Kinetic Expansion of Linear Structural Elements: A Hybrid Method for Floorplan Reconstruction From Indoor Scene Point Cloud

Yunlin Tu, Wenzhong Shi, Yangjie Sun, and Min Zhang

***Abstract*—Indoor floorplans are widely used in fields like building information modeling, indoor navigation, emergency response, smart buildings, and architectural design simulation. However, reconstructing accurate floorplans from indoor laser point clouds is challenging due to the complexity, clutter, and occlusions of indoor structures. We propose kinetic expansion of linear structural elements (KELSE), an indoor scene floorplan reconstruction method to address these challenges. We design a structural element extraction method that integrates geometric feature constraints with semantic information to identify structural elements such as walls, doors, windows, ceilings, and floors in complex indoor scenes. A kinetic data structure expansion and undirected graph optimization are then used to reconstruct the complete floorplan. Experimental results show that KELSE achieves high accuracy and completeness, with room reconstruction reaching 0.98 and 0.95, respectively. KELSE provides an efficient and precise solution for floorplan reconstruction from indoor LiDAR point cloud data.**

***Index Terms*—Indoor floorplan reconstruction, kinetic data structure (KDS), kinetic expansion, lidar point cloud, structural element extraction.**

I. Introduction

AUTOMATED floorplan reconstruction from 3D point cloud data has been a central research topic in computer graphics and computer vision, with growing importance in areas like building information modeling (BIM), indoor navigation, smart buildings, and architectural design simulation [1], [2]. Recent studies have focused on reconstructing floorplans from indoor scene structural elements [3], [4]. This method involves identifying and extracting primary structures, such as ceilings, floors, and walls, from point clouds [5], [6]. However, accurately identifying and extracting these structures in complex scenes with furniture occlusions [7], [8] remains a significant challenge. Scenes often contain noise and missing regions limit the accuracy and automation of floorplan reconstruction. As a result, high-quality floorplans are still often generated manually, which is resource-intensive and inefficient [9], [10].

Existing methods primarily focus on accurately extracting indoor structural elements from point clouds and assembling them into scene floorplans. To overcome the interference caused by scene complexity and missing regions. Researchers have employed traditional detection methods, such as random sample consensus (RANSAC) [11] and region growing [12], [13], as well as deep learning approaches [14], [15], [16] to detect and extract structural elements from point clouds. Structural elements are often represented as planes to denote wall positions or as corner points to mark room corners, forming the basis for constructing room geometry. Traditional extraction methods offer high processing efficiency and broad applicability, while deep learning methods perform better in handling complex scenes. However, both approaches struggle to effectively distinguish between structural and nonstructural elements in scenes with complex layouts, cluttered noise, and missing regions.

After extracting the structural elements, researchers have proposed various methods to combine them into accurate room floorplans. One popular combination strategy is to design an optimization framework that directly connects the detected structural elements [17], [18], [19]. However, this strategy is typically time-consuming and highly dependent on the accuracy of the structural element detection results. Another combination strategy involves partitioning the 2D space into polygons and assigning a room instance label to each polygon [2], [20], [21]. Compared to the former, this strategy is more robust. However, it relies on an accurate estimation of the room instance label map, which means that, in addition to the point cloud, additional instance label data must be provided as input.

Despite the advances in these methods, challenges remain, especially when dealing with complex scenes and missing regions. To address these issues, we propose KELSE, a method that enhances floorplan reconstruction, driven by kinetic expansion. The method begins by performing voxel downsampling and polynomial fitting on the point cloud, reducing

Received 12 June 2025; revised 27 August 2025; accepted 23 September 2025. Date of publication 29 September 2025; date of current version 14 October 2025. This work was supported in part by the "Theories for Spatiotemporal Intelligence and Reliable Data Analysis" (Project ID: HZQSWS-KCCYB-2024058); in part by the Otto Poon Charitable Foundation Smart Cities Research Institute, The Hong Kong Polytechnic University (Work Program: CD06); in part by The Hong Kong Polytechnic University (U-ZECR); and in part by the RGC Grant for Theme-based Research Scheme Project (T43-513/23-N). *(Corresponding author: Wenzhong Shi.)*

Yunlin Tu is with the School of Environment and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China (e-mail: arno.yl.tu@cumt.edu.cn).

Wenzhong Shi and Yangjie Sun are with The Hong Kong Polytechnic University-Shenzhen Technology and Innovation Research Institute (Futian), Shenzhen 518000, China, also with the Otto Poon Charitable Foundation Smart Cities Research Institute, The Hong Kong Polytechnic University, Hong Kong, and also with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong (e-mail: john.wz.shi@polyu.edu.hk; yangjie.sun@polyu.edu.hk).

Min Zhang is with the School of Geosciences and Info-Physics, Central South University, Changsha 410083, China (e-mail: zhang.min@csu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2025.3615609

points while retaining key structural information for improved algorithm efficiency. A novel structural element extraction technique combines semantic and geometric constraints to classify the point cloud into structural and nonstructural elements. Once the structural elements are identified, we draw on the work of Bauchet and Lafarge [22], utilizing quadratic programming (QP) and kinetic data structure (KDS) to optimize the relationships between elements and perform kinetic expansion to form closed polygons. Finally, a graph-based optimization algorithm identifies and retains polygons that accurately represent room areas, automatically excluding invalid or redundant polygons caused by noise and occlusions. In summary, KELSE enhances efficiency and accuracy in floorplan reconstruction by addressing key challenges in complex scenes. The key contributions of this work are as follows.

1) An automatic floorplan generation framework, KELSE, is proposed to address challenges such as complex indoor layouts and missing regions. It also allows floorplans to be reconstructed without requiring explicit prior knowledge of the scene.
2) A novel structural element extraction method is proposed, integrating semantic information with geometric feature constraints to improve accuracy in cluttered and incomplete point clouds.
3) An innovative floorplan simplification algorithm is proposed to remove redundant polygons by eliminating erroneous edges and merging intersections, ensuring the final floorplan's accuracy and conciseness.

II. Related Work

Reconstructing 2D floorplans from point clouds is crucial for indoor scene reconstruction and is significant across various domains, including computer vision, BIM, and smart buildings. Given the complexity of a comprehensive review in this domain, we will provide a brief overview of two key areas central to our research: structural element extraction and floorplan generation.

A. Structural Element Extraction

In indoor scene reconstruction, structural point extraction involves identifying points within a point cloud that represent the scene's structural elements, such as ceilings, walls, and floors. The goal is to accurately extract and classify the features of these points. Research on this topic can be broadly categorized into two main approaches: (i) geometry-based methods and (ii) deep learning-based methods.

Geometric feature-based methods have been extensively studied and applied. These methods extract structural points by analyzing local geometric features such as normal vectors, plane geometry, and curvature. For example, as shown in Table I, Jenke et al. [24] reconstructed point cloud surfaces using fundamental geometric principles, Li et al. [23] leveraged local geometric features for target recognition, and Sanchez and Zakhor [27] employed PCA and RANSAC to classify and segment basic indoor structures. These methods perform well in scenes with distinct geometric features but face significant limitations in complex environments. Their accuracy declines in the presence of occlusions, complex shapes, or uneven data densities.

TABLE I
Overview of Representative Structural Point Extraction Method

Categories	Methods
Geometric feature-based	Object recognition based on local geometric features of points [23].
	Point cloud surface reconstruction based on simple geometric shapes [24].
	Identifying pipes from cluttered environments using curvature features [25].
	Indoor scene reconstruction based on planar geometric features [26].
	Hybrid segmentation method based on region growth and geometric shape to extract structural elements [12].
	Classification and segmentation of indoor basic structures via PCA and RANSAC [27].
	Using RANSAC for the extraction of indoor planar structures [28].
Deep learning-based	Multi layer perceptron-based indoor scene classification and segmentation [29].
	Local feature aggregation (LFA)-based large-scale indoor scene classification and segmentation [30].
	Self-attention layer-based indoor scene classification and segmentation [15].
	Prompt and semantic information-based collaborative point cloud feature extraction [31].

Additionally, they are prone to getting stuck in local optima, failing to capture the holistic structure of the scene when confronted with multiple similar structures or overlapping shapes. This results in a decrease in the overall quality of the reconstruction.

Deep learning-based methods have made substantial progress in recent years. Prominent methods such as PointNet [14], [26], RandLA-Net [30], and Point Transformer [15] utilize neural networks to learn both global and local features of point clouds, enabling more accurate classification. However, despite their advancements, these methods still encounter several challenges. First, they require large-scale labeled datasets, which are time-consuming and costly to create. Additionally, complex models are prone to overfitting, necessitating manual parameter tuning and significant computational resources. Techniques such as transfer learning and data augmentation [32] mitigate these issues to some extent, but the generalization capability of these models remains limited, particularly in real-world applications.

B. Floorplan Generation

Recent advancements in the field of floorplan generation have led to the development of various methods. Among these, two intuitive approaches can be broadly categorized: (i) 2D space partitioning methods and (ii) structural element composition methods.

2D space partitioning methods segment and optimize indoor spaces to generate floorplans based on constraints such as user-defined boundaries, internal unit sizes, shapes, and their

TABLE II
OVERVIEW OF CATEGORIES IN FLOORPLAN RECONSTRUCTION METHODS WITH VARYING WORKFLOWS

Categories	Methods
Partitioning of 2D space method	Utilize various features, such as Voronoi diagrams, distance metrics, and morphological techniques, to partition indoor scenes [33].
	Using MIQP methods to generate indoor floorplans [34].
	Automatically generate floorplans based on user-specified architectural boundaries and design constraints [35].
	Using GANs to assemble individual rooms and their adjacency relationships into a floorplan [36], [37].
Composition of structural elements method	Extract room corners and structural elements from RGB-D data and integrate them into a floorplan using neural networks [18].
	Segment room instances from RGB-D data and integrate them into a floorplan using neural networks [17].
	Leverage neural networks to identify the number of rooms and internal corners, and integrate these components into a floorplan [19].
	FloorUSG [38] extracts walls via point cloud geometry and image semantics, converts them to 2D line segments, and optimizes the floorplan using integer linear programming.
	SceneScript [39] reformulates 3D reconstruction as sequence generation by predicting human-readable commands (e.g., "make wall") via an encoder-decoder Transformer, enabling compact, interpretable, and editable scene representation.
	FRI-Net [40] employs a Transformer-based encoder-decoder to reconstruct 2D floorplans from 3D point clouds using structurally regularized implicit representations, trained in two stages to learn axis-aligned and diagonal lines for robust and consistent room polygon generation.
	PolyRoom [41] reconstructs vectorized floorplans from point clouds by optimizing room queries step-by-step, using fixed-length vertex sequences, Transformer-based encoding, room-aware initialization, and intra-/inter-room attention to ensure accuracy and topological consistency.

interrelationships. For instance, Bormann et al. [33] effectively partitioned indoor scenes using various features, including Voronoi diagrams, distance metrics, and morphological techniques. Wu et al. [34] employed a mixed-integer quadratic programming (MIQP) approach to propose an accurate mathematical model for generating indoor floorplans. Hu et al. [35] further investigated methods for automatically generating floorplans based on user-defined building boundaries and design constraints, improving both generation efficiency and accuracy. Additionally, the authors in [36] and [37] introduced House-GAN and House-GAN++, which use generative adversarial networks (GANs) to partition and combine room layouts, generating diverse and high-quality floorplans. Although these methods perform well when room instance labels are accurate and complete, their performance significantly declines when label information is inaccurate or missing.

