RESEARCH ARTICLE

MorphCut: An Efficient Convex Decomposition Method of 3D Building Models for Urban Morphological Analytics

Yijie Wu[a, d, e], Fan Xue[a,b*], Liangliang Nan[c], Longyong Wu[a], Jantien Stoter[c], and Anthony G. O. Yeh[a,b]

[a]Faculty of Architecture, The University of Hong Kong, Pokfulam, Hong Kong, China;
[b]National Center of Technology Innovation for Digital Construction Hong Kong Branch, The University of Hong Kong, Pokfulam, Hong Kong, China;
[c]Urban Data Science, Delft University of Technology, 2628 BL, Delft, The Netherlands;
[d]JC STEM Lab of Earth Observations, Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China;
[e]Research Centre for Artificial Intelligence in Geomatics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China.

ABSTRACT
Urban morphological analytics on buildings is informative for sustainable development. 3D building massing features, such as courtyards and setbacks, reflect spatial organizations and circulations, while influence daylight access, ventilation, and shading. However, existing 3D GIS methods usually overlook such 3D massing features, further obscure morphological analytics and environmental assessment. This paper proposes MorphCut, an efficient convex decomposition method that segments 3D shapes into mass-aligned parts. MorphCut leverages key morphological properties—planarity, regularity, and Gestalt laws—after a topological preprocessing step to enable mass-aware decomposition. Experiments on representative samples, ranging from small houses to complex skyscrapers, showed that MorphCut outperformed four baseline methods in (i) balancing convexity and compactness, (ii) aligning decomposed parts with building masses, and (iii) preserving geometric fidelity (average deviation: 0.25 m). An urban-scale validation on datasets from Delft and Hong Kong, comprising over 30,000 buildings across 18.3 km^2, demonstrated MorphCut's robustness, scalability, and generalizability. MorphCut successfully decomposed 98% of buildings in low-rise regions (+78% over the second-best method) and 93% in high-rise areas (+2%), completing processing in 13 hours (3 hours faster). These results position MorphCut as a foundational 3D GIS tool for large-scale, mass-aware morphological analysis, with implications for digital twins, sustainable planning, and environmental modeling.

KEYWORDS
3D urban GIS; building morphology; shape decomposition; 3D building model; branch and bound

1. Introduction

Urban morphology plays a key role in developing sustainable cities and communities, aligning with the United Nations' Sustainable Development Goal No. 11 (UN, 2023). Within this context, buildings serve as critical components in urban morphological

*Corresponding author. Tel: +852 3917 4174, Fax: +852 2559 9457, Email: xuef@hku.hk

studies (Ratti, Frenchman, Pulselli, & Williams, 2006). Extensive building analytics have been developed to address various urban sustainability challenges, including energy consumption (Francisco, Mohammadi, & Taylor, 2020), microclimate simulation (Y. Yan, Wang, & Qu, 2024), environmental assessment (Li, Xue, Wu, & Yeh, 2022; Meng, Su, Sun, Li, & Xue, 2025), and risk assessment (Lu, Gu, Xu, Xiong, & Tian, 2020).

Despite the increasing availability of 3D building models, current urban morphological analytics using 3D GIS face difficulties in handling concave geometries. Most approaches either simplify buildings into prismatic volumes or reduce them to overall geometric indicators (Labetski, Vitalis, Biljecki, Arroyo Ohori, & Stoter, 2023), thereby overlooking architectural complexities in massing such as setbacks, courtyards, and facade protrusions. As shown in Fig. 1, these volumetric arrangements are not merely architectural embellishments but essential indicators of functionality and urban performance (Ndiaye, 2018). However, existing methods often assign similar descriptors to buildings with different massing, or fail to recognize similarity when patterns are alike but differ in scale, thus obscuring key massing information. Addressing these challenges requires new analytical tools capable of decomposing and interpreting building massing into meaningful and manageable geometric components.

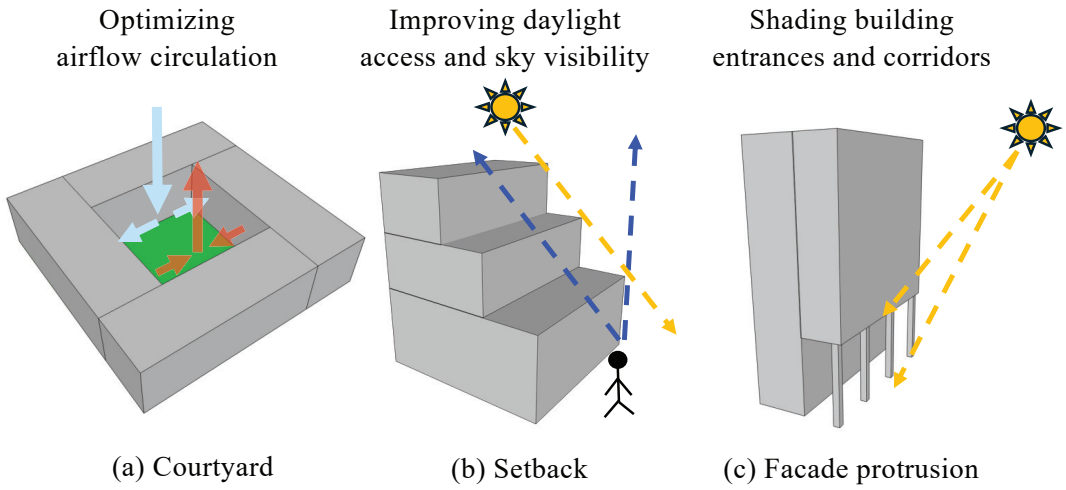


Figure 1. Three examples showing the impact of building massing on urban environment.

Convex decomposition is a fundamental technique in geometry processing and computer graphics, aiming to partition concave shapes into simpler convex or approximately convex parts (Chazelle, Dobkin, Shouraboura, & Tal, 1995). For buildings, the resulting convex parts and their topological relationships provide valuable data for interpreting massing compositions in urban morphological analytics. However, most existing methods were developed for organic or articulated objects, such as animals, furniture, or mechanical assemblies (J. Liu, Savva, & Mahdavi-Amiri, 2024). Given the distinct nature of building geometry, these methods often fail to identify precise cutting planes or to achieve scalable optimal solutions (see Section 2.2). This study therefore examines the morphological properties of buildings to design decomposition methods tailored to architectural forms.

Unlike general shapes, most buildings share three morphological properties. First, they consist primarily of planar surfaces (Ching, 2023), which enables the precise detection of concave edges. Second, building massing follows structured patterns shaped

by functional, performance, and aesthetic requirements, from which potential cutting planes can be derived. Third, architectural design often reflects Gestalt principles, with hierarchical or sequential arrangements (Nan et al., 2011; Xue, Lu, Chen, & Webster, 2020). These massing patterns link the number of potential parts to concave edges, inspiring heuristic approaches for building-specific convex decomposition.

We propose a novel method for the convex decomposition of 3D building models that leverages these morphological properties of buildings. We name this method *MorphCut*, highlighting its goal of segmenting buildings into convex parts for urban morphological analytics. The pipeline begins with topological correction to enable reliable concave edge detection and mesh cutting, then formulates the decomposition problem by defining a solution space and an optimization objective based on geometric compactness and convexity. To solve this, MorphCut applies best-first branch and bound (BBnB) search guided by Gestalt principles, balancing convexity and compactness. A level-of-detail (LoD) scheme is also integrated to accelerate computation. Comprehensive experiments on 24 representative buildings and urban-scale datasets (over 30,000 buildings) demonstrate that MorphCut outperforms four state-of-the-art baselines in effectiveness, efficiency, scalability, and generalizability.

Beyond its technical innovation, MorphCut offers strong practical value. By decomposing 3D building models into convex components aligned with architectural masses (e.g., cores, wings, and setbacks), it enables form-aware digital twins, supports environmental simulations, and enhances city-scale semantic mapping. The resulting components can also be represented as graphs to study morphological patterns and urban impacts. Furthermore, convex decomposition reduces geometric complexity, enabling scalable parallel processing for 3D reconstruction (Bauchet & Lafarge, 2020; Nan & Wonka, 2017), map generalization (Burghardt & Cecconi, 2007), indoor mapping (Wu et al., 2021), and path planning for drone-based reality capture (Y. Liu et al., 2022).

In summary, our main contributions are as follows.

- A novel 3D GIS tool for urban morphological analytics based on convex decomposition, providing new possibilities for analyzing building morphology at urban scales and assessing its impact on urban environment.
- A robust preprocessing procedure for resolving topological defects in 3D building models, ensuring accurate mesh operations for convex decomposition.
- An efficient MorphCut method that produces optimal or near-optimal convex decomposition solutions.

2. Literature Review

2.1. 3D building models and urban morphological analytics

Advances in reality capture and 3D reconstruction have made 3D building models, across various levels of detail, increasingly available at urban scales (Biljecki, Stoter, Ledoux, Zlatanova, & Çöltekin, 2015). These models are usually represented by their boundaries and typically formatted as triangle meshes, polygon meshes, or CityGML scheme databases (Arroyo Ohori, Biljecki, Kumar, Ledoux, & Stoter, 2018). However, such datasets do not guarantee topological consistency, closure, or watertight manifoldness. Self-intersections, open bottoms, or subtle coordinate rounding errors are common in 3D building models, necessitating additional geometry processing to enable topologically robust shape analytics.

