2.5D Building Modeling with Topology Control

Qian-Yi Zhou
University of Southern California
qianyizh@usc.edu

Ulrich Neumann
University of Southern California
uneumann@graphics.usc.edu

Abstract

2.5D building reconstruction aims at creating building models composed of complex roofs and vertical walls. In this paper, we define 2.5D building topology as a set of roof features, wall features, and point features; together with the associations between them. Based on this definition, we extend 2.5D dual contouring into a 2.5D modeling method with topology control. Comparing with the previous method, we put less restrictions on the adaptive simplification process. We show results under intense geometry simplifications. Our results preserve significant topology structures while the number of triangles is comparable to that of manually created models or primitive-based models.

1. Introduction

Building reconstruction lies in the heart of urban modeling which is the basis of various applications such as urban planning, virtual city tourism, and computer games. Many research efforts have addressed the problem of creating building models from city scans captured from nadir perspective, which are known as 2.5D LiDAR point clouds in [14]. In particular, most of these approaches tend to produce building models composed of complex roofs and vertical walls. These facts determine the 2.5D characteristic of typical building modeling problem.

Although most previous work focuses on roof pattern extraction and geometry simplification, the key observation we have made here is that human vision tend to be more sensitive to building topology rather than building geometry. Intuitively, building topology determines the existence of structural pieces and the connections between them; while building geometry describes where these structural pieces appear in the three dimensional space. We notice that humans are more aware of changes in topology even if the related structural piece is small. For example, Figure 1(c,d) demonstrate two building models created targeting to achieve more precise geometry and more precise topology respectively. Although the left model fits the input point cloud better under typical geometrical error measurements (*e.g.*, average quadratic distance), it is visually less convincing than the right one because a roof piece (the chimney) is missing.

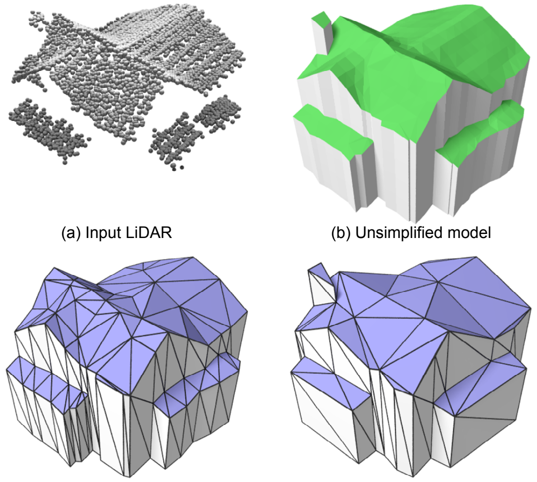

(a) Input LiDAR
(b) Unsimplified model
(c) Geometrically more precise
(d) Topologically more precise

Figure 1. Building models reconstructed targeting to obtain (c) more precise geometry and (d) more precise topology respectively. Compared with (a) the input LiDAR and (b) unsimplified building model, the missing of the chimney makes the former one visually less convincing than the latter one.

The topology issue in 2.5D building modeling is first noticed by Zhou and Neumann [14] and alleviated by introducing a topology test in the 2.5D dual contouring method. Their adaptive simplification process first collapses quadtree cells and optimizes an anchor point completely based on geometric errors without considering building topology; then rewinds the collapse operation if the topology test reveals a possible topology change. This strategy performs well under strong geometric control (*i.e.*, with a small geometry error tolerance). However, in cases where simpler models are desired thus looser geometric control is given, the number of topology test failures increases rapidly and they become the dominant factor in preventing quadtree collapse. Figure 2(a) shows such an example in which topology test frequently detects possible roof layer cracks

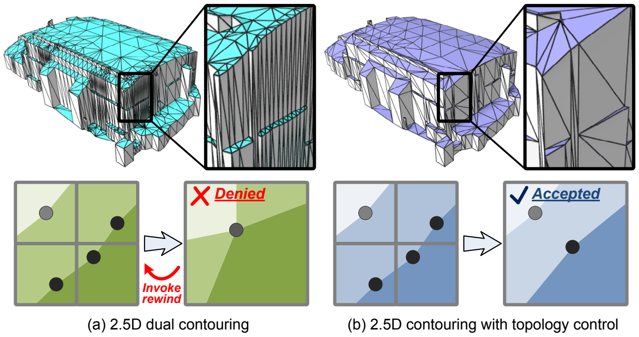


Figure 2. Comparison between (a) 2.5D dual contouring [14] and (b) our contouring method with topology control. While the uniqueness of hyper-point in one cell prevents a flexible simplification in dual contouring, our method detects and controls building topology beyond the rigid quadtree structure.

and denies the cell collapse; therefore, numerous insignificant triangles are produced along the thin long roof features as shown in the closeup. The deep reason behind this problem is that the optimization process is completely unaware of building topology. It produces exact one hyper-point[1] per quadtree cell without discrimination. Hence, the most complicated topological structure that can exist in one cell is a conjunction hyper-point with star-shaped roof boundaries, as shown in Figure 2(a) bottom right. The adaptive simplification becomes problematic in producing building structures with topology that is more complex than a conjunction hyper-point. Collapse rewind is invoked frequently.

We propose an extension to the 2.5D dual contouring method to enable building topology control. The key idea is to maintain multiple hyper-points in one quadtree cell. Therefore complicated in-cell building topology is allowed. With this extension, the adaptive model creation procedure becomes less restrictive, and thus generates simpler building models in a flexible manner, *e.g.*, Figure 2(b). In particular, without changing building topology, our method can produce building models with triangles as few as manually created models or primitive-based models; while it still provides a similar geometric optimization scheme as the data-driven modeling approaches.

