Architectural Co-LOD Generation

RUNZE ZHANG, Shenzhen University, China
SHANSHAN PAN, Shenzhen University, China
CHENLEI LV, Shenzhen University, China
MINGLUN GONG, University of Guelph, Canada
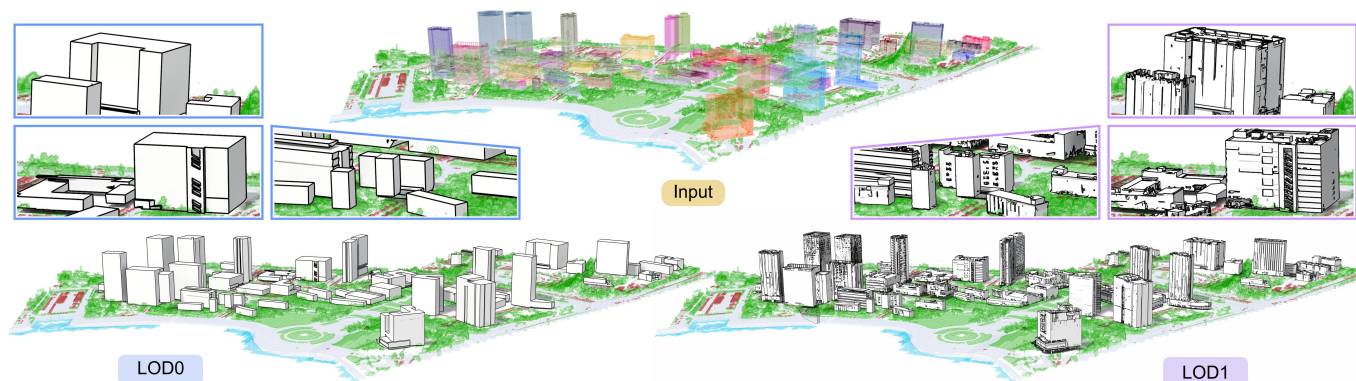HUI HUANG*, Shenzhen University, China

Fig. 1. Co-LOD generation for architectural models from raw real-world point clouds.

Managing the level-of-detail (LOD) in architectural models is crucial yet challenging, particularly for effective representation and visualization of buildings. Traditional approaches often fail to deliver controllable detail alongside semantic consistency, especially when dealing with noisy and inconsistent inputs. We address these limitations with *Co-LOD*, a new approach specifically designed for effective LOD management in architectural modeling. Co-LOD employs shape co-analysis to standardize geometric structures across multiple buildings, facilitating the progressive and consistent generation of LODs. This method allows for precise detailing in both individual models and model collections, ensuring semantic integrity. Extensive experiments demonstrate that Co-LOD effectively applies accurate LOD across a variety of architectural inputs, consistently delivering superior detail and quality in LOD representations.

CCS Concepts: • **Computing methodologies → Shape inference**.

Additional Key Words and Phrases: 3D generation, structural reconstruction, level-of-detail, co-analysis, architectural models

*Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Authors' addresses: Runze Zhang, oliverzrz.cyber@gmail.com, Shenzhen University, China; Shanshan Pan, psshappystar@gmail.com, Shenzhen University, China; Chenlei Lv, lvchenlei@gmail.com, Shenzhen University, China; Minglun Gong, minglun@uoguelph.ca, University of Guelph, Canada; Hui Huang, hhzhiyan@gmail.com, College of Computer Science & Software Engineering, Shenzhen University, China.

1 INTRODUCTION

In the realm of architectural model reconstruction, the creation of structural details at various levels is pivotal for a range of applications such as urban planning, autonomous driving, virtual reality, and digital entertainment. As the pioneering work, CityGML [Gröger and Plümer 2012] outlines that a 3D digital building can be depicted through multiple levels-of-detail (LODs), with each level providing progressively more detailed structures. This forms the basis of LOD generation with two significant challenges. Firstly, the intricate and diverse geometric structures of architectural models make it difficult to robustly control the generation of LODs. Secondly, maintaining semantic consistency within the same LOD level across different building models is a demanding task.

Existing approaches focusing on the LOD generation of building models have been attempting to address the aforementioned two challenges [Bauchet and Lafarge 2020; Gao et al. 2022; Verdie et al. 2015]. To ensure semantic consistency across different models, they use predefined rules to control LOD generation. However, these rules often falter when dealing with complex geometric structures and non-standard inputs. To overcome the limitations of external rules, some methods utilize visual error control and geometric constraints to guide the LOD generation. The aim of these methods is to achieve a more universally applicable and robust reconstruction process that also maintains semantic consistency. Despite these efforts, consistently producing uniform LODs from diverse and intricate inputs remains challenging. The success of maintaining semantic

consistency heavily relies on the quality of the input architectural models. At present, there is a lack of an effective mechanism for controlling LOD generation that can simultaneously meet the demands for robustness and semantic consistency.

In response to these challenges, we introduce Co-LOD, a novel approach that utilizes a co-analysis strategy. Co-LOD operates under the assumption that architectural models contain semantically meaningful details at different hierarchical levels, which are discernible through patterns of repetitiveness and correlation. By conducting a co-analysis of these repetitive and autocorrelated geometric structures across different input models, we can more precisely control the different LOD layers. Co-LOD harnesses the structural association relationships inherent in architectural models to develop correlation rules for LODs, thereby achieving semantic consistency.

The process encompasses two primary stages: initially, raw inputs (*e.g.* point clouds or triangle meshes) are segmented into structural segments to identify the main structure and categorize geometric details at varying levels. Subsequently, we formulate an energy function for these segments, which guides the LOD generation process while ensuring semantic consistency. When applied to a collection of building models, Co-LOD enhances the robustness against individual sample variations, offering an effective way to manage LOD generation. It strikes a balanced chord between robustness and semantic consistency, inheriting the advantages of structured reconstruction with concise and clear mesh results. Co-LOD is pioneering in its use of co-analysis to implement precise LOD controlling, an example of which can be seen in Fig. 1. The key contributions of this approach are summarized as follows:

- **Innovative Co-Analysis Approach.** Co-LOD represents a novel approach in integrating co-analysis for Level-of-Detail (LOD) generation. This initiative is the first to employ a building set-based cross-analysis specifically tailored for LOD generation tasks. Our method opens new research avenues, especially in achieving semantic consistency in the representation and optimization of architectural models.
- **Structural Segments Generation.** Our methodology features a meticulously designed segmentation process, capable of handling a wide array of architectural inputs and producing uniform structural segments. This approach excels in handling complex geometric structures and minimizes dependency on specific parameters, thus significantly enhancing the robustness and adaptability for processing raw inputs.
- **Joint Structural Analysis.** To augment our co-analysis, we have developed a comprehensive segment-based quantitative analysis method, tailored for architectural models. It establishes rules based on the hierarchy and similarity of structural details, which supports precise LOD control across LOD layers. This method effectively balances maintaining semantic consistency with ensuring robustness in LOD generation.

2 RELATED WORK

Low-Poly Reconstruction. To achieve structural model from architectural point clouds, some researchers develop low-poly reconstruction methods that are used to generate initial LOD information. Mehra et al. [2009] utilized characteristic curves and contours to establish building blocks for low-poly representation. Yumer and Kara [2012] implemented consistent abstraction of man-made models based on subvolumes. Lafarge and Alliez [2013] proposed a structure-preserving approach to detect planar components and construct polygonal mesh. Li et al. [2016] presented a fully automatic approach for reconstructing urban scene from point samples. It implemented plane hypothesis and generated candidate boxes to approximate geometry structures of buildings. With the similar technical route, they improved the framework [Nan and Wonka 2017] for reconstructing lightweight polygonal surfaces from point clouds. Kelly et al. [2017] proposed a high-quality structural modeling method for city blocks, which formulates a binary integer program to produce semantic models with associated surface elements. Bauchet and Lafarge [2020] designed an efficient shape assembling mechanism to reconstruct watertight polygonal meshes from point clouds. It is capable of accurately representing piecewise planar structures while also approximating freeform objects from incomplete geometric structures. Fang et al. [2018] proposed a structural method to detect planar shapes from 3D data. Then, they designed an hybrid approach [Fang and Lafarge 2020] to successively connects and slices planes for polygonal mesh reconstruction. Such solutions focus on individual model reconstruction without semantic consistency and precise LOD control.

LOD Generation. A series of detailed re-factoring methods are proposed to generate LODs, which can be concluded into two groups: geometric-based and human observation-based. Methods of first group implement LOD generation based on geometric feature analysis. Some of them [Chen et al. 2023; Gao et al. 2022; Garland and Heckbert 1997; Hoppe 1996; Li and Nan 2021; Lindstrom 2000; Lindstrom and Turk 2000] utilized edge collapse strategy to control LOD with geometric error optimization. Inspired by marching cubes [Lorensen and Cline 1987], some solutions [Takikawa et al. 2022, 2021] generated LOD according to implicit surface estimation. Such methods can handle most 3D shape for LOD generation. However, they lack semantic perception abilities that may produce structural defects or inconsistencies in architectural models. Another group utilize human observations to improve ability of semantic-aware in LOD generation. Verdie et al. [2015] learned geometric attributes and semantic rules to extract facades, roofs, and superstructures. Based on CityGML's [Gröger and Plümer 2012] requirements, such elements are allocated to suitable levels for LOD generation. Huang et al. [2023] classified architectural point clouds into floor, wall, ceiling, and window components with similar purpose. Such methods rely on manually designed rules that take some instability factors for individual building models.

Structural Segmentation. Another important technical route is to utilize segmented components to guide LOD reconstruction. Initial methods for architectural segmentation utilized shallow pipelines incorporating hand-crafted point descriptors and rules [Lin et al. 2013; Martinović et al. 2015; Toshev et al. 2010]. To improve the accuracy for repetitive structural segmentation, Demir et al. [2015b] formulated a weighted minimum set to implement segment optimization. Kobyshev et al. [2016] leveraged symmetry for architectural structural segmentation. Demir et al. [2015a] proposed a user-assisted segmentation method to further reduce ambiguous structures. Recently, Selvaraju et al. [2021] introduced a BuildingNet
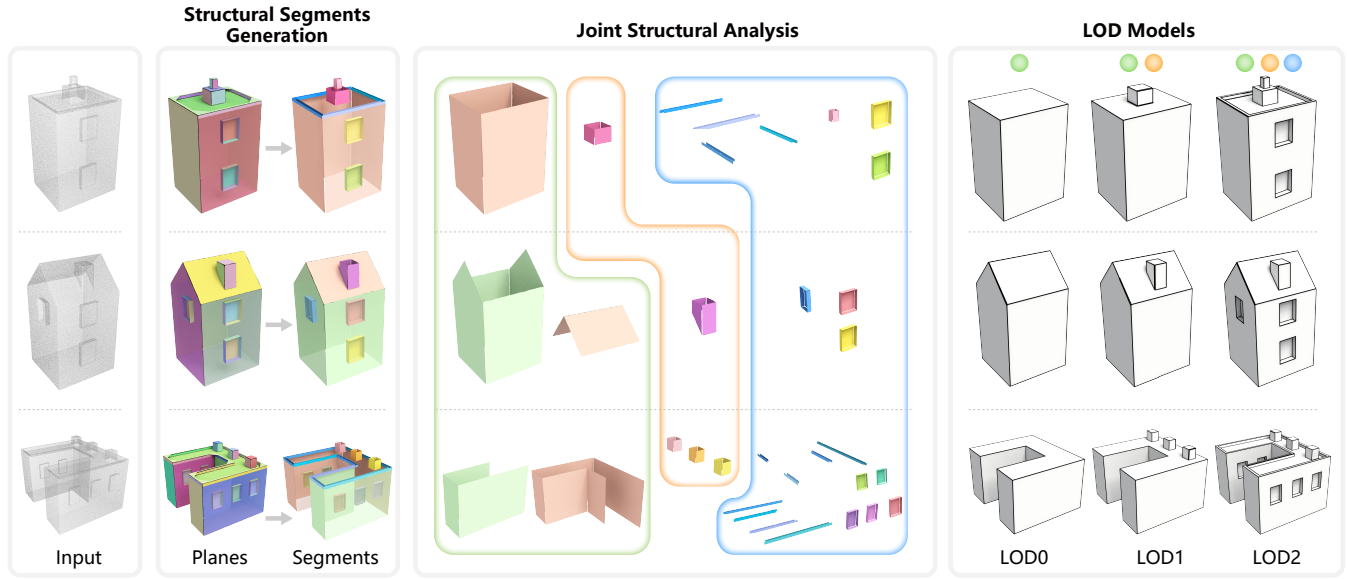
Fig. 2. Overview of Co-LOD workflow, which contains two main modules. The first module involves detecting primary planes and aggregating them to form structural segments from raw inputs. The second module quantitatively measures the similarity of these segments and clusters them to define LOD layers, a process akin to joint structural analysis. Together, these modules enable Co-LOD to effectively perform co-analysis for controlled LOD generation with semantic consistency.