The structural element composition methods, on the other hand, focus on connecting detected structural elements to form a complete floorplan. A representative method in this category is FloorNet [18], which uses neural networks to extract room corner points and structural elements, linking them together for floorplan reconstruction. Chen et al. [17] proposed Floor-SP, which employs a room-wise coordinate descent algorithm to combine segmented room instances with corner/edge likelihood maps, resulting in the integration of a complete floorplan. In addition, FloorUSG proposed by Han et al. [38] integrates image semantics and point cloud geometry to generate candidate wall segments, and selects the optimal combination via integer linear programming to produce structurally consistent indoor floorplans. More recent studies, such as RoomFormer [19], SceneScript [39], FRI-Net [40], and PolyRoom [41], apply Transformer architectures to reconstruct floorplans by predicting sets of polygons based on room and corner point information. These methods leverage encoder-decoder structures to represent scenes as sequences or structured queries, often incorporating geometric priors, dense supervision, and room-aware attention mechanisms to improve reconstruction accuracy, regularity, and topological consistency.

These methods perform well in non-Manhattan structures and large-scale scenes but face challenges with irregular structures and fine details. Their reliance on predefined nodes and postprocessing steps often limits the precision of floorplans in complex scenes.

Despite significant progress in various methods for structural point extraction and floorplan generation, inherent limitations and challenges remain to be addressed. In structural point extraction, geometry-based methods often struggle in complex scenes, particularly when dealing with occlusions, complex shapes, or uneven data densities. Deep learning-based methods typically require large labeled datasets and may exhibit limited generalization capabilities. In floorplan generation, the aforementioned methods also face challenges in handling complex constraints, boundary conditions, and insufficient feature capture. These challenges can negatively impact the accuracy of generated floorplans, particularly in representing spatial relationships.

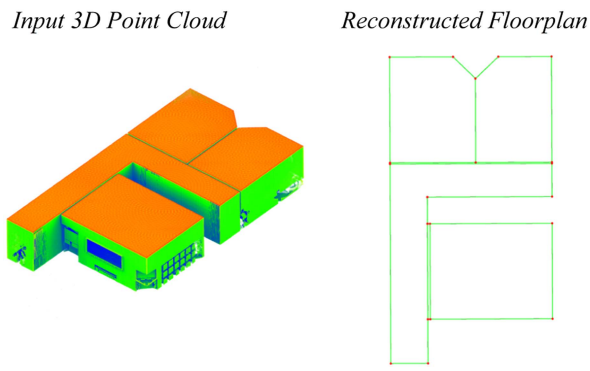


Fig. 1. Indoor floorplan reconstruction: KELSE extracts structural elements from the given indoor scene point cloud, converts them into a KDS, and performs kinetic expansion to reconstruct the indoor floorplan.

III. Methodology

The proposed method, KELSE, processes point clouds from real-world indoor scenes to generate a floorplan, represented as a planar graph with each closed polygon delineating room boundaries. The coordinates of the input point cloud are simply aligned, with the upward direction being aligned to the Z-axis of the world coordinate system.

As illustrated in Fig. 2, KELSE encompasses three primary steps. Initially, to enhance reconstruction accuracy, a suite of preprocessing techniques is implemented, including downsampling, surface smoothing, and semantic segmentation. These steps substantially diminish redundant points, noise, and outliers, while concurrently retaining essential structural points and semantic information. Subsequently, we introduce a multifeature integrated structural element extraction method that leverages semantic information, geometric features, and positional data to accurately identify the structural elements within the scene. Ultimately, once the principal structural element points are identified, they are projected onto a 2D plane. Employing linear fitting, kinetic expansion, and an undirected graph-based optimization algorithm, we generate and refine the closed polygons. This process ensures that the floorplan accurately and rationally mirrors the indoor space's layout.

A. Data Preprocessing

Generating a floorplan directly from raw point cloud is impractical due to the significant impact of redundant information, noise, and outliers on the efficiency and accuracy of KELSE's results. To mitigate these issues, we have implemented a series of preprocessing steps, such as downsampling, surface smoothing, and semantic segmentation. These steps effectively reduce redundant points, noise, and outliers while preserving key structural points and incorporating semantic information, which facilitates subsequent processing.

Noise reduction: The presence of noise, outliers, redundant points, and inaccuracies during data collection can significantly impair the efficiency of KELSE and the accuracy of the generated floorplan. Therefore, we have designed a preprocessing pipeline that includes voxel downsampling and moving least squares smoothing (MLS) [42]. This pipeline eliminates these disturbances effectively. Voxel downsampling segments the input point cloud into numerous small cubes, computes the centroid coordinates for each set of points within a cube, and employs these centroids to simplify the point cloud, effectively reducing the data volume while preserving the structural information of the room point cloud. However, downsampling alone cannot eliminate inaccuracies in the data, so we apply MLS to the downsampled point cloud to obtain a point cloud with uniform density and smooth surfaces. MLS is a widely used technique for adaptive fitting of discrete data. This method fits a local smooth surface within the neighborhood of each point using a moving window and projects the point coordinates onto the fitted surface, thus achieving point cloud smoothing. The following equation represents the process:

$$E\ (\phi_p) = \sum_i \Theta\left(||p_i - p||_2\right)\left(\phi_p\left(p_i\right) - \phi_i\right)^2 \tag{1}$$

where ϕ_p represents the polynomial surface function fitted at point P, and Θ is a weight function that calculates the weighted distance between point P and its nearest neighbors. The term $(\phi_p(p_i) - \phi_i)^2$ represents the gap between the smoothed point coordinates and the original coordinates. For each point p_i in the input point cloud, MLS first identifies a specified number of neighboring points using a KD-tree. A polynomial function ϕ_p is then fitted to these neighboring points, and the optimal surface is obtained by minimizing (1). Finally, each point p_i is projected onto the fitted surface, resulting in a smooth and uniform point cloud.

Semantic segmentation: To accommodate multifeature constraints, semantically annotated points are required as input. For this purpose, we utilize RandLA-Net [30], trained on the S3DIS dataset [43], to segment indoor scene point clouds. The goal of this work is to extract structural elements that describe the spatial distribution of the scene, rather than perform detailed object classification. To achieve this, we follow the structural element classification proposed by Nikoohemat et al. [6] and reclassify the S3DIS category labels accordingly. Specifically, points labeled as walls, doors, windows, and columns are designated as primary structural elements, while ceilings and floors are treated as secondary structural elements. Nonstructural elements include points for bookshelves, chairs, tables, sofas, beams, and clutter. This reclassification reduces computational overhead during network training and improves generalization by reducing the number of categories. Fig. 3 compares the reclassified labels with the original ones. Since structural and nonstructural elements differ mainly in spatial geometric features rather than texture, we simplify the input features to only include ***(x, y, z)*** spatial coordinates, omitting the color components ***(r, g, b)***. The decision not to incorporate color information from the training dataset as a learnable input is based on the observation that the spatial distribution of structural elements provides sufficient distinguishing information for segmentation tasks, and including color would not significantly improve performance in this context.

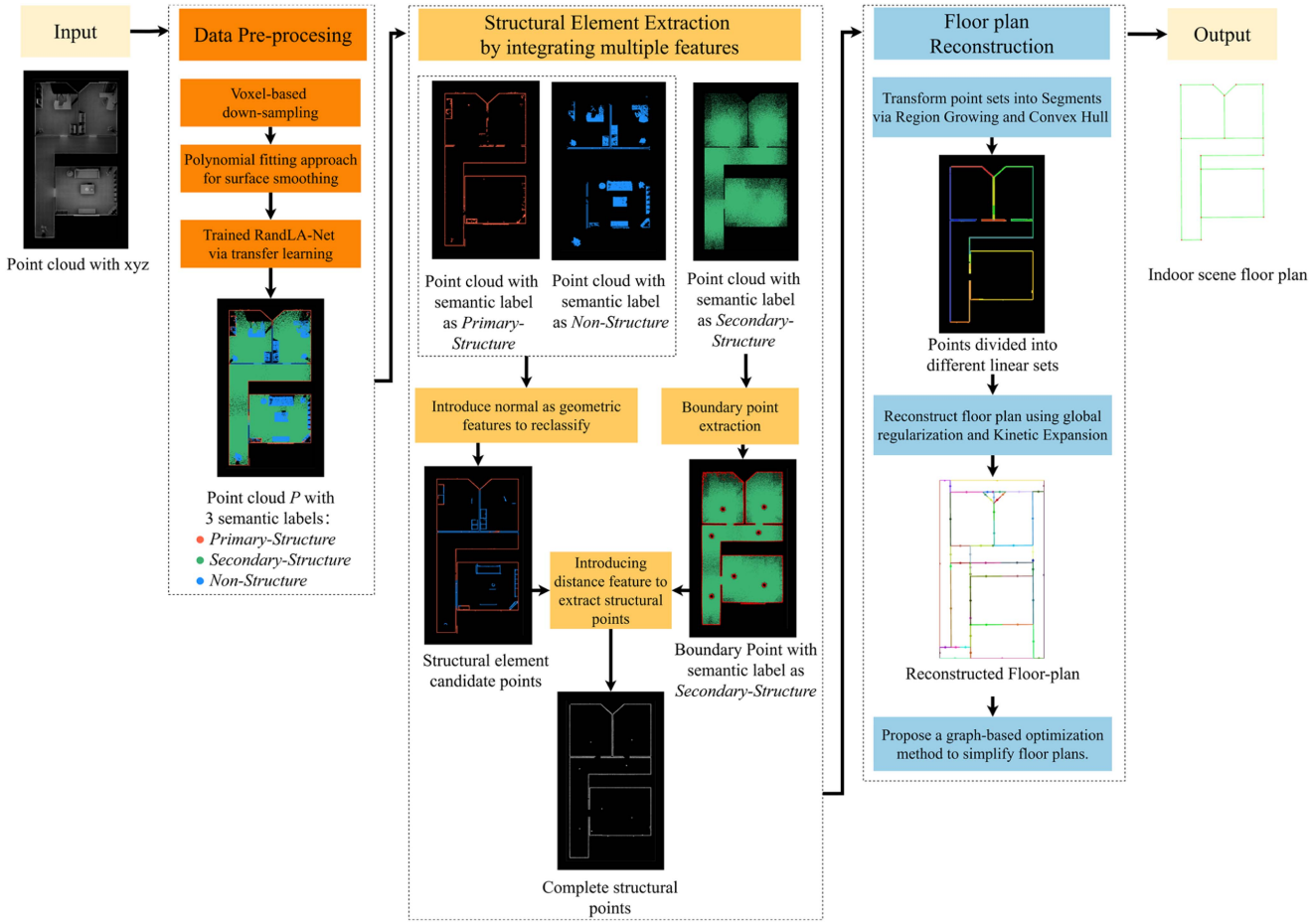


Fig. 2. Overview of KELSE.

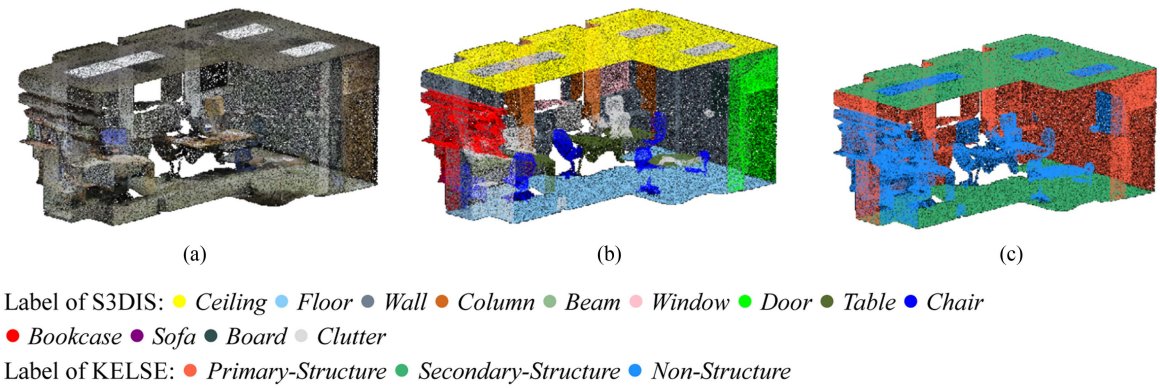


Fig. 3. Comparison of reclassification based on structural element definition. (a) Indoor scene point cloud of the S3DIS dataset. (b) Original labels of the S3DIS dataset. (c) Labels obtained through division based on structural elements.