At the same time, urban morphological analytics using 3D building geometry is

still in its early stages (Batty, 2013). Current applications are mostly limited to visualization and semantic attributes. Progress is constrained by the lack of advanced shape analysis methods, and existing analytics often rely on oversimplified 3D models, despite the substantial differences between low- and high-LoD models (Biljecki, Ledoux, Stoter, & Vosselman, 2016). Most approaches compute geometric metrics–such as height, volume, surface roughness, sky-view factor, and convexity–for entire buildings and then combine these values as features for classification or clustering (Basaraner & Cetinkaya, 2017). This process may yield similar features for buildings with distinctly different massing, thereby reducing classification accuracy. Recent learning-based methods (X. Yan, Ai, Yang, & Tong, 2021; D. Yu, Hu, Li, & Zhao, 2024) attempt to analyze patterns, but either lack interpretability or depend on predefined forms.

2.2. Convex decomposition for 3D building models

Approximate convex decomposition (ACD), a commonly used technique for shape decomposition, seeks to divide a non-convex shape into a minimal number of highly convex parts, defined by the volume ratio of a part p to its convex hull (see Eqn. C.2 in the Appendix) (Lien & Amato, 2007). Strategies include cutting along concave edges (Lien & Amato, 2004; Thul, Ladicky, Jeong, & Pollefeys, 2018) or merging mesh triangles until concavity errors exceed a threshold (Mamou & Ghorbel, 2009). Open-source methods such as VHACD (Mamou, Lengyel, & Peters, 2016) and COACD (Wei, Liu, Ling, & Su, 2022) employ axis-aligned cutting planes to accelerate determining the optimal cutting sequence. However, existing methods face challenges with 3D building decomposition, which requires precise cutting planes beyond the capabilities of generic shape algorithms (see Fig. C.3 in the Appendix). Moreover, they typically guarantee only one-step (Mamou et al., 2016) or multi-step (Wei et al., 2022) optimality, often producing sub-optimal results when applied to complex buildings.

Low-poly building reconstruction through space partitioning is closely related to convex decomposition, as partitioned convex cells form the basis for interior-exterior labeling (Bauchet & Lafarge, 2020; Nan & Wonka, 2017). Adaptive space partitioning (Chen, Ledoux, Khademi, & Nan, 2022) and kinetic plane extension (Bauchet & Lafarge, 2020) effectively control cell numbers while maintaining topological consistency. In COMPOD (Sulzer & Lafarge, 2025), adjacent interior cells are merged to produce concise building decompositions; however the local greedy strategy does not ensure optimal convexity, compactness, or alignment with building masses.

Advances in foundation models of computer vision (CV) demonstrate remarkable potential for shape segmentation. Projecting 2D predictions from general models such as SAM (Kirillov et al., 2023) onto 3D shapes has enabled promising progress in part segmentation under limited supervision. Leading methods like PartSlip (M. Liu et al., 2023) and SAMPart3D (Yang et al., 2024) achieve low-shot or even zero-shot segmentation on textured 3D objects. Nevertheless, their performance deteriorates on textureless building models, where SAM produces unstable decompositions, thereby highlighting the gap in robust zero-shot segmentation for buildings.

In summary, existing ACD algorithms lack precise cuts that guarantee optimality, efficiency, and topological robustness. Spatial partitioning provides a robust pipeline for handling topological issues but does not optimize part numbers. Foundation model-based methods also remain unstable on textureless building models. In the following section, we elaborate on our proposed approach.

3. Methodology

3.1. Overview

Fig. 2 provides an overview of our approach to the convex decomposition of 3D building models. The method takes a 3D building model in polygonal or triangle mesh format as input and outputs approximate convex parts, each represented as a closed 2D manifold. Before decomposition, the topologically inconsistent input is preprocessed into a piecewise 3D singleton building cell, with concave edges detected (Section 3.2). After preprocessing, the proposed MorphCut method defines the ACD problem for buildings, formalizing its objective and solution space based on the concave edges (Section 3.3.1). MorphCut then solves the problem using a deterministic BBnB algorithm, incorporating with heuristics derived from the common morphological properties of buildings (Section 3.3.2). To further accelerate the BBnB search, MorphCut amplifies the lower bound during the search process and integrates a LoD mechanism to decouple the solution space (Section 3.3.3).

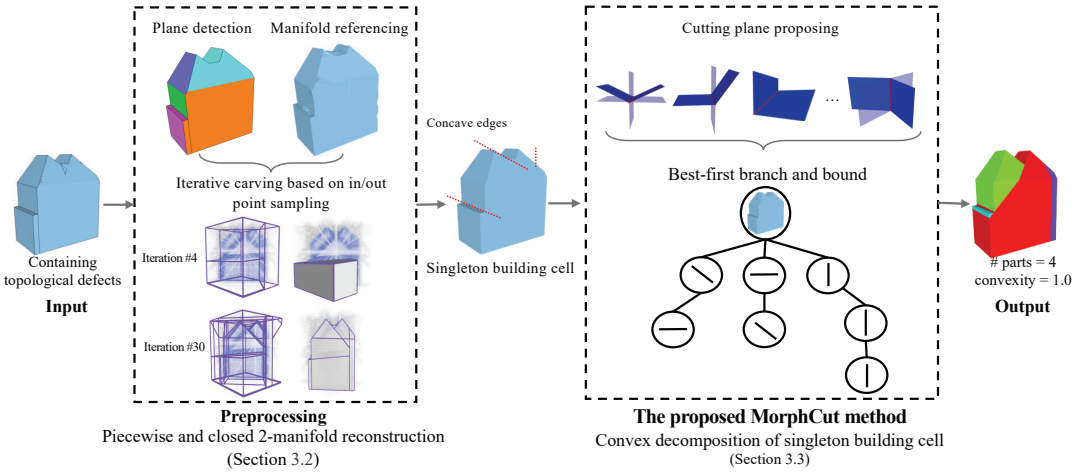


Figure 2. Overview of the preprocessing and the proposed MorphCut method for convex decomposition of 3D building models.

3.2. Preprocessing

The preprocessing procedure reconstructs a piecewise closed 2D manifold from an input mesh model with potential topological errors. This step mitigates the impact of such defects on concave edge detection and convex decomposition. As illustrated in Fig. 3, the iterative space carving shown in the middle represents the core step in preprocessing, supported by two prerequisite steps–plane segmentation (Fig. 3(a)) and point-based space classification (Fig. 3(b))–and followed by two subsequent steps–singleton merging (Fig. 3(d)) and concave edge detection (Fig. 3(e)).

As prerequisites, we first segment planar regions from the input mesh using region growing (Lafarge & Mallet, 2012). Next, point-based space classification determines whether each unit space within the bounding box is interior or exterior to the building. To handle topological defects (e.g., self-intersections), we apply a voxel-based method, ManifoldPlus (Huang, Zhou, & Guibas, 2020), to generate an approximate manifold that serves as a proxy surface for robust classification. From each sampled point, rays

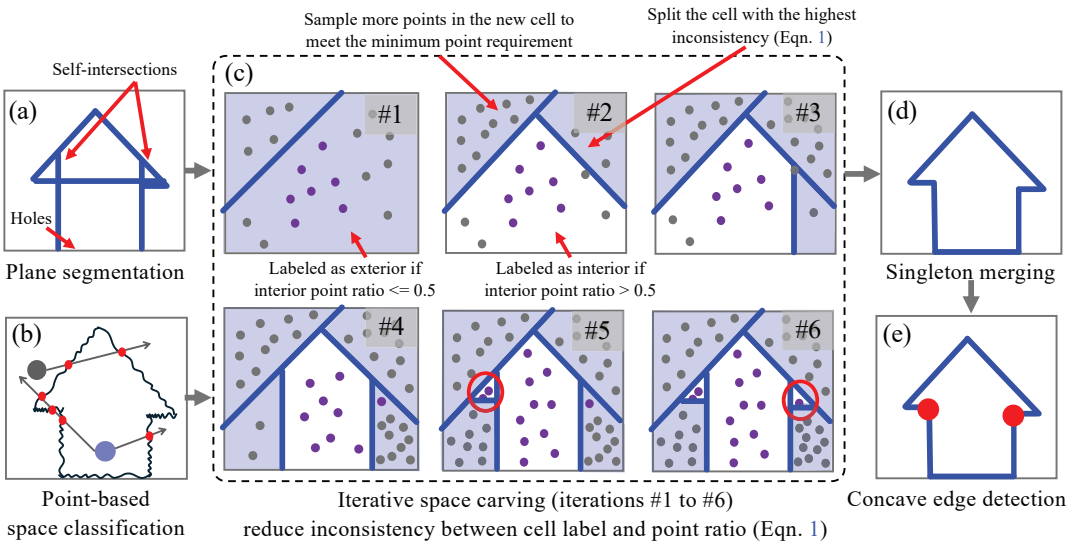


Figure 3. Preprocessing steps. (a) Planar regions are segmented using region growing. (b) A point is labeled as interior (in purple) if its rays intersect the rough manifold surface by an odd number of times; otherwise, it is labeled as exterior (in gray). (c) Iterative space carving, with six iterations shown. Cells are labeled as interior (in white) or exterior (in purple) according to the interior/exterior ratio of their sampled points. This procedure produces a singleton building cell in (d), which removes the interior boundaries of Iteration#6 in (c). (e) Concave edge detection from the singleton building cell. The vertices associated with the detected concave edges in the 2D side view are highlighted in red.

are cast upward; if a ray intersects the mesh an odd number of times, the unit space is classified as interior, and if it intersects an even number of times, the space is classified as exterior.

The iterative carving process extends the point-based space classification to the entire space within the bounding box. This step adopts the same iterative space-partitioning approach as Chen et al. (2022), but the classification of interior and exterior spaces is determined through point voting rather than occupancy learning. The procedure consists of three operations:

(1) **Plane insertion**. The carving begins with the building's bounding box as the initial cell, as shown in Fig. 3(c)#1. The largest planar segments are inserted to split the input cell into two subcells. To avoid unnecessary cuts, a cell is only split by the planar segments that intersect it.