dataset that is used to support learning-based schemes. It is useful to define accurate segmentation or LOD component. However, such methods don't consider the relationships between different segmentations which limit the performance for accurate LOD controlling and semantic analysis. In non-architectural segmentation domains, some approaches [Golovinskiy and Funkhouser 2009; Hu et al. 2012; Huang et al. 2011; Sidi et al. 2011; Wang et al. 2012; Xu et al. 2010] employed co-analysis to improve the accuracy of segmenting components. They initially employed mesh segmentation technique [Golovinskiy and Funkhouser 2008; Katz and Tal 2003; Shapira et al. 2008] to over-segment the input model, followed by feature extraction and clustering to achieve a consistent structural segmentation. Our approach draws inspiration from this scheme and proposes the prior work to implement co-analysis for architectural LOD generation.

3 METHODOLOGY

Overview. Co-LOD is designed to facilitate LOD generation for architectural models with controllable structural accuracy and semantic consistency. Specifically, Co-LOD takes a group of architectural models as input and generates LOD representations for these models that adhere to the following criteria:

i) Each input architectural model corresponds to an entity within each layer;

ii) Architectural models at the same layer exhibit LOD-based semantic consistency;

iii) Structures become progressively enriched with the increase in LOD;

iv) LODs are represented by watertight polygonal meshes.

Essentially, Co-LOD achieves LOD generation that meets the specified criteria through three core steps, as illustrated in Fig. 2. The first stage involves structural segmentation to identify basic components. This is followed by a joint structural analysis that conducts a co-analysis on these segments, organizing them into LOD layers with semantic consistency. Finally, LODs represented by polygonal meshes are extracted from these layered segments. The following sections will delve into the implementation details.

3.1 Structural Segments Generation

The importance of structural segmentation in LOD generation is well-recognized. Existing methods, however, often rely on low-level primitives for LOD generation, limiting their capacity to generate models at different levels-of-detail. To overcome this limitation, we propose a novel segmentation scheme aimed at grouping interrelated planes to obtain meaningful structural segments. It begins with the extraction of primary planes from a raw point cloud or mesh, which aligns with the initial steps of traditional methods [Bauchet and Lafarge 2020; Nan and Wonka 2017]. The scheme then progresses by using the detected planes to form structural segments, serving as the foundation for subsequent LOD control.

A challenge in performing structural segmentation on architectural models is the presence of two distinct types of structures: alcove and closed

(see right). As outlined in Algorithm 1, we address this challenge by initially inverting planes smaller than A_ϵ to convert alcoves into
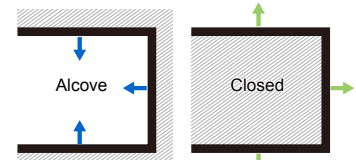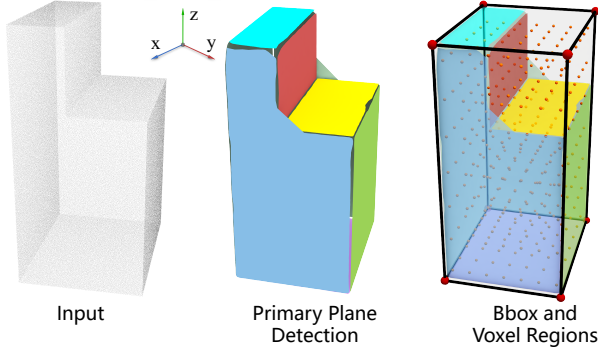
Fig. 3. Visualization of primary planes and Bbox with voxel samples.

closed structures for extraction, and then utilizing the remaining planes for closed structure extraction. These generated segments effectively facilitate subsequent LOD-based analysis.

Initial Segmentation. To initiate structural segmentation, the process begins by detecting primary planes through region growing, using the α shape [Bernardini and Bajaj 1997] to define the region for each detected plane. Once these planes are identified, a voxel-based visibility analysis is conducted to group strongly correlated planes into plane-groups.
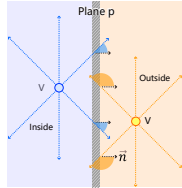
The procedure is outlined as follows: 1) Bounding Box Sampling: As illustrated in Fig. 3, we start by computing a bounding box (Bbox) for the detected planes and divide it uniformly into $n_s \times n_s \times n_s$ voxels, each with a sampled centroid. 2) Ray Shooting: From each voxel centroid v, n_r rays are emitted, uniformly sampled on the XY plane. These rays are used to determine the planes surrounding the voxel. Specifically, the first plane intersected by a ray is denoted as p. If v is inside p (assuming the normal of the plane points outward), the interaction is counted as a valid hit. 3) Visibility Analysis: For each voxel v, the visibility information regarding the planes is recorded in an n_p-dimensional vector $Hitlist_v$, where n_p is the number of planes. Each entry in $Hitlist_v$ records the number of valid hits from all rays emitted from voxel v. 4) Voxel Validation: A voxel with valid hits less than half of the total emitted rays (n_r) suggests that the voxel might be outside of the building, leading to its invalidation. 5) Grouping Planes into Plane-Groups: For each valid voxel v, all planes with non-zero entries in $Hitlist_v$ form a plane-group. This group represents a subset of planes that define the characteristics of the space occupied v, encapsulating essential spatial features for further architectural analysis.

By following these steps, the segmentation yields groups of interrelated planes, termed plane-groups, which are foundational for subsequent levels-of-detail analysis in architectural modeling.

Segments Aggregation. Once the plane-groups are obtained, we further aggregate them to generate structural segments as depicted in Fig. 4. Each segment represents a collection of planes surrounding
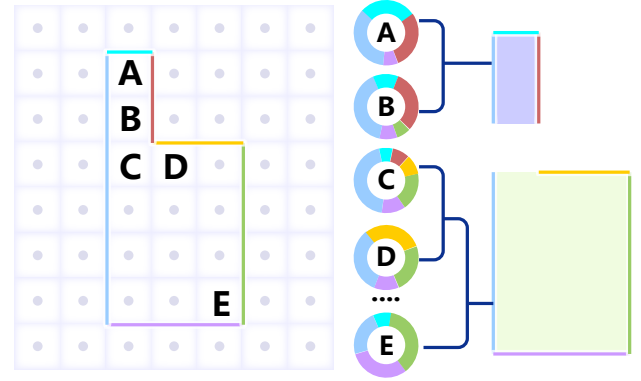
Fig. 4. An instance of structural segments aggregation. Plane-groups A, B, C, D, and E are concentrated based on related voxel regions. According to the overlap in $Hitlist$, interrelated planes (blue, cyan, and brown lines in the upper right corner) is detected to aggregate structures (A, B), then a structural segment (corresponds to the pale purple area) is generated.

a standard component composed of multiple voxels. The aggregation mainly involves organizing the plane-groups to obtain more compact and concise information, corresponding to the architectural LOD. Let g_a and g_b represent the two plane-groups enclosing voxels V_a and V_b, respectively. The total valid hits of a plane-group is calculated by

$$hSum(g, V) = \sum_{v \in V} \sum_{p \in g} Hitlist_v[p], \qquad (1)$$

where g represents a plane-group, p is a plane belong to g, V is the related voxels. Then, the pre-condition for merging can be written as $hSum(g_a \cap g_b, V_a) > 0.5 \times hSum(g_a, V_a)$ and $hSum(g_a \cap g_b, V_b) > 0.5 \times hSum(g_b, V_b)$. It means that the merging of two plane-groups should satisfy a preset overlapping ratio. We iteratively assess the potential merging of plane-groups in pairs and update the merged plane-group as $g_{ab} = g_a \bigcup g_b$, continuing this process until no further pairs are available for merging. A merging queue is maintained to keep current segments. Only the adjacent segments are checked in the queue which avoid exhaustive searching. Finally, we achieve the merged plane-groups to be structural segments. It should be noticed that a single plane could be part of multiple structural segments. To achieve disjointed segmentation results, we assign each plane to the structural segment with largest validly hit number. These structural segments serve as primitives for subsequent co-analysis.

3.2 Joint Structural Analysis

Based on the structural segments, Co-LOD performs co-analysis to ensure semantic consistency in LOD generation across different buildings. To implement the scheme of joint structural analysis, we have designed a two-step method. In the first phase, we identify segments belong to LOD0 through joint analysis of structural segments. It is used to define the initial main structure of the building. Next, we employ spectral clustering to jointly analyze segments beyond LOD0, assigning them to different layers to generate LODs. In this way, Co-LOD achieves progressive LOD generation layer by layer.

ALGORITHM 1: Structural Segments Generation

Input: Building model, denoted as m
Output: A set of segments of m, denoted as S
1   $P \leftarrow$ RegionGrowing(m);
2   // Handle alcove structures
3   Reverse the orientation of planes $p \in P$ with an area less than A_ϵ
4   $G \leftarrow$ InitialSegmentation(P)
5   $S_a \leftarrow$ SegmentsAggregating(G)
6   // Handle closed structures
7   Recover the orientation for all planes $p \in P$
8   **for** $p \in S_a$ **do**
9     $P \leftarrow P/\{p\}$
10 **end**
11 $G \leftarrow$ InitialSegmentation(P)
12 $S_c \leftarrow$ SegmentsAggregating(G)
13 $S \leftarrow S_a \cup S_c$

Initial LOD0 Generation. Firstly, we provide the implementation of LOD0 generation for initial main structure of building representation. A reasonable LOD0 definition should satisfy following requirements: corresponds to the overall shape, as concise as possible, and keeps semantic consistency. Based on the requirements, we formulate the objective function using three terms, which quantitatively evaluate shape fidelity (f_r), simplicity (f_s), and semantic consistency (f_{co}), respectively. That is:

$$E_0 = \sum_{i=0}^{n} \left(f_r(M_i) + \beta f_s(M_i) + \lambda f_{co}(M_i) \right), \quad (2)$$
$$\text{s.t.} \quad f_r(M_i) > 0.8 \quad \forall i = 1, 2, ..., n,$$

where n represents the number of buildings for co-analysis, I_i represents the set of structural segments associated with the i-th building, M_i represents the set of structural segments identified as the LOD0 of I_i, β and λ are two parameters to tune the influence of f_s and f_{co}. Higher E_0 score corresponds to a better LOD0 structure that balances the three requirements. It can be modeled as a binary linear programming problem, which determines whether a segment belongs to LOD0.