B. Structural Element Extraction by Integrating Multiple Features

To accurately represent the layout of indoor spaces, extracting structural element points from the point cloud is essential before reconstructing the indoor scene's floorplan. Given that indoor scenes vary in height, area, and often contain furniture with similar geometric features, such as desks and bookshelves, methods relying solely on geometric features or prior knowledge, such as height, are insufficient for ensuring accuracy. To overcome this limitation, we propose a segmentation method that integrates multiple feature constraints. This method combines semantic information, geometric features, and spatial positioning to comprehensively identify structural elements, ensuring

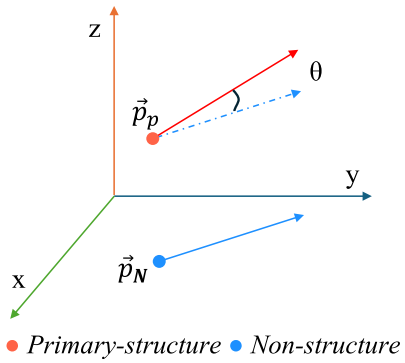


Fig. 4. Extraction of structural elements based on point geometric similarity constraints. Arrows represent the direction of the normal vectors of the points. If the angle $\boldsymbol{\theta}$ formed by the normal vectors of the nonstructural point $\boldsymbol{P_N}$ and the primary structural point $\boldsymbol{P_p}$ satisfies the threshold, then they are considered to have similar geometric features.

the completeness of the extracted results.

$$\theta = arccos\left(\frac{\vec{p}_p \cdot \vec{p}_N}{\left|\vec{p}_p\right| \vec{p}_N|}\right). \tag{2}$$

Structural element extraction with multifeature constraints: In indoor scenes, the complexity of primary structural components often complicates their accurate identification by segmentation networks. In most indoor scenes, walls are perpendicular to the horizontal plane and intersect with the boundaries of ceilings and floors. Building on this observation, our method integrates semantic information, geometric features, and 2D positional information to correct misclassifications made by the segmentation network. These three features work in a complementary manner during the fusion process. Semantic cues provide categorical priors based on learned knowledge, while geometric features [e.g., normals and angles as in (2)] offer local surface characteristics that help distinguish flat, planar structures. Meanwhile, 2D positional information serves as a spatial constraint, aligning structural points with the boundary distribution of ceilings and floors. When inconsistencies arise—for example, when the semantic prediction classifies a point as nonstructural but its geometric and positional features suggest otherwise—the fusion process prioritizes geometric alignment and spatial context over pure semantic labeling. This rule-based resolution strategy ensures that structural consistency is maintained, particularly in ambiguous regions. Although prior studies [4], [5] employed methods such as RANSAC to fit local structural surfaces for detecting and refining elements like walls, the inherent randomness in such sampling-based approaches can lead to inaccuracies. In contrast, our method leverages point-level geometric features to achieve more precise classification of structural elements, as illustrated in (2), Figs. 4 and 6(a).

Where $\vec{p}_p$ and $\vec{p}_N$ represent the normal vectors of the primary structural element point P_p and the nonstructural element point P_N, respectively, and θ denotes the angle between them. Nonstructural points exhibiting geometric compatibility with adjacent primary structural points—as determined by angular deviation θ and a predefined threshold θg—are reclassified as structural elements. Additionally, since walls typically intersect with ceilings and floors, we assume that the positions of all primary structural points in the 2D plane align with the boundaries of secondary structural elements. Based on this assumption, we extract the boundary points of secondary structural elements to eliminate nonstructural elements, as illustrated in Figs. 5 and 6(b).

C. Floorplan Reconstruction

Following the floorplan definition by Fang et al. [3], we reconstruct the indoor scene as a closed polygon floorplan on a 2D plane after extracting primary structural element points. The structural element points from the 3D point cloud are first projected onto a 2D plane, and linear primitives are generated using region growing and convex hull fitting. To improve geometric relationships, we apply QP optimization to enhance orthogonality and parallelism. The optimized linear primitives are then converted into a KDS, with its kinetic expansion used to progressively generate closed polygons. Finally, an undirected graph-based optimization function removes redundant edges and simplifies the polygons, ensuring that the planar graph accurately represents the indoor layout.

Linear primitive fitting: To simplify the point cloud and facilitate floorplan generation, we use geometric model fitting techniques. The primary structural element points are projected onto a 2D plane and simplified into line segment models through linear fitting, transforming the point cloud into manageable geometric primitives. We employ region growing [44] to fit linear models from the projected points, and these linear models serve as the linear primitives for the floorplan. Two criteria are used, as defined in (2): normal vector similarity and point-to-point distance, to assign points to different groups, after which convex hull fitting is used to convert the groups into linear primitives. To eliminate noise and filter the fitted linear primitives, we impose the following rules.

1) The distance from a point to any point within a linear set must be less than the distance threshold Dr.
2) The normal vector difference between a point and any point within the linear set must not exceed the angle threshold θr.
3) The minimum number of points constituting any linear set must be greater than the threshold N, and the length of the fitted linear primitives must exceed the length threshold L.

The fitting process, as shown in Figs. 7 and 8, transforms complex point clouds into more concise linear geometric primitives, significantly reducing the amount of data.

Kinetic expansion: Fitting errors may lead to insufficient orthogonality and parallelism in the linear primitives, resulting in inaccurate polygonal areas in the reconstructed floorplans. This, in turn, affects the precision and overall accuracy of the floorplans. To address these issues, we adopt the global regularization technique and KDS-based expansion strategy proposed by Bauchet and Lafarge [22]. This approach corrects fitting errors and enhances the spatial layout integrity of the linear primitives.

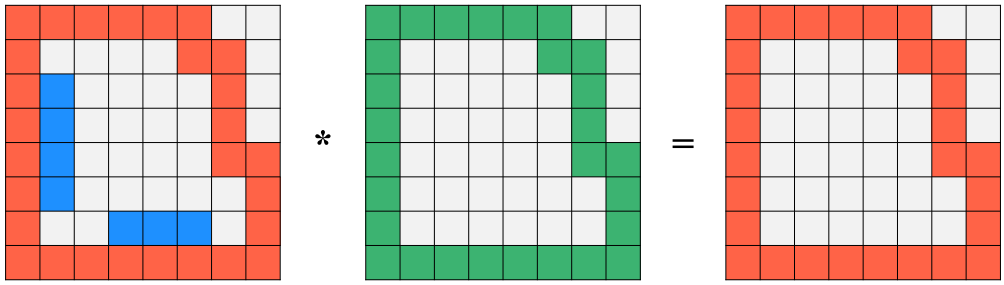


Fig. 5. Structure element extraction based on 2D distance constraints. (a) Primary structure elements containing nonstructural elements. (b) Boundary of secondary structure elements. (c) Primary structure elements corrected according to 2D distance constraints; $*$: 2D distance constraint.

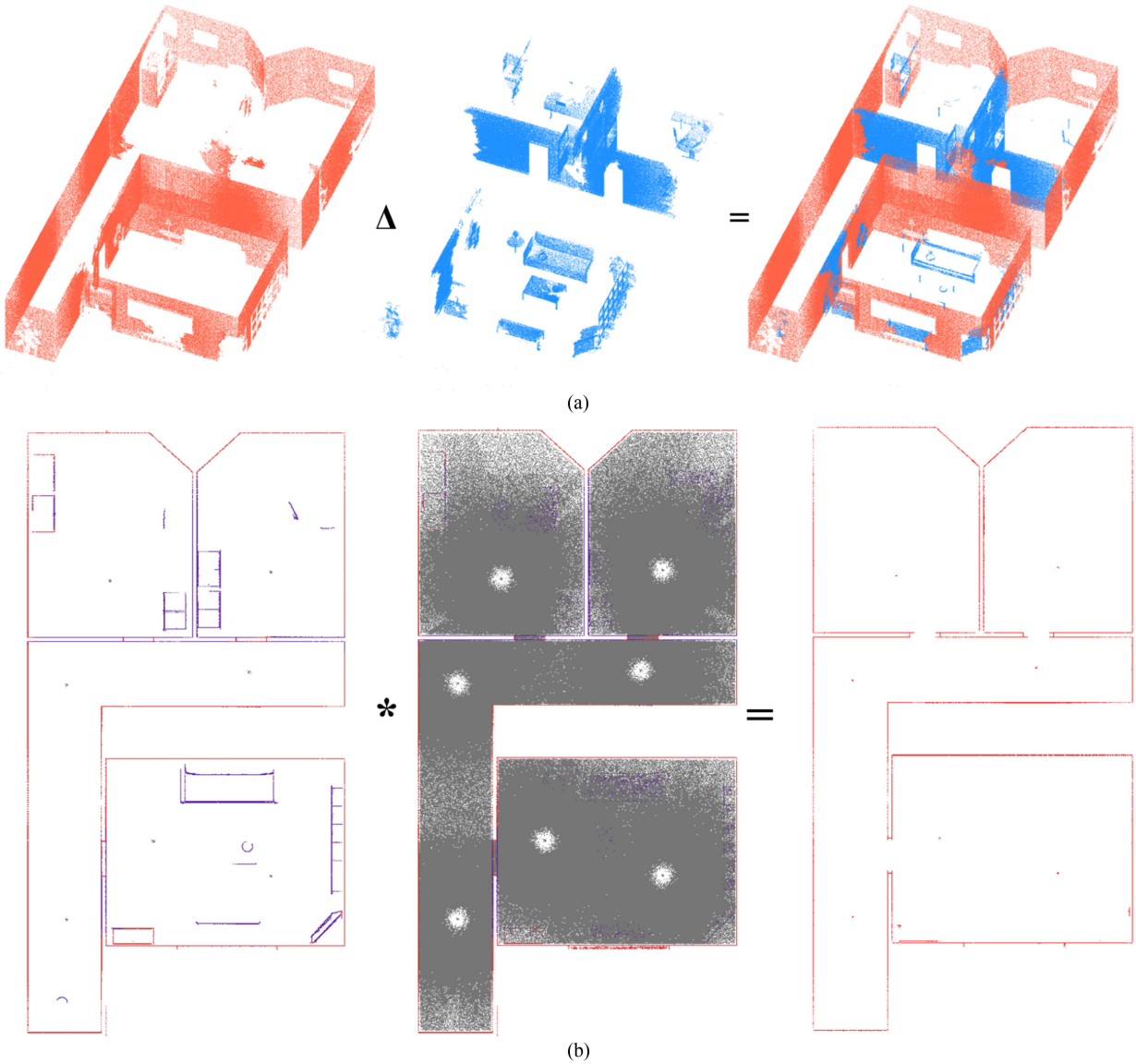


Fig. 6. Multiconstraint structural element extraction process. (a) Structural element extraction based on geometric feature similarity constraints, Δ: Geometric feature similarity constraint. (b) Structural element extraction based on 2D distance constraints.