Contributions: Given that our method is based on some previous work, we explicitly specify our novelties as follows:

1. We formally define three major topology features in 2.5D building models, namely, point features, wall features, and roof features. We reveal the topological relationships between them and present algorithms to detect and control these features.

2. We propose a novel hyper-point clustering algorithm which allows the existence of multiple building topology features in one quadtree cell. We adapt the geometry optimization and polygon generation methods in 2.5D dual contouring to our topology control scheme. Our contouring method produces 2.5D models with less triangles while preserving the building topology.

2. Related Work

We briefly review 2.5D dual contouring, LiDAR-based building reconstruction algorithms, and volumetric modeling approaches with topology control.

2.1. 2.5D Dual Contouring

The 2.5D dual contouring method proposed by Zhou and Neumann [14] is a robust data-driven approach in creating 2.5D building models from aerial LiDAR. In their work, 2.5D characteristic is defined for the building modeling problem as "reconstructing polygonal models composed of detailed roofs and vertical walls". A 2.5D framework is proposed which utilizes a quadtree as the supporting data structure to store Hermite data scan-converted from aerial LiDAR point clouds. The adaptive simplification is then implemented by collapsing quadtree cells and optimizing a quadratic error function (2.5D QEF) to produce exact one hyper-point in each grid cell. The hyper-point is a 2.5D representation of a series of 3D points that have consistent projections on the x-y plane. They are later connected in two different manners to produce roof polygons and vertical walls respectively.

Although the 2.5D dual contouring method achieves advantages such as robustness and sharp feature production, the lack of considering building topology in the optimization process makes it difficult to efficiently handle diverse topology features, as discussed in Section 1.

2.2. LiDAR-based Building Reconstruction

With the fast development of laser scanning technique, dense aerial LiDAR point clouds become accessible as a valuable data source in urban reconstruction. Recent research work (*e.g.*,[5, 9, 13]) has introduced a LiDAR-based pipeline, which removes unimportant elements then segments individual building point cloud as the input of building reconstruction algorithms.

As building reconstruction is the core problem of urban modeling, there are two typical directions to attack this problem. The first research direction is to create building models from a set of pre-defined structural patterns, such as planes [4, 5, 9, 13], primitives [3, 10], or grammars [7]. These methods focus on reconstructing pre-defined patterns and uncovering the structural relevance between them. However, they usually lack of universality and accuracy when dealing with arbitrarily shaped roofs. *E.g.*, Verma *et*

[1] A hyper-point is defined as a series of 3D points having the same x-y coordinates but different z values [14].

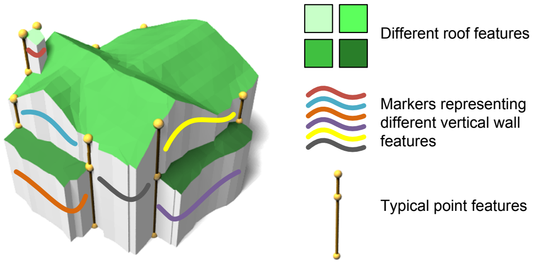


Figure 3. Topological features in an unsimplified 2.5D building model.

al. [9] detects topological relationships only between planar roof patterns, thus is less general than our approach.

Another research direction is with data-driven methods, such as 2.5D dual contouring [14]. They provide building models with optimal geometry regarding the input point clouds. Our approach belongs to this category. Different from the state-of-the-arts, we produce a topology control scheme to enable a more flexible simplification solution.

2.3. Topology Control in Volumetric Modeling

In classic 2D and 3D volumetric modeling methods, the topology issue is first noticed by Ju *et al.* [2]. They propose a topology test mechanism to reject simplification operations yielding possible topology changes. This mechanism is later extended to the 2.5D building modeling framework by Zhou and Neumann [14].

The drawback of creating one vertex per octree/quadtree cell is noticed by researchers and different approaches have been proposed to solve this problem, *e.g.*, [6, 8, 11]. Generally, they all allow one grid cell to have more than one vertices, in order to track contour components which are topologically more complicated than a disk (or in 2D, a line segment) in each cell. Although these methods share some similarities with our approach, we present two key differences: first, our method aims at 2.5D building modeling involving hyper-points that cannot be handled in classic 2D or 3D manner; second, we define and process various building topology features which are more complicated than disk-like features in classic 2D or 3D space.

3. 2.5D Building Topology

Considering a 2.5D dual contouring process without adaptive simplification: taking aerial LiDAR data as input, the contouring method builds up a uniform grid with Hermite data attached; creates one hyper-point (a series of points that are consistent on the x-y plane) in each cell by optimizing a quadratic error function; generates surface polygons and vertical boundary polygons; and produces an unsimplified 2.5D building model such as the one shown in Figure 3 left.

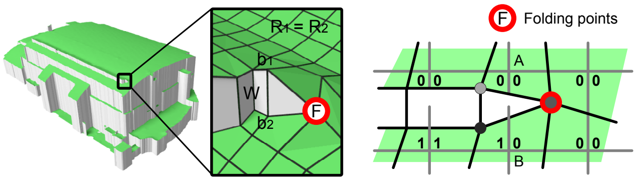


Figure 4. 2.5D building models may contain point features involving only one wall feature (left). These points are produced around grid edges (*e.g.*, AB) which detect inconsistent roof layer assignments in two adjacent cells (right).