Intersection over Union (IoU) between M and I is used to define the shape fidelity term f_r. It can be computed as

$$f_r(M) = IoU(M, I) = IoU(P(M), P(I)), \quad (3)$$

where $P(X)$ represents voxel centroids enclosed by the structural segments collection X. To satisfy the second requirement of concise representation, the simplicity term f_s is introduced by constraining the area of planes, computed as

$$f_s(M) = -area(M)/area(I), \quad (4)$$

where $area(M)$ represents the total area of planes contained in M, $area(I)$ represents the total area of planes contained in I. By maximizing the term, geometric structures of LOD0 can be greatly simplified.

To address the semantic consistency, we first define the measurement between two segments, represented as

$$dis(s, s') = \eta \| d2(norm(s)) - d2(norm(s')) \|_2 + scale_d(s, s'), \quad (5)$$
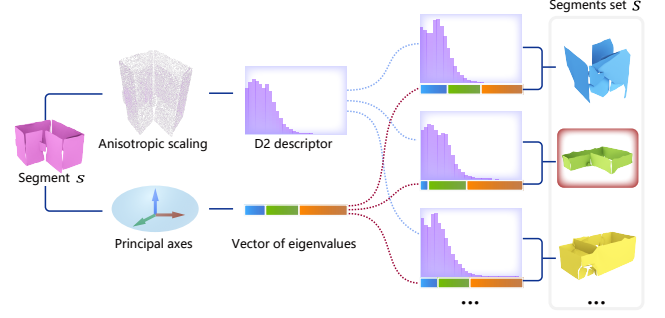

Fig. 5. Illustration of segment-based similarity. With the D2 descriptor-based measurement and principal axes-based eigenvalue analysis, different segments can be quantitative compared, which considers shape and scale differences at the same time.

$$scale_d = \| v_e(s) - v_e(s') \| / (\| v_e(s) \| + \| v_e(s') \|), \quad (6)$$

where dis is a function used to measure the distance between segments s and s'. A smaller distance implies a higher similarity between the two segments. We define the distance with two components, shape distance and scale distance $scale_d$. The $norm(s)$ represents the process of making s isotropic through anisotropic scaling [Kazhdan et al. 2004], $d2(x)$ denotes the D2 descriptor [Osada et al. 2002] for x, and if we denote $\boldsymbol{v}_e(s)$ as the vector of principal eigenvalues of the covariance matrix of s, then $scale_d$ is defined as the normalization of $L1$-distance based on $\boldsymbol{v}_e(s)$ and $\boldsymbol{v}_e(s')$. For intuitively illustrate such similarity, we show an instance in Fig. 5. According to the segment-based similarity, we perform a cross-analysis of segments on different buildings, formulated as

$$Dis_{set}(s, S) = \min_{s' \in S} dis(s, s'), \quad (7)$$

$$Sim_{co}(s, S) = e^{-Dis_{set}(s, S)}, \quad (8)$$

where Dis_{set} represents the distance between s and a segment group S, Sim_{co} represents the cross similarity based on Dis_{set}, it is improved by logarithmic operation for discrimination (Fig. 6). Once the cross similarity is defined, the consistency term f_{co} can be computed. It considers the mutual similarities between different segments according to the definition of LOD [Gröger and Plümer 2012] and real-life scenario (segments beyond LOD0 often comprise elements such as windows, fences, rooftop water tanks, air conditioning units, and other components that are not individually designed). To formulate f_{co}, we encourage segments that match well with others to be removed from LOD0 since they represent repetitive structures. In practice, f_{co} is defined as

$$f_{co}(M) = \sum_{s} \sum_{M', I'} Sim_{co}(s, I' - M'), \quad (9)$$
$$s \in I - M, M' \in \mathcal{M} - M, I' \in \mathcal{I} - I,$$

where $\mathcal{M}$ and M are collections of LOD0-based segments corresponding to all architectural models and current model, $\mathcal{I}$ and I are collections of segments from all architectural models and current model. $I - M$ represents segments beyond LOD0 of current model. By using set operations, we compare segments in certain range by cross similarity Sim_{co} for co-analysis of LOD0.
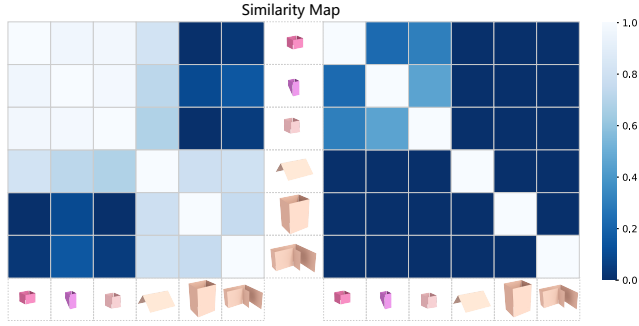
Fig. 6. Structural segment-based similarity map by Dis_{set} (left) and Sim_{co} (right). The latter one significantly distinguishes the difference between segments.

Spectral Clustering. Through the optimization of E_0 in Eq. (2), we obtained an accurate LOD0 architectural model. To further assign more detailed segments into their appropriate LODs, we employed spectral clustering to simultaneously analyze these segments. Based on the cross similarity Sim_{co}, we constructed the similarity matrix M_s to measure the similarity between each pair of segments. Subsequently, we compute the normalized Laplacian matrix L_{norm}, represented as:

$$L_{norm} = M_D^{-1/2}(M_D - M_s)M_D^{-1/2}, M_{D_{ii}} = \sum_j M_{s_{ij}}, \quad (10)$$

where M_s is the similarity matrix constructed by cross similarity Sim_{co}, $M_{s_{ij}} = Sim_{co}(s_i, s_j)$, M_D is the related diagonal matrix. Inspired by spectral clustering [Pothen et al. 1990], we use the eigenvectors corresponding to the first l_n-1 smallest eigenvalues of L_{norm} as features for segments and cluster them by k-means method. Then, such segments are clustered into l_n-1 classes, corresponding to l_n-1 different LOD levels, l_n corresponds to the required LOD layers. In practice, for datasets featuring finely scanned models, we typically set $l_n = 3$ as shown in Fig. 8 and Fig. 23. Conversely, for collections with roughly scanned models, we set $l_n = 2$, as shown in Fig. 1 and Fig. 22. Clusters with smaller average volume are assigned to higher LOD levels, indicating that they represent finer structures. A higher layer of LOD is formed by the planes corresponding to segments smaller than or equal to the current LOD layer with adjacent planes that don't belong to any segments. In this way, we utilize such planes to construct more refined and comprehensive polygonal mesh.

3.3 Polygonal Mesh Extraction

To achieve the desired polygonal meshes, we utilize a space partitioning approach followed by convex hull inside/outside calibration to extract polygonal meshes based on layered segments. Specifically, planes belonging to specific LOD layers undergo Binary Space Partitioning [Fuchs et al. 1980] to divide the architectural model's bounding box into convex hulls. Subsequently, we determine the interior or exterior attributes of each convex hull based on whether
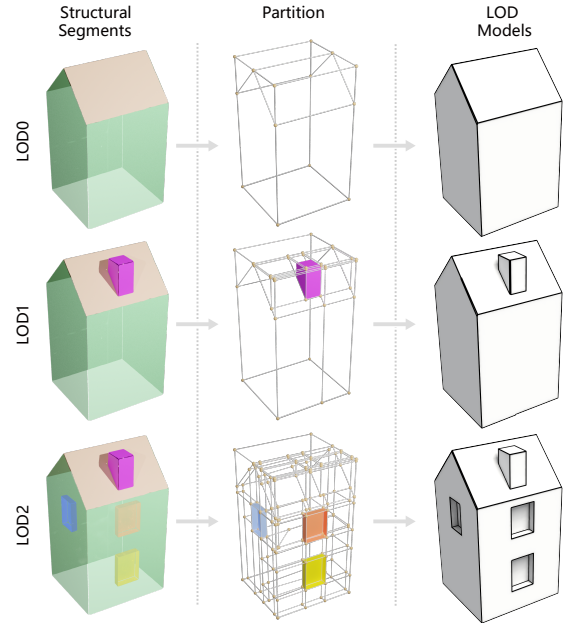

Fig. 7. Illustration of polygonal mesh extraction with different LOD layers. By partitioning scheme with layered structural segments, accurate LOD generation can be implemented.

more than half of the rays see the convex hull as interior. That is:

$$In(C) = \begin{cases} 1, h_v(C) > 0.5n_r \\ 0, other \end{cases}, \quad (11)$$

where $In(C)$ is the index of interior ($In(C) = 1$) or exterior ($In(C) = 0$) attribute, C is the centroid of the related convex hull that also can be regarded as a sub-space, $h_v(C)$ is the total valid hit count from n_r rays uniformly emitted from C. Based on all interior convex hulls, the largest internally connected component forms the watertight space, which produces the polygonal mesh for current layer of LOD. An instance is shown in Fig. 7. Unlike traditional methods that directly use detected planes for polygonal mesh reconstruction, Co-LOD utilizes segment co-analysis to guide clustering and concentrating related planes, which ensure precise and controllable LOD generation with semantic consistency. In experiments, we illustrate the performance of Co-LOD in practical urban scenes.

