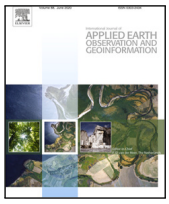
International Journal of Applied Earth Observation and Geoinformation 147 (2026) 105182

Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

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



Constrained planar primitive segmentation for building component extraction from airborne LiDAR point clouds

Hongxin Yang [a], Zhipeng Luo [b], Dedong Zhang [c], Jonathan Li [a,c,*]

[a] School of GeoAI and Hinton STAI Institute, Key Laboratory of Geographic Information Science (Ministry of Education), East China Normal University, Minhang, Shanghai 200241, China
[b] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian 363000, China
[c] Department of Geography and Environmental Management, University of Waterloo, Waterloo, Ontario N2L3G1, Canada

ARTICLE INFO

Keywords:
Planar primitive segmentation
Constrained k-Plane Clustering
Non-convex optimization
Point clouds
ALS
3D building modeling

ABSTRACT

Accurately segmenting planar primitives from airborne LiDAR point clouds is crucial for urban planning applications and three-dimensional (3D) building reconstruction. However, existing approaches often exhibit limited extensibility and diminished robustness when segmenting buildings with diverse architectural styles. To address this issue, this paper proposes an enhanced constrained k-Plane Clustering (kPC) method that segments building point clouds into distinct planar primitives. The kPC algorithm formulates the segmentation task as a mixed-integer non-convex optimization problem incorporating three geometric constraints: point-to-plane distance minimization, cluster-center proximity enforcement, and directional regularization. This problem is solved via an alternating minimization strategy, which iteratively updates cluster assignments and plane parameters using Singular Value Decomposition (SVD) until convergence is reached. The proposed constrained kPC method provides two key advantages. First, it effectively mitigates the infinite extensibility of fitted planes. This issue is common in conventional kPC methods, and its mitigation directly addresses a major source of suboptimal segmentation performance. Second, the framework demonstrates robustness to variations in the optimization objective's coefficients while maintaining consistent performance. Extensive experiments on both synthetic and real-world multi-style building datasets demonstrate that the proposed method achieves superior segmentation accuracy and outperforms state-of-the-art approaches in both qualitative and quantitative evaluations. Furthermore, the performance of the proposed method is robust to variations in its parameters, maintaining effectiveness across a wide range of values.

1. Introduction

Three-dimensional (3D) building models are fundamental to various geospatial applications, such as smart city development (Huang et al., 2022; Nan and Wonka, 2017), urban planning, and automated building reconstruction (Haala and Kada, 2010; Ochmann et al., 2019; Li and Shan, 2022). Therefore, acquiring accurate planar primitives from raw building data is a critical preliminary step.

Current methods for segmenting building planar primitives from LiDAR point clouds fall into two groups: Deep Learning (DL)-based approaches and traditional methods. In DL-based roof segmentation, Zhang and Fan (2022) incorporated mean-shift clustering to group geometrically similar planes, while Li et al. (2024) introduced boundary-aware clustering to refine plane delineation through multi-task learning. While these methods have advanced roof segmentation through integrated clustering techniques, they generally depend on large-scale, finely annotated datasets and involve computationally intensive processing steps. Traditional methods can be further classified into: model fitting-based, region growing-based, and feature clustering-based approaches. Model-fitting approaches extract planar surfaces by identifying point subsets that conform to geometric models. In contrast, region-growing methods iteratively aggregate points into planar segments. This process begins from seed points and employs geometric similarity criteria. Feature clustering-based approaches focus on the feature embedding. However, as these techniques are fundamentally based on point-prototype clustering, they often fail to adequately capture the inherent planar geometry of building roof structures. To address this limitation, Kong et al. (2013) introduced a distinct k-Plane Clustering (kPC) method. This approach fits planes to clusters of laser points. It then computes the intersection lines of these planes. Finally, the method segments the building point cloud into multiple

* Corresponding author at: Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (J. Li).
E-mail address: jli@geoai.ecnu.edu.cn (J. Li).

https://doi.org/10.1016/j.jag.2026.105182
Received 21 November 2025; Received in revised form 3 February 2026; Accepted 15 February 2026
Available online 26 February 2026

planar primitives. Nevertheless, the infinite extensibility of fitted planes in this method leads to misclassification and involves cumbersome intermediate calculations.

To overcome these limitations, this paper proposes an optimized constrained kPC framework for building planar primitive segmentation. The proposed method formulates this primitive extraction task as a mixed-integer non-convex optimization problem, incorporating three geometric constraints: point-to-plane distance, cluster-center proximity, and directional regularization. The directional regularization term specifically minimizes the distance between fitting plane boundaries and cluster centers to prevent unbounded plane extension. This optimization is solved through an alternating minimization algorithm that iteratively refines cluster assignments and plane parameters until convergence. The algorithm begins with random initialization of hyperplane parameters and cluster centers. During the assignment step, cluster labels are allocated to each point by solving the non-convex objective function through Singular Value Decomposition (SVD). The subsequent update step leverages current cluster assignments to optimize normal vectors and inter-plane thresholds. Following each parameter update, points are reassigned to clusters based on the refined geometric constraints. This iterative two-stage process continues until stable convergence is achieved, typically within few iterations.

The contributions of the proposed method are threefold:

- A novel constrained kPC strategy is proposed to mitigate segmentation artifacts arising from the unbounded extensibility of fitted planes in building reconstruction.
- The proposed kPC framework utilizes a non-convex optimization formulation. This formulation integrates three geometric constraints: point-to-plane distance, cluster-center proximity, and directional regularization. Together, these enable the precise identification of building primitives.
- Extensive experimental results demonstrate the superior performance of the proposed method. Compared to other state-of-the-art approaches, it also exhibits stronger robustness when applied to buildings with diverse roof styles.

The paper is structured as follows. Section 2 reviews related work on planar primitive segmentation and the original kPC algorithm. Section 3 delineates the proposed methodology, including its theoretical motivation and optimization framework. Section 4 presents the experimental evaluation, comprising comparative analyses, ablation studies, and segmentation results. Finally, Section 5 concludes the paper and suggests future research directions.

2. Related work

This subsection provides a concise review of related work in building primitive segmentation and plane-prototype clustering methodologies.

2.1. Building primitive segmentation

Traditional methods for building primitive segmentation are briefly reviewed and categorized into three groups.

(1) *Model fitting-based methods:* This category includes widely adopted techniques such as the Hough Transform (HT) (Illingworth and Kittler, 1988) and Random Sample Consensus (RANSAC) (Choi et al., 1997). HT-based implementations (Du et al., 2011), for instance, operate by transforming points into a parameter space where each LiDAR point votes for candidate planes in an accumulator. Final plane parameters are determined by detecting local maxima in this space. Despite their conceptual simplicity, HT-based methods suffer from three major limitations: high computational complexity, sensitivity to parameter settings, and a tendency toward false-positive plane detections. Tarsha-Kurdi et al. (2007) conducted comprehensive evaluations demonstrating that RANSAC-based approaches generally achieve higher segmentation accuracy and lower computational cost compared to HT-based methods, although they remain prone to spurious plane detection.

(2) *Region growing-based methods:* This category of methods typically begins with selected seed points or regions and iteratively expands by aggregating neighboring points that satisfy predefined geometric similarity criteria. For example, Yan et al. (2014) proposed a global plane fitting approach that starts with an initial segmentation using conventional region growing, followed by energy minimization incorporating three core constraints: (a) point-to-plane distance minimization, (b) spatial smoothness regularization, and (c) plane count reduction. Zhang et al. (2025) introduced a voxel-based region growing method enhanced by robust principal component analysis (PCA), which leverages geometric features such as surface smoothness, continuity, and convexity to improve the extraction of roof structures. Despite these refinements, region growing-based methods suffer from two inherent limitations. First, their performance is highly dependent on the initial seed selection, which often leads to inaccurate boundary delineation between adjacent planar surfaces. Second, these methods often yield suboptimal results when segmenting architecturally complex buildings. This is particularly evident at junctions involving multiple surfaces. The suboptimal performance typically manifests as either over-segmentation, where continuous surfaces are excessively fragmented, or under-segmentation, where distinct planes are insufficiently separated.

(3) *Feature clustering-based methods:* Methods in this category typically start by applying an upward-growing algorithm to remove ground points (Sampath and Shan, 2009). Subsequently, the remaining non-ground points are clustered. Finally, planar surfaces such as rooftop facets are estimated using techniques like RANSAC or 2.5D dual contouring. For example, Li et al. (2020) introduced a hierarchical clustering technique that merges adjacent planar patches belonging to the same surface, augmented by a boundary relabeling mechanism for refined edge delineation. In contrast to point-prototype clustering techniques, Kong et al. (2013) proposed the kPC algorithm, which adopts a plane-oriented clustering paradigm. This method comprises three main stages. First, optimal planes are estimated from grouped LiDAR points. Second, intersection lines between adjacent planes are computed. Finally, the building point cloud is segmented based on inferred plane-topology relationships. Despite its advantages, the kPC algorithm suffers from a critical limitation: it relies on reasonably accurate initializations of cluster planes prior to optimization, which can compromise its robustness when segmenting buildings with complex roof geometries.

2.2. Plane-prototype clustering

(1) kPC algorithm (Bradley and Mangasarian, 2000): In contrast to point-prototype clustering methods such as k-means, the kPC algorithm seeks to identify k optimal planes that best represent a given set of data points. The objective function minimizes the orthogonal distances between the input points and their associated fitting planes, resulting in a quadratic programming formulation. This optimization is solved through an iterative procedure. The procedure alternates between two steps. First, each point is assigned to its closest plane based on orthogonal distance. Second, the parameters of each plane are updated using the points currently assigned to its cluster. For the ith cluster, the objective minimizes the sum of squared orthogonal distances between its constituent points and the corresponding fitting plane, subject to the constraint $\|\mathbf{w}_i\|_2^2 = 1$. The plane parameters $(\mathbf{w}_i, \gamma_i)$ are typically initialized randomly at the outset of the algorithm.

(2) Proximal plane clustering algorithm (Shao et al., 2013): Building upon the kPC framework, the Proximal Plane Clustering (PPC) method incorporates inter-cluster proximity information in addition to the intra-cluster coherence considered in the original formulation.