3.1. Topological Feature Definitions

The first observation we have made for 2.5D building topology is that a typical building structural piece (*e.g.*, a chimney) is usually composed of one unique roof patch and its surrounding walls. Hence we define:

Definition 1 A *roof feature* R is a connected component composed of non-vertical surface polygons.

Roof features are the key in determining 2.5D building structures. Figure 3 utilizes green mesh pieces rendered with different color intensities to represent multiple roof features. In particular, neighboring roof patches exhibit a height gap along their common boundary, which is sealed up by vertical boundary polygons (grey vertical polygons). These polygons form wall features that are marked in Figure 3 by curves with various colors.

Definition 2 Given two roof features R_1 and R_2, the corresponding *wall feature* W is defined as the connected component composed of vertical boundary polygons adjacent to R_1 and R_2 simultaneously. W intersects with R_1 and R_2 via non-identical roof boundary polylines.

Here we make a slight modification to 2.5D dual contouring, that we disable triangulation for polygons. Instead, we treat surface polygons as quads connecting all four vertices around a grid corner, and produce boundary polygons with two non-vertical edges linking a pair of neighboring hyper-points. Therefore, we say a vertical boundary polygon "adjacent" to a roof feature as long as they share a non-vertical edge. By tracking the consecutive non-vertical edges, we create one roof boundary polyline regarding each adjacent wall-roof feature pair $(W, R_i), i = 1, 2$, denoted as $b_i = W \cap R_i, i = 1, 2$. According to definition 2, W is a valid feature when b_1 and b_2 are not identical, even if R_1 and R_2 refer to the same roof patch.

Definition 3 Given a wall feature W which shares two consecutive roof boundary polylines b_1 and b_2 with R_1 and R_2 respectively, we define *point features* as hyper-points which contain b_1 and b_2's end points if there is any.

Typically, a hyper-point shared by two neighboring wall features is a point feature. They are rendered in Figure 3 as golden balls connected by vertical lines. Note that definition 3 supports these common hyper-points shared by neighboring wall features; but is not limited to them. In a special case shown in Figure 4, the corner points of grid edge AB have the same roof layer assignment in one adjacent cell (right cell) but different assignments in another (left cell). Thus, a vertical boundary polygon is produced to reflect this significant topology feature. Specifically, R_1 and R_2 denote the same roof patch, which is adjacent with W along b_1 and b_2 representing the upper roof boundary polyline and the bottom roof boundary polyline respectively. Since b_1 and b_2 are non-identical, W is a valid wall feature. Point feature F acts as a "folding point" which connects b_1 and b_2 together and folds up the boundary of W.

3.2. Connections between Topological Features

By projecting the 2.5D building model onto the x-y plane, we can view the building topology with a 2D cell complex representation[2]. In particular, roof features, wall features, and point features are projected onto the 2D x-y plane as 2-cells (regions), 1-cells (polylines), and 0-cells (points) respectively. High dimensional cells are always bounded by a set of low dimensional cells.

Given a projection operator $\mathbb{P}(\cdot)$ which projects a set of 2.5D objects onto the x-y plane and a boundary extraction operator $\partial(\cdot)$, we reveal the connections between roof feature set $\mathcal{R}$, wall feature set $\mathcal{W}$, and point feature set $\mathcal{P}$ with following equations:

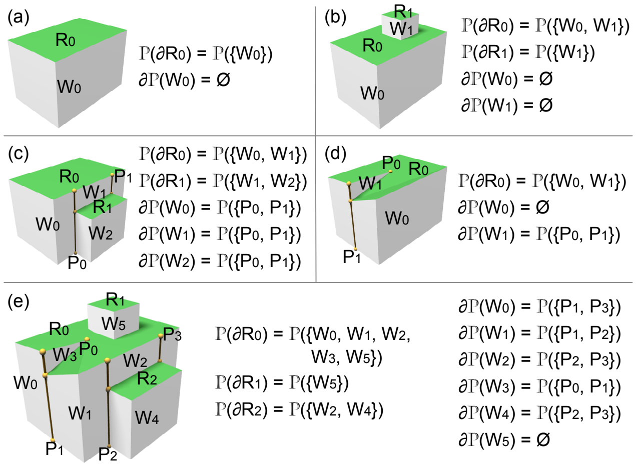
$$\mathbb{P}(\partial R) \subseteq \mathbb{P}(\mathcal{W}), \qquad \text{for any } R \in \mathcal{R}, \tag{1}$$

$$\partial\mathbb{P}(W) \subseteq \mathbb{P}(\mathcal{P}), \qquad \text{for any } W \in \mathcal{W}. \tag{2}$$

These equations can be straightforwardly derived from the definitions of roof, wall, and point features. On the other hand, once topological features and their associations expressed in form of Equation (1) and (2) are fixed, the 2.5D building topology is determined accordingly.

We demonstrate typical building structures in Figure 5 including standing-alone building blocks, vertically attached blocks, horizontally attached blocks, stair shapes, and the combinations of these patterns. Nevertheless, our 2.5D building topology representation describes them in a deterministic and differentiable manner.

In addition, we notice the difference between 2.5D topology representation and classic 2D topology representation. The latter one can be achieved by projecting all building elements onto the x-y plane at first, and treating different roof layers as multiple region materials. This representation, however, is problematic in handling wall features connecting the gap within one roof layer (*e.g.*, W_1 in Figure 5(d)). It eliminates such wall features together with the folding point (*e.g.*, P_0). In contrast, our topology representation faithfully preserves all significant 2.5D topology features which are the basis in our topology control method [3].