(2) **Point voting**. Classification is performed through random point voting. In the simplified example in Fig. 3(c)#1, the larger cell is classified as exterior because the interior point ratio falls below 0.5. The number of sampled points affects the resolution of space classification. The number of points is determined using a given sampling interval with minimum and maximal bounds.

(3) **Inconsistency-guided iteration**. The iterations aim to reduce inconsistencies between the cell label and ratio obtained from point voting as:

$$\text{Inconsistency}(cell) = \frac{|\{pt | pt \in \mathcal{P}, \text{Label}(pt) \neq \text{Label}(cell)\}|}{|\mathcal{P}|} \tag{1}$$

where $\mathcal{P}$ denotes the point set sampled in the cell. If any cell's inconsistency exceeds a given threshold and planar segments are detected within the cell, the cell undergoes further planar carving in the first operation. For instance, a new split occurs in Fig. 3(c)#2. The iterative procedure terminates when either (i) the point-based voting converges (i.e., all the inconsistencies fall below the threshold) or (ii) no planar segments remain in the cells.

Subsequently, all the interior cells are merged together into a singleton building cell, and the shared boundaries of adjacent interior cells are dissolved. The preprocessing stage then finalizes the procedure by detecting concave edges, i.e., edges where the inner angle formed by two adjacent faces is greater than π. Adjacent and collinear concave edges are further merged.

3.3. The proposed MorphCut method

The proposed MorphCut method identifies a sequence of cutting operations to decompose a singleton 3D building cell into approximate convex parts. Section 3.3.1 formalizes the convex decomposition problem using key morphological properties of buildings. Section 3.3.2 introduces a deterministic BBnB search with a heuristic design for finding optimal solutions. Finally, Section 3.3.3 presents a fast approximation version with two acceleration mechanisms that balance optimality and efficiency.

3.3.1. Problem definition of building convex decomposition

The convex decomposition problem of buildings has two aspects: solution space and objective. The solution space is defined by potential cutting planes, while the objective function seeks to minimize both concavity and the number of resulting parts.

3.3.1.1. Solution space defined by potential cutting planes. Inspired by the regularity of building massing, cutting planes with specific orientations are preferred to ensure a structured decomposition of a building. Fig. 4(a) illustrates four representative cases: (a) vertical cutting planes for separating facade protrusions, (b) horizontal planes for detaching setback blocks, (c) inclined cutting planes for isolating dormer windows, and (d) symmetric joint cutting planes for separating curved blocks. These planes coincide with the concave edges of buildings and are therefore selected as potential cutting planes. The corresponding concave edges are categorized into six types for proposing cutting planes, with details provided in Appendix B.1.

The solution space consists of all possible combinations of potential cutting planes, forming a tree structure as shown in Fig. 5. The root node represents the singleton building cell before cutting, and each node $\mathbf{n}$ corresponds to a combination of cutting planes and the resulting parts. Nodes with remaining potential cuts expand into child nodes by applying those cuts to their parts, whereas nodes without potential cuts are leaves, representing feasible decomposition solutions. Each child node $\mathbf{n}_c$ can be further expanded based on the potential cuts of its parts. Assuming c concave edges and two cutting planes per edge, the total number of possible leaves (without deduplication) is estimated as:

$$2^c \times 2^{c-1} \times 2^{c-2} \cdots \times 2 = 2^{c\times(c+1)/2}. \tag{2}$$

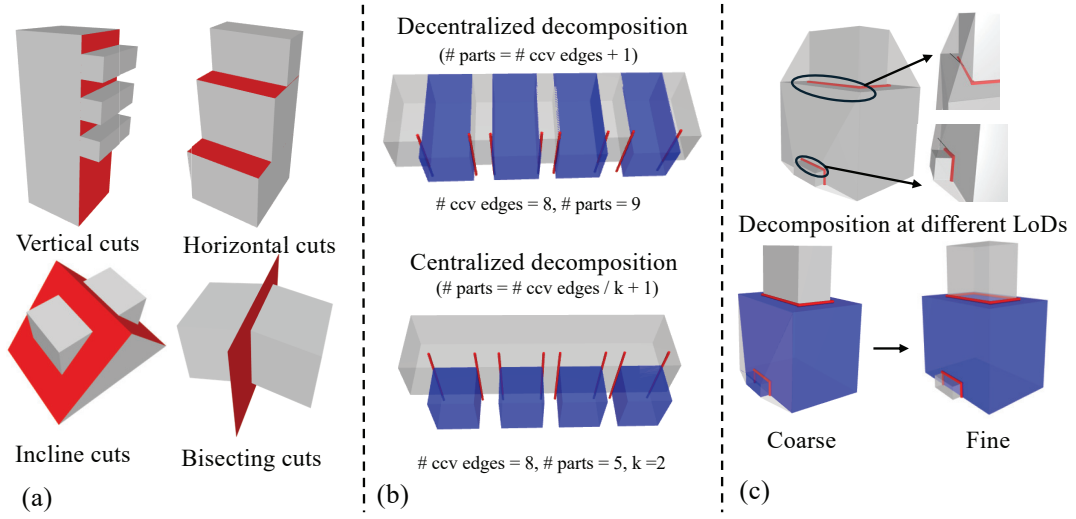


Figure 4. Building morphological properties used by MorphCut to (a) propose potential cutting planes, (b) estimate the number of parts, and (c) guide decomposition from coarse to fine granularity.

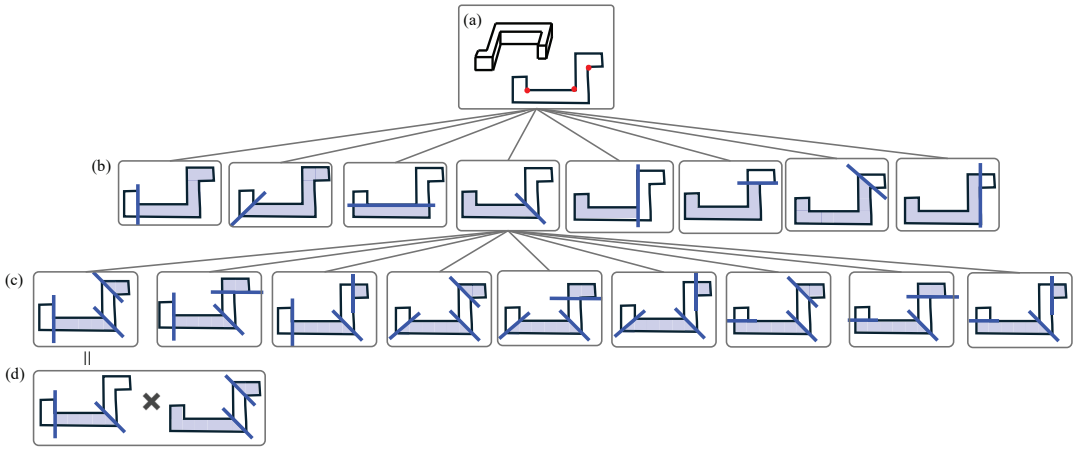


Figure 5. Expanding a part of solution space formed by potential cutting planes as a tree. (a) a given simple (without setbacks and complex roofs) singleton building cell as the root node; (b) expanded child nodes using 8 potential cutting planes; (c) branched nodes of the fourth node in (b); (d) a combination of two parts' potential cutting planes. For clarity in illustration, only a small fraction of the full branching is shown.

3.3.1.2. Objective function for convex decomposition. The goal of building convex decomposition consists of two objectives: (1) maximizing the convexity of decomposed parts and (2) minimizing the number of parts. Accordingly, the objective function $f(\mathbf{n})$ is formulated as:

$$f(\mathbf{n}) = \frac{|\text{Parts}(\mathbf{n})|}{|\text{ConcaveEdges}(\mathbf{r})| + 1} + \left(1 - \frac{\sum_{p \in \text{Parts}(\mathbf{n})} \Delta\text{CH}(p)}{\Delta\text{CH}(\mathbf{r})}\right), \tag{3}$$

where $\text{Parts}(\mathbf{n})$ denotes the decomposed parts at node $\mathbf{n}$. $|\text{ConcaveEdges}(\mathbf{r})|$ denotes the number of concave edges in the root node $\mathbf{r}$. $\Delta\text{CH}(p)$ is the volume difference between the convex hull and the part, i.e., $\text{Volume}(\text{CH}(p)) - \text{Volume}(p)$. The first term

in Eqn. 3 normalizes the number of parts, drawing inspiration from the Gestalt laws in building massing. As shown in Fig. 4(b), decentralized decompositions tend to generate more parts, whereas centralized ones yield fewer. Therefore, $|\text{ConcaveEdges}(\mathbf{r})| + 1$ is used for normalization. In the second term, convexity is measured as the sum of volume differences, normalized by the initial volume difference of $\mathbf{r}$. Note that the complement of the second term indicates the remaining concavity needed to be decomposed.

Along the expansion paths in the solution space, both terms increase. The optimization procedure is to find a leaf $\mathbf{n}_\text{l}$ in the tree with the minimum value of $f(\mathbf{n}_\text{l})$.

3.3.2. Problem solving by best-first branch and bound search

The optimal solution to Eqn. 3 can be found using the classic BBnB search (Land & Doig, 1960), which is deterministic and guarantees optimality. Given a node $\mathbf{n}$ and an objective function $f(\mathbf{n})$ to minimize, BBnB iteratively expands a node, i.e., branches, on the solution tree defined in Section 3.3.1.1. Once a leaf node is reached, we record or update the upper bound B of $f(\mathbf{n})$. Since the optimal solution $f(\mathbf{n}^*)$ will be no greater than B, any branch with a lower bound greater than B is pruned because it cannot lead to a better solution than $\mathbf{n}^*$ (Land & Doig, 1960). The detailed procedure is provided in Algorithm 1 in Appendix A.