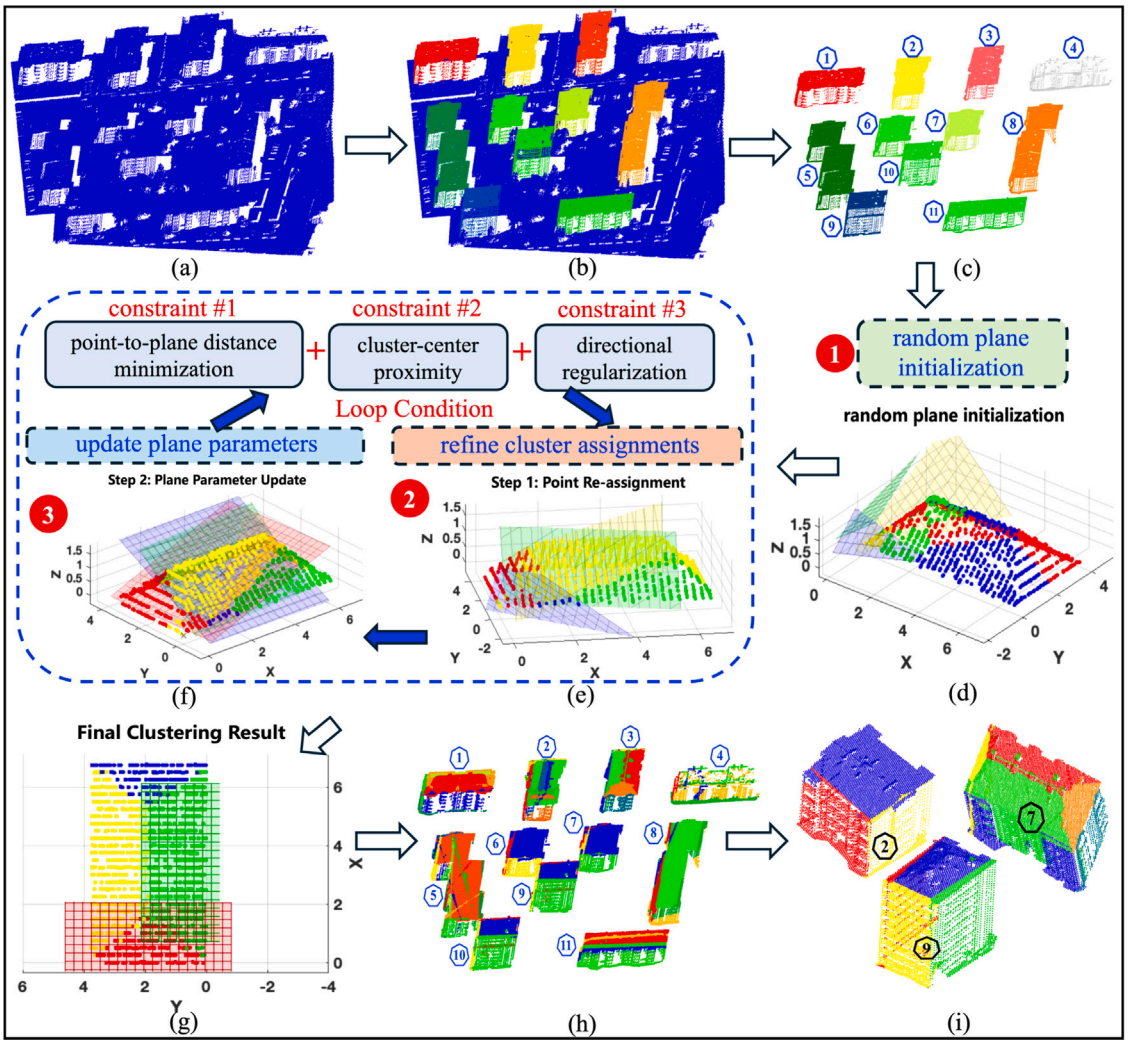


Fig. 1. Pipeline of the proposed building planar segmentation method.

The quadratic matrix in the objective function is expressed as a difference between two positive semi-definite matrices. These matrices are derived from intra-cluster and inter-cluster samples, respectively. Consequently, this matrix is not guaranteed to be positive definite, rendering the resulting objective function non-convex due to the indefinite nature of the quadratic form. This non-convexity implies that the solution generally lacks theoretical guarantees regarding global optimality.

3. Methodology

3.1. Preliminaries

We first outline the standard kPC algorithm before presenting our constrained framework.

kPC algorithm: Let the input dataset be denoted as $\mathbf{X} \in R^{N\times d}$, where N is the number of points and d represents the dimensionality of the feature space. The set of candidate plane is formally defined as:

$$plane_i = \left\{\mathbf{x}|\mathbf{w}_i^T\mathbf{x}_i + \gamma_i = 0\right\}, i = 1, 2, \ldots, k, \tag{1}$$

where $\mathbf{w}_i$ and γ_i represent the normal vector and the intercept (threshold) of the ith fitting plane, respectively. The term $\mathbf{x}_i$ denotes a data point assigned to that plane. The superscript $^{\mathrm{T}}$ indicates the transpose operator. Based on this formulation, the objective function of the kPC algorithm is defined as:

$$\frac{1}{2}\sum_{i=1}^{k} \underset{\mathbf{w}_i,\gamma_i}{min} \|\mathbf{w}_i^T\mathbf{x}_i + \gamma_i\mathbf{e}\|_2^2 \tag{2}$$

$$s.t. \quad \|\mathbf{w}_i\|_2^2 = 1, i = 1, 2, \ldots, k. \tag{3}$$

The notation follows Eq. (1). In this formulation, $\mathbf{e}$ is a column vector of ones with the appropriate dimension. Additionally, $\mathbf{x}_i$ comprises the points assigned to the ith plane.

3.2. Optimization problem

The standard kPC framework suffers from unbounded plane extension, as it considers only point-to-plane distance and ignores misclassifications induced by the infinite scalability of fitted planes. To overcome this limitation, we introduce a constrained kPC formulation. A brief example illustrates this issue: for a roof structure composed of multiple peaks and flat surfaces, the conventional kPC method models the flat regions as an indefinitely extensible plane. Consequently, sections of adjacent peaks are often incorrectly merged into the same planar segment. In contrast, the proposed method incorporates three geometrically grounded constraints to prevent such errors. The first constraint minimizes the orthogonal point-to-plane distance: $(\|\mathbf{w}_i^T\mathbf{x}_j + \gamma_i\|)_2^2$. The second constraint preserves spatial coherence by enforcing cluster-center proximity: $(\|\mathbf{x}_j - \mu_i\|)_2$. The third constraint provides

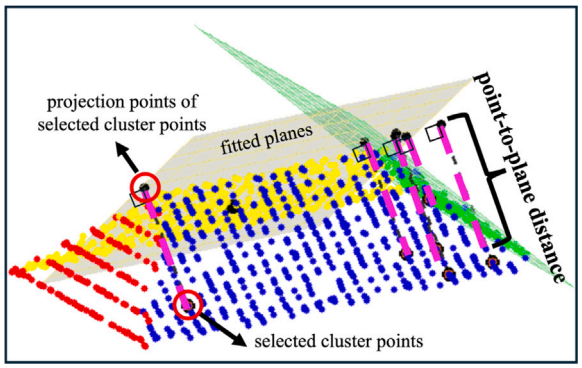


Fig. 2. Illustration of the orthogonal point-to-plane distance constraint.

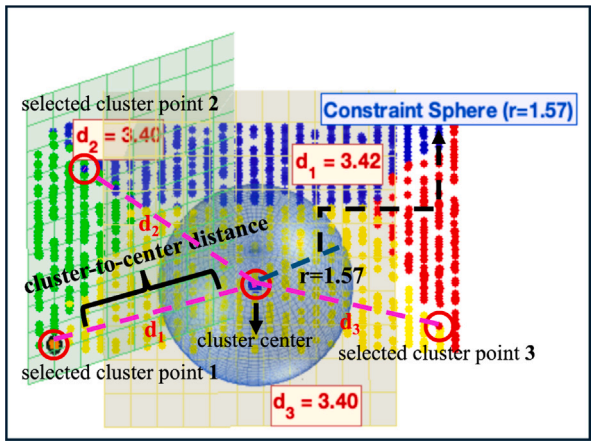


Fig. 3. Illustration of the cluster-centroid proximity constraint.

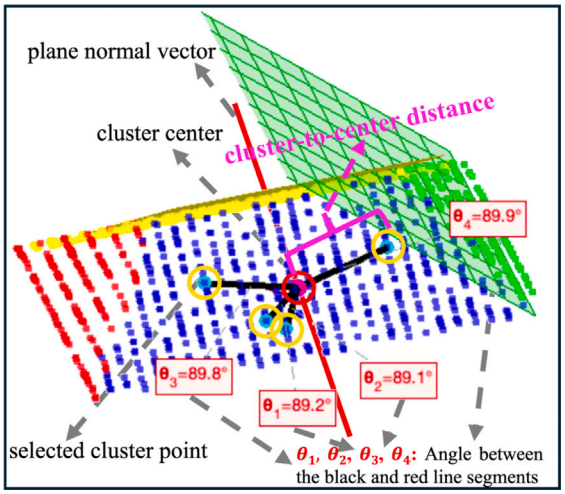


Fig. 4. Illustration of the directional regularization constraint.

directional regularization aligned with architectural structures: $\cos(\theta_{ij})$. The symbol $\mathbf{x}_j$ denotes the set of points assigned to the jth fitted plane. This leads to the following optimization formulation:

$$\mathbf{L}_{(j,i)} = \alpha \cdot (\|\mathbf{w}_i^T \mathbf{x}_j + \gamma_i\|)_2^2 + \beta \cdot (\|\mathbf{x}_j - \mu_i\|)_2 + \lambda \cdot \cos(\theta_{ij}) \tag{4}$$

$$\cos(\theta_{ij}) = \frac{\mathbf{w}_i^T(\mu_i - \mathbf{x}_j)}{\|\mu_i - \mathbf{x}_j\|_2}, \tag{5}$$

where $\alpha, \beta, \lambda \geq 0$ are weighting coefficients governing different penalty terms. Specifically, α, β, and λ control the orthogonal point-to-plane distance, the Euclidean distance to the cluster center, and a directional consistency term, respectively. Here, $\mu_i \in \mathbb{R}^d$ denotes the centroid of the ith cluster, and $\cos(\theta_{ij})$ quantifies the cosine similarity between the vector $(\mu_i - \mathbf{x}_j)$ and the plane's normal vector $\mathbf{w}_i$.

The workflow of the proposed framework is illustrated in Fig. 1. The process begins with a large-scale point cloud scene containing multiple buildings as input (Fig. 1(a)). The point cloud is first classified into building and non-building points, followed by instance segmentation to delineate individual building instances (Fig. 1(b)). Each segmented building is assigned a unique identifier, labeled ① through ⑪. The middle section of Fig. 1 provides a step-by-step visualization of the proposed constrained kPC framework for planar primitive segmentation. The process starts with random initialization of the plane parameters. For example, as shown in Fig. 1(d), a building roof composed of four planar primitives is represented by points colored in red, blue, yellow, and green, with their corresponding fitting planes depicted in the same colors. Subsequently, based on the initial plane parameters, all roof points are reassigned to their optimal fitting planes according to the proposed constrained kPC formulation. The plane parameters are then updated using the newly assigned points. These two steps, cluster assignment refinement and plane parameter update, are illustrated in Fig. 1(e) and (f), respectively. The process iterates until convergence. The final segmentation results, shown in Fig. 1(i), demonstrate the successfully identified planar primitives. All extracted planar components are merged in Fig. 1(h). To further illustrate the effectiveness of the method, three representative building primitives (labeled ②, ⑦, and ⑨) are enlarged in Fig. 1(g).