(a)
R0
W0
$\mathbb{P}(\partial R_0) = \mathbb{P}(\{W_0\})$
$\partial\mathbb{P}(W_0) = \varnothing$
(b)
R1
W1
R0
W0
$\mathbb{P}(\partial R_0) = \mathbb{P}(\{W_0, W_1\})$
$\mathbb{P}(\partial R_1) = \mathbb{P}(\{W_1\})$
$\partial\mathbb{P}(W_0) = \varnothing$
$\partial\mathbb{P}(W_1) = \varnothing$
(c)
R0
P1
W1
R1
W0
W2
P0
$\mathbb{P}(\partial R_0) = \mathbb{P}(\{W_0, W_1\})$
$\mathbb{P}(\partial R_1) = \mathbb{P}(\{W_1, W_2\})$
$\partial\mathbb{P}(W_0) = \mathbb{P}(\{P_0, P_1\})$
$\partial\mathbb{P}(W_1) = \mathbb{P}(\{P_0, P_1\})$
$\partial\mathbb{P}(W_2) = \mathbb{P}(\{P_0, P_1\})$
(d)
P0
R0
W1
W0
P1
$\mathbb{P}(\partial R_0) = \mathbb{P}(\{W_0, W_1\})$
$\partial\mathbb{P}(W_0) = \varnothing$
$\partial\mathbb{P}(W_1) = \mathbb{P}(\{P_0, P_1\})$
(e)
R1
W5
R0
P0
W3
P3
W2
R2
W0
W1
W4
P1
P2
$\mathbb{P}(\partial R_0) = \mathbb{P}(\{W_0, W_1, W_2, W_3, W_5\})$
$\mathbb{P}(\partial R_1) = \mathbb{P}(\{W_5\})$
$\mathbb{P}(\partial R_2) = \mathbb{P}(\{W_2, W_4\})$
$\partial\mathbb{P}(W_0) = \mathbb{P}(\{P_1, P_3\})$
$\partial\mathbb{P}(W_1) = \mathbb{P}(\{P_1, P_2\})$
$\partial\mathbb{P}(W_2) = \mathbb{P}(\{P_2, P_3\})$
$\partial\mathbb{P}(W_3) = \mathbb{P}(\{P_0, P_1\})$
$\partial\mathbb{P}(W_4) = \mathbb{P}(\{P_2, P_3\})$
$\partial\mathbb{P}(W_5) = \varnothing$

Figure 5. 2.5D building topology is represented by topological features, and the associations between them that are presented in form of Equation (1) and (2). Examples include typical building structures such as (a) individual building blocks, (b) blocks with top attachments, (c) blocks with side attachments, (d) stair-shaped structures, and (e) combinations of these patterns.

4. Contouring with Topology Control

So far we have formally defined 2.5D topological features and the associations between them. These mechanisms can be naturally expanded from a uniform grid to a quadtree. Thus, we present our topology control method to maintain the 2.5D building topology during quadtree-based simplification.

4.1. Hyper-points Clustering

The core of the simplification algorithm is to optimize the geometry of hyper-points based on a 2.5D quadratic error function [14]. We start with categorizing hyper-points by the number of its layers.

1. **1-layer points**: A hyper-point containing one vertex is optimized targeting the disk-like geometry in a grid cell. In most cases, it is connected to vertices created in its neighboring cells only by surface polygons. Such a hyper-point is an inner vertex of a roof feature. Thus it can be safely merged into a neighboring vertex without changing the 2.5D building topology. The accepting vertex can be either another 1-layer point or one vertex

[2] Cell complexes are the basic concepts in algebraic topology, see [1] for detailed discussion.

[3] This problem can also be regarded as the result of changing the order in applying boundary extraction operator and projection operator. As 2D topology representation projects all elements onto the x-y plane at first, it attempts to replace $\mathbb{P}(\partial R)$ in Equation (1) with $\partial\mathbb{P}(R)$. This attempt is problematic because $\partial\mathbb{P}(R) \not\equiv \mathbb{P}(\partial R)$ as $\mathbb{P}(\cdot)$ can absorb wall features such as W_1 in Figure 5(d).

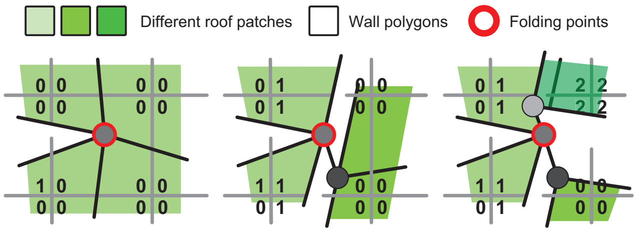


Figure 6. Folding points can be part of a 1-layer hyper-point, a 2-layer hyper-point, or a multi-layer hyper-point (from left to right). They are detected and marked as point features before adaptive simplification starts.

in a hyper-point with more than one layers. The only exception is a 1-layer folding point, which is connected to two different layers of a neighboring hyper-point by a boundary polygon, as shown in Figure 4 and Figure 6 left. In this case the 1-layer point is a point feature, thus should not be merged into other points.

2. **2-layer points**: A hyper-point containing two layers is a typical roof boundary anchor point, which is optimized with surface geometry and boundary geometry simultaneously. Typically, it is an inner element of a wall feature, thus can be merged into another hyper-point that is connected to it by a vertical boundary polygon. The accepting hyper-point can be either another typical 2-layer point or a multi-layer point. Similar to 1-layer points, folding points can exist within 2-layer points. Figure 6 middle shows such an example where the 2-layer point is a point feature. It cannot be merged in typical manner.