BBnB search expands the 'best' node, defined as the one with the lowest $f'(\mathbf{n})$. The objective $f(\mathbf{n})$ defined in Eqn. 3 only reflects only the cost based on the current decomposed parts. A lower current $f(\mathbf{n})$ does not necessarily imply a lower final $f(\mathbf{n})$. Therefore, we redefine an objective function $f'(\mathbf{n})$ for BBnB search as:

$$f'(\mathbf{n}) = f(\mathbf{n}) + h(\mathbf{n}), \tag{4}$$

where $h(\mathbf{n})$ is a heuristic term estimating the future cost to reach termination. To ensure BBnB maintains optimality, $h(\mathbf{n})$ must not overestimate the future cost (Land & Doig, 1960). Similar to the current cost, the future cost contains two components: (1) an estimate of the additional number of parts, and (2) the remaining concavity to reduce.

The second component of $h(\mathbf{n})$ is straightforward to compute: it is the complement of the second term in $f(\mathbf{n})$. However, estimating the additional number of parts for general shapes is challenging. Fortunately, inspired by the Gestalt laws of building perception, we found an effective way to approximate the minimum additional parts required. As shown in Fig. 4(b), a centralized decomposition–where a building is split into a core with attached parts–typically yields the minimum number of parts. The number of the attached parts can usually be approximated as $|\text{ConcaveEdges}(\mathbf{n})|/k$, where k refers to the number of concave edges surrounding an attached part. Accordingly, the heuristic term is formulated as follows:

$$h(\mathbf{n}) = \frac{|\text{ConcaveEdges}(\mathbf{n})|/k}{|\text{ConcaveEdges}(\mathbf{r})| + 1} + \frac{\sum_{p \in \text{Parts}(\mathbf{n})} \Delta\text{CH}(p)}{\Delta\text{CH}(\mathbf{r})}, k > 0. \tag{5}$$

We set k to 4 by default, based on the assumption that typical protrusions, such as cubic balconies or rooftop equipment, are bounded by four concave edges in 3D (see Fig. B.2(d) in the Appendix). While more complex attached parts may require a larger k (Fig. B.2(e) in the Appendix), such cases are relatively uncommon. Parameter analysis of k is discussed in Section 4.3.3.

3.3.3. Acceleration mechanisms balancing optimality and efficiency

Although BBnB search prunes branches in the solution space, it only completes within an acceptable time (several minutes) for relatively simple buildings. Therefore, this section introduces a fast approximation of the BBnB search. The approximate search method is implemented using two acceleration mechanisms on top of the BBnB framework, which trade off between efficiency (lower computational cost) and optimality (fewer parts and higher convexity).

3.3.3.1. Heuristically weighting the lower bound. The first acceleration mechanism heuristically weights the lower bound function. A coefficient ϵ is introduced to emphasize the lower bound $h(\mathbf{n})$ in the cost function. By setting $\epsilon > 1.0$, branching on nodes with relatively high lower bounds is suppressed, i.e.:

$$f'(\mathbf{n}) = f(\mathbf{n}) + \epsilon h(\mathbf{n}), \epsilon > 1.0. \tag{6}$$

This prioritizes branching nodes with smaller $h(\mathbf{n})$–those closer to termination– resulting in earlier termination without exploring as many branches as when $\epsilon = 1.0$. However, when $\epsilon > 1.0$, $h(\mathbf{n})$ may overestimate the future cost towards termination and sacrifice optimality. Therefore, we set ϵ to 2.0 for fast (but sub-optimal) and 1.0 for optimal (but slower) search, by default, based on sensitivity analysis presented in Section 4.3.1.

3.3.3.2. Increased level of detail (LoD). Another acceleration mechanism reduces the number of cutting combinations by structuring the decomposition in terms of different LoDs. For example, at a coarse LoD (denoted as l_1), cutting planes for all potential parts are applied as a complete combination to branch a node. Given the decomposition result Θ at l_1, we search for a finer LoD decomposition for each part in Θ independently, avoiding the need for deeper combinations. This strategy significantly accelerates the search efficiency for detailed decomposition of complex buildings. As shown in Fig. 4(c), to achieve LoD decomposition, we evaluate the significance of a concave edge as follows:

$$\text{Significance}(e) = \frac{\text{Length}(e) \times \text{MaxDist}(e, \text{CH}(\mathbf{n}))}{\text{Area}(\text{DiagPlane}(R))}, \tag{7}$$

where e refers to the concave edge. $\text{MaxDist}(e, \text{CH}(\mathbf{n}))$ represents the maximum distance from e to the convex hull of the current node $\mathbf{n}$. $\text{DiagPlane}(R)$ refers to the vertical diagonal plane of the bounding box of the singleton building cell R. For each LoD, we set a threshold to filter out non-significant edges.

These two acceleration mechanisms are integrated into the fast BBnB search. For the complete procedure, please refer to Algorithm 2 in Appendix A.

4. Experimental tests

4.1. Experimental settings

4.1.1. Test buildings

We selected 24 test buildings from the three public building datasets: City3D (Huang, Stoter, Peters, & Nan, 2022), 3DBIT00 (LandsD, 2024), and NYC3D (DCP, 2018). The selection was based on the representativeness of building morphology, covering diverse styles and levels of shape complexity. The samples range from simple, small houses to highly complex, iconic buildings. The number of planar regions ranges from 16 to 251, while the number of concave edges and potential cutting planes range from fewer than 10 to several hundred. Furthermore, we extended our experiments to the urban scale, including both high-rise and low-rise regions from 3D-BIT00 and City3D, respectively, processing more than 30,000 buildings in total. Additional details on the datasets and selected samples are provided in Appendix C.1.

4.1.2. Key parameters and implementation details

Regarding preprocessing parameters, the depth parameter of ManifoldPlus (Huang et al., 2020) was set to 6. The distance thresholds for region growing and sampling intervals were manually tuned based on the smallest details of the input building model, as discussed in Section 4.3.4. The minimum and maximum sampled points per convex cell in iterative space carving were set to 100 and 10,000, respectively. For the convex decomposition using MorphCut, ϵ in fast BBnB was set to 2.0, and the levels of detail L were set to $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. Additionally, the interior/exterior point classification and node branching were parallelized, as each iteration could involve thousands or more elementary computations. For further details on the method implementation and the computational facilities used, please refer to Appendix C.2 and C.3.

4.1.3. Baseline methods of comparisons

We compared our method with four state-of-the-art approaches: two ACD methods, VHACD (Mamou et al., 2016) and COACD (Wei et al., 2022); one space-partitioning method, COMPOD (Sulzer & Lafarge, 2025), combined with a corresponding high-quality plane detection algorithm PSDR (M. Yu & Lafarge, 2022); and one zero-shot part segmentation based on SAM, SAMPart3D (Yang et al., 2024). For detailed configurations of the baseline methods, please refer to Appendix C.6.

4.1.4. Evaluation metrics

Convexity and part count are the two evaluation metrics for a set of decomposed parts Θ. The average convexity of the decomposed parts is calculated as shown in Eqn. C.1, and the number of parts is normalized. The geometric accuracy of the reconstructed singleton building cells is also evaluated using the Hausdorff distance (HD) between the input building model and the reconstructed closed 2D manifold of the 3D cell. The detailed calculation of the evaluation metrics is provided in Appendix C.5.

4.2. Results

Fig. 6 shows the decomposition results of four LoDs, $L = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, produced by the proposed MorphCut on 24 selected samples. The complete process required an average of 53 seconds, achieving a convexity of 0.99 and a compactness of 0.53 at the finest LoD ($l = 10^{-4}$). In the following subsections, we evaluate our results from four principal perspectives: (i) balance between convexity and compactness, (ii) alignment with building masses, (iii) fidelity to original building shapes, and (iv) efficiency. We also compared our method with VHACD, COACD, COMPOD-PSDR, and SAMPart3D for each perspective.

4.2.1. Balance between convexity and compactness

The primary goal of building decomposition is to segment buildings into as few parts as possible while maximizing convexity at a given LoD, ideally balancing convexity and compactness. Across all LoDs, MorphCut achieved this balance more effectively than baseline methods. As shown in Fig. 7(a), MorphCut consistently outperformed VHACD, COACD, and COMPOD-PSDR, covering a wider LoD range in terms of compactness, whereas COMPOD-PSDR was limited to medium and fine LoDs. Fig. 7(b) further shows that at the finest LoD ($L = 10^{-4}$), MorphCut attained the highest average convexity (i.e., 0.99) with the lowest deviation (i.e., 0.02). SAMPart3D was excluded, as it only supports coarse-level decomposition without reliable control at fine LoDs.

MorphCut's advantage in balancing convexity and compactness highlights the effectiveness of the proposed BBnB search. By estimating the potential optimum of each branch based on Gestalt principles, the algorithm prioritizes promising branches and ensures results remain within a bounded deviation from the global optimum. In contrast, VHACD relies on one-step greedy decisions, COACD employs multi-step Monte Carlo tree search (MCTS), and COMPOD relies on local merging—none of which guarantee for global optimality. SAMPart3D depends on the image-based SAM and is therefore unstable for building models without texture or cases where texture is not a reliable cue for decomposition.

4.2.2. Alignment with building masses

MorphCut decomposes buildings at concave edges using cutting planes derived from planar surfaces, ensuring alignment between parts and building masses. At coarse LoDs (i.e., $l \in \{10^{-1}, 10^{-2}\}$), it isolated the main structural components in Samples (3), (6), (7), (9), (13), (14), (16), and (23), and accurately captured the large setbacks in Samples (15), (18), and (19). At medium LoDs (i.e., $l \in \{10^{-2}, 10^{-3}\}$), MorphCut reliably segmented interlocking elements, such as the pitched roofs in Samples (1) and (3) and the double-ring prisms in Sample (10). At fine LoDs (i.e., $l \in \{10^{-3}, 10^{-4}\}$), it successfully extracted intricate details, including the facade protrusions in Samples (6), (8), (21), and (24); the dormer windows in Samples (2) and (4); the rooftop equipment in Samples (7) and (10); and the successive setbacks in Sample (14).