The Eq. (4) can be decomposed into three distinct terms: (1) a point-to-plane distance minimization term $\|\mathbf{w}_i^T \mathbf{x}_j + \gamma_i\|_2^2$, (2) a cluster-center proximity term $\|\mathbf{x}_j - \mu_i\|_2$, and (3) a directional regularization term $\cos(\theta_{ij})$. These components are visualized in Fig. 2, Fig. 3, and Fig. 4, where they are labeled as 'constraint #1', 'constraint #2', and 'constraint #3', respectively. As shown in Fig. 2 for 'constraint #1', a set of clustered points (highlighted in yellow) and the associated fitted plane (depicted as a gray mesh) are labeled as 'fitted planes'. From the cluster (marked in blue), five representative points (shown in orange) are selected. Their corresponding orthogonal projections onto the gray mesh are designated as 'selected cluster points' and 'projection points of selected cluster points', respectively. The purpose of this constraint term is twofold: first, to compute the orthogonal distances, indicated by magenta dashed lines, between the yellow points and the gray plane; second, to ensure that points assigned to a given cluster are located closer to their corresponding fitted plane than to any other fitted plane. These orthogonal distances are labeled as 'point-to-plane distance'.

The objective of 'constraint #2' ($|\mathbf{x}_j - \mu_i|_2$) is to minimize the Euclidean distance between each point $\mathbf{x}_j$ and the centroid μ_i of its assigned cluster. As shown in Fig. 3, the centroid of the cluster corresponding to the yellow points is marked with a red circle and labeled as 'cluster center'. For visualization purposes, up to three points — also highlighted with red circles and labeled as 'selected cluster point 1–3' — are randomly chosen from distinct clusters within the sphere. The Euclidean distances between these points and the cluster centroid are computed. These distances are represented by magenta dashed line segments and annotated as d_1, d_2, and d_3, respectively. To illustrate the average spatial extent of the cluster, a translucent sphere centered at the centroid is generated, with its radius set to the mean Euclidean distance of the selected points. This constraint mitigates misclassification caused by the unbounded extension of fitted planes. It achieves this by geometrically tethering points to their respective cluster centers. This tethering is particularly beneficial for points located near building edges.

As shown in Fig. 4, the objective of 'constraint #3' is represented by the term $\cos(\theta_{ij})$ and aims to regularize the angular relationship

Table 1
Iterative solution procedure for the optimization formulation in Eq. (4).

Iterative solution procedure for the optimization formulation in Eq. (4)	
Input:	Point cloud $\mathbf{x}_j$; parameters $\alpha, \beta, \lambda \geq 0$; initialization $\mathbf{w}_i^{(0)} \in \mathbb{R}^d$, $\mu_i^{(0)} \in \mathbb{R}^d$, $\gamma_i^{(0)} = -\frac{1}{N}\sum_{j=1}^{N}(\mathbf{w}_i^{(0)})^\top \mathbf{x}_j$
Output:	Optimal parameters: $\mathbf{w}_i^* \leftarrow \mathbf{w}_i^{(t)}$, $\mu_i^* \leftarrow \mu_i^{(t)}$, $\gamma_i^* \leftarrow \gamma_i^{(t)}$
Iterative optimization procedure ($t = 1, 2, \ldots, T$)	
Step 1: Update bias term	Update the bias term γ_i using the current normal vector: $\gamma_i^{(t)} \leftarrow -\frac{1}{N}\sum_{j=1}^{N}(\mathbf{w}_i^{(t-1)})^\top \mathbf{x}_j$
Step 2: Update cluster centroid	Update the cluster centroid μ_i via gradient descent with smoothing factor ϵ: $g_\mu \leftarrow \nabla_{\mu_i}\sum_{j=1}^{N}\left[\beta\|\mathbf{x}_j - \mu_i^{(t-1)}\|_2 + \lambda\left(1 - \sqrt{\left(\frac{(\mathbf{w}_i^{(t-1)})^\top(\mu_i^{(t-1)} - \mathbf{x}_j)}{\|\mu_i^{(t-1)} - \mathbf{x}_j\|_2}\right)^2 + \epsilon}\right)\right]$ $\mu_i^{(t)} \leftarrow \mu_i^{(t-1)} - \eta_\mu \cdot g_\mu$
Step 3: Update normal vector	Update the normal vector $\mathbf{w}_i$ via projected gradient descent: $g_\mathbf{w} \leftarrow \nabla_{\mathbf{w}_i}\sum_{j=1}^{N}\left[\alpha((\mathbf{w}_i^{(t-1)})^\top \mathbf{x}_j + \gamma_i^{(t)})^2 + \lambda\left(1 - \sqrt{\left(\frac{(\mathbf{w}_i^{(t-1)})^\top(\mu_i^{(t)} - \mathbf{x}_j)}{\|\mu_i^{(t)} - \mathbf{x}_j\|_2}\right)^2 + \epsilon}\right)\right]$ $\mathbf{w}_i^{(t)} \leftarrow \mathbf{w}_i^{(t-1)} - \eta_\mathbf{w} \cdot g_\mathbf{w}$ $\mathbf{w}_i^{(t)} \leftarrow \mathbf{w}_i^{(t)} / \|\mathbf{w}_i^{(t)}\|_2$ *(unit norm projection)*
Convergence criteria:	$\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(t-1)}\|_2 < \epsilon$ and $\|\mu_i^{(t)} - \mu_i^{(t-1)}\|_2 < \epsilon$ or $t \geq T$

between the points assigned to a cluster and the normal vector $\mathbf{w}_i$ of its associated fitted plane. This plane normal is indicated by a red solid line and labeled as 'plane normal vector' in the figure. The purpose of this constraint is to improve the segmentation of points located near the intersection lines of adjacent planes by promoting directional consistency with the respective plane's orientation. The visualization in Fig. 4 is organized as follows: the cluster centroid is indicated by a red circle. Four representative points, labeled as 'selected cluster point' and marked with yellow circles, are randomly chosen from the cluster. For each point, the direction vector from the centroid to the point is computed (displayed as a black line segment) and normalized to a unit vector. The cosine similarity between each unit direction vector and the plane's normal vector is then calculated via the dot product. The corresponding angles θ_{ij} (in degrees) are displayed in a magenta text box alongside the visualization.

The optimization problem formulated in Eq. (4) results in a mixed-integer non-convex program, which is solved using an alternating minimization strategy. Owing to the random initialization of the hyperplane parameters, the optimal value of γ_i is determined by differentiating the expression in Eq. (6) with respect to γ_i and setting the derivative to zero:

$$\gamma_i^* = arg\min_{\gamma_i} \sum_j \cdot(\|\mathbf{w}_i^T \mathbf{x}_j + \gamma_i\|)^2 \tag{6}$$

$$\frac{\partial}{\partial \gamma_i}\sum_j(\|\mathbf{w}_i^T \mathbf{x}_j + \gamma_i\|)^2 = 2\sum_j(\|\mathbf{w}_i^T \mathbf{x}_j + \gamma_i\|) = 0 \tag{7}$$

Therefore, the closed-form solution for γ_i is given by:

$$\gamma_i^* = -\frac{1}{N}\sum_{j=1}^{N}\mathbf{w}_i^T \mathbf{x}_j \tag{8}$$

With $\mathbf{w}_i$ and γ_i fixed, the cluster centroid $\boldsymbol{\mu}_i$ is determined through gradient-based optimization. A smooth approximation is applied to handle the non-differentiability arising from the absolute value terms in the objective. Similarly, with μ_i and γ_i held constant, the normal vector $\mathbf{w}_i$ is updated using a projected gradient method to maintain the unit norm constraint. Let $\mathbf{w}_i^{(t)}$, $\boldsymbol{\mu}_i^{(t)}$, and $\gamma_i^{(t)}$ denote the values for the ith plane at the tth iteration. These respectively represent its normal vector, cluster centroid, and plane threshold. The superscript (0) indicates the initialized values. The proposed algorithm alternates iteratively between two core steps: (1) reassigning points to clusters based on current plane hypotheses, and (2) updating the corresponding plane parameters using the newly formed clusters. This iterative process continues until convergence is achieved or a predetermined maximum number of iterations is completed. A complete summary of the optimization procedure for solving Eq. (4) is provided in Table 1.

4. Experiments

4.1. Datasets

In this section, the proposed method is evaluated using three roof-specific datasets and two large-scale building datasets. A detailed description of these experimental datasets is provided as follows:

Roof Datasets: *(1) Synthetic Roof Dataset (Li et al., 2020):* This dataset includes 14 non-flat roof types, each comprising multiple planar surfaces. Points belonging to the same planar roof segment are assigned a shared instance label. The dataset contains 15,400 samples, divided into training and test sets at a 10:1 ratio. *(2) RoofN3D Dataset (Wichmann et al., 2018):* A publicly available dataset featuring real-world 3D building scans from New York City. The dataset encompasses three common roof types: gable, pyramid, and hip. It provides LiDAR point clouds with both semantic and instance-level annotations. These annotations cover roof and façade structures. *(3) Building3D Dataset (Wang et al., 2023):* A large-scale benchmark for urban 3D reconstruction, offering multi-modal data such as point clouds, wireframes, and mesh models. Instance labels for roof planes are derived by computing point-to-mesh distances between LiDAR points and the corresponding mesh surfaces.

Building datasets: To further evaluate the proposed method, two distinct geographic regions with diverse architectural styles were selected for testing. The building datasets are divided into two categories to assess planar primitive segmentation performance: (1) identification of planar primitives within individual buildings, and (2) segmentation of roof primitives across complex multi-structure scenes. The six small-scale building datasets used in this study, labeled 'Area 1' through 'Area 6', contain only building and non-building points. They were acquired using two different dual-channel LiDAR systems: Areas 1–3 (Feng et al., 2022) was captured with a RIEGL VQ-1560i (first row of Fig. 5(a)), and Areas 4–6 with a RIEGL-VZ-600i (second row of Fig. 5(a)). In both rows, building and non-building points are visually distinguished by color. Representative examples of the color point cloud collected with the RIEGL-VZ-600i and their corresponding roof structures are shown in the first and second rows of Fig. 5(b).

4.2. Evaluation metric

We quantitatively evaluated the proposed method using seven widely adopted performance metrics, in line with previous studies (Wang et al., 2019; Li et al., 2020): **Coverage (Cov):** Measures the instance-wise Intersection over Union (IoU) between predicted and ground-truth (GT) planes, averaged across all instances. **Weighted**

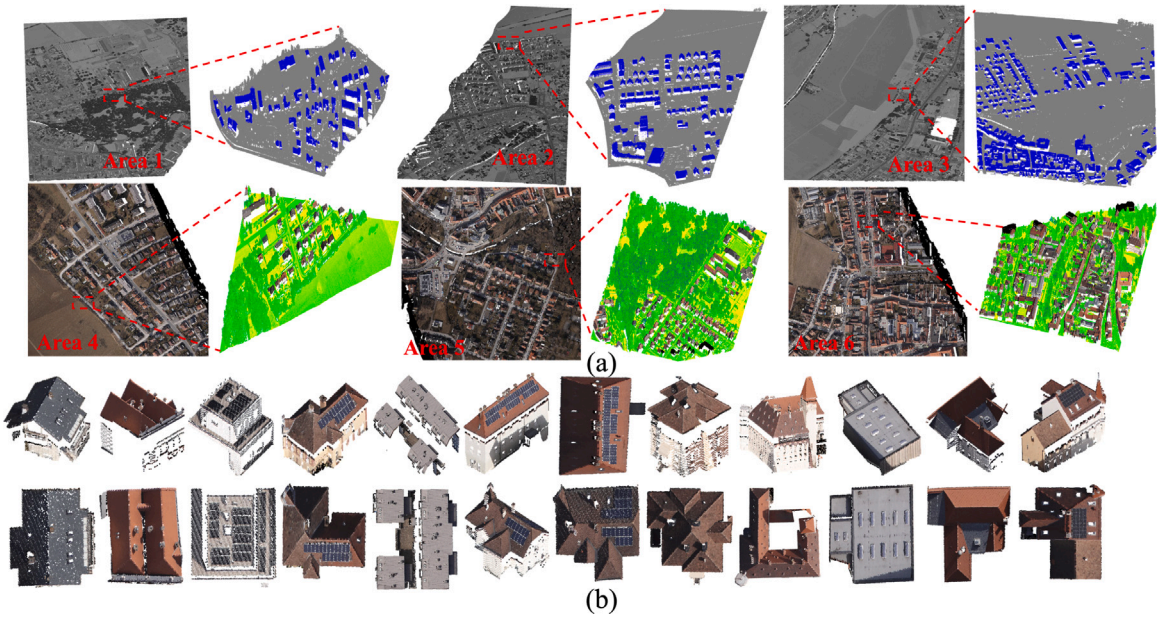


Fig. 5. Dataset overview: architectural style diversity. (a) building point clouds captured by the RIEGL VQ-1560i (first row) and RIEGL VZ-600i (second row) terrestrial laser scanners. (b) representative samples from the RIEGL VZ-600i dataset, illustrating the variety of structural types and architectural styles.