3. **Multi-layer points**: A hyper-point with more than two layers can be regarded as a conjunction point of more than two regions, if we view the problem on the 2D projection plane and treat roof patches as regions with different materials. We find that any multi-layer point is a point feature. Proof is straightforward as they cannot be the inner elements of wall features; thus always stand at boundaries of wall features' 2D projections. Therefore, multi-layer points can only accept 1-layer points and 2-layer points joining it, but cannot be merged with other point features.

Since folding points can exist in 1-layer points, 2-layer points, and multi-layer points, as demonstrated in Figure 6, they are detected and marked as point features during preprocessing. The detection algorithm is implemented by uncovering grid edges with inconsistent roof layer assignments in adjacent cells.

Apart from folding points, we denote typical 1-layer points, typical 2-layer points, and multi-layer points as p^1, p^2, and p^m respectively. We introduce a set of hyper-point clustering operations to merge components connected by surface polygons, denoted as $\Phi_{\mathrm{S}}^{1\to 1}$, $\Phi_{\mathrm{S}}^{1\to 2}$, and $\Phi_{\mathrm{S}}^{1\to m}$, with following functions:

$$\begin{aligned} \Phi_{\mathrm{S}}^{1\to 1}: &\quad \{p_1^1, p_2^1, \ldots, p_n^1\} &&\Rightarrow p^{1*}, && (3)\\ \Phi_{\mathrm{S}}^{1\to 2}: &\quad \{p_1^1, p_2^1, \ldots, p_n^1\}, p^2 &&\Rightarrow p^{2*}, && (4)\\ \Phi_{\mathrm{S}}^{1\to m}: &\quad \{p_1^1, p_2^1, \ldots, p_n^1\}, p^m &&\Rightarrow p^{m*}. && (5)\end{aligned}$$

Each operation merges a connected component within a roof feature, and produces one hyper-point (*e.g.*, p^{1*}, p^{2*}, or p^{m*}) per cluster. In particular, the geometric coordinates of output hyper-points are obtained by optimizing a 2.5D QEF matrix which is the combination of QEF matrices from input hyper-points. The third column from p_i^1's matrices are placed with corresponding matrix column from p^2 or p^m. Details of QEF matrices combination can refer to [14].

Similarly, we can define clustering operations for components that are connected by vertical boundary polygons:

$$\begin{aligned} \Phi_{\mathrm{B}}^{2\to 2}: &\quad \{p_1^2, p_2^2, \ldots, p_n^2\} &&\Rightarrow p^{2*}, && (6)\\ \Phi_{\mathrm{B}}^{2\to m}: &\quad \{p_1^2, p_2^2, \ldots, p_n^2\}, p^m &&\Rightarrow p^{m*}. && (7)\end{aligned}$$

These operations merge connected components within a wall feature.

4.2. Handling Degenerate Cases

Although these clustering operations aim at simplifying continuous roof and boundary features, they risk in making topological features degenerate. For example, with intense geometry simplification, the building structure in Figure 5(a) may degenerate into a single vertical line with its top vertex collapsed from R_0.

To address this problem, we employ degenerate tests after each clustering operation. Given $f(C)$ and $e(C)$ as the number of faces and edges in a cell complex C, we require:

$$\begin{aligned} f(R) &\geq 1, &&\text{for any } R \in \mathcal{R}, && (8)\\ e(\mathbb{P}(W)) &\geq 1, &&\text{for any } W \in \mathcal{W}. && (9)\end{aligned}$$

Initially, these two criteria are fulfilled due to Definition 1 and 2. In adaptive simplification phase, for each possible hyper-point clustering operation, we check all the modified roof features and wall features, and test if these two criteria are still satisfied. If any of them is violated, we rewind the clustering operation.

In practice, we found this degenerate test inefficient because it involves polygon recreation and topological feature detection after every clustering operation. We propose an equivalent criterion which is much easier to be implemented:

Degenerate test A hyper-point clustering operation passes the degenerate test if the following criterion stays true:

$$e(\partial\mathbb{P}(R)) \geq 3, \qquad \text{for any } R \in \mathcal{R}. \qquad (10)$$

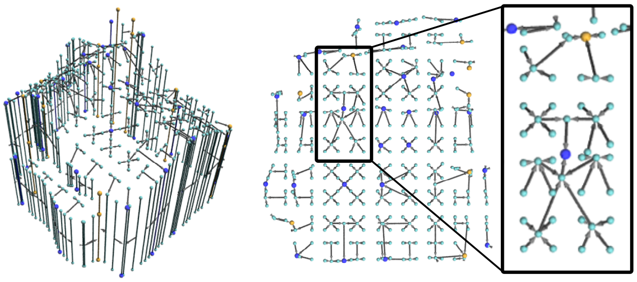


Figure 7. Hyper-point cluster forest viewed from oblique and orthogonal perspectives.

The equivalence between this degenerate test and the one based on Equation (8) and (9) is proved in Appendix A.