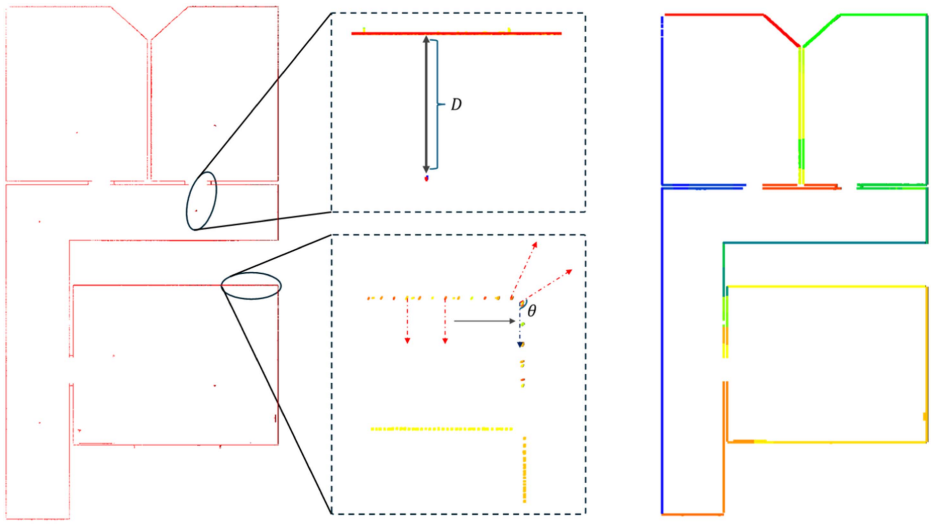


Fig. 7. Linear fitting of projected points using fitting rules through region growing. (a) Unpartitioned projected points. (b) Upper: Calculating the distance between points and points in the linear set, lower: Calculating the normal vector differences between.

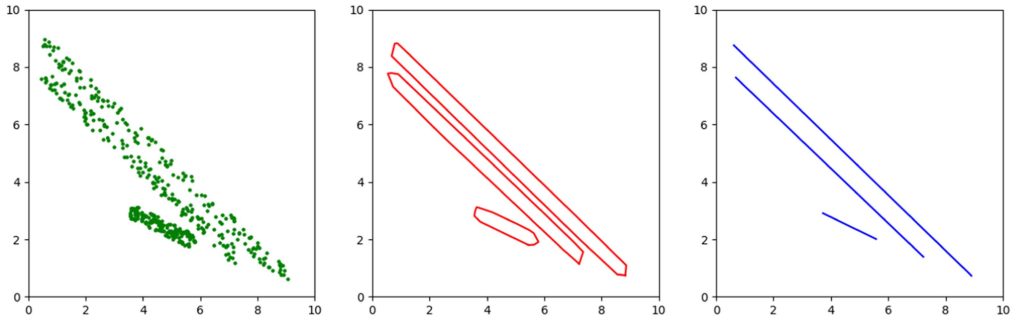


Fig. 8. Conversion of linear point sets into linear primitives through convex hull transformation.

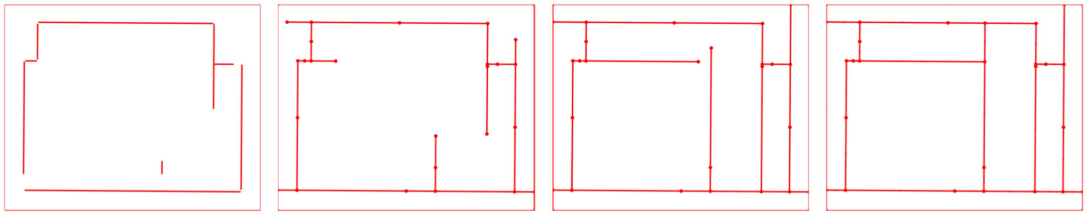


Fig. 9. Reconstruction process of the floorplan is based on kinetic expansion. (a) Initialization of kinetic expansion, (b) the 30th expansion, (c) the 120th expansion, and (d) finalization of kinetic expansion.

When reconstructing the floorplan from fitted linear primitives, it is crucial to establish their interconnections. Additionally, due to structural omissions caused by furniture obstructions, the fitted primitives often fail to fully capture the indoor scene's layout. As illustrated in Fig. 9, we expand the linear primitives using KDS, linking them to form closed polygons, which effectively compensates for structural omissions due to these obstructions.

Floorplan simplification: Due to the complexity of indoor scenes and potential data loss, the resulting floorplans may contain redundant polygons. Therefore, this step aims to eliminate redundant edges in the floorplans to ensure that the final polygons accurately reflect the indoor environment. We transform the generated polygons into an undirected graph $G = \langle V, E \rangle$, where the vertex set V represents the endpoints of the linear primitives, and the edge set E represents the linear primitives themselves. Subsequently, we analyze and simplify the edges and vertices in the graph, with the simplification process consisting of two parts: (i) removing erroneous edges from the graph, and (ii) merging the remaining edges.

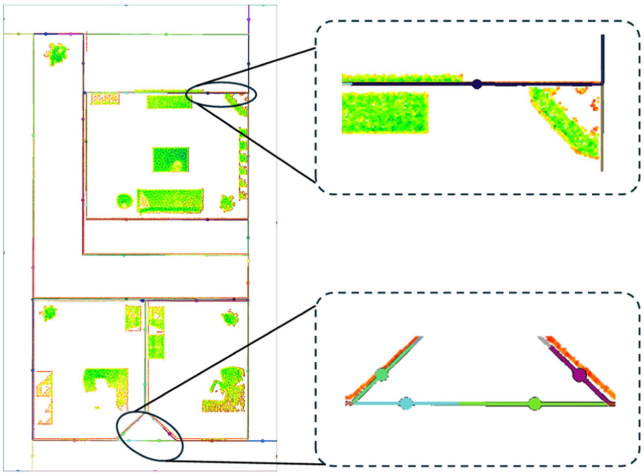


Fig. 10. Calculating the validity of edges $V(S)$, superimposing the undirected graph G with the point cloud to determine whether the edges in G are invalid or valid; Upper image: The edges in G highly coincide with the point cloud, indicating valid edges; Lower image: The edges in G cannot find enough points in the point cloud, indicating invalid edges.

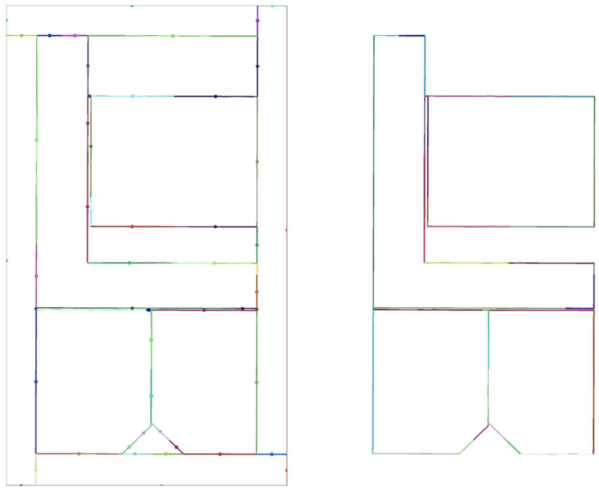


Fig. 11. Undirected graph G after removing invalid edges based on the energy function $\boldsymbol{E}(\boldsymbol{S})$. (a) Before invalid edge removal. (b) After invalid edge removal.

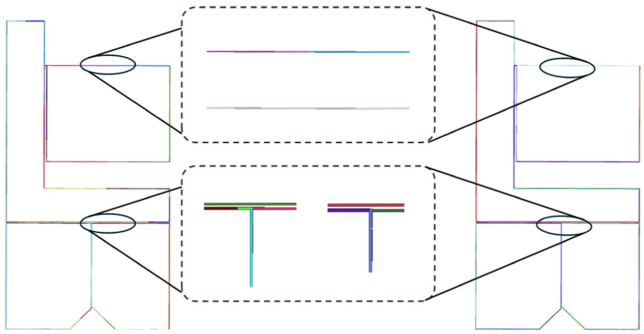


Fig. 12. Simplified undirected graph $\boldsymbol{G}$, with dashed lines illustrating our merging criteria: edges that are collinear and have an intersection degree of 2 are merged into a single edge; otherwise, they are not merged.

For the elimination step, we designed an energy function *E(S)* to determine whether the linear primitive *S* should be discarded. This function includes a validity term *V* and a degree term *D*, as shown in the following equation:

$$E\ (S) = V\,(S) + \lambda \cdot D\,(S)\,. \tag{3}$$

The validity term *V(S)* is defined as shown in (4) and Fig. 10, and is used to calculate the proportion of valid points for the linear primitive *S* within the point cloud. Each linear primitive *S* is divided into multiple segment points based on its length, with the number of points denoted by *n*. For each segment point P_s, the number of valid points is calculated using a predefined threshold T and an indicator function $\mathbb{1}$ to determine whether the segment point is valid. The higher the proportion of valid points, the lower the energy. The indicator function $\mathbb{1}$ is shown in (5).

$$V\ (S) = 1 - \frac{1}{n}\sum_{i=0}^{n} \mathbb{1}\,(\,P_s\,) \tag{4}$$

$$1\ (\,P_s\,) = \begin{cases} 1,\ if\ N\,(\,P_s\,) \geq T \\ 0,\ if\ N\,(\,P_s\,) < T \end{cases}. \tag{5}$$

In an undirected graph, the degree of an endpoint represents the number of connected edges. We observe that connection counts in actual valid structures typically cluster within a specific range. Based on this, the degree term $D(S)$ in (6) evaluates primitive validity by quantifying endpoint connection deviations at P_{tar} and P_{sou}

$$D\ (S) = \text{Max}\,(0, |\deg\,(P_{tar}) - 2| + |\deg\,(P_{sou}) - 2|) \tag{6}$$

Elevated $D(S)$ values indicate that the primitive's connectivity pattern substantially deviates from normative structural elements, suggesting potential reconstruction errors. To visualize the effect of this energy-based pruning, Fig. 11 shows the effect of invalid edge removal based on the energy function.

After the removal of erroneous edges, the remaining edges are merged to simplify the planar graph. Specifically, if two edges E_i and $E_j$intersect at exactly one point and the angle between them satisfies the condition in (7), the edges are collinear, and thus merged into a single edge, as shown in Fig. 12.

$$\left|\cos\theta_{\substack{\rightarrow \\ E_i}\ \substack{\rightarrow \\ E_j}}\right| = 1. \tag{7}$$

IV. Experiment and Analysis

A. Overview

This section introduces the datasets used in the KELSE algorithm experiments and the evaluation metrics for assessing the method's accuracy. The dataset comprises eight indoor scene point clouds, including six multiroom scenes and two single-room scenes. Among the six multiroom scenes, three are virtually generated from 3D models [8], and the other three are captured using terrestrial LiDAR. For the two single-room scenes, one is obtained with terrestrial LiDAR, while the other is scanned using a backpack LiDAR system. We conducted

qualitative and quantitative comparisons of KELSE with other state-of-the-art methods on the 3 synthetic multiroom scene datasets and performed quantitative assessments on all datasets. The KELSE method is implemented using C++ and Python, relying on the Computational Geometry Algorithms Library (CGAL) [45], Point Cloud Library (PCL) [46], open-source computer vision library (OpenCV) [47], and Open3D [48] to realize all methods. All experiments were conducted on a PC equipped with an Intel Core i9-10900K CPU (3.70 GHz) and an NVIDIA RTX 3090 GPU.

B. Dataset

The eight indoor scene datasets used to assess the effectiveness of KELSE are presented in Table III, which provides their visual representations, while Table IV offers detailed information on their size, scene complexity, point density, and degrees of occlusion and absence in primary structures.

C. Parameters Setting

The KELSE algorithm employs a defined set of parameters to govern its core processing stages, with the specific values utilized in our experiments detailed in Table V. These parameter values were selected to achieve an optimal balance between reconstruction accuracy and computational efficiency applicable across diverse indoor scenes. Core parameters associated with preprocessing and feature extraction—namely the downsampling voxel size Vs, the polynomial degree k for surface smoothing, and the feature extraction thresholds θg, Db, θr, Dr—were maintained at fixed values.