4 EXPERIMENTAL RESULTS

We evaluate the performance of Co-LOD in the context of urban scene-based architectural LOD generation tasks. All experiments were conducted on a machine equipped with an Intel i9-13900K processor, 128GB RAM, and an RTX4090 graphics card, running Windows 10 and Visual Studio 2022. For linear algebra operations, basic geometry processing, and numerical optimization, we utilized public libraries such as Eigen [Guennebaud et al. 2010], CGAL [Fabri et al. 2000], and LPSolve. Comparative analysis between Co-LOD and several classical solutions, as well as single building processing and ablation studies, are used to comprehensively assess the effectiveness of Co-LOD.
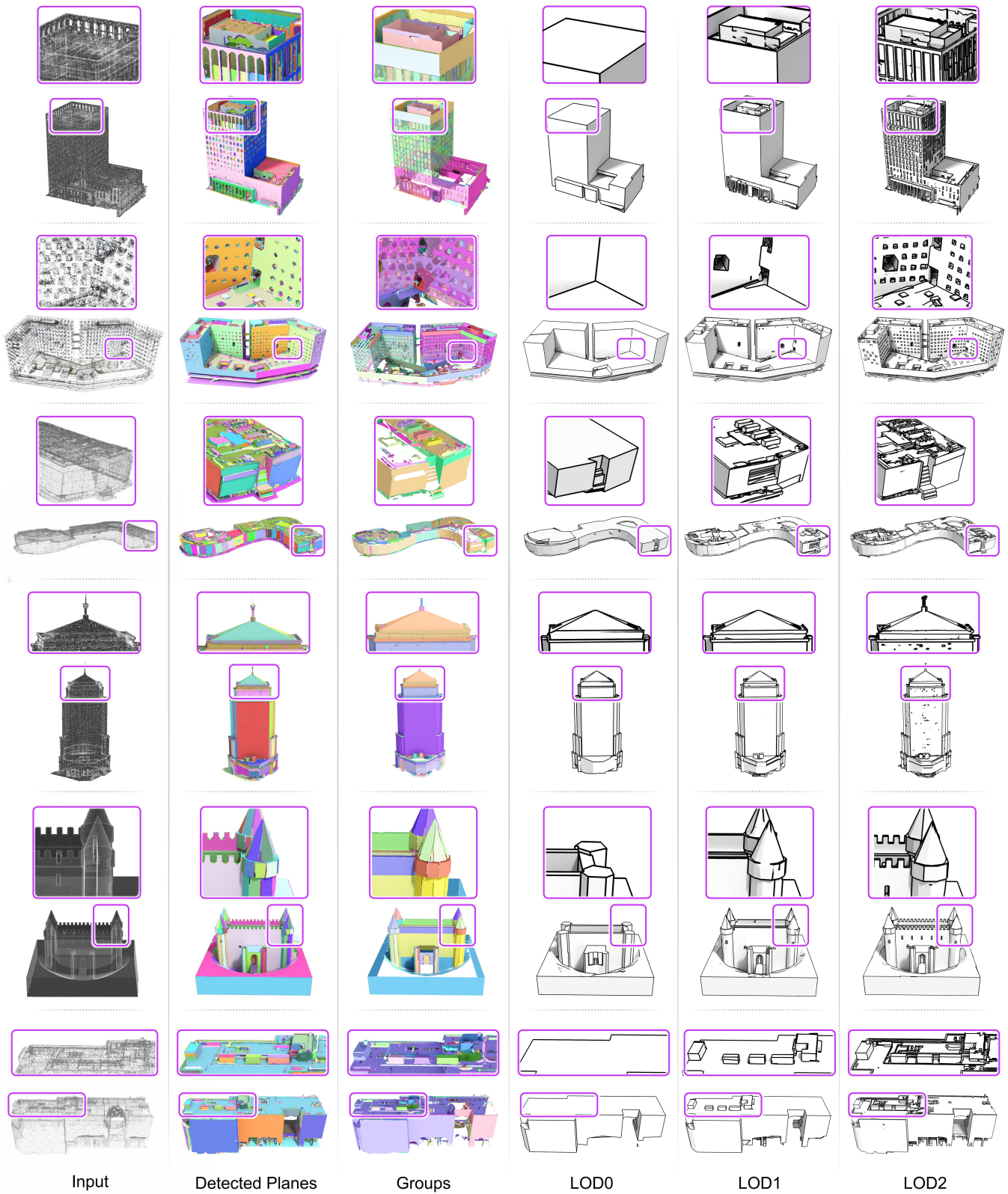
Fig. 8. LOD generation results for *Composite Scene*, demonstrating Co-LOD's ability to handle both diverse styles and delicate structures.
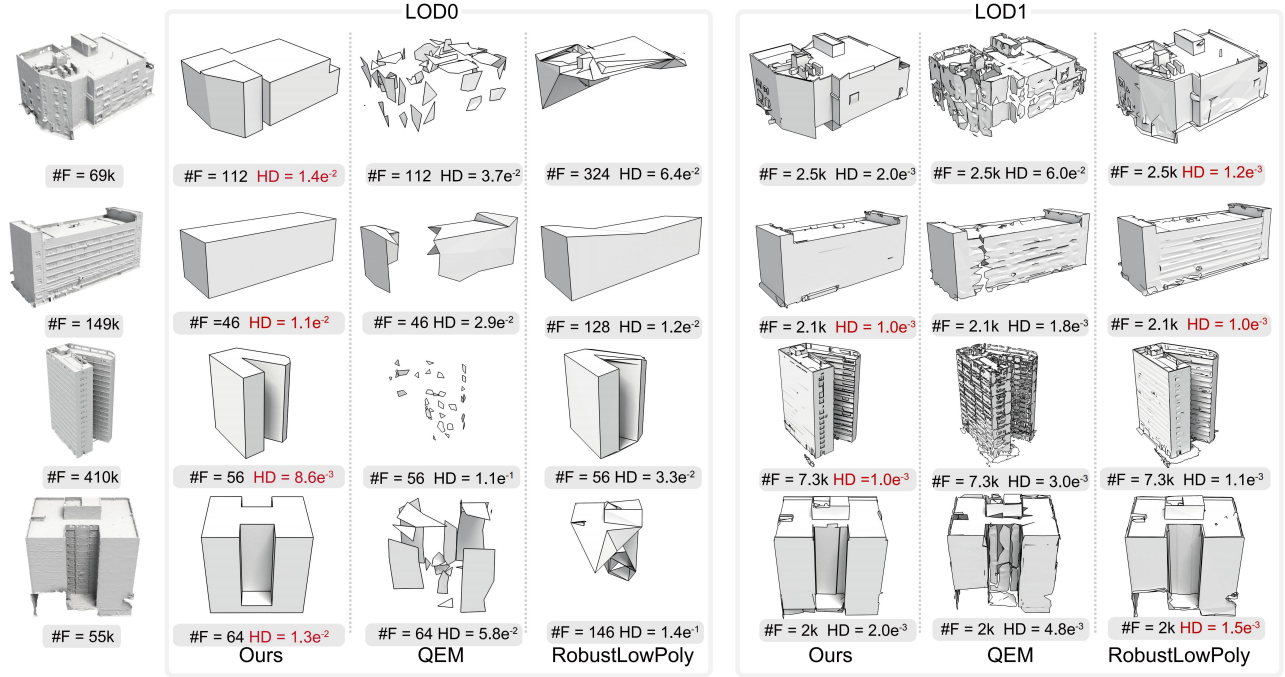
Fig. 9. Comparisons with QEM and RobustLowPoly for LOD generation. Co-LOD achieves better balance between simplicity (#F: face number) and accuracy (HD: Hausdroff distance between the raw input and related reconstruction result).

Table 1. Statistics of input scenes, including $\overline{V}$: average vertices, $\overline{F}$ average faces, $\overline{C}$: average connected components, $\overline{I}$: average self-intersections, $\overline{P}$: average detected planes, $\overline{S}$: average segments, $\overline{T}$: average runtime, M: model numbers.

Scenes	$\overline{V}$	$\overline{F}$	$\overline{C}$	$\overline{I}$	$\overline{P}$	$\overline{S}$	$\overline{T}$	M
Metropolis	199k	271k	5.9k	363k	2476	22	229.8s	54
Town	148k	288k	1k	15k	1378	13	75.6s	27
Research Center	371k	543k	7k	971k	3127	72	776.9s	14
Composite Scene	588k	867k	16k	964k	3468	93	987.8s	13
European City	59k	86k	2.3k	96k	735	67	522.7s	10
Campus	56k	70k	3.6k	125k	501	22	144.8s	37
Suburbia	1k	2k	1	3	18	3	56.7s	266

Datasets. To thoroughly evaluate the effectiveness of our method, we investigated building datasets published post-2020, including Toronto-3D [Tan et al. 2020], SensatUrban [Hu et al. 2022], SUM [Gao et al. 2021], STPLS3D [Chen et al. 2022], InstanceBuilding [Xu et al. 2021], and UrbanBIS [Yang et al. 2023]. We conducted experiments on the datasets with normal information, totaling 421 architectural data samples of various styles and captured from diverse locations. These models were reconstructed using UAV photogrammetry, which introduced inherent noise and incomplete information. We organized the samples into seven scenes. Following the approach in LowPoly [Gao et al. 2022], we compiled the statistics for the scenes used in Table 1. As Co-LOD takes a collection of building as input, the seven building scenes are individually fed into the pipeline for co-analysis.
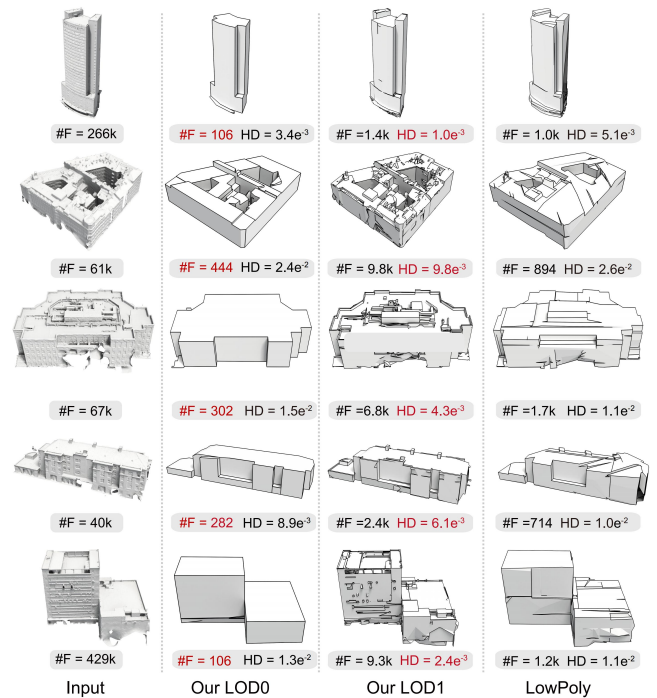

Fig. 10. Comparisons with LowPoly for LOD generation. LowPoly fails to keep some main structures. Co-LOD achieves more concise and clear main structures.
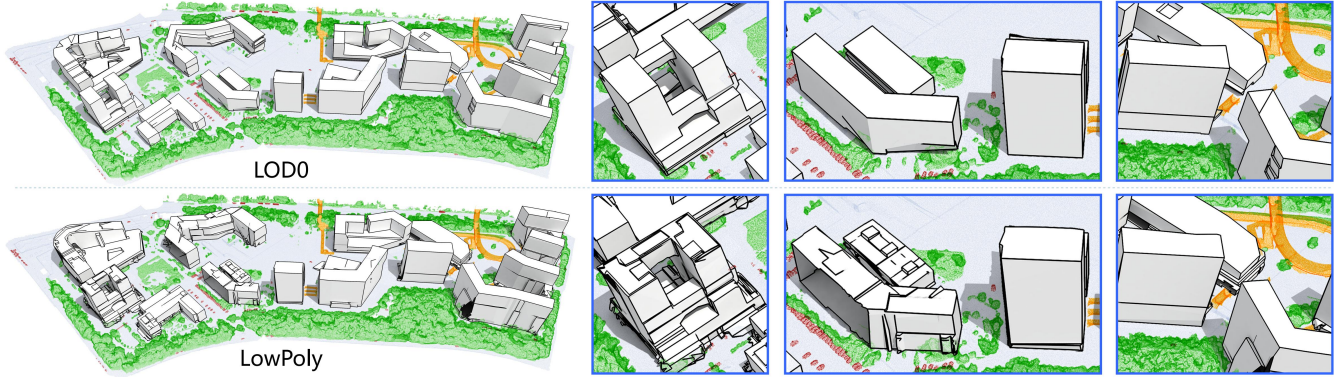
Fig. 11. Comparisons with LowPoly for LOD generation using *Research Center*. The zoomed-in views highlight the notable inconsistencies in the amount of details used by LowPoly to model different buildings.