Since the degenerate test is irrelevant to typical 1-layer points, clustering operations based on surface components are always allowed (*i.e.*, $\Phi_{\mathrm{S}}^{1\to 1}$, $\Phi_{\mathrm{S}}^{1\to 2}$, $\Phi_{\mathrm{S}}^{1\to m}$). For clustering among 2-layer points and multi-layer points, we precompute boundaries of roof feature projections, *i.e.*, $\partial\mathbb{P}(R)$, and keep tracking of the edge numbers $e(\partial\mathbb{P}(R))$ during the complete simplification process. Boundary component clustering operations (*i.e.*, $\Phi_{\mathrm{B}}^{2\to 2}$ and $\Phi_{\mathrm{B}}^{2\to 2}$) decrease the corresponding edge numbers. Once a boundary edge number is less than 3, the latest operation is rewound.

4.3. Adaptive Contouring

We choose to extend 2.5D dual contouring by allowing multiple hyper-points in each grid cell of the quadtree Q. In addition to the adaptive structure of quadtree, we maintain a hyper-point forest to allow topology-preserving clustering operations which are detailed in previous sections. As illustrated in Figure 7, trees in the forest are connected components that can be simplified via a series of clustering operations, and each of them is finally represented by its root (blue and gold hyper-points) whose coordinates are determined by optimizing a 2.5D QEF combining geometric information from leaf points.

We build this hyper-point cluster forest in a bottom-up manner. In a quadtree cell c composed of four leaf cells $c_{0,0}, c_{0,1}, c_{1,0}, c_{1,1}$, assume each leaf cell has a set of cluster roots that are available for further clustering (*i.e.*, without exceeding the geometry error tolerance or violating degenerate test). We first traverse all the corner points in c that are shared by two of the four leaf cells. At each grid corner, four vertices in adjacent cells are connected via a surface polygon. We retrieve the roots of these vertices and detect possible clustering operations based on surface component, *i.e.*, $\Phi_{\mathrm{S}}^{1\to 1}$, $\Phi_{\mathrm{S}}^{1\to 2}$, or $\Phi_{\mathrm{S}}^{1\to m}$. Similar approaches can be applied to minimal grid edges which exhibit roof layer gaps. They imply boundary-neighborships leading to possible operations $\Phi_{\mathrm{B}}^{2\to 2}$ and $\Phi_{\mathrm{B}}^{2\to m}$.

With possible clustering operation detected, we sequentially test them against the geometry error tolerance and the degenerate test. Since the test sequence may affect the modeling quality, we specify a priority to each clustering operation. In particular, we assign high priority to $\Phi_{\mathrm{S}}^{1\to 1}$ and $\Phi_{\mathrm{B}}^{2\to 2}$; medium priority to $\Phi_{\mathrm{S}}^{1\to 2}$ and $\Phi_{\mathrm{S}}^{1\to m}$; and low priority to $\Phi_{\mathrm{B}}^{2\to m}$. The reason behind this priority assignment is that we expect 1-hyper points and 2-hyper points to be first clustered together to form meaningful geometric patterns (*e.g.*, roof ridges and straight vertical walls), before they are merged into key features with higher dimensional topology. As for clustering operations with same priority, the test sequence is determined by the addition to quadratic errors in ascending order.

Polygon generation of 2.5D dual contouring is adapted to the hyper-point cluster forest in a straightforward manner. Considering the unsimplified polygonal model created from the uniform grid, we replace each point in the model by the root of its cluster (or the corresponding portion of the root if that has more layers than the leaf point). Numerous polygons become degenerate and are removed automatically, *e.g.*, a triangle whose three vertices belong to the same cluster and thus map to the same root point. A simple polygonal model with small amount of triangles is produced which has the same 2.5D building topology as the unsimplified model.

5. Experiment Results

Figure 8 shows a stadium model reconstructed using different approaches, namely, 2.5D dual contouring (a,b), our method (c,d), manual creation (e), and plane-based method such as the one proposed in [12] (f). In particular, we vary the geometry error tolerance δ in order to trade-off between model scale and fitting quality. The relation curve between δ and the number of triangles produced by different approaches is illustrated in Figure 8(g). Quantitative measurements are given in Table 1. With error tolerance δ increasing, our method constantly decreases the triangle number of reconstructed models. Reasonable cost is paid in fitting quality as the trade-off. On the contrary, 2.5D dual contouring reaches the simplification barrier around 3000 triangles. This barrier can be explained by the last column of Table 1, showing the percentage of unsuccessful collapses caused by topology test among all unsuccessful collapses. Many of them happen in small cells that create trivial triangles as shown in Figure 8(a,b) closeups.

Figure 9 shows the building reconstruction for a 5km-by-7km urban area of Denver, from 73M input aerial LiDAR points with 6 samples/sq.m. resolution. We employ the urban modeling pipeline in [13] to extract individual building patches, and test our method, 2.5D dual contouring, and a plane-based method independently. We utilize a fairly large error tolerance for both our method and 2.5D dual contouring. Our method successfully reconstructs 2,099 2.5D building models within 5 minutes on a consumer level lap-