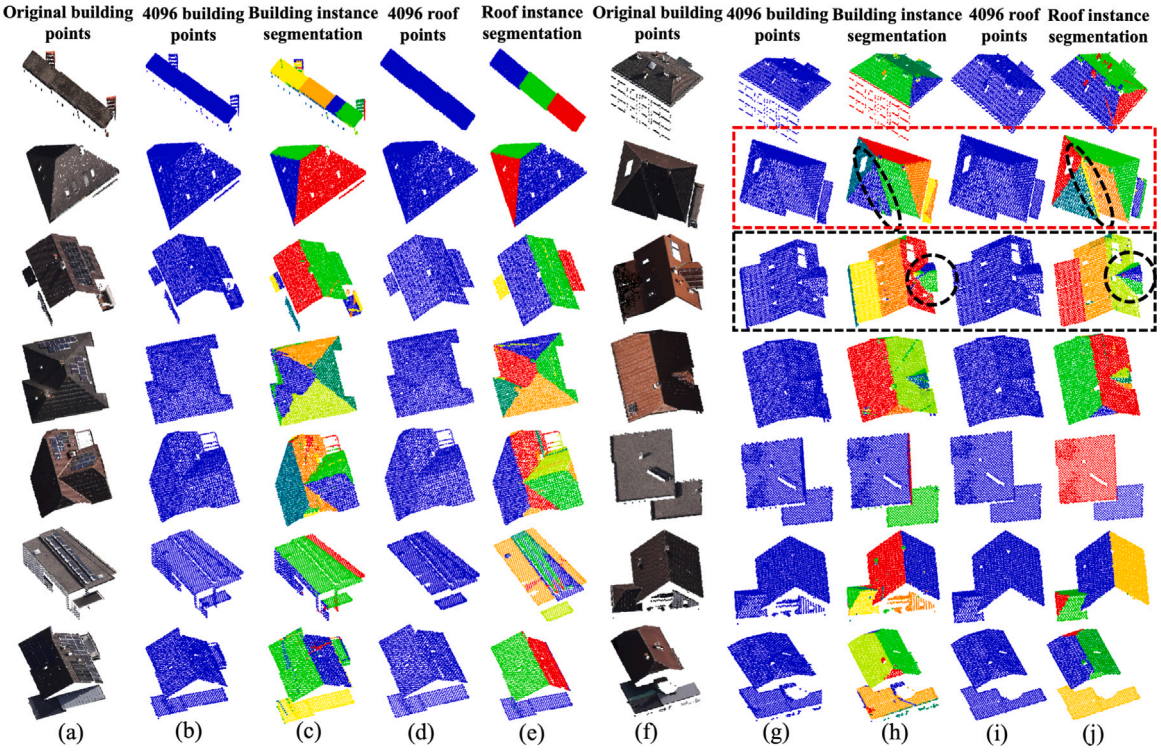


Fig. 6. Workflow and outcomes of the proposed planar primitive segmentation framework. (a, f) Original building point clouds. (b, g, d, i) Downsampled point clouds (4096 points each). (c, h, e, j) Final planar primitive segmentation results. Misclassification cases are indicated by red and black dotted rectangles and circles.

Coverage (WCov): A version of Cov where each instance is weighted by its point count, placing greater emphasis on larger roof planes. **Mean Precision (MPrec)** and **Mean Recall (MRec):** Standard instance segmentation metrics evaluated at an IoU threshold of 0.5, reflecting the precision and recall of plane-level predictions. **Completeness** (C_m): The ratio of GT roof planes that are correctly detected in the prediction. **Correctness** (C_r): The proportion of predicted planes that match actual GT instances. **Quality Index (QI):** A holistic metric combining both completeness and correctness to evaluate overall segmentation quality. These metrics collectively offer a comprehensive assessment of segmentation performance and are formally defined in the equations below:

$$Cov(G,P) = \sum_{i=1}^{|G|} \frac{1}{|G|} \max_{j} IoU(N_G, N_P) \tag{9}$$

$$WCov(G,P) = \sum_{i=1}^{|G|} \frac{1}{G} \frac{|N_G|}{\sum_i |N_G|} \max_{j} IoU(N_G, N_P) \tag{10}$$

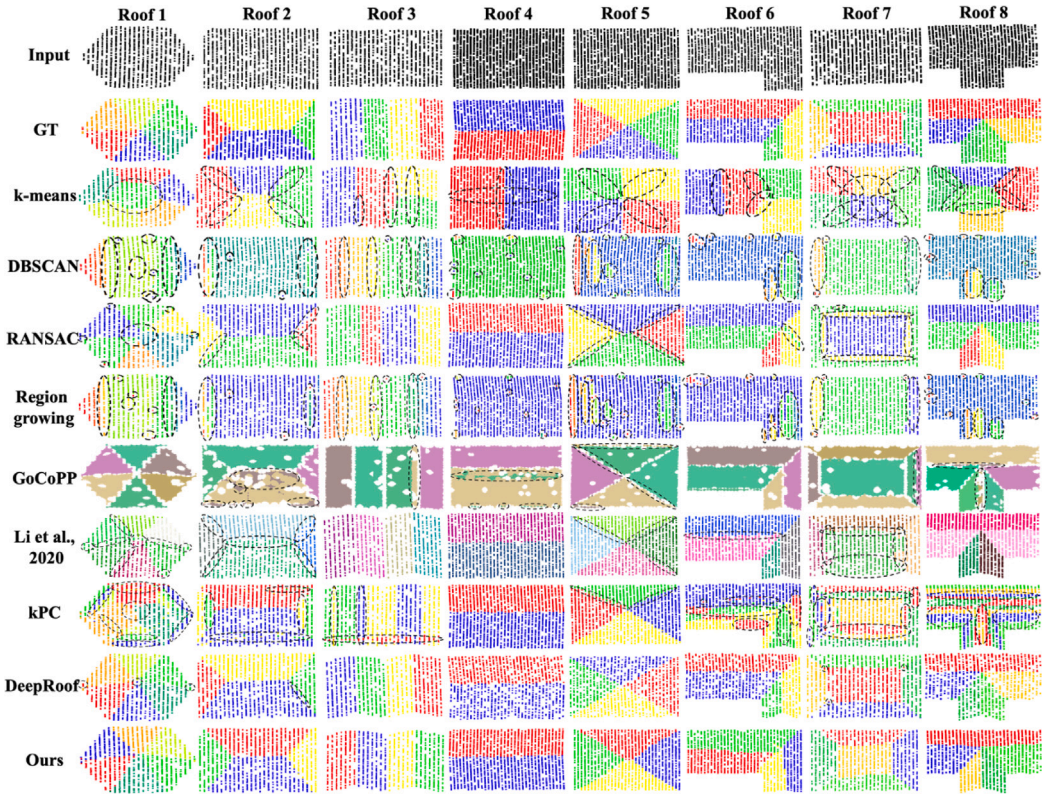


Fig. 7. Comparative evaluation of planar primitive segmentation methods on synthetic roof dataset. (The distinct colors indicate different roof plane instances (or categories). The black dots denote points that are misclassified when comparing the predicted primitive segmentation results with the GT).

$$C_m = \frac{TP}{TP + FN} \tag{11}$$

$$C_r = \frac{TP}{TP + FP} \tag{12}$$

$$QI = \frac{TP}{TP + FN + FP} \tag{13}$$

where, N_G and N_P denote the number of points in the GT cluster and the predicted cluster i, respectively. The point classification categories are defined as follows: True Positive (TP): roof points correctly identified in both the GT and predicted sets. False Negative (FN): GT roof points that were not detected in the prediction. False Positive (FP): Non-roof points incorrectly predicted as roof segments. These metrics collectively evaluate segmentation accuracy from both instance-level and point-level perspectives. Higher values of Cov, $WCov$, C_m, C_r, and QI correspond to better segmentation quality, indicating more accurate plane extraction and fewer classification errors.

4.3. Quantitative and qualitative results

The experimental evaluation is structured as follows: (1) Method Validation: The segmentation effectiveness is first validated on building and roof point clouds from a proprietary dataset. (2) Comparative Analysis: Performance is compared against conventional methods using eight roof instances selected from the Synthetic Roof and Building3D datasets. (3) Capability Assessment: The method is further evaluated on four small-scale urban scenes extracted from the region shown in Fig. 5(a). These scenes contain both roof and façade structures. In the first experimental analysis, 14 building instances from our proprietary dataset were selected to visualize planar primitive segmentation performance. Each column in Fig. 6 is labeled from (a) to (j). Columns (a) and (f) display the original segmented building point clouds. To mitigate memory explosion issues caused by matrix operations, the original point clouds were downsampled to 4096 points, as shown in columns (b) and (g), respectively. The building instances were segmented into roof planes and facades, with basic directional information retained to support 3D building reconstruction. The roof plane instance segmentation results are presented in columns (e) and (j). Two representative cases are highlighted with red and black dotted rectangles. The red rectangle shows that even small roof regions can be successfully detected. In contrast, the black rectangle indicates an area composed of two triangular planes that is occasionally misclassified. This error arises when the dihedral angle between adjacent planes is small, as smaller angles increase the likelihood of misclassification.

In the second experimental phase, the effectiveness of the proposed method is evaluated through comparison with eight representative baseline approaches: k-means (Hamerly and Elkan, 2003), DBSCAN (Ferrara et al., 2018), RANSAC (Fischler and Bolles, 1981), RG (Region Growing) (Lafarge and Mallet, 2012), GoCoPP (Yu and Lafarge, 2022), the method by Li et al. (2020), the original kPC algorithm (Bradley and Mangasarian, 2000), and the DL-based DeepRoof framework (Li et al., 2024). For implementation, k-means and DBSCAN were executed using MATLAB's built-in functions. RANSAC and Region Growing were implemented using the Computational Geometry Algorithms Library (CGAL; https://www.cgal.org/). The source code for GoCoPP was obtained from the authors' official repository (https://team.inria.fr/titane/). As no public implementation of the original kPC method was available, we re-implemented it based on the description in the original publication. The methods by Li et al. (2020) and DeepRoof were evaluated using the code provided by the respective authors. The quantitative results for the eight synthetic and Building3D roof datasets are presented in Figs. 7 and 8, respectively.