In contrast, as shown in Fig. 8, the four baseline methods failed to consistently align decomposed parts with the underlying building masses. VHACD and COACD performed comparably to MorphCut at coarse LoDs but lose alignment at medium and fine LoDs, additionally limiting convexity (see Fig. C.3 in the Appendix). COMPOD-PSDR, although generally better aligned than VHACD and COACD, still failed to maintain consistency and shows suboptimal alignment in fine-scale structures at LoD 2

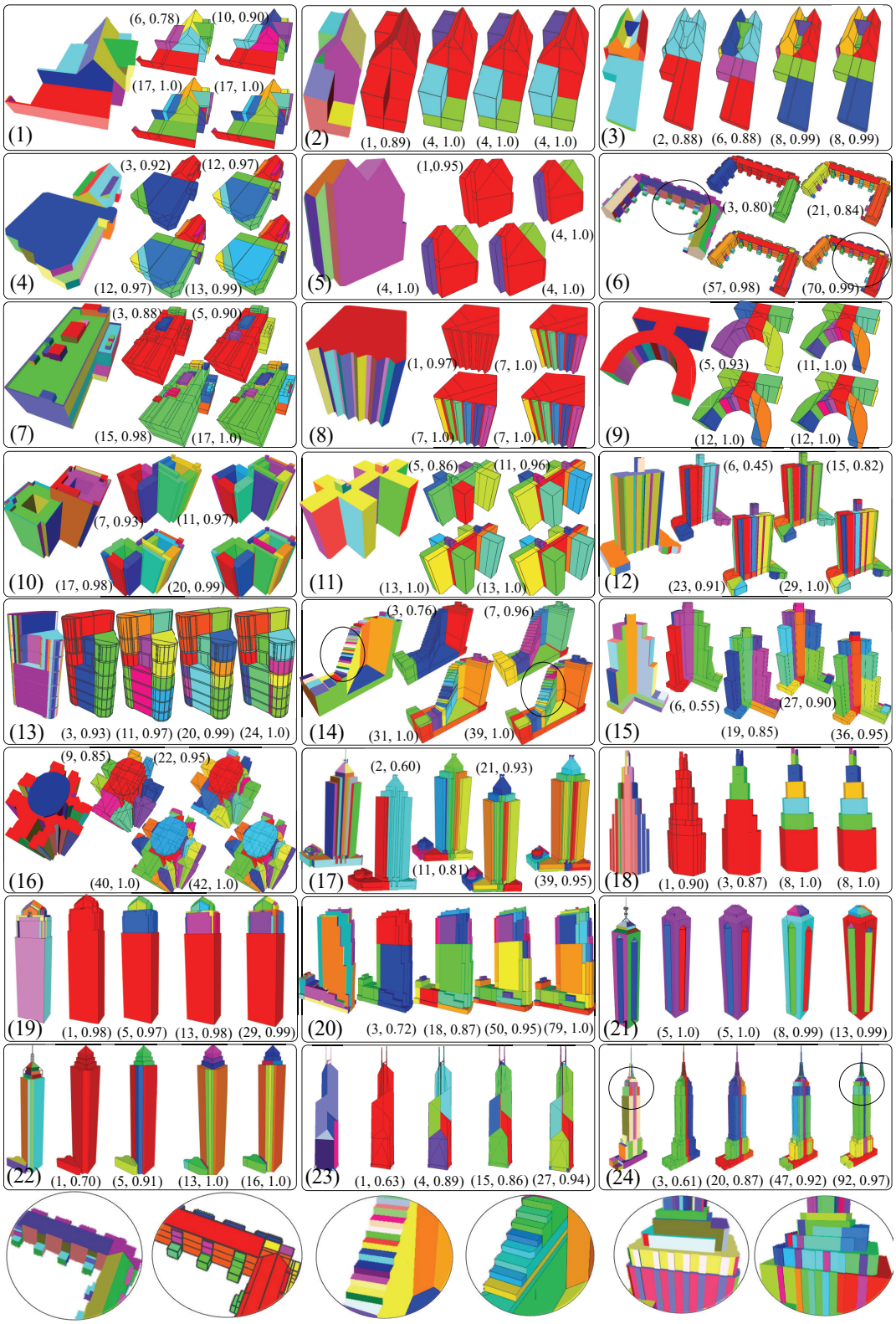


Figure 6. Results of four LoDs in $L = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ using MorphCut on 24 samples. (•, •) indicate the number of decomposed parts and the average convexity, respectively.

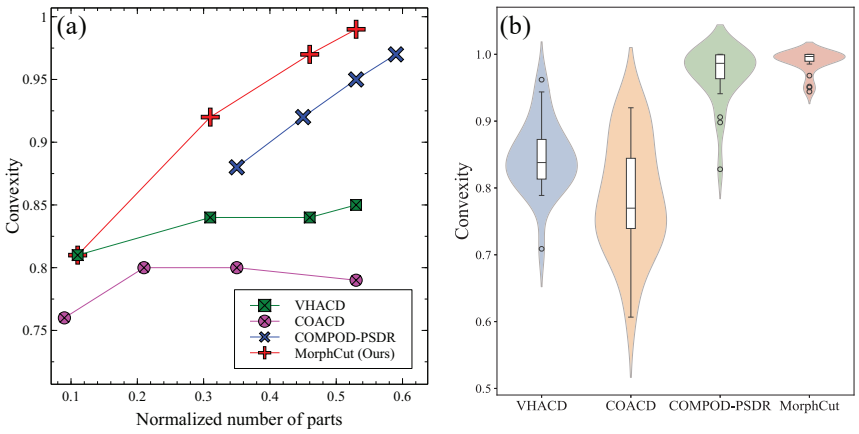


Figure 7. Comparisons between VHACD, COACD, COMPOD-PSDR, and MorphCut in terms of convexity and compactness. (a) Convexity versus compactness across four LoDs. (b) Convexity distribution at the finest LoD ($l = 10^{-4}$). Note: SAMPart3D was excluded from comparisons because it can only perform decomposition at coarse LoDs and does not reliably support finer-level control.

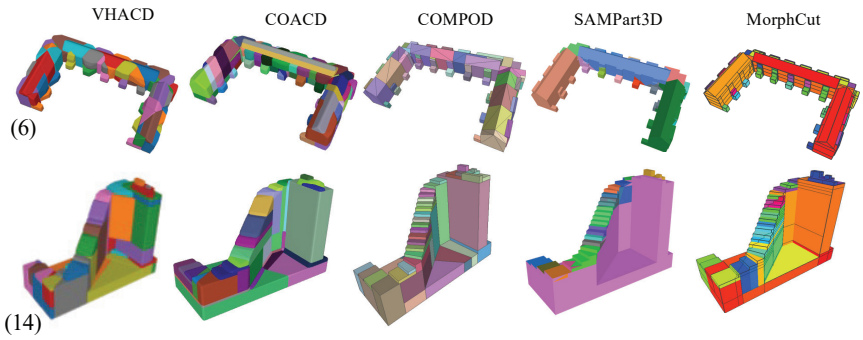


Figure 8. Results of the four methods at the highest LoD for Samples (6) and (14). For additional results, please refer to Appendix C.6.

(see Fig. C.4 in the Appendix). SAMPart3D produced noisy boundaries and unreliable LoD control due to the absence of texture or color cues (see Fig. C.5 in the Appendix).

4.2.3. Fidelity to original building shapes

Fidelity to building shapes is essential for precise building morphology analytics. The decomposed parts by MorphCut preserved the fidelity to the original building shapes. The assembly of these decomposed parts exactly reconstructs the singleton cell during the preprocessing. The proposed preprocessing addresses topological issues, such as open bottom and self-intersection, while maintaining shape fidelity. The average Hausdorff distance between the input building models and reconstructed cells was 0.25 m. Significant geometric deviations occurred only in trivial details omitted during reconstruction (see Fig. C.6 in the Appendix). In contrast, baseline methods exhibited persistent fidelity issues. VHACD and COACD failed to capture the block structures of input building models lacking a bottom. In the meantime, since COMPOD-PSDR relies on consistently outward-facing input normals, inconsistent orientations can result in missing interior parts (see Fig. E.1 in the Appendix).

4.2.4. Runtime efficiency

Runtime efficiency is essential for applying building decomposition at the urban scale. Fig. 9 summarizes preprocessing and decomposition times relative to model complexity. Preprocessing time grew linearly with the number of planar regions (Fig. 9(a)), taking 180 seconds for the most complex case (251 regions), which remains feasible for large-scale applications. Decomposition time, driven by concave edges, was significantly reduced by our BBnB acceleration, lowering complexity from exponential to polynomial (Fig. 9(b)). The small quadratic coefficient indicates near-linear scaling; even the most complex case (402 concave edges, 214 planes) was completed in 222.15 seconds. Fig. 9(c) shows that the proportion of preprocessing versus decomposition time varies with the ratios of regions, concave edges, and planes, which jointly determine overall complexity.