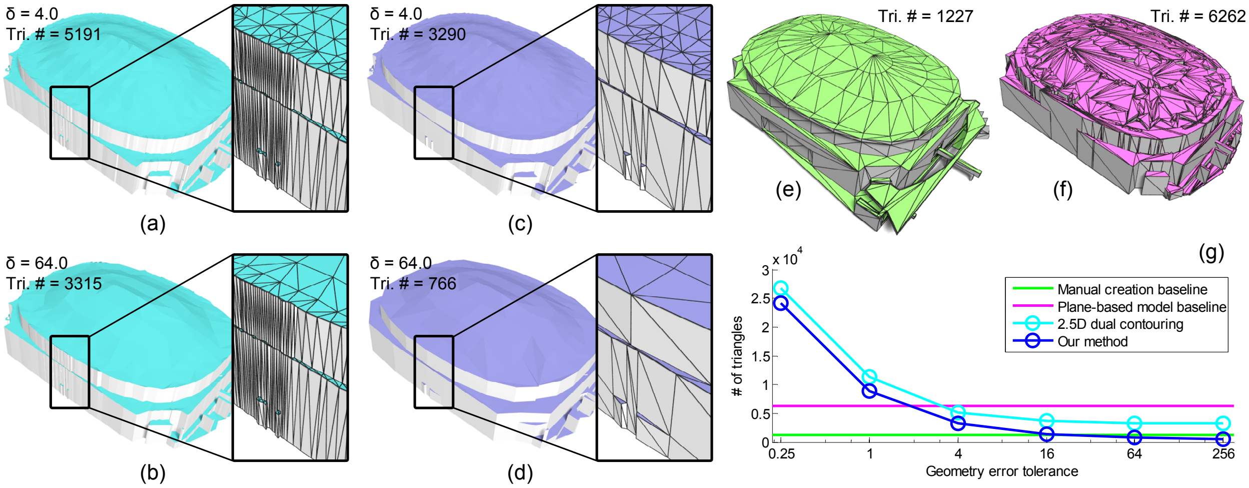


Figure 8. A stadium model created using (a,b) 2.5D dual contouring [14], (c,d) our method, (e) manual creation, and (f) plane-based approach [12]. The relation curve between error tolerance and the triangle number of reconstructed models is illustrated in (g). With larger geometry error tolerance given, our method can always produce simpler models with less triangles; while the overstrict topology test in 2.5D dual contouring creates numerous trivial triangles along thin roof features shown in closeups of (a,b).

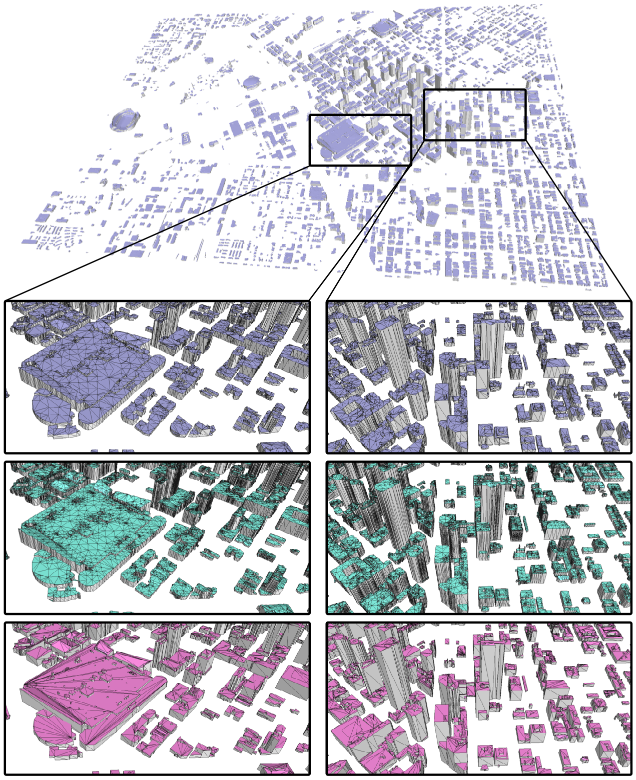


Figure 9. 2,099 building models are created for an urban area in Denver using (top) our method, (middle) 2.5D dual contouring, and (bottom) plane-based method. Our method produces as few triangles as the plane-based method while recovering and preserving the topological features in each building structure.

top (Intel i-7 CPU 1.60GHz with 6GB memory). We produce 227,566 triangles for the building models which are rendered in the top rows of Figure 9. Our output triangle number is comparable to plane-based results (181,752 triangles rendered in the bottom row). However, unlike the plane-based method, our method detects and preserves 2.5D building topology, thus avoids producing cracks and inconsistencies between building blocks. *E.g.*, the roof of the large structure shown in the bottom left closeup intersects with small features on top of it; while our method does not have such problem. The middle row of Figure 9 shows 2.5D dual contouring result, it produces twice as many triangles (551,341 triangles) as the other two approaches.

Since the scale of our result is inversely proportional to geometry error tolerance δ. It is beneficial to study the evolution of building model with respect to δ. In particular, we create 2.5D building models for the same LiDAR point cloud using an exponentially increasing δ, shown in Figure 10. Although the model geometry constantly becomes simpler, the building topology is faithfully preserved. Even in the extreme simplification case where $\delta = \infty$, we generate a model with 32 vertices and 52 triangles, which is the smallest amount of vertices and triangles that can represent the building topology of this model. [4]

6. Conclusion

We define 2.5D building topology as a combination of topological features and the associations between them. We propose convenient tools to change model geometry without modifying the topology. In addition, we extend 2.5D dual contouring with our topology control strategies, to achieve a more flexible adaptive structure for simplification.