In the first row of each figure, the original input point clouds are shown in a uniform color, while the second row displays the segmented planar primitives, each depicted in a distinct color. Misclassified regions are identified through comparison with the GT. They are

Table 2
Benchmarking quantitative results on the Synthetic Roof Dataset. The best performances are highlighted in bold.

Method	Metrics	Roof1	Roof2	Roof3	Roof4	Roof5	Roof6	Roof7	Roof8
k-means	Cov//WCov	51.47//51.59	44.28//47.57	51.28//51.86	33.75//33.75	34.95//34.95	35.44//32.86	36.62//33.71	41.16//41.84
	C_m//C_r	66.67//66.67	50.00//50.00	75.00//75.00	100.0//100.0	100.0//100.0	75.00//75.00	80.00//80.00	80.00//80.00
	QI	50.00	33.33	60.00	100.0	100.0	60.00	66.67	66.67
DBSCAN	Cov//WCov	20.29//20.10	31.30//37.31	16.33//16.34	49.68//49.68	23.63//23.75	26.08//33.65	40.20//40.70	20.88//26.98
	C_m//C_r	83.33//12.20	75.00//27.27	100.0//7.41	50.00//14.29	50.00//4.88	50.00//8.33	60.00//33.33	40.00//6.90
	QI	11.90	25.00	7.41	12.50	4.65	7.69	27.27	6.25
RANSAC	Cov//WCov	74.91//75.04	86.46//88.15	80.94//81.16	97.14//97.14	87.33//87.35	87.14//87.77	31.43//41.30	75.05//79.41
	C_m//C_r	83.33//71.43	100.0//100.0	50.00//50.00	100.0//100.0	100.0//100.0	100.0//100.0	60.00//75.00	80.00//100.0
	QI	62.50	100.0	33.33	100.0	100.0	100.0	50.00	80.00
RG	Cov//WCov	20.29//20.10	31.30//37.31	16.33//16.34	49.68//49.68	23.63//23.75	26.08//33.65	40.20//40.70	20.88//26.98
	C_m//C_r	83.33//12.82	75.00//27.27	100.0//7.550	50.00//16.67	50.00//9.090	50.00//9.090	60.00//42.86	40.00//7.690
	QI	12.50	25.00	7.550	14.29	8.330	8.330	33.33	6.900
GoCoPP	Cov//WCov	16.37//16.26	24.85//33.23	24.84//25.04	33.02//33.02	22.97//22.80	21.60//22.05	19.86//25.65	18.41//22.10
	C_m//C_r	16.67//100.0	25.00//100.0	25.00//100.0	50.00//100.0	25.00//100.0	25.00//100.0	20.00//100.0	20.00//100.0
	QI	16.67	25.00	25.00	50.00	25.00	25.00	20.00	20.00
Li et al. 2020	Cov//WCov	30.11//30.15	38.30//35.59	87.14//87.51	33.62//33.63	42.48//42.35	30.98//31.04	31.01//27.70	28.68//28.21
	C_m//C_r	83.33//83.33	75.00//75.00	75.00//75.00	100.0//100.0	100.0//100.0	50.00//50.00	80.00//100.0	60.00//75.00
	QI	71.43	60.00	60.00	100.0	100.0	33.33	80.00	50.00
Li et al. 2024	Cov//WCov	**99.47//99.46**	89.74//89.74	**100.0//100.0**	98.85//98.85	98.38//98.38	**98.66//98.30**	89.18//89.04	**99.02//98.92**
	C_m//C_r	**100.0//100.0**	100.0//100.0	**100.0//100.0**	100.0//100.0	100.0//100.0	**100.0//100.0**	100.0//100.0	**100.0//100.0**
	QI	**100.0**	100.0	**100.0**	100.0	100.0	**100.0**	100.0	**100.0**
kPC	Cov//WCov	38.94//38.91	59.39//75.20	36.25//36.19	99.73//99.73	94.72//94.70	38.91//50.06	33.12//37.16	15.72//18.31
	C_m//C_r	66.67//66.67	100.0//100.0	100.0//100.0	100.0//100.0	100.0//100.0	75.00//75.00	80.00//80.00	100.0//100.0
	QI	50.00	100.0	100.0	100.0	100.0	60.00	66.67	100.0
Ours	Cov//WCov	99.26//99.25	**90.71//95.01**	**100.0//100.0**	**99.80//99.80**	**98.50//98.50**	93.33//95.83	**94.68//94.53**	95.52//95.42
	C_m//C_r	100.0//100.0	**100.0//100.0**	**100.0//100.0**	**100.0//100.0**	**100.0//100.0**	100.0//100.0	**100.0//100.0**	100.0//100.0
	QI	100.0	**100.0**	**100.0**	**100.0**	**100.0**	100.0	**100.0**	100.0

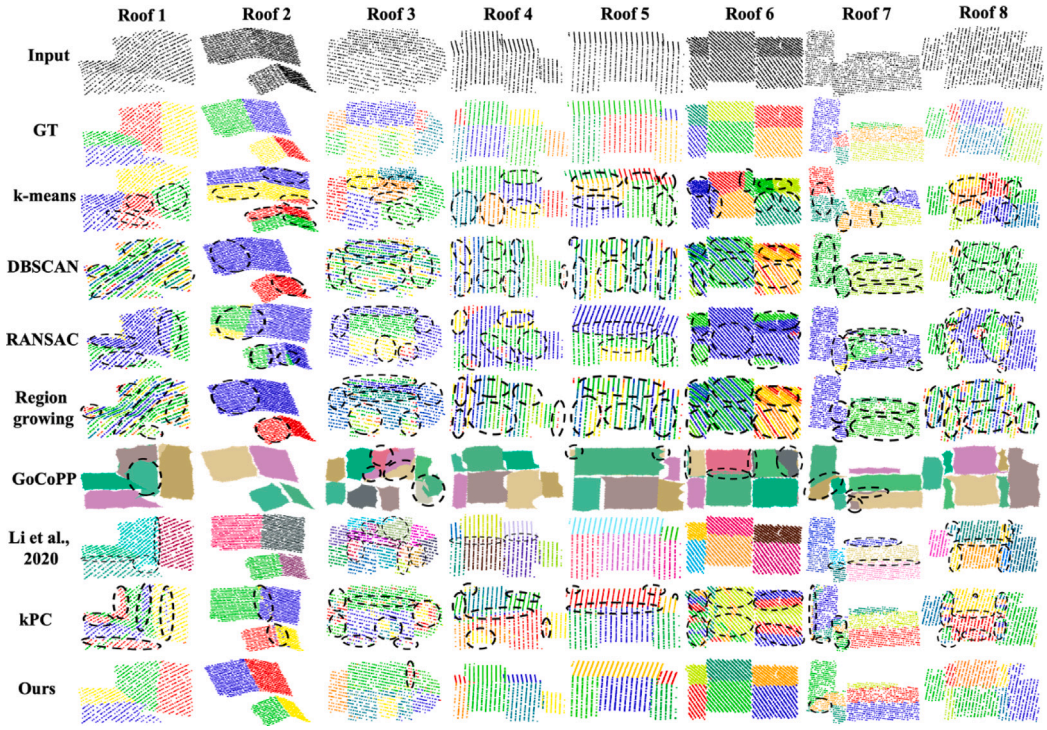


Fig. 8. Comparative evaluation of planar primitive segmentation methods on Building3D dataset. (The distinct colors indicate different roof plane instances (or categories). The black dots denote points that are misclassified when comparing the predicted primitive segmentation results with the GT).

then highlighted using black dotted irregular ovals. The experimental findings can be summarized as follows: k-means, which relies on Euclidean distance-based clustering, frequently incorrectly splits large roof surfaces into multiple segments or merges adjacent yet distinct planar patches. DBSCAN is sensitive to variations in point density and struggles to separate adjacent planar surfaces with small inter-plane distances. It is also highly susceptible to noise and outliers, as visible in the fourth row of Fig. 8. RANSAC performs well at extracting a single dominant plane but is ill-suited for multi-planar segmentation. Its performance declines notably when planes are of

Table 3
Benchmarking Quantitative Results on the Building3D Roof Dataset. The best performances are highlighted in bold.

Method	Metrics	Roof1	Roof2	Roof3	Roof4	Roof5	Roof6	Roof7	Roof8
k-means	Cov//WCov	47.93//50.00	33.70//33.78	44.28//42.74	58.63//63.28	34.65//35.85	60.21//64.36	44.52//46.93	43.00//49.53
	C_m//C_r	100.0//57.14	100.0//100.0	71.43//62.50	71.43//71.43	100.0//100.0	83.33//83.33	100.0//85.71	87.50//87.50
	QI	57.14	100.0	50.00	55.56	100.0	71.43	85.71	77.78
DBSCAN	Cov//WCov	9.86//9.38	50.00//50.23	14.79//12.92	18.83//15.47	16.57//11.58	11.12//9.90	49.60//58.30	32.21//33.65
	C_m//C_r	50.00//2.60	50.00//100.0	85.71//4.41	100.0//16.28	80.00//7.84	33.33//2.94	50.00//75.00	37.50//75.00
	QI	2.53	50.00	4.38	16.28	7.69	2.78	42.86	33.33
RANSAC	Cov//WCov	28.54//31.23	35.79//35.85	29.77//41.10	26.10//27.26	37.73//51.11	22.70//24.76	25.43//28.77	22.48//27.47
	C_m//C_r	50.00//50.00	50.00//50.00	42.86//75.00	42.86//75.00	60.00//100.0	33.33//100.0	50.00//100.0	62.50//100.0
	QI	33.33	33.33	37.50	37.50	60.00	33.33	50.00	
RG	Cov//WCov	9.99//9.56	50.00//50.22	42.65//39.14	18.74//15.42	16.40//11.43	11.11//9.90	49.58//58.22	16.50//12.92
	C_m//C_r	50.00//2.90	50.00//100.0	71.43//13.16	100.0//15.91	80.00//7.69	33.33//2.94	50.00//60.00	75.00//5.22
	QI	2.82	50.00	12.50	15.91	7.55	2.78	37.50	5.13
GoCoPP	Cov//WCov	25.00//26.66	25.00//25.19	14.29//23.70	14.29//16.79	20.00//27.78	16.67//20.08	16.67//24.31	12.50//18.34
	C_m//C_r	25.00//100.0	25.00//100.0	14.29//100.0	14.29//100.0	20.00//100.0	16.67//100.0	16.67//100.0	12.50//100.0
	QI	25.00	25.00	14.29	14.29	20.00	16.67	16.67	12.50
Li et al. 2020	Cov//WCov	25.00//26.54	24.73//24.81	18.80//23.90	25.31//33.81	25.78//35.75	16.61//20.55	30.25//43.53	15.55//20.91
	C_m//C_r	25.00//100.0	25.00//50.00	28.57//50.00	28.57//100.0	40.00//100.0	16.67//50.00	33.33//100.0	25.00//66.67
	QI	25.00	20.00	22.22	28.57	40.00	14.29	33.33	22.22
kPC	Cov//WCov	43.02//48.21	95.80//95.70	36.63//55.27	52.80//58.22	74.15//83.76	26.44//25.99	**67.18//78.23**	48.86//72.28
	C_m//C_r	100.0//100.0	100.0//100.0	85.71//85.71	71.43//71.43	80.00//80.00	83.33//83.33	**100.0//100.0**	87.50//87.50
	QI	100.0	100.0	75.00	55.56	66.67	71.43	**100.0**	77.78
Ours	Cov//WCov	**92.21//92.70**	**99.98//99.13**	**45.52//46.15**	**79.28//83.62**	**77.73//80.47**	**63.50//70.24**	49.04//56.89	**50.69//59.38**
	C_m//C_r	**100.0//100.0**	**100.0// 100.0**	**85.71//85.71**	**100.0//100.0**	**100.0//100.0**	**83.33//83.33**	66.67//66.67	**75.00//75.00**
	QI	**100.0**	**100.0**	75.00	**100.0**	**100.0**	**71.43**	50.00	**60.00**