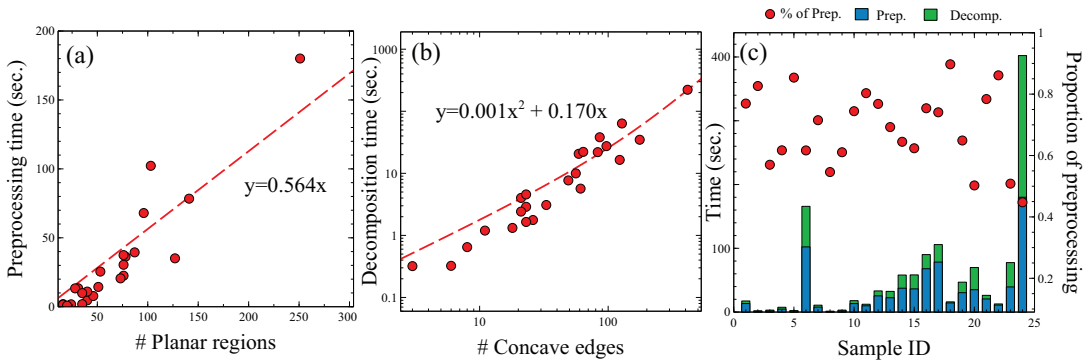


Figure 9. Processing time of preprocessing and decomposition as a function of building complexity.

We compared MorphCut's runtime efficiency with four baseline methods. As shown in Fig. 10(a), MorphCut's average processing time across 24 samples was longer than VHACD and COACD at coarse LoDs, and shorter than COACD but longer than VHACD and COMPOD-PSDR at fine LoDs. Fig. 10(b) shows VHACD and COACD runtimes remained constant at a given LoD, with little correlation to the number of planar regions, and were higher than those of COMPOD-PSDR and MorphCut for simpler samples. By contrast, COMPOD-PSDR and MorphCut scaled with shape complexity, but MorphCut remained highly competitive for low- to medium-complexity models (fewer than 50–100 planar regions), which dominate urban datasets. SAM-Part3D is excluded because it was tested under a different GPU-enabled environment and required 800–1,200 seconds per sample.

In summary, MorphCut decomposes simple buildings more efficiently while delivering superior convexity-compactness trade-offs, better alignment with building masses, and higher shape fidelity—demonstrating strong potential for urban-scale building morphology analytics. Urban-scale validation is presented in Section 4.4.

4.3. Parameter sensitivity analysis

We analyzed four key parameters of our solution in this section: (1) ϵ in $f'(\mathbf{n})$ (Eqn. 4) which weighs the heuristic term; (2) the number of LoD levels in the fast BBnB search (Algorithm 2 in the Appendix A) balancing optimality and efficiency; (3) k in $h(\mathbf{n})$ (Eqn. 5), estimating the number of concave edges surrounding a convex part; and (4) the resolution setting during preprocessing. Detailed results and visualization are

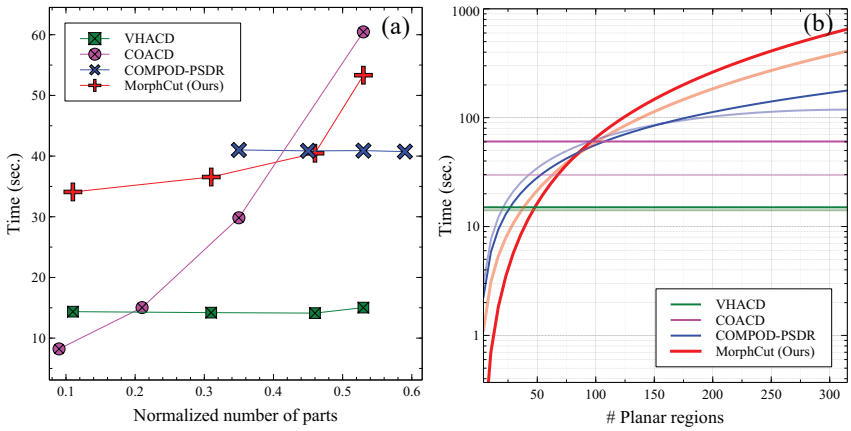


Figure 10. Comparisons between VHACD, COACD, COMPOD-PSDR, and MorphCut on processing time. (a) Average processing time across four LoDs. (b) Processing time at two LoDs relative to the shape complexity of building models measured by the number of planar regions (dark and light colors correspond to $l = 10^{-4}$ and $l = 10^{-3}$, respectively). SAMPart3D is excluded, as it was tested in a different computational environment with a GPU and required significantly more computational time (800 to 1,200 seconds).

provided in Appendix C.7.

4.3.1. ϵ in $f(\mathbf{n})$

We tested our method with the optimal search setting ($\epsilon = 1.0$ with a single LoD $L = \{10^{-4}\}$). Of the 24 samples examined, only the three simplest samples were completed within the one-hour threshold, while the remaining 21 samples exceeded this limit. Both approaches achieved the same level of convexity and number of parts; however, the decomposition time increased drastically with shape complexity.

We further compared MorphCut with $\epsilon = 2.0$ and 4.0, using a single LoD $L = 10^{-2}$ to remove LoD interference. MorphCut failed to complete within 5 minutes for five complex samples at $\epsilon = 2.0$. For the remaining 19 samples, $\epsilon = 2.0$ provided a balance between convexity and part count (see Tab. C.1 and Fig. C.8 in the Appendix). The normalized part count for $\epsilon = 2.0$ was 0.04 lower than $\epsilon = 4.0$, while convexity with $\epsilon = 4.0$ improved only 0.01. Moreover, the efficiency gain from $\epsilon = 4.0$ was marginal compared to $\epsilon = 2.0$ over $\epsilon = 1.0$. Considering convexity, part count, and decomposition time, we set $\epsilon = 2.0$ as the default for the fast BBnB search.

4.3.2. Number of LoDs in fast BBnB

Another key parameter in the fast BBnB search is the number of LoDs. Integrating LoDs can effectively decouple the combination of potential cutting planes, though it may compromise optimality. We tested three different numbers of LoDs, denoted as L. All settings terminate at 10^{-4}. With $|L| = 2$, the method produced a low normalized part count and maintained convexity above 0.99. However, increasing the number of LoDs led to greater irregularity and higher part counts (see Tab. C.2 and Fig. C.9 in the Appendix). Decomposition with $|L| = 2$ also required longer processing times, while efficiency gains diminished as $|L|$ grew. Because branching in the BBnB search is parallelized but the LoD loops in the fast BBnB search are not, excessive LoDs reduce parallelism. This explains why $|L| = 8$ required only slightly more time than $|L| = 4$. Balancing quality and efficiency, we set $|L| = 4$ as the default.

4.3.3. Number of concave edges surrounding a convex part

The number of concave edges surrounding a convex part, denoted as k in $h(\mathbf{n})$ (Eqn. 5), is a predefined parameter used to estimate the minimal number of parts based on the total number of concave edges. We tested values of $k = 2$, 4, and 8. The decomposition time for $k = 8$ was longer than that for $k = 4$ or $k = 2$, and it also resulted in a higher number of decomposed parts (see Tab. C.3 in the Appendix). This parameter analysis indicates that $k = 4$ achieves the most favorable trade-off among convexity, compactness, and computational efficiency.

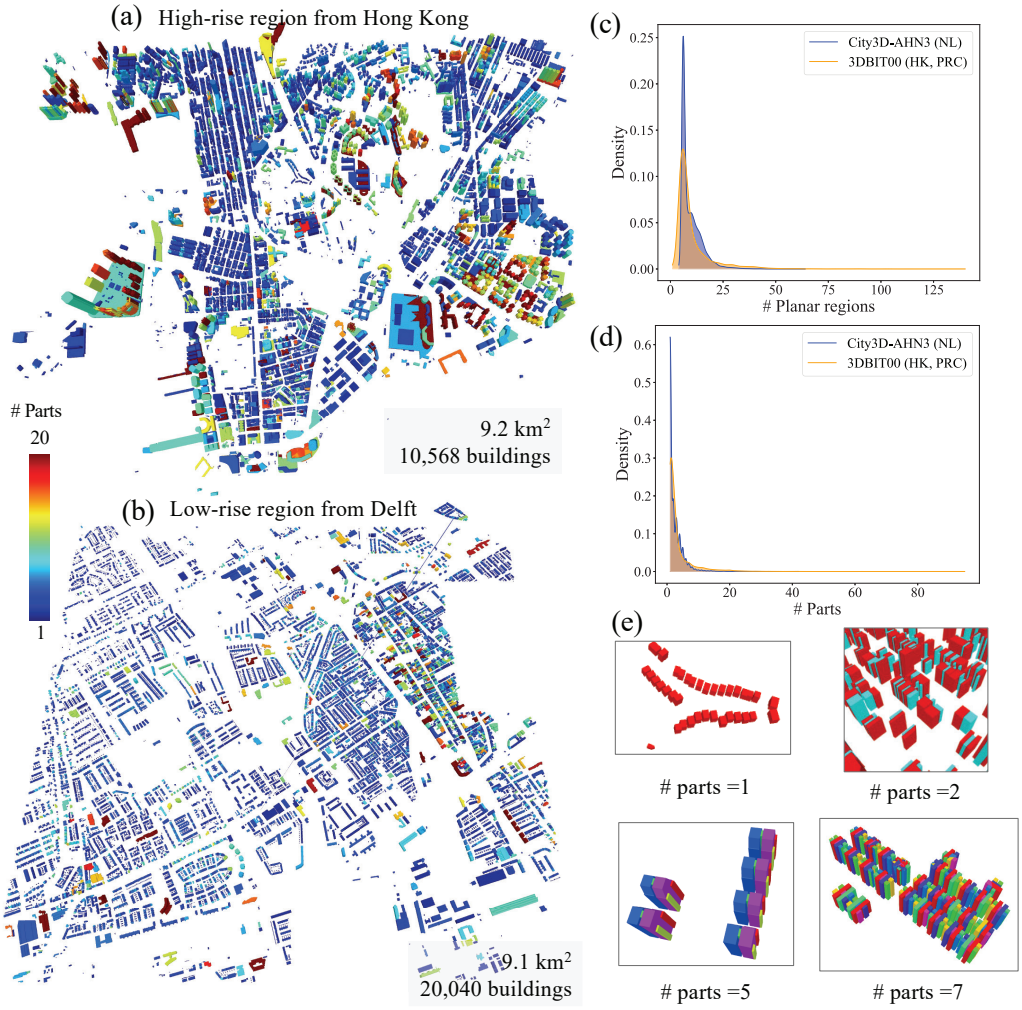
4.3.4. Resolution of preprocessing

The resolution of a singleton building cell during preprocessing is determined by the sampling interval i and the distance threshold d used for planar region growing. The reconstruction resolution is directly controlled by i, with $d < i$ to ensure sufficient planar detail. In Section 4.2, both parameters were manually tuned to preserve the smallest input features. Without manual tuning, i can be set based on the bounding box diagonal, $i = f_{\mathrm{d}} \times \mathrm{Diagonal}(\mathrm{BBox}(\mathrm{sample}))$, with $d = i/4$ by default. Increasing f_{d} from 0.00625 to 0.1 raised the geometric deviation (i.e., HD) from 0.24 m to 2.57 m, where 0.24 m matches the manually tuned result. Finer resolutions reduced deviation but increased preprocessing time, decomposition time, and part count (see Tab. C.4 and Fig. C.10 in the Appendix).