[4] Visit our project homepage for more results and demonstrations: http://graphics.usc.edu/~qianyizh/projects/buildingtopology.html

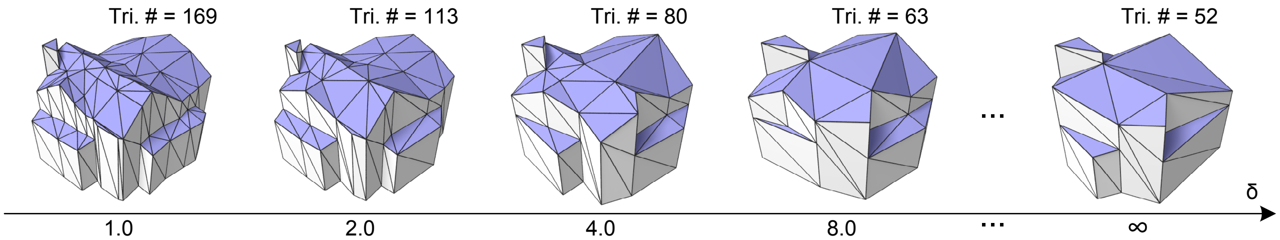


Figure 10. Model evolution with error tolerance growing from 1.0 to infinite.

Geometry error tolerance	Our method		2.5D dual contouring [14]		
	Triangle #	Ave. distance^2	Triangle #	Ave. distance^2	Topology test failure rate
$\delta = 0.25$	24161	0.0157	26776	0.0156	3.02% (82 out of 2713)
$\delta = 1.00$	8864	0.0202	11280	0.0182	11.83% (109 out of 921)
$\delta = 4.00$	3290	0.0849	5191	0.0316	39.10% (149 out of 381)
$\delta = 16.00$	1374	0.1259	3644	0.0988	76.30% (161 out of 211)
$\delta = 64.00$	766	0.4812	3315	0.2104	90.45% (161 out of 178)

Table 1. Quantitative comparison between our method and 2.5D dual contouring using the experiment shown in Figure 8. The last column reports the percentage of cell collapses rejected by topology test among all rejected collapses. The topology test becomes dominant in 2.5D dual contouring with large error tolerance.

Our outputs have the same representability as models created by 2.5D dual contouring, but contain fewer vertices and triangles.

7. Acknowledgement

We thank Airborne 1 Corp. and Sanborn Corp. for providing data. We are grateful for support from Dr. Mark Pritt at Lockheed Martin Corp. We thank Na Chen, Suya You, and anonymous reviewers for their valuable comments.

Appendix A

Proof of equivalence between (8)+(9) and (10):

First, we notice that all three equations stand true in the initial unsimplified model.

After a clustering operation, if both equation (8) and (9) are true, for each $R \in \mathcal{R}$, we have $f(\mathbb{P}(R)) = f(R) \geq 1$. *I.e.*, $\mathbb{P}(R)$ contains at least one 2-cell. Thus, the boundary of $\mathbb{P}(R)$ contains at least 3 edges, $e(\partial\mathbb{P}(R)) \geq 3$.

Conversely, when Equation (10) is true, $\mathbb{P}(R)$ has at least one 2-cell. Therefore, $f(R) = f(\mathbb{P}(R)) \geq 1$, *i.e.*, (8). As for a wall feature W with $|\partial\mathbb{P}(W)| \geq 2$, it is bounded by two point features, and (9) is true by definitions of hyper-point clustering operations. Now we consider a wall feature W with $|\partial\mathbb{P}(W)| \leq 1$ (*e.g.*, W_0 in Figure 5(d) and W_1 in Figure 5(b)), $\mathbb{P}(W)$ is a close loop on 2D space. It divides $\mathcal{R}$ into partition $\mathcal{R}_{in}$ and $\mathcal{R}_{out}$, where $\mathcal{R}_{in} \neq \emptyset$. We have:

$$\mathbb{P}(W) = \partial \sum_{R \in \mathcal{R}_{in}} \mathbb{P}(R). \tag{11}$$

Since the right part sums at least one 2-cell before boundary extraction, the boundary of the cell complex contains at least 3 edges, *i.e.*, $e(\mathbb{P}(W)) \geq 3$. □

References

[1] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[2] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring on hermite data. In *ACM SIGGRAPH*, 2002.

[3] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny. Building reconstruction from a single dem. In *CVPR*, 2008.

[4] B. Matei, H. Sawhney, S. Samarasekera, J. Kim, and R. Kumar. Building segmentation for densely built urban regions using aerial lidar data. In *CVPR*, 2008.

[5] C. Poullis and S. You. Automatic reconstruction of cities from remote sensor data. In *CVPR*, 2009.

[6] S. Schaefer, T. Ju, and J. Warren. Manifold dual contouring. *IEEE Trans. Vis. Comput. Graph.*, 2007.

[7] A. Toshev, P. Mordohai, and B. Taskar. Detecting and parsing architecture at city scale from range data. In *CVPR*, 2010.

[8] G. Varadhan, S. Krishnan, Y. Kim, and D. Manocha. Feature-sensitive subdivision and iso-surface reconstruction. In *IEEE Visualization*, 2003.

[9] V. Verma, R. Kumar, and S. Hsu. 3d building detection and modeling from aerial lidar data. In *CVPR*, 2006.

[10] L. Zebedin, J. Bauer, K. Karner, and H. Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *ECCV*, 2008.

[11] N. Zhang, W. Hong, and A. Kaufman. Dual contouring with topology preserving simplification using enhanced cell representation. In *IEEE Visualization*, 2004.

[12] Q.-Y. Zhou and U. Neumann. Fast and extensible building modeling from airborne lidar data. In *ACM GIS*, 2008.

[13] Q.-Y. Zhou and U. Neumann. A streaming framework for seamless building reconstruction from large-scale aerial lidar data. In *CVPR*, 2009.

[14] Q.-Y. Zhou and U. Neumann. 2.5d dual contouring: A robust approach to creating building models from aerial lidar point clouds. In *ECCV*, 2010.