These values were selected based on empirical observations made during method development. Specifically, $Vs = 0.025$ was found to provide a good trade-off between point cloud simplification and the preservation of key geometric details, significantly reducing computation time while retaining structural elements necessary for downstream processing. Similarly, the polynomial order $k = 2$ offered sufficient surface smoothing without over-smoothing local geometry. Other thresholds such as θg, Db, θr and Dr were incrementally adjusted and validated through controlled experiments to achieve consistent performance across scenes. The choice of these fixed parameters reflects a compromise between runtime efficiency and reconstruction quality, which was iteratively optimized based on experimental results.

Parameters governing the scale and density of extracted structural elements—specifically the minimum length *L* and the minimum point count *N*—exhibit inherent dependencies on input data characteristics such as point density, room size, and level of detail. In our implementation, these parameters were manually adjusted based on qualitative observations of the point cloud data. Specifically, we examined the typical geometric extent and density distribution of linear structures in each dataset and selected *L* and *N* values that could effectively filter out small, noisy segments while preserving valid structural elements. The parameter settings were guided by empirical observations and demonstrated consistent performance across different datasets. The specific values used in our experiments are listed in Table VI.

D. Evaluation Metrics

To verify the accuracy of the floorplans generated by KELSE, we designed an evaluation method comprising six assessment metrics. These criteria comprehensively assess the effectiveness of KELSE from two aspects: geometric measurement and room reconstruction.

In terms of geometric measurement, we annotated key structural points in the dataset and employed two assessment metrics to evaluate the reconstruction performance of the model: the root mean square error (RMSE) and the difference in room area. The calculation formulas for these metrics are as follows:

$$RMSE_{2D} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(P_{re,i} - P_{ori,i}\right)^2} \tag{8}$$

$$RMSE_{3D} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d^2\left(P_{re,i}, M\right)} \tag{9}$$

$$Room\ Area = \frac{\sum_{i=1}^{n}|G_i - A_i|}{\sum_{i=1}^{n} A_i}. \tag{10}$$

Equations (8) and (9) provide a detailed description of the calculation method for the RMSE, which is used to assess the degree of fit between the 2D floorplans reconstructed by KELSE and the corresponding 3D point cloud. Since the objective of KELSE is to generate 2D floorplans from 3D point clouds of indoor scenes, we calculate the RMSE in both 2D and 3D to ensure a more comprehensive and accurate evaluation. The 2D RMSE is calculated as shown in (8) by determining the minimum distance between the generated point $P_{re,i}$ and the reference point $P_{ori,i}$ to obtain the RMSE value. This method of calculation directly reflects the degree of fit between the 2D floorplan and the actual point cloud on the plane. In (9), $d^2(P_{re,i}, M)$ represents the distance from the generated point $P_{re,i}$ to the local surface model M of the reference point cloud. Considering the potential for voids in the 3D point cloud due to occlusion by furniture or insufficient scanning, this method of calculation can more accurately assess the differences between the KELSE reconstruction results and the actual scene. (10) describes the method for calculating the difference in room area, where G_i represents the true area of the scene's room, and A_i represents the area of the room in the reconstructed floorplan; these are all effective metrics for evaluating the quality of KELSE's reconstruction results.

In room reconstruction, the floorplan visually captures the number of rooms and their spatial relationships. We assess the accuracy of our generated floorplans by comparing them with the actual room. To evaluate KELSE's performance, we utilize four key metrics: accuracy for correctly identified rooms, recall for detected existing rooms, F1 score as a balance of precision and recall, and mean Intersection over Union (mIoU) for the average spatial overlap between predicted and actual room areas. These metrics, detailed in (11)–(14), gauge the effectiveness of KELSE's room reconstruction capabilities.

$$precision = \frac{TP}{TP + FP} \tag{11}$$

$$ecall = \frac{TP}{TP + FN} \tag{12}$$

TABLE III
ILLUSTRATIONS OF THE 6 MULTIROOM INDOOR SCENE DATASETS FOR KELSEY'S EFFECTIVENESS EVALUATION

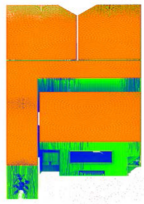
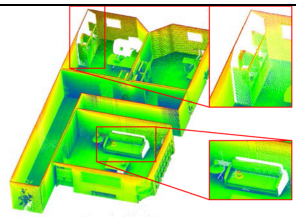
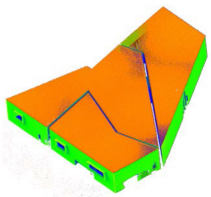
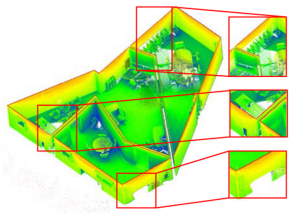
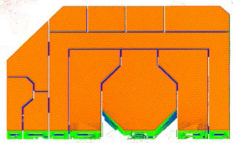
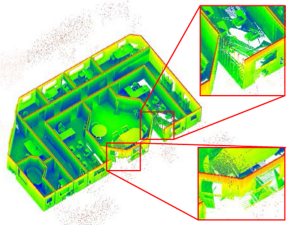
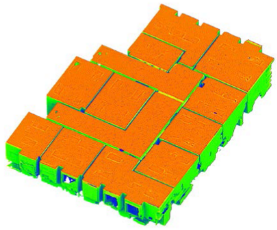
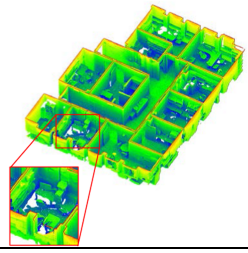
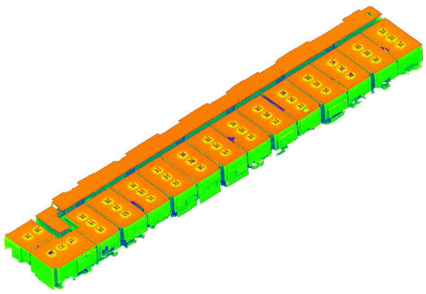
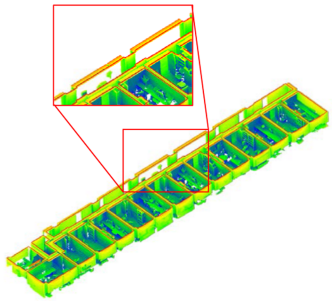
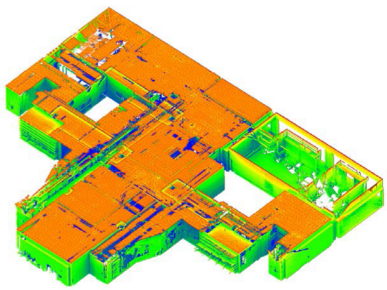
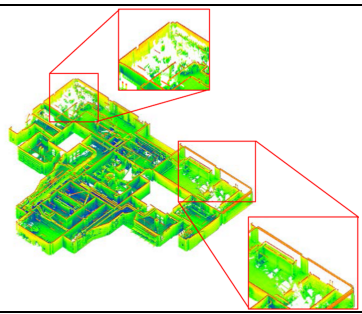
Name	Perspective view	Indoor view
Synth 1		
Synth 2		
Synth 3		
Office 3		
Office 4		
Factory		

TABLE IV
ILLUSTRATIONS OF THE 2 SINGLE ROOM INDOOR SCENE DATASETS FOR KELSEY'S EFFECTIVENESS EVALUATION

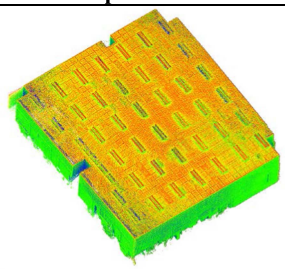
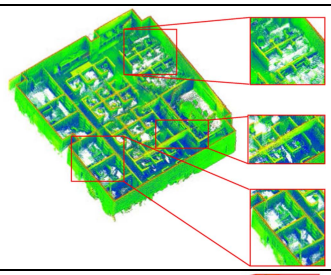
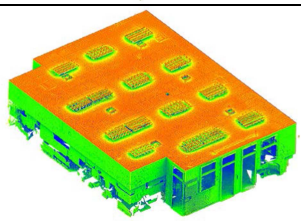
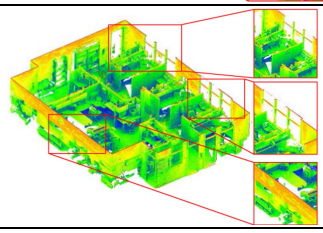
Name	Perspective view	Indoor view
Office 1		
Office 2		

TABLE V
DETAILED INFORMATION OF THE 8 INDOOR SCENE DATASETS USED TO EVALUATE THE EFFECTIVENESS OF KELSE

Name	Points	Area [m²]	Density [m]	Characteristics
Synth 1	19.34×10^6	80.26	0.007	Missing Level: Low Complexity Level: Low Noise Level: Low, mainly outside the scene
Synth 2	19.25×10^6	214.95	0.014	Missing Level: Low Complexity Level: Middle Noise Level: Low, mainly outside the scene
Synth 3	68.53×10^6	287.28	0.004	Missing Level: Middle Complexity Level: High Noise Level: Low, mainly outside the scene
Office 1	18.70×10^6	221.74	0.008	Missing Level: Middle Complexity Level: Middle Noise Level: High, present within the scene
Office 2	4.97×10^6	111.79	0.012	Missing Level: High Complexity Level: Middle Noise Level: Low
Office 3	11.0×10^6	199.4	0.001	Missing Level: Low Complexity Level: Middle Noise Level: Low
Office 4	15.1×10^6	310.3	0.001	Missing Level: Low Complexity Level: Middle Noise Level: Low
Factory	41.1×10^6	1006.9	0.001	Missing Level: High Complexity Level: Middle Noise Level: High

TABLE VI
PARAMETERS FOR KEY STEPS OF THE KELSE IN OUR EXPERIMENT

Workflow	Parameter	Value	Description
Data Pre-processing	Vs	0.05	Voxel size for down sampling
	k	2	Degree of the polynomial function ϕ_p used for surface smoothing
Structural element extraction	θg	10°	Maximum allowed angle between the normal vectors of primary and secondary structural element points
	Db	0.1	Maximum allowed distance between primary structural element points and secondary structural element boundary points
Line primitives fitting	θr	2.5°	Angle threshold for fitting linear primitives using region growing
	Dr	0.1	Distance threshold for fitting linear primitives using region growing
	L	*	Minimum length of linear primitives
	N	*	Minimum number of points included in linear primitives
Floorplan optimization	T	5	Minimum number of points required to consider a segment point of a linear primitive as valid
	λ	0.5	Weight balancing the validity term V and the degree term D

$$F1score = \frac{2 * Precision * Recall}{recision + Recall} \tag{13}$$

$$\text{mIoU} = \frac{1}{N} \sum \frac{TP}{TP + FP + FN}. \tag{14}$$

In these metrics, TP represents the overlapping area between the predicted rooms in the floorplan and the actual rooms within the intersection and union region, FP denotes the area of the predicted rooms that is not covered by the actual rooms, and FN indicates the area of the actual rooms that is not covered by the predictions.