Table 2. Statistics and evaluations of different LOD generation methods for related LOD layers. Since PolyFit can only handle buildings from D7 successfully, we separately list quantitative data in D7. Other results are reported based on all scenes. #V: vertices, #F: triangles, #P: polygons, mean: average value of related metric, sd: standard deviation of related metric. For statistic analysis of success rate r_s, a failure case means that if the method cannot generate valid output or the running time exceeds one hour. I-LOD and S-LOD present user study outcomes for individual buildings and scenes, respectively. The values represent the percentage each method was selected as the best.

| LOD Layer | Method | #V | #F | #P | r_s | LFD ↓ | | HD ↓ | | I-LOD ↑ | S-LOD ↑ |
						mean	sd	mean	sd		
LOD0 (D7)	PolyFit	65	51	9	42%	6757.7	2886.0	0.110	0.060	——	——
	Ours	14	22	9	42%	**5063.8**	1875.6	**0.035**	0.025	——	——
LOD0	KSR	232	427	124	47.8%	5963.1	4126.3	0.048	0.053	10.9%	8.7%
	QEM	189	125	113	100%	5388.7	2577.4	0.037	0.028	3.2%	4.7%
	RobustLowPoly	117	254	250	100%	7146.5	4349.8	0.097	0.069	11.7%	5.3%
	Ours	66	125	40	100%	**4507**	1785.7	**0.025**	0.024	**77.3%**	**81.3%**
LOD1	KSR	1010	1932	681	98.8%	3793.2	1429.4	0.010	0.011	10.9%	8.7%
	LowPoly	542	1046	645	100%	3899.5	1235.5	0.012	0.006	——	——
	QEM	5159	3753	3584	100%	3889.2	4075.2	**0.005**	0.005	3.2%	4.7%
	RobustLowPoly	1865	3750	3594	100%	4961.7	3770.9	0.073	0.068	11.7%	5.3%
	Ours	1888	3755	1352	100%	**3626.5**	1481.2	**0.005**	0.004	**77.3%**	**81.3%**
LOD2	KSR	6648	12898	7723	70.0%	3026.3	1103.3	0.007	0.013	10.9%	8.7%
	QEM	53629	31501	31239	100%	9892	12607.1	0.002	0.001	3.2%	4.7%
	RobustLowPoly	15719	31535	29982	100%	**1975.4**	405.7	**0.001**	0.001	11.7%	5.3%
	Ours	15811	31535	14516	100%	2237.2	432.4	**0.001**	0.001	**77.3%**	**81.3%**

Metrics & Configurations. To evaluate the performance of the LOD generation methods, we employ Hausdorff distance (HD) and Light Field distance (LFD) [Chen et al. 2003] to measure geometric and visual errors. We also report success rate (r_s) and user study at both individual and scene levels to reveal the practicality. For parameter configurations of Co-LOD, we set $n_s = 150$, $n_r = 100$, $A_\epsilon = 80m^2$, $\beta = 0.3$, $\lambda = 0.4$, $\eta = 4.0$ as the default specification. The rationality of parameter selection are discussed in the subsequent sections. For the plane detection, we employ the Region Growing algorithm [Rabbani et al. 2006] provided by CGAL. The distance and angle-based thresholds are set to 0.2m and 30°, respectively, and the minimum region size requiring 20 inside points. The results generated using Co-LOD for some instances are shown in Fig. 8.

4.1 Comparisons

To validate the effectiveness of Co-LOD, we conducted comparisons with several mainstream approaches, including QEM [Garland and Heckbert 1997], RobustLowPoly [Chen et al. 2023], LowPoly [Gao et al. 2022], NeuralLOD [Takikawa et al. 2021], PolyFit [Nan and Wonka 2017], and KSR [Bauchet and Lafarge 2020]. These comparisons were based on the previously mentioned metrics and included user studies at both individual and scene levels. Additionally, we presented the generated LOD models visually to demonstrate the hierarchical consistency advantage of Co-LOD.

Comparisons with QEM and RobustLowPoly. We conducted comparative experiments using QEM, RobustLowPoly, and Co-LOD,
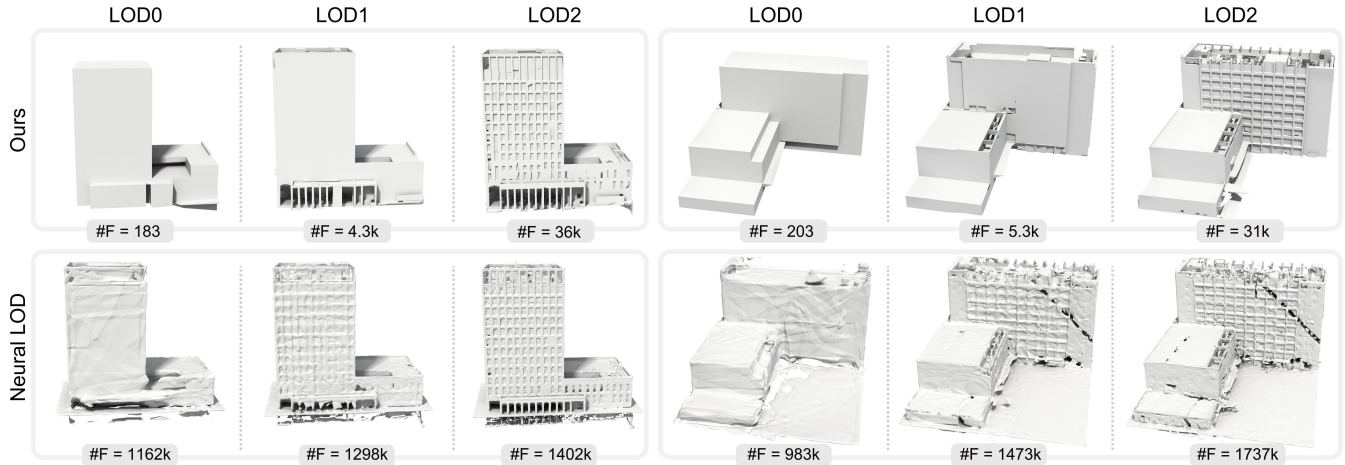
Fig. 12. Comparisons with NeuralLOD. The face number of NeuralLOD-based results are huge even in models with LOD0 layer. Co-LOD effectively reduces the face number while maintaining accurate geometric structures in different LOD layers.
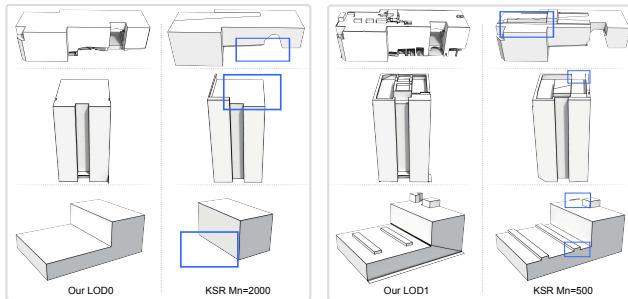

Fig. 13. Comparisons with Alternative Pipeline. With the parameter changing, KSR can generate different levels of geometric structures. However, the main structure can not be guaranteed which poses significant drawback for geometric and semantic consistency.

focusing on geometric and visual error optimization. Due to limitations in the ability of the comparison methods to implement accurate LOD control, we simplified the input facets to simulate the function of Co-LOD generation for a fair assessment. As demonstrated in Fig. 9, QEM demonstrates limitations in preserving architectural integrity, particularly struggling to retain sharp features during LOD generation, which results in poor visual effects. RobustLowPoly, while preserving basic geometric details more effectively, still falls short in ensuring the consistency of building structures. Additionally, it lacks the capability to provide structural completion and to smooth noisy planes. Moreover, both methods struggle to maintain the main structure when simplified to a very low number of faces, leading to lower geometric and visual fidelity and inconsistencies within the LOD0 layer.

Benefited from the co-analysis, Co-LOD excels in maintaining structural integrity and achieving better semantic consistency with fewer planes. These advantages are evidenced by quantitative evaluations of HD and LFD, as detailed in Table 2.

Comparisons with LowPoly. LowPoly can construct the main structure of an architectural model, making it a suitable benchmark for evaluating the LOD generation capabilities of Co-LOD. However, it has some limitations in maintaining semantic consistency within the LOD layer, as shown in Fig. 10 and Fig. 11. Different models exhibit varying levels of simplification, and repetitive structures are not consistently removed. In contrast, Co-LOD can generate a more concise and uniform LOD0 layer across various architectural models. For LOD1 generation shown in Fig. 10, Co-LOD produces more accurate geometric structures and better semantic consistency. Further comprehensive results are reported in Table 2.

Comparisons with NeuralLOD. The advent of deep implicit surface reconstruction has brought new possibilities to the generation of various LOD layers for architectural models, as exemplified by NeuralLOD. In Fig. 12, we present a comparison of architectural models generated by both NeuralLOD and Co-LOD, across different LOD layers. NeuralLOD, particularly at lower LOD layers, tends to smooth out geometric details. While this is consistent with the principles of LOD generation, the numbers of faces needed to represent these smooth surfaces are significantly larger than the ones needed by Co-LOD. In addition, NeuralLOD does not operate as a structured reconstruction scheme and lacks the capability to extract primary planes. Consequently, even at higher LOD layers, Co-LOD demonstrates a significant advantage in terms of simplification, while simultaneously maintaining the accuracy of the geometric structures. This comparison underscores the efficiency and precision of Co-LOD in generating LOD layers for architectural models.

Comparisons with PolyFit and KSR. For architectural model reconstruction, classical methods [Bauchet and Lafarge 2020; Chauve et al. 2010; Nan and Wonka 2017] focus on detecting primary planes to generate polygonal models. While these methods are not tailored for precise LOD control, they offer some level of accuracy adjustment in reconstructed buildings through parameter tuning. The
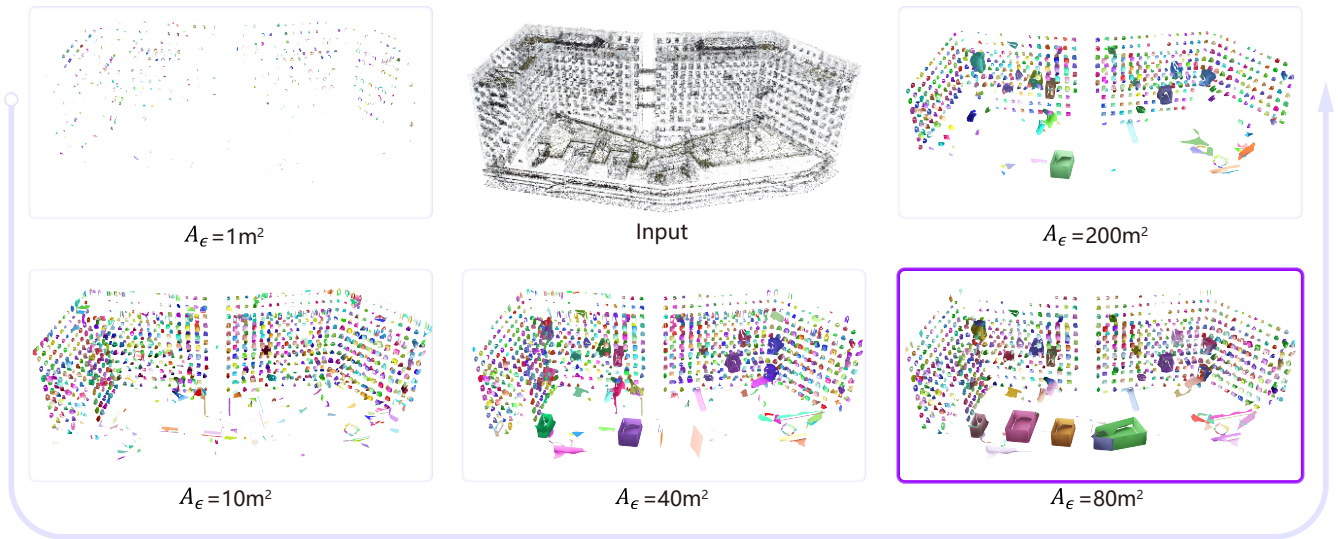
Fig. 14. The impact of A_ϵ on alcove structure detection. While a larger A_ϵ facilitates the detection of larger alcove structures, excessively increasing A_ϵ to $200m^2$ can impede detection by causing the inversion of large planes.
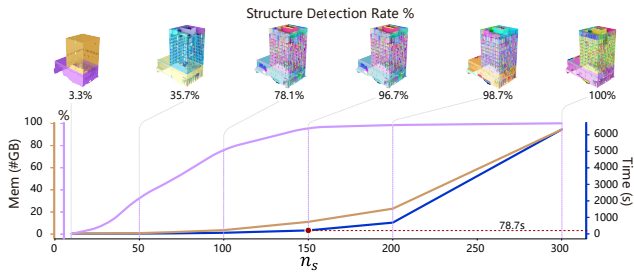

Fig. 15. Relationship between n_s and computational cost, benchmarked at $n_s = 300$ for SDR calculations. Increasing n_s improves detection of fine structures but incurs a non-linear rise in computational costs.