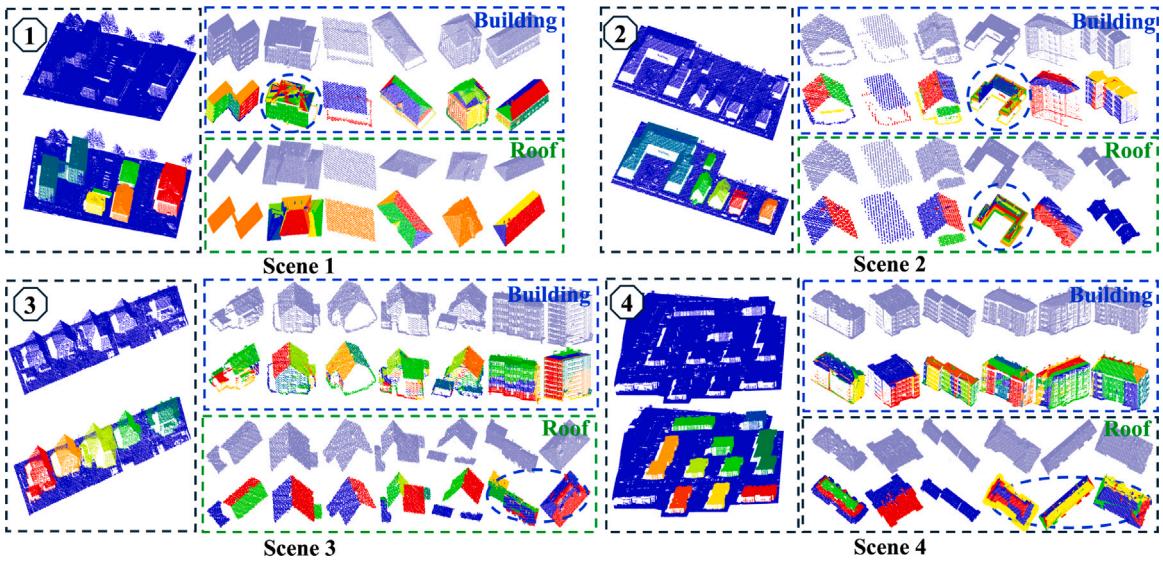


Fig. 9. Comprehensive evaluation of the proposed planar primitive segmentation method across four small-scale building datasets.

similar size or contain noise. RG is heavily influenced by initial seed selection and tolerance thresholds. Poor choices often lead to over- or under-segmentation. GoCoPP, as a geometry-based method, incorporates strong geometric constraints but tends to be computationally intensive and may depend on heuristics that lack generalizability across diverse architectural styles. DeepRoof, a DL-based approach, requires large volumes of annotated data, which are often expensive and labor-intensive to obtain. kPC optimizes a set of k infinite planes, making it inherently more suitable for capturing the geometric essence of roof structures than proximity- or density-based methods. The proposed method enhances structural consistency by enforcing coplanarity and encouraging parallel or orthogonal relationships among roof segments. This leads to more semantically coherent and architecturally plausible segmentation results.

The proposed method demonstrates state-of-the-art performance, as evidenced in Tables 2 and 3. It achieves perfect segmentation accuracy (100%) on Roofs 3, 4, 5, and 8 in Table 2, and establishes new performance benchmarks on Roofs 2, 4, and 8—even outperforming the recent deep learning approach introduced by Li et al. (2024). The robustness of the method is further confirmed in Table 3, where it attains the highest scores across six roof types (e.g., 99.98//99.13 on Roof 2). In contrast, conventional methods (k-means, DBSCAN, region growing) exhibit limited adaptability, with performance declining to as low as 33.75%–51.47% on geometrically complex roofs. While the competing learning-based method delivers strong results in certain scenarios, our approach achieves comparable or superior accuracy without relying on large annotated datasets or intensive computational resources, utilizing only geometric constraints. This further highlights the practicality and efficiency of the proposed framework for building

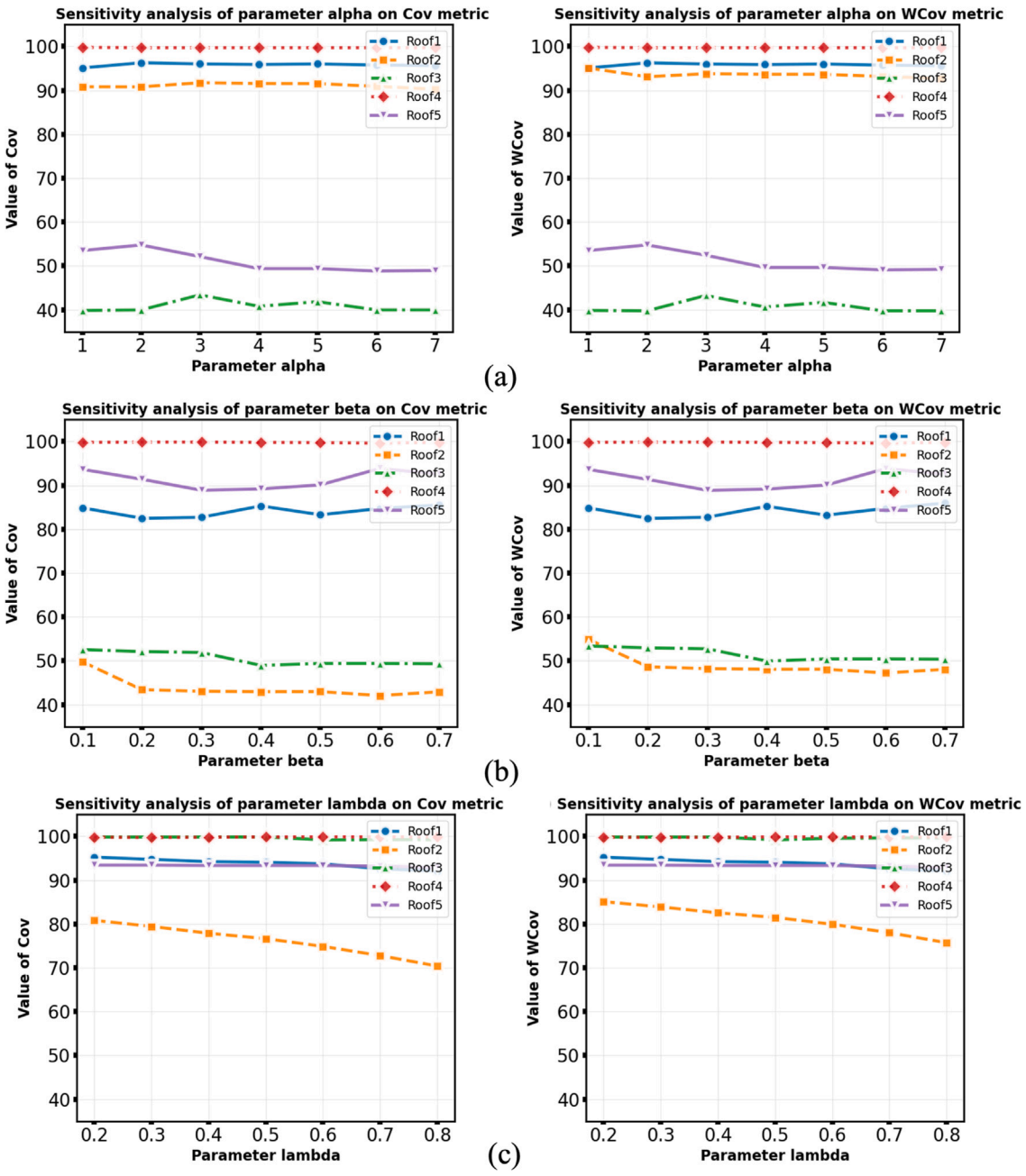


Fig. 10. Parameter sensitivity analysis of α, β, and λ on Cov and WCov metrics across representative roof datasets.

planar primitive segmentation. It should be noted that quantitative and qualitative results for Deep-Roof on the Building3D dataset are not included in Table 2 and Fig. 8. This omission is due to the fact that the Building3D dataset does not provide ground-truth annotations for roof primitives. Manually annotating more than 36,000 buildings would be prohibitively labor-intensive; therefore, comparisons on this dataset were not conducted.

In addition to the experimental results presented earlier, four small-scale scenes designated as 'Scene 1' through 'Scene 4' were selected from the proprietary dataset (Feng et al. (2022)), as illustrated in Fig. 9. In each input scene, the point cloud contains both building and non-building objects, all displayed without instance-level labels or GT annotations and visualized using distinct colors. To demonstrate the effectiveness of the proposed planar primitive segmentation method, building and roof segments are extracted from these scenes and highlighted with blue ("Building") and green ("Roof") labels, respectively. Results show that the proposed method successfully segments most building primitives into roof and façade instances, which can effectively support building reconstruction tasks. The evaluated buildings exhibit diverse architectural styles, including apartments, townhouses, and detached houses with varied roof types. Two buildings from Scenes 1 and 2, along with three roofs from Scenes 2–4, exhibit misclassifications. These errors primarily occur when a roof consists of multiple intersecting planes with very small inclination angles between them, or when certain roof tiles occupy an extremely small surface area. In such cases, individual roof tiles are prone to being incorrectly segmented.

4.4. Parameter analysis

A comprehensive parameter sensitivity analysis was conducted to evaluate the robustness of the proposed method across five distinct roof datasets (Roof1–Roof5), with performance assessed through *Cov*

Table 4
Ablation study on the contribution of individual constraints.

