of Scene2.



(a) Graph-cut (b) LSD

(c) FLD

Figure 10. Visualization result of Scene3.

3.4 Result of 3D line extraction

To address the issue of disconnected intersections caused by incomplete point cloud data, we propose a line segment

optimization algorithm that incorporates structural collinear and perpendicular regularization. Specifically, thresholds for intersection distance and angle are set to 0.3m and 15°, respectively; for collinearity, the distance and angle thresholds are 0.3 meters and 40°; and for orthogonality, they are set to 0.3 meters and 70°. As illustrated in Figure 9, most line segments belonging to the same linear structure are successfully merged, and intersections at orthogonal directions are effectively closed, resulting in improved structural continuity and regularity.

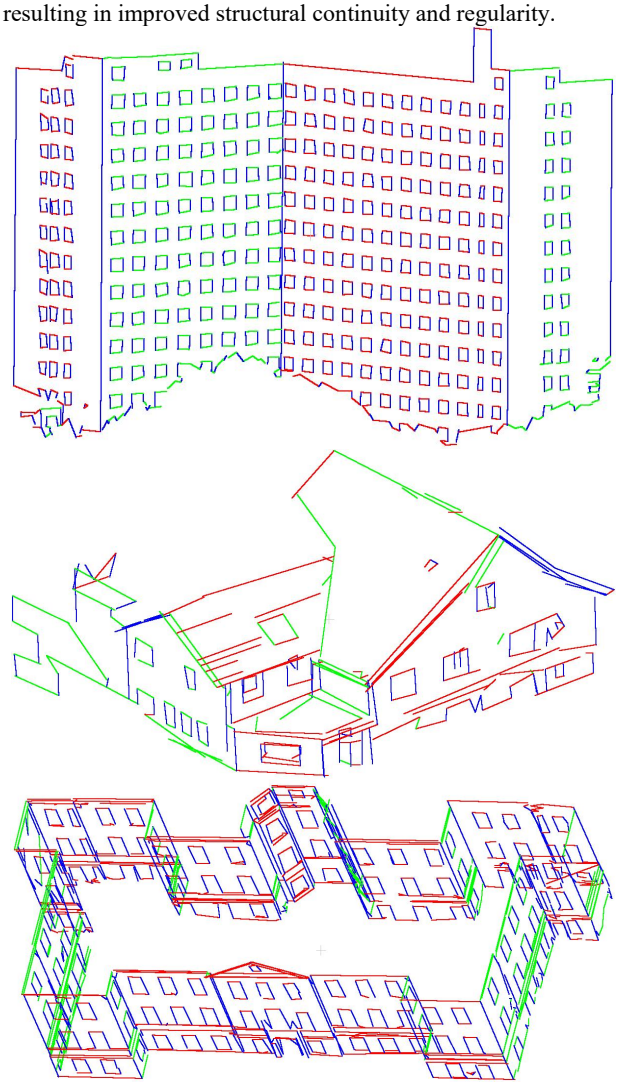


Figure 9. Visualization results of structural regularization.

4. Conclusion

This paper proposes a point cloud line feature extraction algorithm based on planar graph-cut segmentation, aiming to provide structurally consistent and geometrically accurate architectural line features for 3D reconstruction. Experiments conducted on three outdoor architectural scenes demonstrate that the proposed method effectively alleviates the issue of unclosed orthogonal line segments through graph-cut segmentation. The incorporation of adjacency constraints successfully merges redundant line segments and reduces angular deviation to some extent. Furthermore, collinearity and orthogonality optimization constraints help address the common problems of line discontinuity and non-closure, significantly reducing data redundancy. Overall, the proposed method improves the geometric completeness of extracted line features and enhances the accuracy of architectural contour extraction. However, the method still exhibits limitations in extracting line segments in regions with sparse or missing point cloud data. Further optimization is required, particularly regarding the adaptive selection of parameters in the architectural geometric regularization constraints.

References

Chen D, Li J, Di S, et al. Critical Points Extraction from Building Façades by Analyzing Gradient Structure Tensor[J]. Remote Sensing, 2021, 13(16): 3146.

Chen D, Wan L, Hu F, et al. Semantic-aware Room-level Indoor Modeling from Point Clouds[J]. International Journal of Applied Earth Observation and Geoinformation, 2024, 127: 103685.

Chen X, Zhao B. An Efficient Global Constraint Approach for Robust Contour Feature Points Extraction of Point Cloud[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-16.

Cui Y, Li Q, Yang B, et al. Automatic 3-D Reconstruction of Indoor Environment with Mobile Laser Scanning Point Clouds[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(8): 3117-3130.

Hackel T, Savinov N, Ladicky L, et al. Semantic3d. net: A new large-scale point cloud classification benchmark[J]. arXiv preprint arXiv:1704.03847, 2017.

Han J, Rong M, Jiang H, et al. Vectorized Indoor Surface Reconstruction from 3D Point Cloud with Multistep 2D Optimization[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2021, 177: 57-74.

Hu Z, Chen C, Yang B, et al. Geometric Feature Enhanced Line Segment Extraction from Large-scale Point Clouds with Hierarchical Topological Optimization[J]. International Journal of Applied Earth Observation and Geoinformation, 2022, 112: 102858.

Jiang T, Wang Y, Zhang Z, et al. Extracting 3-D Structural Lines of Building from ALS Point Clouds Using Graph Neural Network Embedded with Corner Information[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-28.

Kase T, Hasegawa K, Watanabe K, et al. Real-time Point Cloud Visualization for Sustainable Spatial Digital Twins[C]//2025 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2025: 1-2.

Mi X, Yang B, Dong Z, et al. Automated 3D Road Boundary Extraction and Vectorization Using MLS Point Clouds[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(6): 5287-5297.

Tao Y, Muñoz-Bañón M Á, Zhang L, et al. The Oxford Spires Dataset: Benchmarking Large-Scale LiDAR-Visual Localisation, Reconstruction and Radiance Field Methods[J]. arXiv preprint arXiv:2411.10546, 2024.

Tian P, Hua X, Tao W, et al. Robust Extraction of 3D Line Segment Features from Unorganized Building Point Clouds[J]. Remote Sensing, 2022, 14(14): 3279.

Ursini A, Grazzini A, Matrone F, et al. From Scan-to-BIM to A Structural Finite Elements Model of Built Heritage for Dynamic Simulation[J]. Automation in Construction, 2022, 142: 104518.

Xin X, Huang W, Zhong S, et al. Accurate and Complete Line Segment Extraction for Large-scale Point Clouds[J]. International Journal of Applied Earth Observation and Geoinformation, 2024, 128: 103728.

Xu Y, Stilla U. Toward Building and Civil Infrastructure Reconstruction from Point Clouds: A Review on Data and Key Techniques[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2021, 14: 2857-2885.

Zang Y, Chen B, Xia Y, et al. LCE-NET: Contour Extraction for Large-scale 3-D Point Clouds[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-13.

Zhao J, Cui Y, Niu X, et al. Point Cloud Slicing-based Extraction of Indoor Components[J]. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2022, 48: 103-108.