Fig. 16. Impact of the shape-scale balance weight η on segment similarity: Increasing η shifts the metric emphasis from scale to shape similarity.

key parameter is the minimum number of inliers (Mn), which influences the scale of detected planes by dictating the minimum points needed for a valid plane. We use KSR [Bauchet and Lafarge 2020] as a baseline method and adjust the Mn to simulate LOD control. The reconstructed architectural models shown in Fig. 13 illustrate that different Mn values for KSR can indeed control geometric details. However, it cannot ensure the integrity of the main structure. Precise geometric details obtained with lower Mn are inconsistent. Conversely, Co-LOD effectively addresses the issue, achieving accurate and uniform LODs. Furthermore, Table 2 quantitatively demonstrates that Co-LOD achieves higher modeling accuracy at similar complexities and handles complex inputs more effectively.

User Study. We conducted a user study utilizing all scenes and randomly selected 30 individual models, to evaluate the quality of LOD models generated by various methods at both the individual model and scene levels. This study involved 108 participants who chose the best set of LODs from four groups, each provided by
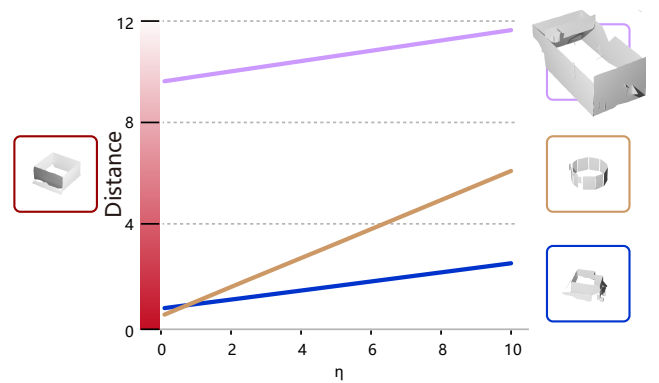
a different method, based on their balance between fidelity and simplicity. Detailed results are presented in Table 2. Our method was preferred for both individual buildings and scenes, receiving approval ratings of 77.3% and 81.3%, respectively. This confirms the effectiveness of our approach in generating LOD models that are both geometrically and visually accurate, as well as aesthetically appealing. For visualizations of the comparison results, please refer to our supplementary materials.

Co-LOD for Single Building. By default, Co-LOD is used to control LOD generation in scenes with multiple buildings while maintaining semantic consistency. When processing single buildings, Co-LOD can benefit from incorporating an additional database to build prior knowledge. To demonstrate this, we utilized *Composite Scene* to compile a database consisting of 90 segments with
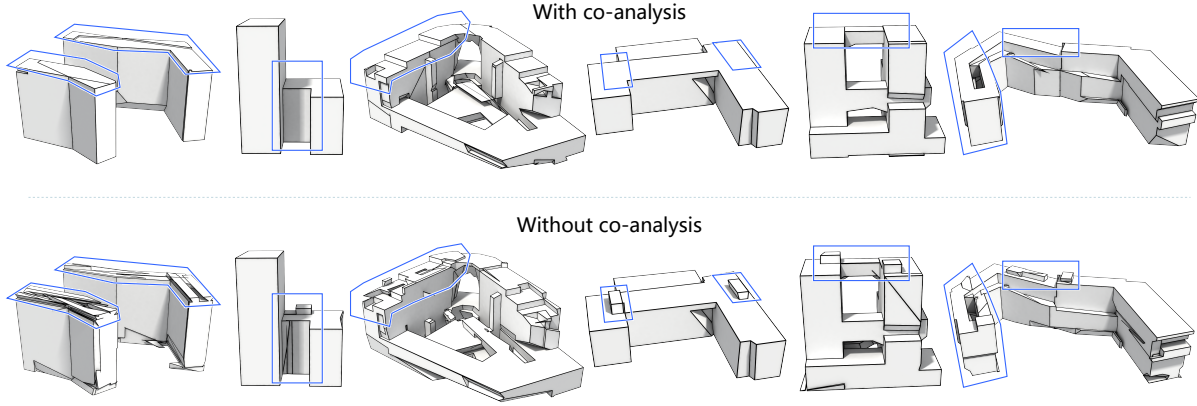
Fig. 17. Ablation study on the impact of consistency term f_{co} on LOD0 generation. The top row shows results with the term, where LOD0 exhibits concise and consistent geometric structures. In contrast, the bottom row, without f_{co}, displays irregular geometric structures that introduce disturbances.
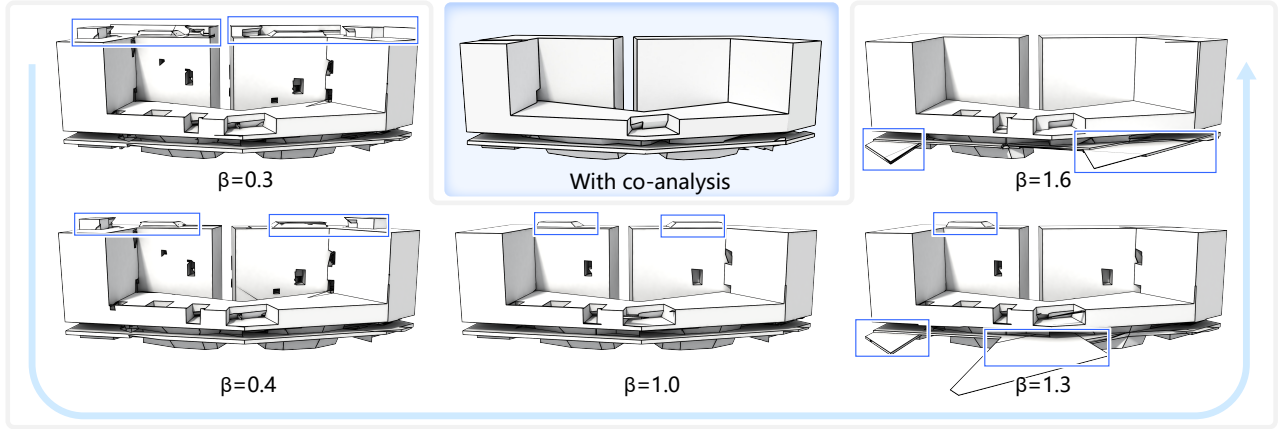

Fig. 18. Evaluation of shape term f_r and simplicity term f_s in Co-LOD without function of f_{co}. Tuning the ratio between the two terms cannot effectively generate a desired LOD0 model, underscoring the indispensable role of co-analysis in Co-LOD controlling.

various shapes. The experiment, conducted using *Campus*, assessed the effectiveness of co-analysis when supplemented with additional database knowledge. Table 3 demonstrates that using a database significantly reduces calculation time, primarily by speeding up the co-analysis stage. However, a downside is that non-primary structures not matched within the constructed database may be erroneously assigned to lower LOD layers, resulting in insufficient simplification. Further details of this implementation can be found in the supplementary materials.

Efficiency. Across the tested dataset, Co-LOD takes an average of 151.6s to process an input, including stages for structural segment generation (78.7s), co-analysis (51.8s), and polygonal mesh extraction (21.1s). In contrast, comparative methods include KSR, PolyFit, LowPoly, QEM, and RobustLowPoly take 87.3s, 5.1s, 513.3s, 2.1s and 603.2s, respectively. Overall, Co-LOD achieves a better balance in terms of computational efficiency, LOD control, semantic consistency, and practicality.

Table 3. Statistics of Co-LOD in different LOD generation tasks on *Campus*. S: single building analysis, Co: co-analysis based on the scene, superscript numbers: LOD layer, #F: triangles, #V: vertices, T_s: segments generation time cost, T_c: co-analysis time cost, T_p: polygonal mesh extraction time cost.

Task	#F	#V	HD	LFD	T_s	T_c	T_p
S^0	421	188	0.023	3633	2827.1s	167.3s	534.6s
S^1	6898	3510	0.001	1873			
Co^0	346	179	0.021	3656	2834.6s	1972.2s	549.3s
Co^1	6775	3417	0.001	1889			

4.2 Ablation & Limitations

Co-LOD delivers accurate LOD generation with consistency for architectural models. Given the multiple control parameters and optimization elements within Co-LOD, we conducted an ablation study to elucidate their impacts and discuss some limitations.
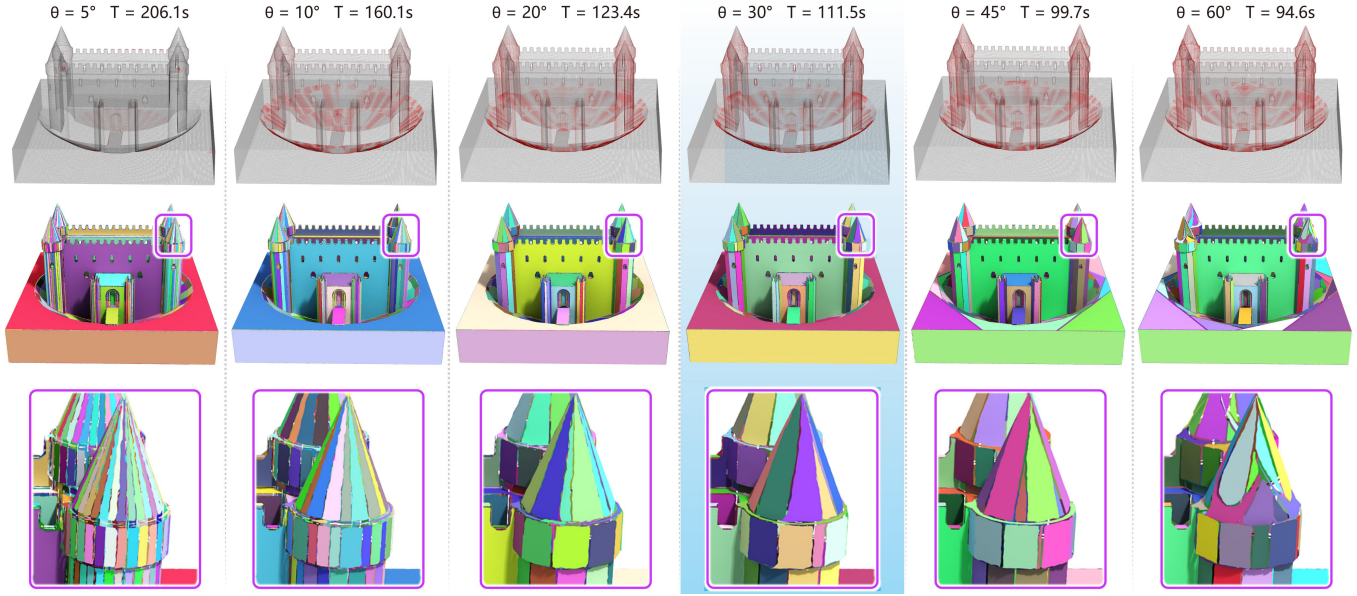
Fig. 19. Errors from plane-based surface approximation. Reducing the angle-based threshold (θ) mitigates these errors but increases computational time (T). Deeper reds in the point cloud signify higher reconstruction errors.