Components				Metrics	Roof types							
	C_1	C_2	C_3		Roof1	Roof2	Roof3	Roof4	Roof5	Roof6	Roof7	Roof8
(a)	✓			Cov//WCov	18.54//18.53	89.36//92.06	41.17//40.89	99.73//99.73	94.72//94.70	17.02//19.94	29.19//27.15	46.07//55.86
				Cm//Cr	100.0//100.0	100.0//100.0	100.0//100.0	100.0//100.0	100.0//100.0	100.0//100.0	80.00//80.00	100.0//100.0
				QI	100.0	100.0	100.0	100.0	100.0	100.0	66.67	100.0
(b)		✓		Cov//WCov	45.48//45.52	35.61//35.80	49.60//50.48	33.75//33.75	34.93//34.97	35.37//32.79	37.51//34.37	51.30//48.69
				Cm//Cr	100.0//100.0	75.00//75.00	75.00//75.00	100.0//100.0	100.0//100.0	100.0//100.0	100.0//100.0	100.0//100.0
				QI	100.0	60.00	60.00	100.0	100.0	100.0	100.0	100.0
(c)			✓	Cov//WCov	38.07//38.06	34.99//46.29	34.70//34.89	99.73//99.73	29.75//29.70	28.26//32.18	21.78//25.24	34.96//39.70
				Cm//Cr	83.33//83.33	75.00//75.00	100.0//100.0	100.0//100.0	75.00//75.00	100.0//100.0	60.00//60.00	80.00//80.00
				QI	71.43	60.00	100.0	100.0	60.00	100.0	42.86	66.67
(d)	✓	✓		Cov//WCov	66.73//67.49	83.42//87.24	57.42//56.71	99.73//99.73	93.86//93.85	38.50//44.51	**75.77//75.60**	67.32//74.75
				Cm//Cr	100.0//100.0	100.0//100.0	75.00//75.00	100.0//100.0	100.0//100.0	75.00//75.00	**100.0//100.0**	80.00//80.00
				QI	100.0	100.0	60.00	100.0	100.0	60.00	**100.0**	66.67
(e)	✓		✓	Cov//WCov	94.53//94.51	**89.52//92.19**	36.35//36.28	99.73//99.73	**94.86//94.84**	38.52//49.52	59.62//64.69	28.05//29.34
				Cm//Cr	100.0//100.0	**100.0//100.0**	100.0//100.0	100.0//100.0	**100.0//100.0**	100.0//100.0	80.00//80.00	80.00//80.00
				QI	100.0	**100.0**	100.0	100.0	**100.0**	100.0	66.67	66.67
(f)		✓	✓	Cov//WCov	38.33//38.35	29.95//29.51	45.52//46.56	33.75//33.75	35.25//35.33	35.64//32.90	37.73//35.27	42.92//40.04
				Cm//Cr	83.33//83.33	75.00//75.00	75.00//75.00	100.0//100.0	100.0//100.0	100.0//100.0	80.00//80.00	80.00//80.00
				QI	71.43	60.00	60.00	100.0	100.0	100.0	66.67	66.67
(g)	✓	✓	✓	Cov//WCov	**95.26//95.25**	80.71//85.01	**99.84//99.84**	**99.80//99.80**	93.44//93.43	**83.39//85.91**	50.51//43.85	**95.52//95.42**
				Cm//Cr	**100.0//100.0**	100.0//100.0	**100.0//100.0**	**100.0//100.0**	100.0//100.0	**100.0//100.0**	80.00//80.00	**100.0//100.0**
				QI	**100.0**	100.0	**100.0**	**100.0**	100.0	**100.0**	66.67	**100.0**

and $WCov$ metrics under varying configurations of parameters α, β, and λ, as illustrated in Fig. 10(a-c), respectively. The analysis reveals differentiated sensitivity patterns across the three parameters α, β, and λ. The proposed method demonstrates strong robustness to variations in α across a range of 1 to 7. The performance on Roof1 and Roof4 varies by less than 2%, while other roof types exhibit only minor declines. Sensitivity to β (0.1–0.7) is moderate: structurally simple roofs (Roof1, 4, 5) remain stable, whereas more complex structures (Roof2, 3) show more pronounced reductions (e.g., the coverage (Cov) of Roof2 declines from approximately 49.76 to 42.07). The greatest sensitivity is observed with respect to λ (0.2–0.8); nevertheless, Roof1, 4, and 5 maintain high tolerance. Roof2 is the most affected, with its Cov dropping from 80.84 to 70.40. Overall, the method operates reliably across a broad parameter range, showing minimal variance (< 2%) on key roof structures (Roof1, 4), which confirms its stability for practical applications.

The proposed method demonstrates strong robustness to variations in α across the tested range (1–7), with Roof1 showing minimal fluctuation (Cov: 95.14–96.31; $WCov$: 95.13–96.29) and Roof4 maintaining near-perfect stability ($Cov/WCov \approx$ 99.73–99.80). While Roof2 and Roof3 exhibit moderate variations, only Roof5 shows slight degradation at higher α values, with Cov declining from 53.50 to 48.96. In contrast, the proposed method displays moderate sensitivity to β variations (0.1–0.7). Well-structured roofs including Roof1, Roof4, and Roof5 maintain strong performance across all β values, whereas Roof2 and Roof3 show more pronounced sensitivity, with Cov declining from approximately 49.76 to 42.07 and 52.55 to 49.33, respectively. This pattern suggests that structural complexity influences parameter dependency, with simpler roof forms demonstrating better robustness to β variations. The most significant sensitivity is observed with λ variations (0.2–0.8), where gradual performance degradation occurs across most datasets. Roof2 exhibits the strongest dependency, with Cov decreasing from 80.84 to 70.40, while Roof3 maintains excellent performance ($Cov > 99.2$) until $\lambda = 0.6$ before showing minor decline. Roof1, Roof4, and Roof5 demonstrate superior tolerance to λ variations, indicating that the method's sensitivity to regularization parameters is highly dependent on roof structural characteristics. Overall, the proposed method exhibits high robustness to α variations, moderate robustness to β changes, and controlled sensitivity to λ parameters. The method demonstrates consistent performance across diverse roof structures. This validates its stability and practical applicability. In particular, Roof1 and Roof4 exhibit minimal performance variance (<2%) across all parameter configurations. These findings indicate that parameter tuning can optimize performance for specific roof types. Nevertheless, the method maintains reliable operation across a broad parameter range. This demonstrates its robustness for real-world applications.

4.5. Ablation study

This subsection integrates three geometric constraints into the objective function to identify building components: minimization of point-to-plane distance (C_1), cluster-center proximity (C_2), and directional regularization (C_3). Seven combinations of these constraints are evaluated on the Synthetic Roof Dataset using the previously defined metrics. As summarized in Table 4, the full combination (g) achieves state-of-the-art or highly competitive performance across most roof types. This result is not achieved by any individual constraint or partial combination. It sets new benchmarks on complex structures such as Roof1 (95.26//95.25), Roof3 (99.84//99.84), Roof4 (99.80//99.80), Roof6 (83.39//85.91), and Roof8 (95.52//95.42), while attaining perfect scores (100%) in Completeness (C_m), Correctness (C_r), and QI. Importantly, the full model (g) exhibits consistent performance without the significant fluctuations observed in other configurations. For instance, configuration (e) (C_1+C_3) performs strongly on Roof1, Roof2, and Roof5, but deteriorates sharply on Roof3 (36.35) and Roof8 (28.05). Similarly, configuration (d) (C_1+C_2) performs poorly on Roof3 (57.42). The ablation study reveals the distinct and complementary contributions of each component: (1) C_1: When used alone (a), it performs well on regular roofs (e.g., Roof4, Roof5) but fails on complex cases (e.g., Roof1: 18.54, Roof6: 17.02). While it achieves high correctness in some cases, it lacks overall robustness. C_2: Alone (b), it offers more stable coverage than C_1 on certain roofs (e.g., Roof1: 45.48, Roof8: 51.30), but at the cost of lower overall accuracy, especially on complex shapes (e.g., Roof2: 35.61). It appears to enforce basic geometric consistency but struggles with fine details. C_3: Used in isolation (c), it delivers inconsistent and generally mediocre results, suggesting it serves a supporting rather than a leading role.

Among partial combinations: C_1+C_2 (d): Shows substantial improvement over either component alone, particularly on Roof1 (66.73)

and Roof7 (75.77), indicating that C_2 effectively regularizes the features extracted via C_1. C_1+C_3(e): Achieves near-optimal results on several roofs (e.g., Roof1, Roof2, Roof5), suggesting that C_3 enhances or refines the representations learned through C_1. C_2+C_3(f): Yields minimal improvement over C_2 alone and is generally inferior to combinations involving C_1, underscoring that C_1 provides the essential foundation upon which C_2 and C_3 effectively build. In summary, each constraint contributes uniquely. Here, C_1 serves as the foundational component, whereas C_2 and C_3 provide complementary regularization and refinement. Together, they enable robust and accurate planar primitive segmentation.

4.6. Discussion

4.6.1. Comparative advantage over existing methods

Conventional region growing algorithms (e.g., Xia et al., 2020; Huang et al., 2024; Wang et al., 2018) often lack sufficient geometric regularization, leading to poorly defined planar boundaries and topological inconsistencies. In contrast, while DL-based approaches (Li et al., 2024) show strong potential, their effectiveness is typically constrained by a reliance on large, precisely annotated datasets. This requirement presents a significant practical challenge in Earth Observation (EO), where acquiring comprehensive GT data is often costly and labor intensive.

To address these limitations, our framework introduces three geometrically interpretable constraints grounded in photogrammetric principles: point to plane distance minimization, cluster center proximity preservation, and directional regularization. Unlike the heuristic rules used in some methods or the data driven principles of others (e.g., Bohg et al., 2013), our constraints establish a physically transparent and principled segmentation mechanism. This design collectively enhances boundary precision, improves semantic coherence within regions, and mitigates common artifacts such as plane over extension and boundary leakage (e.g., Landrieu and Boussaha, 2019; Lin et al., 2018). Specifically, standard k planes clustering implementations can exhibit unbounded plane growth in complex urban scenes. Our method counters this through explicit directional regularization, achieving superior boundary fidelity and directly reducing the over segmentation of continuous roof surfaces, a known issue in urban reconstruction. Furthermore, the cluster center proximity constraint minimizes the characteristic leakage error often observed when region growing techniques are applied to airborne LiDAR data. By integrating the rigorous parametric modeling tradition of photogrammetry with modern numerical optimization, the framework provides a scalable, interpretable, and robust solution suited for large scale urban analysis in EO applications.

4.6.2. Limitations and challenges

While effective, the proposed framework exhibits three primary methodological limitations that reflect broader challenges in data-driven 3D building reconstruction. First, the reliable separation of adjacent planar patches with very low inclination angles remains difficult. This is a common geometric ambiguity in urban scenes, where subtle surface variations or material differences cannot be resolved by geometry alone. Second, the method displays a tendency toward under-segmentation in regions of near-coplanar surface distributions, where minimal geometric contrast may cause distinct planar entities to merge. Third, the segmentation results are sensitive to the initial parameter estimates. This characteristic, shared by many optimization-based clustering methods, necessitates careful initialization to ensure the stability and reproducibility of the outcomes.

These limitations are not unique to our framework but are indicative of persistent challenges within the field. They also highlight strategic opportunities for synergistic integration with complementary EO data sources. For example, multispectral satellite imagery could provide material-based or functional spectral cues to distinguish between geometrically similar, low-inclination surfaces. Concurrently, higher-resolution data from UAV platforms could enhance the geometric fidelity of point clouds, thereby improving the discrimination of near-coplanar structures. Furthermore, the sensitivity to initialization could be mitigated by employing transfer-learning strategies to adapt parameters across different urban contexts, or by integrating prior semantic knowledge from established urban classification products and GIS databases. Addressing these limitations through multi-source data fusion and knowledge-informed refinement constitutes a promising direction for enhancing the robustness and general applicability of urban 3D reconstruction methods.