4.4. Urban-scale validation

To assess scalability and generalizability, we tested the approach on two urban datasets: 3DBIT00 and City3D. The 3DBIT00 subset covers Hong Kong's Yau Tsim Mong and Kowloon City districts, characterized by 150 years of dense high-rise development and architectural diversity. The City3D subset represents Delft, the Netherlands, with similarly dense but predominantly low- to mid-rise buildings. The selected regions contain 10,568 buildings in 3DBIT00 and 20,040 in City3D. Parameter settings for the urban-scale validation are provided in Appendix C.8.

As shown in Fig. 11, MorphCut decomposed 98% of City3D buildings and 93% of 3DBIT00 buildings, with failures primarily due to severe topological defects (Section 5.2). Average processing times were 1.01 s for City3D and 2.51 s for 3DBIT00, with convexity close to 1.0. Most buildings were decomposed into fewer than five parts (Fig. 11(c)), and six sample groups illustrate representative urban building patterns (Fig. 11(e)). In the urban-scale comparison, SAMPart3D was prohibitively slow (800–1200 s/sample). VHACD and COACD exhibited stable runtimes but would still require about 5 and 21 days, respectively, to process all buildings at fine LoDs. In contrast, COMPOD-PSDR and MorphCut proved far more efficient, so only COMPOD-PSDR was included for comparison. MorphCut outperformed COMPOD-PSDR on most metrics, particularly success rate (Fig. 11(f)), as COMPOD-PSDR often misses interior parts or fails when inputs contain many inconsistently oriented faces. These results demonstrate MorphCut's superior scalability, generalizability, and robustness.



(f)

	Successful rate ↑		Convexity ↑		# Parts normalized ↓		Time (sec.) ↓	
	City3D	3DBIT00	City3D	3DBIT00	City3D	3DBIT00	City3D	3DBIT00
COMPOD -PSDR	0.20	0.91	0.98± 0.01	0.99± 0.02	**0.57± 0.39**	0.94± 0.57	1.25± 2.71	3.09± 8.29
MorphCut (Ours)	**0.98**	**0.93**	**1.00 ± 0.0**	**1.00 ± 0.0**	0.80± 0.19	**0.86± 0.17**	**1.01± 1.45**	**2.51± 6.39**

(g)

	HD (m) ↓
City3D	0.08 ± 0.14
3DBIT00	0.47± 3.3

Figure 11. Urban-scale validation in Hong Kong and Delft. (a, b) Overview of the two study areas, with buildings colored according to the number of parts decomposed by MorphCut. (c, d) Distribution of the number of parts identified by MorphCut and the planar regions of tested buildings. (e) Zoomed-in views of selected areas in Hong Kong. (f) Quantitative comparison between COMPOD-PSDR and MorphCut, where ↑ indicates higher is better, ↓ indicates lower is better, and the best values are in bold. (g) Average geometric deviation between input meshes and singleton building cells reconstructed by the proposed preprocessing.

5. Discussions

5.1. Applications in urban morphological analytics

Section 4 demonstrates the effectiveness, efficiency, scalability, and generalizability of MorphCut, highlighting its strong potential for application in urban morphological analytics. In a comparison of four baseline methods on 24 representative samples, Mor-

phCut delivers higher-quality results, achieving an advantageous balance between convexity and compactness while maintaining satisfactory alignment with building masses. Meanwhile, MorphCut also achieved geometric deviations below 0.5 m on 24 representative samples, as well as in the City3D and 3DBIT00 tests, meeting the general resolution requirement tested in prior work on 3D morphological indicators (Labetski et al., 2023). The processing time of MorphCut scales with the shape complexity of buildings, which could be longer than that of baseline methods when processing highly complex buildings. The decomposition quality and efficiency were further validated in the urban-scale tests involving more than 30,000 buildings from high-rise and low-rise urban regions in 3DBIT00 and City3D. Overall, MorphCut required approximately 13 hours to successfully decompose most buildings into a small number of convex parts, whereas baseline methods took significantly longer or failed due to topological issues in the inputs.

Furthermore, the optimal or sub-optimal convex decomposition of 3D building models can produce multiple solutions, leading to ambiguity. Various decomposition patterns emerge from this ambiguity. MorphCut can be readily extended to accommodate numerous decomposition patterns by incorporating additional terms for specific patterns in $f(\mathbf{n})$. Details of the formulation and visualization are provided in Appendix D.

5.2. Limitations and future research directions

The proposed approach has several limitations, including the need for manual resolution tuning for optimal preprocessing, occasional sub-optimal results due to acceleration (Fig. E.2), cutting or classification errors (Fig. E.3), limited robustness to severe topological defects and non-manifold geometry (Fig. E.4), long processing time for complex iconic buildings, and the lack of semantic or appearance-based reasoning. Visual details are provided in Appendix E. To address these issues, future work will explore data-driven and learning-based techniques (Park, Florence, Straub, Newcombe, & Lovegrove, 2019) for adaptive resolution selection, decomposition at appropriate LoDs, and the consistent prediction of functionally relevant cuts, even in the presence of interference from exterior-mounted equipment. Such methods also show promise in handling complex topological challenges and enhancing robustness across diverse building geometries for urban morphological analytics.

6. Conclusion

This paper presents MorphCut, a convex decomposition method designed to overcome the limitations of current 3D GIS approaches in capturing 3D building massing features. By segmenting 3D shapes into (near-)convex parts aligned with architectural massing, MorphCut enables mass-aware morphological analysis at both building and urban scales. Experiments on 24 representative buildings and over 30,000 urban-scale models demonstrated that MorphCut outperformed four state-of-the-art baseline methods in convexity, building mass alignment, geometric fidelity, and scalability.

MorphCut contributes to GIScience by integrating 3D shape decomposition with urban morphology analytics, thereby supporting scalable analysis of urban building forms. It also reduces geometry complexity, facilitating urban-scale computation, such as building reconstruction, generalization, and path planning for drones. Limitations include the need for manual resolution tuning, sensitivity to mesh topology, and limited handling of fine-scale architectural details. Future work will explore learning-based

decomposition, enhanced robustness to topological errors, and the integration of semantic cues to further broaden its applicability in real-world 3D urban modeling.

Acknowledgments

The authors sincerely thank the anonymous reviewers and editor for their valuable feedback, which greatly improved this article. Y.W. gratefully acknowledges the support of her postdoctoral supervisor, Prof. Qihao Weng, in finalizing the revision for paper production.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Supplementary materials

The Appendix is available at https://doi.org/10.6084/m9.figshare.28382141.

Data and codes availability statement

The code and data are available at https://doi.org/10.6084/m9.figshare.28382141 and https://github.com/eiiijiiiy/MorphCut.

Additional information

Funding

The work was supported by the Hong Kong Research Grants Council [27200520 and T22-504/21-R], the Natural Science Foundation of Guangdong, China [2023A1515010757], the Research Talent Hub of the Innovation and Technology Fund and the Global STEM Professorship of the Hong Kong SAR Government [P0039329]. The computations of experiments were performed using research computing facilities offered by Information Technology Services, the University of Hong Kong.

Notes on contributors

Yijie Wu is a Postdoctoral Fellow at The Hong Kong Polytechnic University and received her PhD from The University of Hong Kong in July 2025. This study forms part of her PhD thesis and was partially conducted during her visit to TU Delft. Her research interests include 3D geometry processing and urban morphology.

Fan Xue is an Associate Professor with the Department of Real Estate and Construction, The University of Hong Kong. He has an interdisciplinary background in Automation, Computer Science, Systems Engineering, and IT in Construction. His

research interests include building/city informatics, optimization algorithms, urban sensing, and machine learning.

Liangliang Nan received his PhD degree in mechatronics engineering from the Graduate University of the Chinese Academy of Sciences, China, in 2009. He is currently an Associate Professor at the Delft University of Technology, The Netherlands, where he is leading the AI Laboratory on 3D Urban Understanding. His research interests lie at the intersection of computer vision, computer graphics, and 3D geoinformation, with a focus on understanding and modeling real-world scenes.

Longyong Wu is a PhD student in the Department of Real Estate and Construction within the Faculty of Architecture at The University of Hong Kong. His research interests include point cloud processing and 3D modeling.

Jantien Stoter is a full professor and chairs the 3D Geoinformation research group at the Delft University of Technology. She (co) leads/has led many national and international projects on GeoBIM, 3D city modeling and urban applications. She has fulfilled several roles in the Open Geospatial Consortium and the European Spatial Data Research association.