Table 4. Statistical data for varying n_r values: HD and LFD between the original models and related highest LOD layer models, T_s: structural segment generation time cost, T_p: polygonal mesh extraction time cost, SDR: calculated using $n_r = 200$ as the benchmark.

Values of n_r	HD		LFD	T_s	T_p	SDR
	mean	sd				
5	0.236	0.12	5305	39.3s	2.7s	2.3%
10	0.103	0.036	3803.2	42.5s	2.9s	67.8%
30	0.003	0.002	2499.6	49.6s	7.3s	96.3%
60	0.002	0.002	2397.3	61.1s	12.5s	97.5%
100	0.001	0.001	2375.7	78.3s	21.6s	97.3%
150	0.001	0.001	2374.9	102.7s	43.2s	97.2%
200	0.001	0.001	2375.2	125.3s	95.3s	100%

Fig. 20. Limitation in detecting finer structures. The inability to extract the circled structure led to inconsistency in the LOD1 model.

Ablation Study. Co-LOD implementation involves various parameters and optimization items that require careful consideration, including alcove structure area A_ϵ, BBox resolution n_s, number of Rays Emitted n_r, shape-scale balance weight η, and terms in Eq. (2).
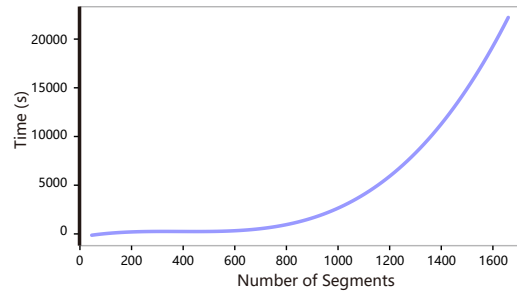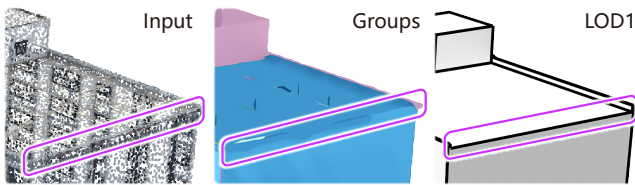

Fig. 21. The relationship between the calculation time of the co-analysis stage and the number of segments analyzed simultaneously.

The selection of A_ϵ primarily influences the detection of alcove structures, setting the upper size limit of detectable alcoves, as illustrated in Fig. 14. Through extensive experimentation, we have set A_ϵ to 80m^2, which effectively handles the alcove structures contained in our datasets.

The BBox resolution influences structural segmentation, which in turn impacts LOD generation. In Fig. 15, we explored various n_s values across our dataset and introduced the Structure Detection Rate (SDR) as our custom metric. SDR measures how accurately the Structural Segment Generation algorithm identifies relevant structures. It is calculated as the percentage of segments generated with the current parameter settings compared to those generated using the optimal parameters. Our findings indicate that increasing n_s detects finer structures but also raises computational consumption non-linearly. Therefore, we have set $n_s = 150$ to achieve the optimal balance between efficiency and performance.
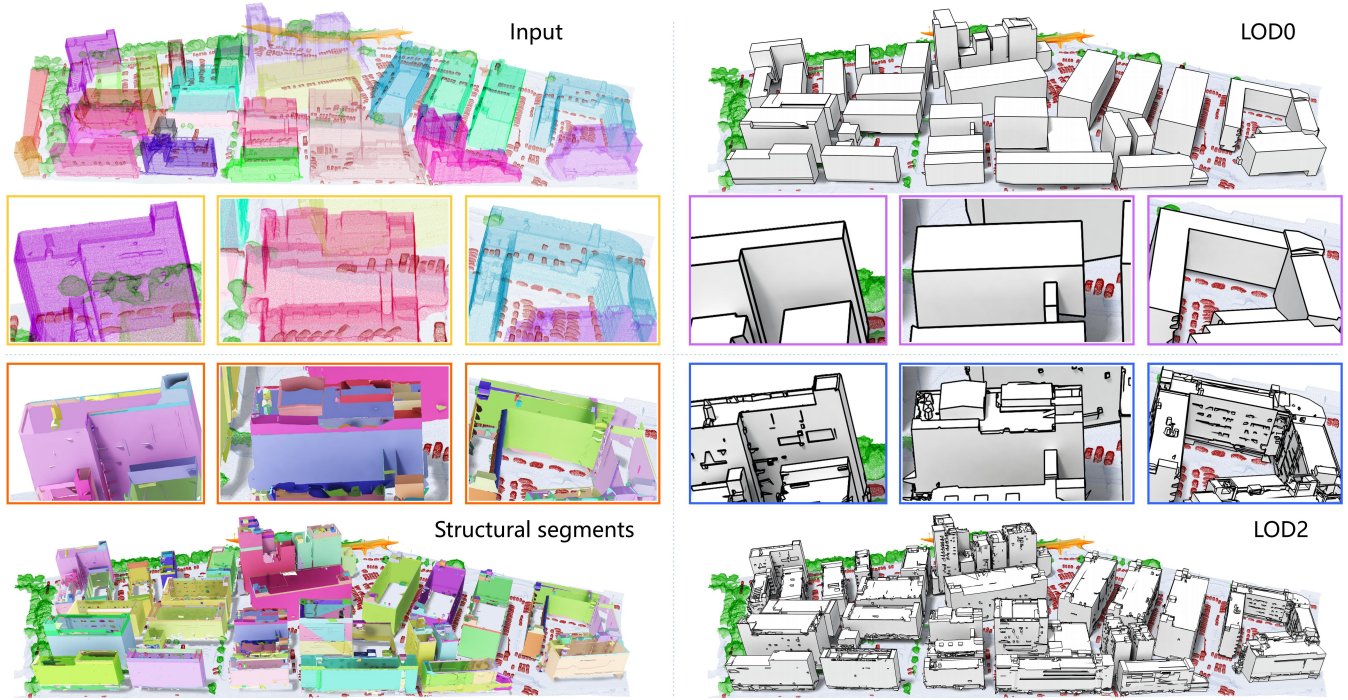
Fig. 22. LOD generation results by Co-LOD for *Town*.

The number of emitted rays n_r affects structural segment generation and polygonal mesh extraction, as shown in Table 4. This parameter is crucial for analyzing how a voxel is surrounded by planes; fewer rays increase the likelihood of missing surrounding planes, leading to inferior structural segmentation and mesh extraction. Considering that increasing n_r does not lead to non-linear growth in computational costs, we have chosen a larger default value (100) to enhance the robustness and HD-based evaluation.

Different values of η yield varied similarity measurements that influence LOD hierarchy as depicted in Fig. 16. Larger value of η prioritizes the shape and smaller one emphasizes the scale. Based on the extensive experiments, we set $\eta = 4.0$ to align shape and scale measurement space.

There are several optimization terms in Eq. (2) and their functions should be demonstrated, which take important influence for LOD0 generation. First, we evaluate the function of consistency term f_{co} that provides semantic consistency. In Fig. 17, we compare the results with and without f_{co} in Co-LOD. The f_{co} suppresses the intricate geometric details for LOD0 generation, thereby ensuring consistency within the level. Due to the Eq. (2) contains three terms, a reasonable assumption is to simulate the function of third term by changing the ratio of weights for the first two terms f_r and f_s. In Fig. 18, we compare the results with different weight ratios based on Co-LOD framework. As β increases, the main structure of selected architectural model is indeed simplified. However, the removal of essential structural segments resulted in the introduction of random noisy planes to close off the main structure. Overall, co-analysis cannot be simply replaced in accurate LOD generation task. We

believe that it is a significant discovery for future research works on architectural reconstruction.

Limitations. Co-LOD can handle curved structures, but its reliance on piecewise planar approximation introduces more errors in curved areas. Tweaking Region Growing [Rabbani et al. 2006] parameters can control the precision of surface approximation and, consequently, the reconstruction error, as demonstrated in Fig. 19. A reduced angle threshold enhances accuracy but also results in longer computation times and a higher count of output faces. The Structural Segment Generation algorithm performs well with noisy inputs. However, due to its reliance on spatial discretization for visibility analysis, it overlooks finer structures, as shown in Fig. 20. Even increasing voxel density can help, the computational cost is correspondingly increased. Especially for large-scale scenes, such computational increase for both segment generation and co-analysis is inevitable, as observed in Fig. 21. The nonlinear growth stems from the co-analysis stage, mainly due to solving the optimization equation Eq. 2. Fortunately, Co-LOD for single building provides an potential solution to address this nonlinear growth issue. The computationally expensive co-analysis can be accomplished through pre-training on large-scale databases, enabling independent control over the LOD generation of each building.

5 CONCLUSIONS

In this paper, we introduce Co-LOD, a novel method for controllable LOD generation with semantic consistency. It signifies pioneering efforts in data-driven LOD generation for urban scenes, aimed at
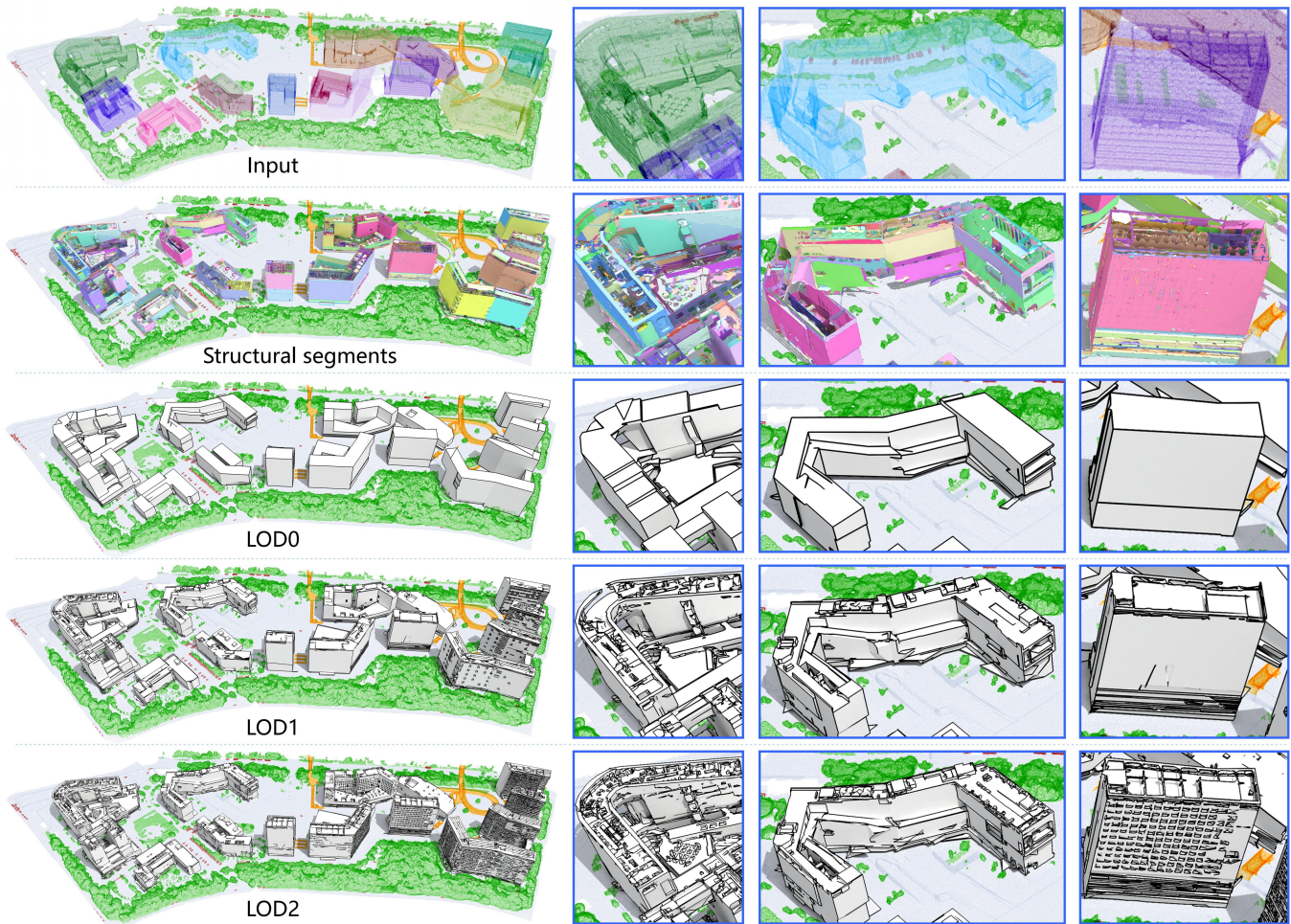
Fig. 23. LOD generation results by Co-LOD for *Research Center*.