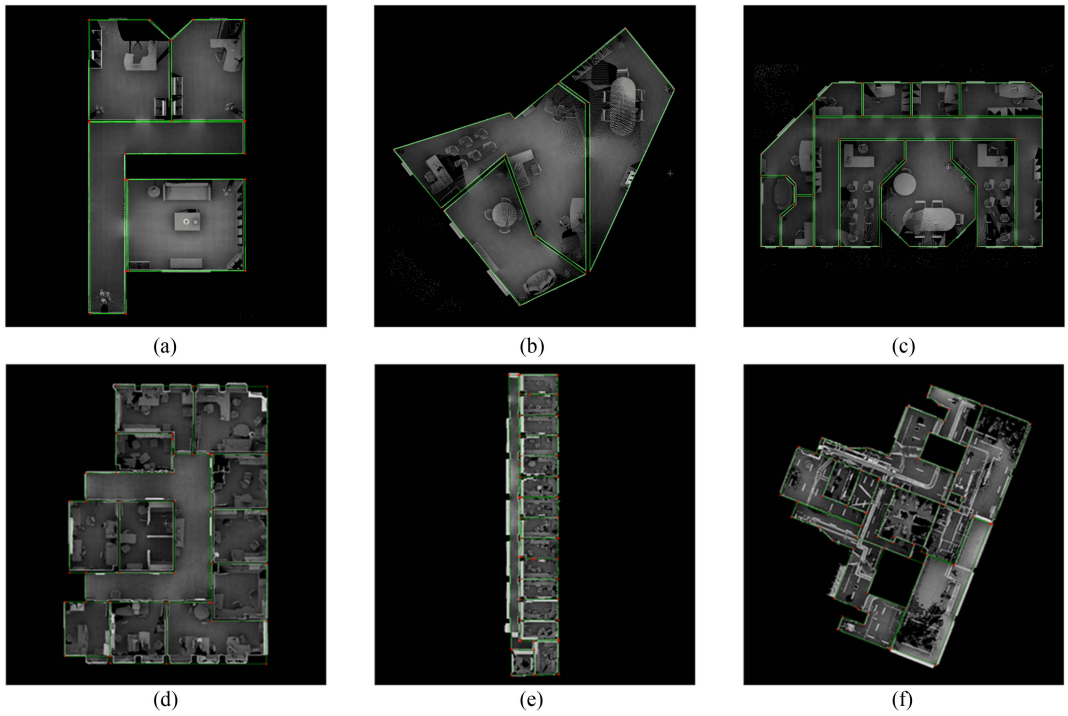


Fig. 13. Reconstructed floorplan for 6 multiroom scenes. (a) Synth 1. (b) Synth 2. (c) Synth 3. (d) Office 3; (e) Office 4; (f) Factory

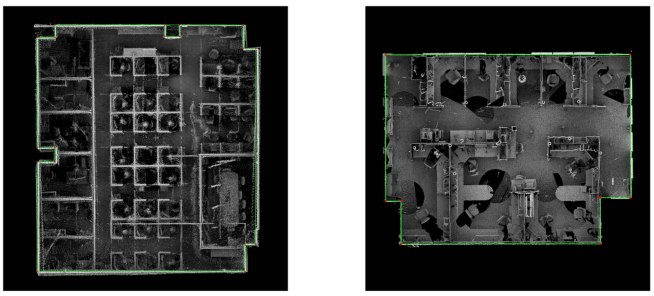


Fig. 14. Reconstructed floorplan for 2 single-room scenes. (a) Office 1. (b) Office 2.

TABLE VII
EVALUATION RESULTS OF THE RMSE IN 2D AND 3D

Test scenes	2D RMSE (m)	3D RMSE (m)
Synth 1	0.013	0.051
Synth 2	0.014	0.052
Synth 3	0.011	0.053
Office 1	0.003	0.036
Office 2	0.002	0.119
Office 3	0.040	0.121
Office 4	0.054	0.159
Factory	0.031	0.148

E. Result and Analysis

Geometric measurement: We evaluated the reconstruction performance of the KELSE method on eight datasets by calculating metrics such as the RMSE and differences in room area. Figs. 13 and 14 illustrate the reconstruction results, with detailed metrics provided in Tables VII and VIII. Visual inspection and statistical analysis indicate that the method performs well across a variety of indoor scenes.

Figs. 13 and 14 visually present the reconstruction outcomes of KELSE across eight distinct indoor scenarios, encompassing diverse spatial layouts and structural complexities, including synthetic datasets and real office environments, reflecting a range from relatively simple to complex indoor configurations. In the synthetic datasets, despite irregular room shapes, varying numbers of rooms, and some missing regions, KELSE accurately captures wall boundaries and reconstructs floorplans with high precision and stability. The real-world office scenarios include multiple offices and factory settings of varying sizes, with scene areas ranging from approximately 112 square meters to over 1000 square meters, featuring complex and diverse spatial arrangements. For example, the Office 3 scenario contains a U-shaped corridor, while the Office 4 scenario includes an L-shaped corridor approximately 41 m long. Furthermore, in the factory scenario covering roughly 1007 square meters, KELSE consistently achieves high-quality reconstruction results, demonstrating its versatility and robustness in large-scale and complex environments.

Table VII lists the RMSE values for the eight datasets in both 2D and 3D. The 2D RMSE values are generally low, with Office 1 showing an especially small 2D RMSE of only 0.003 m, indicating extremely high geometric reconstruction accuracy. Although 3D RMSE values are higher than those in 2D, errors

TABLE VIII
COMPARISON OF ROOM AREA DIFFERENCE BETWEEN KELSE RECONSTRUCTED RESULT AND GROUND TRUTH

Test scenes	Ground truth (m²)	Reconstruction (m²)	Difference (m²)	Precision
Synth 1	80.26	81.72	1.46	0.98
Synth 2	214.95	215.18	0.23	0.99
Synth 3	287.28	289.54	2.26	0.99
Office 1	221.74	224.01	2.27	0.98
Office 2	111.79	112.74	0.96	0.99
Office 3	199.43	206.57	7.14	0.96
Office 4	310.27	301.89	8.38	0.97
Factory	1006.91	1015.06	8.15	0.99

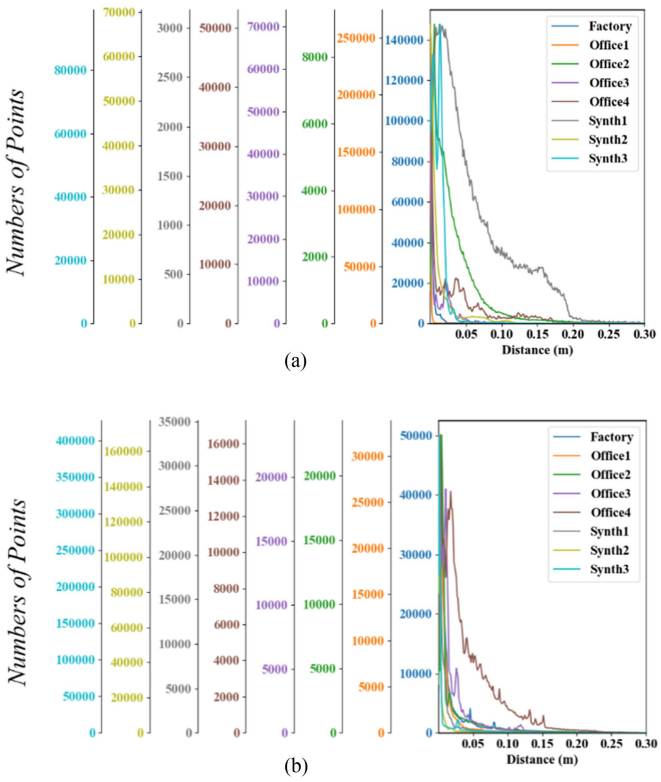


Fig. 15. Reconstruction result distance distribution histograms for five scenes. (a) 2D distance distribution histogram. (b) 3D distance distribution histogram.

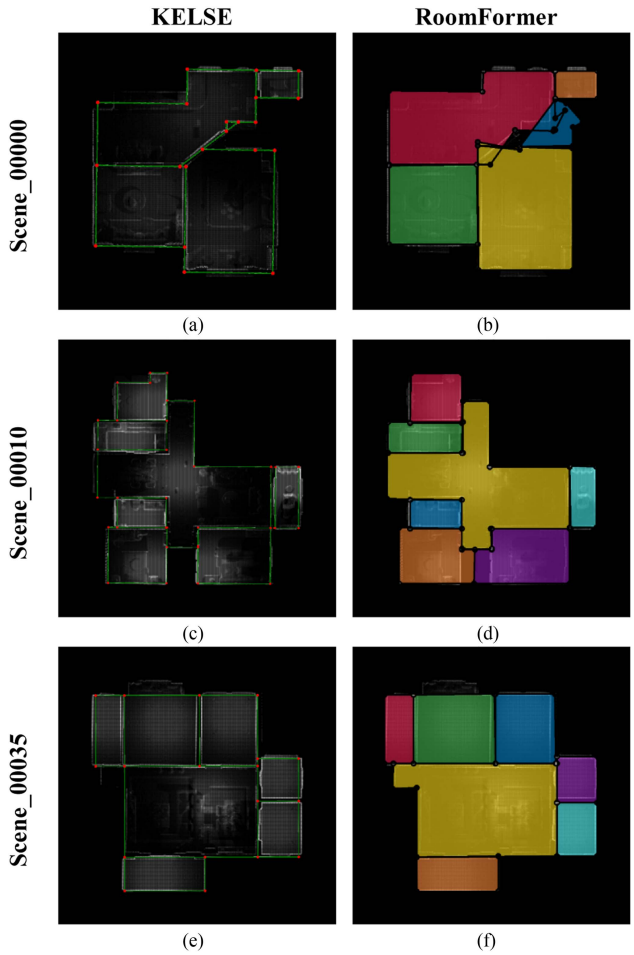


Fig. 16. Comparison of KELSE and RoomFormer on the Structured3D dataset. KELSE results (left column) are depicted using red vertices and green edges, while RoomFormer (right column) uses distinct color blocks to represent each room area.

across all scenes remain within reasonable bounds. The main sources of error include occlusions by furniture, scanning voids, and openings such as doors and windows. Combined with the point distance distribution histograms in Fig. 15, the reconstruction results demonstrate consistent accuracy across a variety of complex scenes, underscoring the reliability and robustness of the KELSE method.

Table VIII quantitatively presents the area reconstruction accuracy of KELSE across these eight scenarios. The results show that the area differences remain small and accuracy is high across all scenes. Among the synthetic datasets, Synth 2 performs exceptionally well, with an area difference of only 0.23 square meters and an accuracy of 0.99. In real scenarios, despite increased complexity, Office 2 maintains high accuracy with an area difference of 0.96 square meters and an accuracy of 0.99. Larger-scale scenes such as Synth 3, Office 3, Office 4, and the factory also exhibit controlled area discrepancies, reflecting

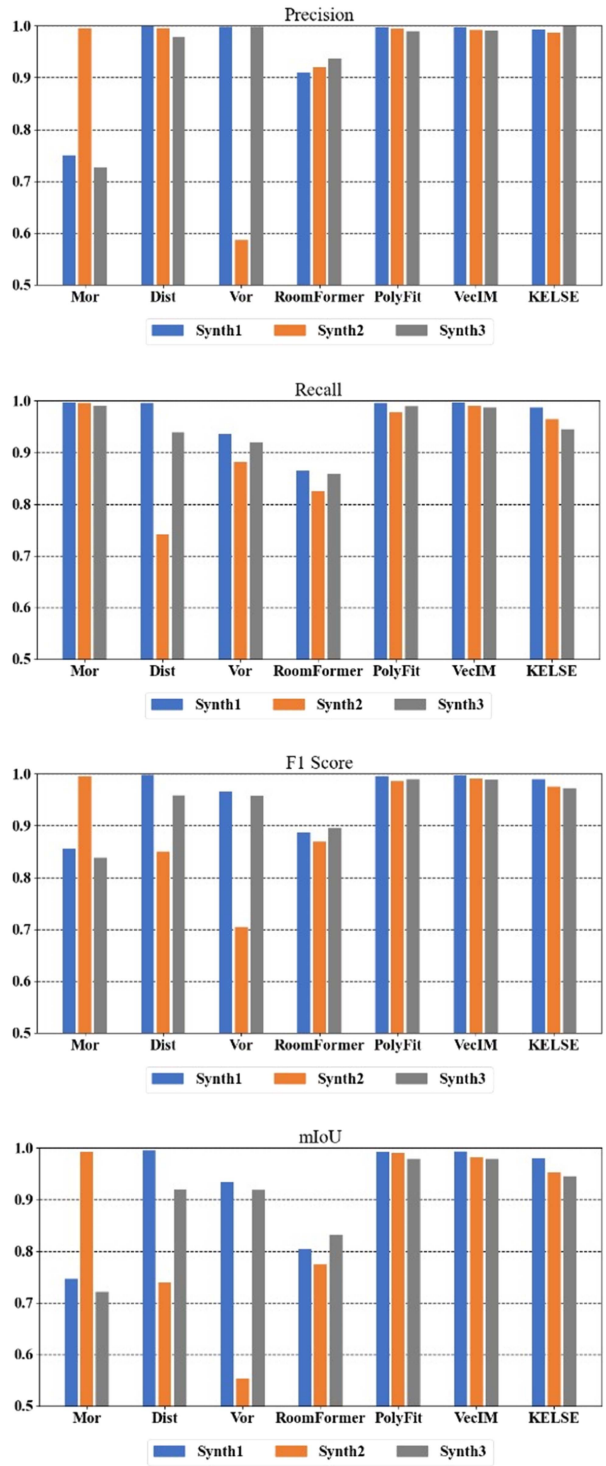


Fig. 17. Quantitative comparison bar charts on three multiroom scenes.