Anthony G.O. Yeh is the Chair Professor in Urban Planning and Geographic Information System and the Chan To Haan Professor in Urban Planning and Design with the Department of Urban Planning and Design, The University of Hong Kong. He is also an Academician with the Chinese Academy of Sciences and the Academy of Social Sciences in the UK.

References

Arroyo Ohori, K., Biljecki, F., Kumar, K., Ledoux, H., & Stoter, J. (2018). Modeling cities and landscapes in 3D with CityGML. *Building Information Modeling: Technology Foundations and Industry Practice*, *09*, 199-215. (doi:10.1007/978-3-319-92862-3_11)

Basaraner, M., & Cetinkaya, S. (2017). Performance of shape indices and classification schemes for characterising perceptual shape complexity of building footprints in gis. *International Journal of Geographical Information Science*, *31*(10), 1952–1977. (doi:10.1080/13658816.2017.1346257)

Batty, M. (2013). *The New Science of Cities*. MIT press.

Bauchet, J.-P., & Lafarge, F. (2020). Kinetic shape reconstruction. *ACM Transactions on Graphics*, *39*(5), 1–14. (doi:10.1145/3376918)

Biljecki, F., Ledoux, H., Stoter, J., & Vosselman, G. (2016). The variants of an LOD of a 3D building model and their influence on spatial analyses. *ISPRS Journal of Photogrammetry and Remote Sensing*, *116*, 42-54. (doi:10.1016/j.isprsjprs.2016.03.003)

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, *4*(4), 2842–2889. (doi:10.3390/ijgi4042842)

Burghardt, D., & Cecconi, A. (2007). Mesh simplification for building typification. *International Journal of Geographical Information Science*, *21*(3), 283–298. (doi:10.1080/13658810600912323)

Chazelle, B., Dobkin, D. P., Shouraboura, N., & Tal, A. (1995). Strategies for polyhedral surface decomposition: An experimental study. In *Proceedings of the 11th Annual Symposium on Computational Geometry* (pp. 297–305). (doi:10.1016/S0925-7721(96)00024-7)

Chen, Z., Ledoux, H., Khademi, S., & Nan, L. (2022). Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS Journal of Photogrammetry and Remote Sensing*, *194*, 58–73. (doi:10.1016/j.isprsjprs.2022.09.017)

Ching, F. D. (2023). *Architecture: Form, Space, and Order (5th ed.)*. John Wiley & Sons.

DCP, N. (2018). *NYC 3D Model by Community District.* `https://www.nyc.gov/site/planning/data-maps/open-data/dwn-nyc-3d-model-download.page`.

Francisco, A., Mohammadi, N., & Taylor, J. E. (2020). Smart city digital twin–enabled energy management: Toward real-time urban building energy benchmarking. *Journal of Management in Engineering*, *36*(2), 04019045. (doi:10.1061/(ASCE)ME.1943-5479.0000741)

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

Huang, J., Zhou, Y., & Guibas, L. (2020). ManifoldPlus: A Robust and Scalable Watertight Manifold Surface Generation Method for Triangle Soups. *arXiv preprint arXiv:2005.11621*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... others (2023). Segment anything. In *Proceedings of the IEEE/CVF international Conference on Computer Vision* (pp. 4015–4026). (doi:10.1109/ICCV51070.2023.00371)

Labetski, A., Vitalis, S., Biljecki, F., Arroyo Ohori, K., & Stoter, J. (2023). 3D building metrics for urban morphology. *International Journal of Geographical Information Science*, *37*(1), 36–67. (doi:10.1080/13658816.2022.2103818)

Lafarge, F., & Mallet, C. (2012). Creating large-scale city models from 3D point clouds: A robust approach with hybrid representation. *International Journal of Computer Vision*, *99*(1), 69–85. (doi:10.1007/s11263-012-0517-8)

Land, A., & Doig, A. (1960). An automatic method of solving discrete programming problems. *Econometrica*, *28*(3), 497–520. (doi:10.2307/1910129)

LandsD, H. (2024). *3D-BIT00.* `https://www.landsd.gov.hk/en/survey-mapping/mapping/3d-mapping.html`.

Li, M., Xue, F., Wu, Y., & Yeh, A. G. O. (2022). A room with a view: Automatic assessment of window views for high-rise high-density areas using City Information Models and deep transfer learning. *Landscape and Urban Planning*, *226*, 104505. (doi:10.1016/j.landurbplan.2022.104505)

Lien, J.-M., & Amato, N. M. (2004). Approximate convex decomposition. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry* (pp. 457–458). (doi:10.1145/997817.997889)

Lien, J.-M., & Amato, N. M. (2007). Approximate convex decomposition of polyhedra. In *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling* (pp. 121–131). (doi:10.1145/1236246.1236265)

Liu, J., Savva, M., & Mahdavi-Amiri, A. (2024). Survey on modeling of articulated objects. *arXiv preprint arXiv:2403.14937*.

Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., & Su, H. (2023). PartSlip: Low-shot part segmentation for 3D point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition* (pp. 21736–21746). (doi:10.1109/CVPR52729.2023.02082)

Liu, Y., Lin, L., Hu, Y., Xie, K., Fu, C.-W., Zhang, H., & Huang, H. (2022). Learning reconstructability for drone aerial path planning. *ACM Transactions on Graphics*, *41*(6), 197:1–197:17. (doi:10.1145/3550454.3555433)

Lu, X., Gu, D., Xu, Z., Xiong, C., & Tian, Y. (2020). CIM-powered multi-hazard simulation framework covering both individual buildings and urban areas. *Sustainability*, *12*(12), 5059. (doi:10.3390/su12125059)

Mamou, K., & Ghorbel, F. (2009). A simple and efficient approach for 3D mesh approximate convex decomposition. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (pp. 3501–3504). (doi:10.1109/ICIP.2009.5414068)

Mamou, K., Lengyel, E., & Peters, A. (2016). Volumetric hierarchical approximate convex decomposition. In *Game Engine Gems.* CRC Press.

Meng, S., Su, X., Sun, G., Li, M., & Xue, F. (2025). From 3D pedestrian networks to wheelable networks: An automatic wheelability assessment method for high-density urban areas using contrastive deep learning of smartphone point clouds. *Computers, Environment and Urban Systems*, *117*, 102255. (doi:10.1016/j.compenvurbsys.2025.102255)

Nan, L., Sharf, A., Xie, K., Wong, T.-T., Deussen, O., Cohen-Or, D., & Chen, B. (2011). Conjoining gestalt rules for abstraction of architectural drawings. *ACM Transactions on Graphics*, *30*(6), 1–10. (doi:10.1145/2070781.2024219)

Nan, L., & Wonka, P. (2017). PolyFit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2353–2361). (doi: 10.1109/ICCV.2017.258)

Ndiaye, D. (2018). The impact of building massing on net-zero achievability for office buildings. In *Building Simulation* (Vol. 11, pp. 435–438). (doi:10.1007/s12273-017-0417-5)

Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019). DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 165–174). (doi:10.1109/CVPR.2019.00025)

Ratti, C., Frenchman, D., Pulselli, R. M., & Williams, S. (2006). Mobile landscapes: Using location data from cell phones for urban analysis. *Environment and Planning B: Planning and Design*, *33*(5), 727-748. (doi:10.1068/b32047)

Sulzer, R., & Lafarge, F. (2025). Concise plane arrangements for low-poly surface and volume modelling. In *European Conference on Computer Vision* (pp. 357–373). (doi:10.1007/978-3-031-72904-$1_2$1)

Thul, D., Ladicky, L., Jeong, S., & Pollefeys, M. (2018). Approximate convex decomposition and transfer for animated meshes. *ACM Transactions on Graphics*, *37*(6), 226. (doi:10.1145/3272127.3275029)

UN. (2023). *The Sustainable Development Goals Report, Special Edition.* `https://unstats.un.org/sdgs/report/2023/The-Sustainable-Development-Goals-Report-2023.pdf`.

Wei, X., Liu, M., Ling, Z., & Su, H. (2022). Approximate convex decomposition for 3D meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics*, *41*(4), 1–18. (doi:10.1145/3528223.3530103)

Wu, Y., Shang, J., Chen, P., Zlatanova, S., Hu, X., & Zhou, Z. (2021). Indoor mapping and modeling by parsing floor plan images. *International Journal of Geographical Information Science*, *35*(6), 1205–1231. (doi:10.1080/13658816.2020.1781130)

Xue, F., Lu, W., Chen, Z., & Webster, C. J. (2020). From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles. *ISPRS Journal of Photogrammetry and Remote Sensing*, *167*, 418–431. (doi:10.1016/j.isprsjprs.2020.07.020)

Yan, X., Ai, T., Yang, M., & Tong, X. (2021). Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *International Journal of Geographical Information Science*, *35*(3), 490–512. (doi:10.1080/13658816.2020.1768260)

Yan, Y., Wang, K., & Qu, X. (2024). Urban air mobility (UAM) and ground transportation integration: A survey. *Frontiers of Engineering Management*, *11*, 734–758. (doi:10.1007/s42524-024-0298-0)

Yang, Y., Huang, Y., Guo, Y.-C., Lu, L., Wu, X., Lam, E. Y., ... Liu, X. (2024). SAMPart3D: Segment any part in 3D objects. *arXiv preprint arXiv:2411.07184*.

Yu, D., Hu, Y., Li, Y., & Zhao, L. (2024). PolygonGNN: Representation learning for polygonal geometries with heterogeneous visibility graph. New York, NY, USA: Association for Computing Machinery. (doi:10.1145/3637528.3671738)

Yu, M., & Lafarge, F. (2022). Finding good configurations of planar primitives in unorganized point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6367–6376). (doi:10.1109/CVPR52688.2022.00626)