addressing the challenges faced by deep learning solutions in precise reconstruction and LOD control under the current insufficiency of data. The process begins with a robust segmentation method to extract fundamental structural units from architectural models. Subsequently, Co-LOD employs joint structural analysis to facilitate progressive LOD generation. This involves a detailed comparison of segment-based similarities, establishing precise constraints that guarantee semantic consistency across various models. Our experiments validate that Co-LOD uniquely achieves LOD generation with significant semantic consistency. Looking ahead, we plan to expand Co-LOD's capabilities by integrating a generative framework to enhance its completion ability. Additionally, we aim to develop multi-resolution texture mapping for different LOD layers, which will contribute to more realistic rendering of architectural models.

ACKNOWLEDGMENTS

REFERENCES

Jean-Philippe Bauchet and Florent Lafarge. 2020. Kinetic shape reconstruction. *ACM Trans. on Graphics* 39, 5 (2020), 156:1–156:14.

Fausto Bernardini and Chandrajit Bajaj. 1997. Sampling and reconstructing manifolds using alpha-shapes. In *Proc. Canadian Conf. on Computational Geometry*.

Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. 2010. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*. 1261–1268.

Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3 (2003), 223–232.

Meida Chen, Qingyong Hu, Zifan Yu, Hugues Thomas, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. 2022. Stpls3d: a large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. *arXiv preprint arXiv:2203.09065* (2022).

Zhen Chen, Zherong Pan, Kui Wu, Etienne Vouga, and Xifeng Gao. 2023. Robust Low-poly meshing for general 3D models. *ACM Trans. on Graphics* 42, 4 (2023), 119:1–1119:20.

Ilke Demir, Daniel G. Aliaga, and Bedrich Benes. 2015a. Procedural editing of 3D building point clouds. In *Proc. Int. Conf. on Computer Vision*. 2147–2155.

undefinedlke Demir, Daniel G. Aliaga, and Bedrich Benes. 2015b. Coupled segmentation and similarity detection for architectural models. *ACM Trans. on Graphics* 34, 4 (2015), 104:1–l04:11.

Andreas Fabri, Geert-Jan Giezeman, Lutz Kettner, Stefan Schirra, and Sven Schönherr. 2000. On the design of CGAL a computational geometry algorithms library. *Software: Practice and Experience* 30, 11 (2000), 1167–1202.

Hao Fang and Florent Lafarge. 2020. Connect-and-Slice: an hybrid approach for reconstructing 3D objects. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*. 13490–13498.

Hao Fang, Florent Lafarge, and Mathieu Desbrun. 2018. Planar shape detection at structural scales. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*. 2965–2973.

Henry Fuchs, Zvi M. Kedem, and Bruce F. Naylor. 1980. On visible surface generation by a priori tree structures. *Proc. SIGGRAPH* 14, 3 (1980), 124–133.

Weixiao Gao, Liangliang Nan, Bas Boom, and Hugo Ledoux. 2021. SUM: a benchmark dataset of semantic urban meshes. *ISPRS J. Photogrammetry and Remote Sensing* 179 (2021), 108–120.

Xifeng Gao, Kui Wu, and Zherong Pan. 2022. Low-poly mesh generation for building models. In *Proc. SIGGRAPH*. 3:1–3:9.

Michael Garland and Paul S Heckbert. 1997. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH*. 209–216.

Aleksey Golovinskiy and Thomas Funkhouser. 2008. Randomized cuts for 3D mesh analysis. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 27, 5 (dec 2008), 145:1–145:12.

Aleksey Golovinskiy and Thomas Funkhouser. 2009. Consistent segmentation of 3D models. *Computers & Graphics* 33, 3 (2009), 262–269.

Gerhard Gröger and Lutz Plümer. 2012. CityGML – Interoperable semantic 3D city models. *ISPRS J. Photogrammetry and Remote Sensing* 71 (2012), 12–33.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Hugues Hoppe. 1996. Progressive meshes. In *Proc. SIGGRAPH*. 99–108.

Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. 2022. Sensaturban: learning semantics from urban-scale photogrammetric point clouds. *Int. J. Computer Vision* 130, 2 (2022), 316–343.

Ruizhen Hu, Lubin Fan, and Ligang Liu. 2012. Co-segmentation of 3D shapes via subspace clustering. *Computer Graphics Forum* 31, 5 (2012), 1703–1713.

Jingwei Huang, Shanshan Zhang, Bo Duan, Yanfeng Zhang, Xiaoyang Guo, Mingwei Sun, and Li Yi. 2023. ArrangementNet: learning scene arrangements for vectorized indoor scene modeling. *ACM Trans. on Graphics* 42, 4 (2023), 1–15.

Qixing Huang, Vladlen Koltun, and Leonidas Guibas. 2011. Joint shape segmentation with linear programming. In *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*. 1–12.

Sagi Katz and Ayellet Tal. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. on Graphics* 22, 3 (2003), 954–961.

Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. 2004. Shape matching and anisotropy. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 23, 3 (2004), 623–629.

Tom Kelly, John Femiani, Peter Wonka, and Niloy J Mitra. 2017. BigSUR: large-scale structured urban reconstruction. *ACM Trans. on Graphics* 36, 6 (2017), 1–16.

Nikolay Kobyshev, Hayko Riemenschneider, Andras Bodis-Szomoru, and Luc Van Gool. 2016. Architectural decomposition for 3D landmark building understanding. In *IEEE Winter Conf. on Applications of Computer Vision*. 1–10.

Florent Lafarge and Pierre Alliez. 2013. Surface reconstruction through point set structuring. *Computer Graphics Forum* 32, 2pt2 (2013), 225–234.

Minglei Li and Liangliang Nan. 2021. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS J. Photogrammetry and Remote Sensing* 173 (2021), 135–150.

Minglei Li, Peter Wonka, and Liangliang Nan. 2016. Manhattan-world urban reconstruction from point clouds. In *Proc. Euro. Conf. on Computer Vision*. 54–69.

Hui Lin, Jizhou Gao, Yu Zhou, Guiliang Lu, Mao Ye, Chenxi Zhang, Ligang Liu, and Ruigang Yang. 2013. Semantic decomposition and reconstruction of residential scenes from liDAR data. *ACM Trans. on Graphics* 32, 4 (2013), 66:1–66:10.

Peter Lindstrom. 2000. Out-of-core simplification of large polygonal models. In *Proc. SIGGRAPH*. 259–262.

Peter Lindstrom and Greg Turk. 2000. Image-driven simplification. *ACM Trans. on Graphics* 19, 3 (2000), 204–241.

William E Lorensen and Harvey E Cline. 1987. Marching cubes: a high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH*. 163–169.

Anđelo Martinović, Jan Knopp, Hayko Riemenschneider, and Luc Van Gool. 2015. 3D all the way: semantic segmentation of urban scenes from start to end in 3D. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*. 4456–4465.

Ravish Mehra, Qingnan Zhou, Jeremy Long, Alla Sheffer, Amy Gooch, and Niloy J Mitra. 2009. Abstraction of man-made shapes. *ACM Trans. on Graphics* 28, 5 (2009), 137:1–137:10.

Liangliang Nan and Peter Wonka. 2017. Polyfit: polygonal surface reconstruction from point clouds. In *Proc. Int. Conf. on Computer Vision*. 2353–2361.

Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. 2002. Shape distributions. *ACM Trans. on Graphics* 21, 4 (2002), 807–832.

Alex Pothen, Horst D. Simon, and Kang-Pu Liou. 1990. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Analysis and Applications* 11, 3 (1990), 430–452.

Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann. 2006. Segmentation of point clouds using smoothness constraint. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36, 5 (2006), 248–253.

Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. 2021. BuildingNet: learning to label 3D buildings. In *Proc. Int. Conf. on Computer Vision*. 10397–10407.

Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer* 24, 4 (2008), 249–259.

Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 30, 6 (2011), 126:1–126:10.

Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. 2022. Variable bitrate neural fields. In *Proc. SIGGRAPH*. 41:1–41:9.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. 2021. Neural geometric level of detail: real-time rendering with implicit 3D shapes. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*. 11358–11367.

Weikai Tan, Nannan Qin, Lingfei Ma, Ying Li, Jing Du, Guorong Cai, Ke Yang, and Jonathan Li. 2020. Toronto-3D: a large-scale mobile lidar dataset for semantic segmentation of urban roadways. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*. 202–203.

Alexander Toshev, Philippos Mordohai, and Ben Taskar. 2010. Detecting and parsing architecture at city scale from range data. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*. 398–405.

Yannick Verdie, Florent Lafarge, and Pierre Alliez. 2015. LOD generation for urban scenes. *ACM Trans. on Graphics* 34, 3 (2015), 30:1–30:14.

Yunhai Wang, Shmulik Asafi, Oliver van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2012. Active co-analysis of a set of shapes. *ACM Trans. on Graphics* 31, 6 (2012), 165:1–165:10.

AH Xu, Jiazhou Chen, Shufang Lu, Ronghua Liang, and Liangliang Nan. 2021. 3D instance segmentation of MVS buildings. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, Vol. 60. 1–14.

Kai Xu, Honghua Li, Hao Zhang, Daniel Cohen-Or, Yueshan Xiong, and Zhiquan Cheng. 2010. Style-content separation by anisotropic part scales. *ACM Trans. on Graphics* 29, 6 (2010), 184:1–184:10.

Guoqing Yang, Fuyou Xue, Qi Zhang, Ke Xie, Chi-Wing Fu, and Hui Huang. 2023. UrbanBIS: a large-scale benchmark for fine-grained urban building instance segmentation. In *Proc. SIGGRAPH*. 16:1–16:11.

Mehmet Ersin Yumer and Levent Burak Kara. 2012. Co-abstraction of shape collections. *ACM Trans. on Graphics* 31, 6 (2012), 1–11.