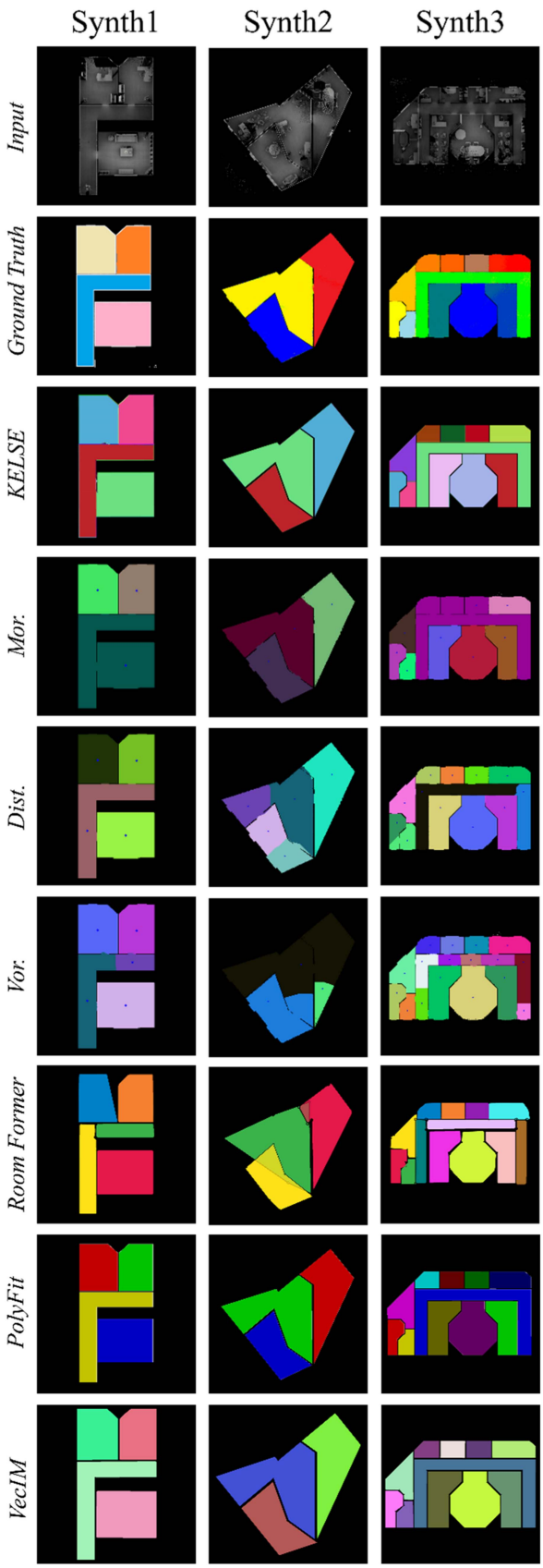


Fig. 18. Qualitative comparison of three multiroom indoor scenes. Compared to other methods, our algorithm is less affected by different room shapes, demonstrating advantages in reconstructing room boundaries and maintaining room topological information.

the method's stable performance in complex structures and large environments.

Room reconstruction: Multiroom scenes not only contain more room boundaries, doors, windows, and other structural details but also frequently encounter issues such as occlusions and overlaps. Moreover, in multiroom scenes, the algorithm must accurately identify the boundaries of different

TABLE IX
QUANTITATIVE COMPARISON ON 3 MULTIROOM INDOOR SCENES

Test method	Precision	Recall	F1 score	mIoU
Mor	0.82	0.99	0.90	0.82
Dist	0.99	0.89	0.94	0.88
Vor	0.86	0.91	0.89	0.80
RoomFormer	0.92	0.85	0.88	0.80
Polyfit	0.99	0.99	0.99	0.99
VecIM	0.99	0.99	0.99	0.98
KELSE	0.99	0.97	0.98	0.96

TABLE X
QUANTITATIVE COMPARISON ON 3 INDOOR SCENES OF STRUCTURED 3D DATASET

Test method	Test scene	Precision	Recall	F1 score	mIoU
RoomFormer	Scene_00000	0.93	0.90	0.92	0.85
	Scene_00010	0.92	0.93	0.92	0.85
	Scene_00035	**0.97**	0.96	**0.97**	**0.94**
	Mean Value	**0.94**	0.93	**0.94**	0.88
KELSE	Scene_00000	0.95	0.91	0.93	0.88
	Scene_00010	0.93	**0.98**	0.95	0.91
	Scene_00035	0.91	**0.98**	0.94	0.89
	MeanValue	0.93	**0.96**	0.94	**0.89**

rooms and their adjacent rooms, ensuring the correctness of the overall topological structure. To assess the performance of the KELSE algorithm, we qualitatively and quantitatively compared it with three popular floorplan partitioning methods. These methods include morphological segmentation (Mor) [49], distance transformation-based segmentation (Dist) [33], and Voronoi graph-based segmentation (Vor) [50]. Additionally, we compared KELSE with the current state-of-the-art floorplan reconstruction methods, including RoomFormer [19], PolyFit [51], and VecIM [5]. Considering that all the test datasets vary in scale, data volume, and shape layout, we selected Synth1, Synth2, and Synth3 as representative benchmarks for comparison. These three datasets are relatively moderate in both scene scale and data volume, while also featuring distinctive and representative shape layouts. The specific results can be seen in Table IX and Fig. 17, while Fig. 18 showcases the qualitative comparison results between KELSE and these methods.

The experimental results demonstrate that KELSE consistently performs well across all four reconstruction metrics, showing clear advantages in multiroom scenarios. As shown in Fig. 17, it maintains stable performance across Synth1, Synth2, and Synth3, accurately identifying and reconstructing rooms of diverse shapes. In comparison, Mor achieves high recall but low precision, Dist shows competitive precision with reduced recall, Vor performs moderately, and RoomFormer generalizes reasonably well but is limited by training data and low input resolution (256×256). Notably, these methods rely on point cloud density images, which inevitably lose spatial information. PolyFit and VecIM, in contrast, directly utilize raw point clouds to reconstruct floorplans via planar primitives, improving reconstruction accuracy. To ensure fair comparison, their input scenes were preprocessed by dividing multiroom scenarios into individual rooms, as planar-primitive-based reconstruction involves complex fitting and combinatorial optimization that is sensitive to input quality and prone to errors in sparse or complex regions.

On Structured3D, we compare KELSE with RoomFormer on three representative scenes using the same reconstruction metrics, with results summarized in Table X. KELSE achieves higher Recall and mIoU, while RoomFormer performs slightly better in Precision, and both obtain comparable F1 scores—showing that KELSE remains competitive. As illustrated in Fig. 16, RoomFormer tends to generate erroneous connections in oblique-wall regions due to limited training data and low input resolution, whereas KELSE struggles in Scene_00035 because a missing wall segment hinders reliable primitive extraction. KELSE employs a simpler linear primitive regularization strategy. Although its mIoU slightly decreases due to minor angular and distance deviations, it maintains strong performance in other metrics. Overall, considering all methods, KELSE demonstrates robust and stable performance in multiroom reconstruction, effectively handling complex point clouds and generating complete floorplans, while outperforming or matching the other approaches in most respects.

Limitations: Although KELSE achieves strong performance across multiple datasets, some limitations remain. In the Structured3D dataset, missing wall segments and boundary information in scenes like Scene_00000 and Scene_00035 lead to reconstruction failures and discontinuous room layouts. In S3DIS, fine-grained structures such as the narrow corridor in Office4 and the protruding pillar in Office3 are not well preserved, showing reduced adaptability to complex layouts. Additionally, KELSE primarily models linear structures, limiting its generalization to environments with nonlinear or curved boundaries.

The method's computational complexity also constrains scalability and real-time performance, with preprocessing, structural element extraction, and reconstruction as the main costs. Complexity roughly scales linearly with input points, but preprocessing can significantly increase runtime, especially for large or multiroom scenes.

These limitations highlight the need to improve robustness to missing data, enhance adaptability to complex geometries, extend representation to nonlinear boundaries, and streamline preprocessing—including simplifying large-scale complex scenes—to balance overall efficiency and reconstruction accuracy.

V. Conclusion

This article presents KELSE, a robust and effective indoor scene floorplan reconstruction method for point clouds without prior knowledge. KELSE introduces a structural element extraction approach that combines multiple features to effectively identify structural elements, even in complex scenes with missing regions. By incorporating KDS optimization, the extracted structural elements are converted into interconnected closed polygons, enhancing the geometric consistency of the floorplan. An undirected graph-based optimization strategy further refines the reconstruction, automatically removing noise and redundant structures to improve accuracy. Experiments in various indoor scenes demonstrate KELSE's robustness and performance on par with current state-of-the-art methods, especially in handling complex and incomplete data. The method also shows strong recovery capabilities in the presence of data deficiencies. Based on both visual inspection and quantitative analysis, KELSE proves highly effective. Future research will explore its extension to large-scale and multiroom scenarios, as well as its potential application in BIM generation to increase its practical value in engineering and real-world applications.

References

[1] M. F. Goodchild et al., "Digital twins in urban informatics," *Urban Inform.*, vol. 3, no. 1, pp. 3–16, May 2024, doi: 10.1007/s44212-024-00048-6.

[2] G. Pintore, C. Mura, F. Ganovelli, L. Fuentes-Perez, R. Pajarola, and E. Gobbetti, "State-of-the-art in automatic 3D reconstruction of structured indoor environments," *Comput. Graph. Forum*, vol. 39, no. 2, pp. 667–699, 2020, doi: 10.1111/cgf.14021.

[3] H. Fang, F. Lafarge, C. Pan, and H. Huang, "Floorplan generation from 3D point clouds: A space partitioning approach," *ISPRS J. Photogrammetry Remote Sens.*, vol. 175, pp. 44–55, 2021, doi: 10.1016/j.isprsjprs.2021.02.012.

[4] Q. Wang, Z. Zhu, R. Chen, W. Xia, and C. Yan, "Building floorplan reconstruction based on integer linear programming," *Remote Sens.*, vol. 14, no. 18, Jan. 2022, Art. no. 4675, doi: 10.3390/rs14184675.