4.6.3. Extensibility and future research directions

The extensibility of the proposed framework is a core design principle, ensuring its adaptability to evolving data and application needs in EO. Its optimization-based architecture provides a natural and flexible structure for integrating multi-modal EO data, thereby reducing reliance on purely geometric information. A straightforward pathway for enhancement involves extending the point feature vector to incorporate complementary data layers. For example, this could include integrating spectral indices from optical imagery to infer material properties, temporal change indicators from multi-epoch acquisitions to model urban dynamics, or semantic attributes from urban GIS databases to introduce functional context. Such multi-modal data fusion is essential for addressing persistent gaps in modeling heterogeneous urban environments, where geometric point clouds alone are often insufficient to resolve complex structural and functional details. The framework's inherent adaptability allows it to capitalize on emerging EO data streams, including those from next-generation satellite constellations and increasingly ubiquitous UAV platforms, thereby enhancing its practical utility and relevance for contemporary urban sensing challenges.

Future research will prioritize the systematic validation and enhancement of the framework's transferability. Key directions include: (1) evaluating performance across diverse sensor types, such as comparing results from airborne LiDAR and satellite-derived point clouds; (2) assessing robustness under varying geographic and architectural contexts; and (3) scaling the method for applications ranging from individual building analysis to district-wide urban modeling. These efforts aim to solidify the framework's position within the advanced toolkit for urban remote sensing and 3D reconstruction, ensuring its applicability to a wide spectrum of real-world scenarios.

5. Conclusion

This study has introduced a constrained kPC framework for robust segmentation of planar primitives from building point clouds. To address the persistent challenges in extracting reliable building primitives, especially for complex and stylistically diverse urban structures, the proposed method incorporates three geometrically meaningful constraints. These constraints are: point-to-plane distance minimization, cluster-center proximity preservation, and directional regularization. By formulating the segmentation task as a mixed-integer non-convex optimization problem, we solve it efficiently using an alternating minimization strategy supported by SVD, which iteratively refines cluster assignments and plane parameters until convergence. Comprehensive experiments were conducted on both synthetic and real-world datasets. The results confirm that the method achieves high segmentation accuracy on synthetic data and the Building3D benchmark. Furthermore, it outperforms current state-of-the-art techniques in both quantitative and qualitative evaluations.

The integration of the three constraints enables the framework to support both semantic and instance segmentation while mitigating the problem of unbounded plane extension common in conventional kPC implementations. These attributes make the approach particularly suitable for real-world applications such as 3D building reconstruction

and urban modeling, where structural accuracy and semantic coherence are critical. Looking forward, the constrained kPC framework offers a flexible and efficient alternative to heavily data-dependent DL-based methods, especially in annotation-scarce scenarios. Future work will focus on extending the formulation to incorporate non-planar primitives and exploring adaptive constraint weighting for handling multi-scale urban scenes.

CRediT authorship contribution statement

Hongxin Yang: Writing – original draft, Methodology, Formal analysis, Conceptualization, Investigation, Project administration, Funding acquisition. **Zhipeng Luo:** Writing – review & editing, Conceptualization, Validation, Visualization. **Dedong Zhang:** Writing – review & editing. **Jonathan Li:** Writing – review & editing, Supervision, Resources, Software.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hongxin Yang reports financial support provided by the Shanghai Science and Technology Program (Grant No. 25ZR1402133), the Postdoctoral Fellowship Program of CPSF (Grant No. GZC20250239), and the Shanghai Magnolia Talent Program-Pujiang Project (Grant No. 25PJD031). Given Jonathan Li's role as Editor-in-Chief of the International Journal of Applied Earth Observation and Geoinformation, he was not involved in the peer review process of this article. The other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Shanghai Science and Technology Program (Grant No. 25ZR1402133), the Postdoctoral Fellowship Program of CPSF (Grant No. GZC20250239), the National Natural Science Foundation of China (Grant No. 42301503), and the Shanghai Magnolia Talent Program-Pujiang Project (Grant No. 25PJD031).

Data availability

The authors do not have permission to share data.

References

Bohg, J., Morales, A., Asfour, T., Kragic, D., 2013. Data-driven grasp synthesis—a survey. IEEE Trans. Robot. 30 (2), 289–309. http://dx.doi.org/10.1109/TRO.2013.2289018.

Bradley, P.S., Mangasarian, O.L., 2000. K-plane clustering. J. Global Optim. 16 (1), 23–32, URL: https://link.springer.com/article/10.1023/A:1008324625522.

Choi, S., Kim, T., Yu, W., 1997. Performance evaluation of RANSAC family. J. Comput. Vis. 24 (3), 271–300. http://dx.doi.org/10.5244/C.23.81.

Du, S., Tu, C., Van Wyk, B.J., Chen, Z., 2011. Collinear segment detection using HT neighborhoods. IEEE Trans. Image Process. 20 (12), 3612–3620. http://dx.doi.org/10.1109/TIP.2011.2155076.

Feng, H., Chen, Y., Luo, Z., Sun, W., Li, W., Li, J., 2022. Automated extraction of building instances from dual-channel airborne LiDAR point clouds. Int. J. Appl. Earth Obs. Geoinf. 114, 103042. http://dx.doi.org/10.1016/j.jag.2022.103042.

Ferrara, R., Virdis, S.G., Ventura, A., Ghisu, T., Duce, P., Pellizzaro, G., 2018. An automated approach for wood-leaf separation from terrestrial LIDAR point clouds using the density based clustering algorithm DBSCAN. Agric. for. Meteorol. 262, 434–444. http://dx.doi.org/10.1016/j.agrformet.2018.04.008.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395. http://dx.doi.org/10.1145/358669.358692.

Haala, N., Kada, M., 2010. An update on automatic 3D building reconstruction. ISPRS J. Photogramm. Remote Sens. 65 (6), 570–580. http://dx.doi.org/10.1016/j.isprsjprs.2010.09.006.

Hamerly, G., Elkan, C., 2003. Learning the k in k-means. Adv. Neural Inf. Process. Syst. 16, 281–288.

Huang, J., Stoter, J., Peters, R., Nan, L., 2022. City3D: Large-scale building reconstruction from airborne LiDAR point clouds. Remote. Sens. 14 (9), 2254. http://dx.doi.org/10.3390/rs14092254.

Huang, Z., Wen, Y., Wang, Z., Ren, J., Jia, K., 2024. Surface reconstruction from point clouds: A survey and a benchmark. IEEE Trans. Pattern Anal. Mach. Intell. 46 (12), 9727–9748. http://dx.doi.org/10.1109/TPAMI.2024.3429209.

Illingworth, J., Kittler, J., 1988. A survey of the hough transform. Comput. Vis. Graph. Image Process. 44 (1), 87–116. http://dx.doi.org/10.1016/S0734-189X(88)80033-1.

Kong, D., Xu, L., Li, X., Li, S., 2013. K-plane-based classification of airborne LiDAR data for accurate building roof measurement. IEEE Trans. Instrum. Meas. 63 (5), 1200–1214. http://dx.doi.org/10.1109/TIM.2013.2292310.

Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. Int. J. Comput. Vis. 99 (1), 69–85, URL: https://link.springer.com/article/10.1007/s11263-012-0517-8.

Landrieu, L., Boussaha, M., 2019. Point cloud oversegmentation with graph-structured deep metric learning. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit... pp. 7440–7449.

Li, L., Li, Q., Xu, G., Zhou, P., Tu, J., Li, J., Li, M., Yao, J., 2024. A boundary-aware point clustering approach in euclidean and embedding spaces for roof plane segmentation. ISPRS J. Photogramm. Remote Sens. 218, 518–530. http://dx.doi.org/10.1016/j.isprsjprs.2024.09.030.

Li, Z., Shan, J., 2022. RANSAC-based multi primitive building reconstruction from 3D point clouds. ISPRS J. Photogramm. Remote Sens. 185, 247–260. http://dx.doi.org/10.1016/j.isprsjprs.2021.12.012.

Li, L., Yao, J., Tu, J., Liu, X., Li, Y., Guo, L., 2020. Roof plane segmentation from airborne LiDAR data using hierarchical clustering and boundary relabeling. Remote. Sens. 12 (9), 1363. http://dx.doi.org/10.3390/rs12091363.

Lin, Y., Wang, C., Zhai, D., Li, W., Li, J., 2018. Toward better boundary preserved supervoxel segmentation for 3D point clouds. ISPRS J. Photogramm. Remote Sens. 143, 39–47. http://dx.doi.org/10.1016/j.isprsjprs.2018.05.004.

Nan, L., Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds. In: Proc. IEEE Int. Conf. Comput. Vis... pp. 2353–2361.

Ochmann, S., Vock, R., Klein, R., 2019. Automatic reconstruction of fully volumetric 3D building models from oriented point clouds. ISPRS J. Photogramm. Remote Sens. 151, 251–262. http://dx.doi.org/10.1016/j.isprsjprs.2019.03.017.

Sampath, A., Shan, J., 2009. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. IEEE Trans. Geosci. Remote Sens. 48 (3), 1554–1567. http://dx.doi.org/10.1109/TGRS.2009.2030180.

Shao, Y.-H., Bai, L., Wang, Z., Hua, X.-Y., Deng, N.-Y., 2013. Proximal plane clustering via eigenvalues. Procedia Comput. Sci. 17, 41–47. http://dx.doi.org/10.1016/j.procs.2013.05.007.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. In: ISPRS Workshop Laser Scan. 2007 Silvilaser 2007. vol. 36, pp. 407–412, URL: https://shs.hal.science/halshs-00264843v1.

Wang, R., Huang, S., Yang, H., 2023. Building3d: A urban-scale dataset and benchmarks for learning roof structures from point clouds. In: Proc. IEEE Int. Conf. Comput. Vis... pp. 20076–20086.

Wang, X., Liu, S., Shen, X., Shen, C., Jia, J., 2019. Associatively segmenting instances and semantics in point clouds. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit... pp. 4096–4105.

Wang, R., Peethambaran, J., Chen, D., 2018. Lidar point clouds to 3-D urban models : a review. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 11 (2), 606–627. http://dx.doi.org/10.1109/JSTARS.2017.2781132.

Wichmann, A., Agoub, A., Kada, M., 2018. Roofn3d: Deep learning training data for 3d building reconstruction. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. 42, 1191–1198. http://dx.doi.org/10.5194/isprs-archives-XLII-2-1191-2018.

Xia, S., Chen, D., Wang, R., Li, J., Zhang, X., 2020. Geometric primitives in LiDAR point clouds: A review. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 13, 685–707. http://dx.doi.org/10.1109/JSTARS.2020.2969119.

Yan, J., Shan, J., Jiang, W., 2014. A global optimization approach to roof segmentation from airborne lidar point clouds. ISPRS J. Photogramm. Remote Sens. 94, 183–193. http://dx.doi.org/10.1016/j.isprsjprs.2014.04.022.

Yu, M., Lafarge, F., 2022. Finding good configurations of planar primitives in unorganized point clouds. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit... pp. 6367–6376.

Zhang, C., Fan, H., 2022. An improved multi-task pointwise network for segmentation of building roofs in airborne laser scanning point clouds. Photogramm. Rec. 37 (179), 260–284. http://dx.doi.org/10.1111/phor.12420.

Zhang, Y., Tian, M., Liu, B., 2025. An action decoding framework combined with deep neural network for predicting the semantics of human actions in videos from evoked brain activities. Front. Neuroinform. 19, 1526259. http://dx.doi.org/10.3389/fninf.2025.1526259.