[5] J. Han, M. Rong, H. Jiang, H. Liu, and S. Shen, "Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization," *ISPRS J. Photogrammetry Remote Sens.*, vol. 177, pp. 57–74, Jul. 2021, doi: 10.1016/j.isprsjprs.2021.04.019.

[6] S. Nikoohemat, M. Peter, S. O. Elberink, and G. Vosselman, "Exploiting indoor mobile laser scanner trajectories for semantic interpretation of point clouds," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. IV-2/W4, pp. 355–362, Sep. 2017, doi: 10.5194/isprs-annals-IV-2-W4-355-2017.

[7] R. Ambrus, S. Claici, and A. Wendt, "Automatic room segmentation from unstructured 3-D data of indoor environments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 749–756, Apr. 2017, doi: 10.1109/LRA.2017.2651939.

[8] C. Mura, O. Mattausch, A. Jaspe Villanueva, E. Gobbetti, and R. Pajarola, "Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts," *Comput. Graph.*, vol. 44, pp. 20–32, Nov. 2014, doi: 10.1016/j.cag.2014.07.005.

[9] L. Fan and Y. Cai, "An efficient filtering approach for removing outdoor point cloud data of Manhattan-world buildings," *Remote Sens.*, vol. 13, no. 19, 2021, Art. no. 3796, doi: 10.3390/rs13193796.

[10] Z. Hua, L. Qi, D. Du, W. Jiang, and Y. Sun, "Dual attention based multiscale feature fusion network for indoor RGBD semantic segmentation," in *Proc. 26th Int. Conf. Pattern Recognit.*, Aug. 2022, pp. 3639–3644, doi: 10.1109/ICPR56361.2022.9956246.

[11] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007, doi: 10.1111/j.1467-8659.2007.01016.x.

[12] W. Shi, W. Ahmed, N. Li, W. Fan, H. Xiang, and M. Wang, "Semantic geometric modelling of unstructured indoor point cloud," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 1, 2019, Art. no. 2352, doi: 10.3390/ijgi8010009.

[13] J. Ntiyakunze and T. Inoue, "Segmentation of structural elements from 3D point cloud using spatial dependencies for sustainability studies," *Sensors*, vol. 23, no. 4, Jan. 2023, Art. no. 1924, doi: 10.3390/s23041924.

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5100–5109.

[15] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point Transformer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16239–16248, doi: 10.1109/ICCV48922.2021.01595.

[16] X. Wu et al., "Point transformer V3: Simpler, faster, stronger," in *Proc. 2024 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2024, pp. 4840–4851, doi: 10.1109/CVPR52733.2024.00463.

[17] J. Chen, C. Liu, J. Wu, and Y. Furukawa, "Floor-SP: Inverse CAD for floorplans by sequential room-wise shortest path," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 2661–2670, doi: 10.1109/ICCV.2019.00275.

[18] C. Liu, J. Wu, and Y. Furukawa, "FloorNet: A unified framework for floorplan reconstruction from 3D scans," 2018, pp. 201–217, Accessed: Sep. 01, 2024. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Chen_Liu_FloorNet_A_Unified_ECCV_2018_paper.html

[19] Y. Yue, T. Kontogianni, K. Schindler, and F. Engelmann, "Connecting the dots: Floorplan reconstruction using two-level queries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 845–854, doi: 10.1109/CVPR52729.2023.00088.

[20] M. Li, F. Lafarge, and R. Marlet, "Approximating shapes in images with low-complexity polygons," 2020, pp. 8633–8641, Accessed: Sep. 01, 2024. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Li_Approximating_shapes_in_images_with_low-complexity_polygons_CVPR_2020_paper.html

[21] S. Ochmann, R. Vock, R. Wessel, M. Tamke, and R. Klein, "Automatic generation of structural building descriptions from 3D point cloud scans," in *Proc. Int. Conf. Comput. Graph. Theory Appl.*, Jan. 2014, pp. 1–8, Accessed: Sep. 01, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/7296039

[22] J. P. Bauchet and F. Lafarge, "KIPPI: KInetic polygonal partitioning of images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3146–3154, doi: 10.1109/CVPR.2018.00332.

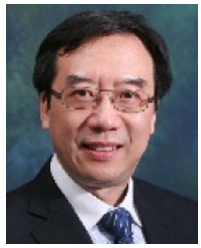
[23] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra, "GlobFit: Consistently fitting primitives by discovering global relations," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 52:1–52:12, Jul. 2011, doi: 10.1145/2010324.1964947.

[24] P. Jenke, B. Krückeberg, and W. Straßer, "Surface reconstruction from fitted shape primitives," in *Proc. 13th Int. Fall Workshop Vis., Model, Vis.*, Jan. 2008, pp. 31–40.

[25] T. Czerniawski, M. Nahangi, C. Haas, and S. Walbridge, "Pipe spool recognition in cluttered point clouds using a curvature-based shape descriptor," *Automat. Construction*, vol. 71, pp. 346–358, Nov. 2016, doi: 10.1016/j.autcon.2016.08.011.
[26] M. Previtali, L. Díaz-Vilariño, and M. Scaioni, "Indoor building reconstruction from occluded point clouds using graph-cut and ray-tracing," *Appl. Sci.*, vol. 8, no. 9, Sep. 2018, Art. no. 1529, doi: 10.3390/app8091529.
[27] V. Sanchez and A. Zakhor, "Planar 3D modeling of building interiors from point cloud data," in *Proc. Int. Conf. Image Process.*, 2012, pp. 1777–1780, doi: 10.1109/ICIP.2012.6467225.
[28] I. Anagnostopoulos, V. Pătrăucean, I. Brilakis, and P. Vela, "Detection of walls, floors, and ceilings in point cloud data," in *Proc. Old New Construction Technol. Converge Historic San Juan*, May 2016, pp. 2302–2311, doi: 10.1061/9780784479827.229.
[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85, doi: 10.1109/CVPR.2017.16.
[30] Q. Hu et al., "Learning semantic segmentation of large-scale point clouds with random sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8338–8354, Nov. 2022, doi: 10.1109/TPAMI.2021.3083288.
[31] X. Wu et al., "Towards large-scale 3D representation learning with multi-dataset point Prompt training," in *Proc. 2024 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2024, pp. 19551–19562, doi: 10.1109/CVPR52733.2024.01849.
[32] A. Xiao, J. Huang, D. Guan, F. Zhan, and S. Lu, "Transfer learning from synthetic to real LiDAR point cloud for semantic segmentation," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, pp. 2795–2803, Jun. 2022, doi: 10.1609/aaai.v36i3.20183.
[33] R. Bormann, F. Jordan, W. Li, J. Hampp, and M. Hägele, "Room segmentation: Survey, implementation, and analysis," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2016, pp. 1019–1026, doi: 10.1109/ICRA.2016.7487234.
[34] W. Wu, L. Fan, L. Liu, and P. Wonka, "MIQP-based layout design for building interiors," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 511–521, May 2018, doi: 10.1111/cgf.13380.
[35] R. Hu, Z. Huang, Y. Tang, O. Van Kaick, H. Zhang, and H. Huang, "Graph2Plan," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 118:118:1–118:118:14, Aug. 2020, doi: 10.1145/3386569.3392391.
[36] N. Nauata, S. Hosseini, K.-H. Chang, H. Chu, C.-Y. Cheng, and Y. Furukawa, "House-GAN++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13627–13636, doi: 10.1109/CVPR46437.2021.01342.
[37] N. Nauata, K.-H. Chang, C.-Y. Cheng, G. Mori, and Y. Furukawa, "House-GAN: Relational generative adversarial networks for graph-constrained house layout generation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 162–177, doi: 10.1007/978-3-030-58452-8_10.
[38] J. Han, Y. Liu, M. Rong, X. Zheng, and S. Shen, "FloorUSG: Indoor floorplan reconstruction by unifying 2D semantics and 3D geometry," *ISPRS J. Photogrammetry Remote Sens.*, vol. 196, pp. 490–501, Feb. 2023, doi: 10.1016/j.isprsjprs.2023.01.020.
[39] A. Avetisyan et al., "SceneScript: Reconstructing scenes with an autoregressive structured language model," in *Proc. Eur. Conf. Comput. Vis.*, 2025, pp. 247–263, doi: 10.1007/978-3-031-73030-6_14.
[40] H. Xu, J. Xu, Z. Huang, P. Xu, H. Huang, and R. Hu, "FRI-Net: Floorplan reconstruction via room-wise implicit representation," in *Proc. Eur. Conf. Comput. Vis.*, 2025, pp. 1–17, doi: 10.1007/978-3-031-73411-3_1.
[41] Y. Liu et al., "PolyRoom: Room-aware transformer for floorplan reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, 2025, pp. 322–339, doi: 10.1007/978-3-031-72973-7_19.
[42] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 1, pp. 3–15, Jan. 2003, doi: 10.1109/TVCG.2003.1175093.
[43] I. Armeni et al., "3D Semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1534–1543, doi: 10.1109/CVPR.2016.170.
[44] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 69–85, Aug. 2012, doi: 10.1007/s11263-012-0517-8.
[45] The CGAL Project. CGAL User and Reference Manual. CGAL Editorial Board, 5.6.3 edition, 2025. [Online]. Available: https://doc.cgal.org/5.6.3/Manual/how_to_cite_cgal.html
[46] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 1–4.
[47] G. Bradski, "The openCV library," *Dr. Dobb's J. Softw. Tools*, 2000.
[48] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.
[49] F. Meyer and S. Beucher, "Morphological segmentation," *J. Vis. Commun. Image Represent*, vol. 1, no. 1, pp. 21–46, Sep. 1990, doi: 10.1016/1047-3203(90)90014-M.
[50] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.*, vol. 99, no. 1, pp. 21–71, Feb. 1998, doi: 10.1016/S0004-3702(97)00078-7.
[51] L. Nan and P. Wonka, "PolyFit: Polygonal surface reconstruction from point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2372–2380, doi: 10.1109/ICCV.2017.258.

Yunlin Tu received the B.S. degree in surveying engineering in 2021 from the China University of Mining and Technology, Xuzhou, China, where he is currently working toward the Ph.D. degree in cartography and geographical information engineering with the School of Environment and Spatial Informatics.

His research interests include laser point cloud processing, artificial intelligence, 3-D object recognition, and 3-D modeling based on LiDAR and images.



Wenzhong Shi received the Ph.D. degree in geomatics from the University of Osnabrück, Vechta, Germany, in 1994.

He is currently the Director of PolyU-Shenzhen Technology and Innovation Research Institute (Futian), the Director of Otto Poon Charitable Foundation Smart Cities Research Institute of PolyU, the Chair Professor in GISci and Remote Sensing, and the Director of Joint Research Laboratory on Spatial Information of PolyU and Wuhan University. He is an Academician of the International Eurasian Academy of Sciences and a Fellow of the Academy of Social Sciences (U.K.). He has authored more than 300 research articles in journals indexed by SCI and 20 books. His research interests include urban informatics for smart cities, GISci and remote sensing, specifically, AI-based object recognition and change detection from satellite imagery, intelligent analytics and quality control for spatial data, mobile mapping and 3-D modeling based on LiDAR and imagery, and 3-D GIS models.

Yangjie Sun received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2021.

He is currently a Research Assistant Professor with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong. His research interests include remote sensing image segmentation, artificial intelligence, 3-D object recognition, and 3-D modeling based on LiDAR and images.

Min Zhang received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2020.

From 2021 to 2025, he served as a Research Assistant Professor with The Hong Kong Polytechnic University. He is currently a Distinguished Associate Professor with the School of Geosciences and Info-Physics, Central South University, Changsha, China. His research interests include spatial data quality, artificial intelligence, change detection, and object recognition in remote sensing.