

# Topologically-aware Building Rooftop Reconstruction from Airborne Laser Scanning Point Clouds

Dong Chen, Ruisheng Wang, and Jiju Peethambaran

**Abstract**—This paper presents a novel topologically-aware 2.5D building modeling methodology from airborne laser scanning (ALS) point clouds. The building reconstruction process consists of three main steps: primitive clustering, boundary representation and geometric modeling. In primitive clustering, we propose an enhanced probability density clustering (PDC) algorithm to cluster the rooftop primitives by taking into account the topological consistency among primitives. In the second step, we employ a novel Voronoi subgraph based algorithm to seamlessly trace the primitive boundaries. This algorithm guarantees the production of geometric models without crack defects among adjacent primitives. The primitive boundaries are further divided into multiple linear segments, from which the key points are generated. These key points help to form a hybrid representation of the boundary by combining the projected points with part of the original boundary points. The model representation by the hybrid key points is flexible and well captures the rooftop details to generate lightweight and highly regular building models. Finally, we assemble the primitive boundaries to form the topologically correct entities which are regarded as the basic units for primitive triangulation. The reconstructed models not only have accurate geometry and correct topology but more importantly have abundant semantics, by which five levels of building models can be generated in real time. The proposed reconstruction method has been comprehensively evaluated on Toronto data set in terms of model compactness, multilevel model representation and geometric accuracy.

**Index Terms**—LiDAR, ALS, Rooftop Clustering, Building Rooftop Model, LoD, Topological Consistency

## I. INTRODUCTION

THREE dimensional models of buildings are of great interest to many applications such as 3D animation [1], indoor or outdoor navigation [2], building energy modeling [3], disaster simulation [4] and evaluation of the built environment [5], among others. As an obvious consequence,

This work was jointly supported by the Open Funds of the State Key Laboratory of Remote Sensing Science under Grant OFSLRSS201516, the National Natural Science Foundation of China under Grant 41301521, the China Scholarship Council (CSC) under Grant 201506710075 and the Eyes-High Postdoctoral fellowship at the University of Calgary.

D. Chen is now with the Nanjing Forestry University, Nanjing 210037, China, and also with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: chendong@njfu.edu.cn; dong.chen@ucalgary.ca).

R. Wang is with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada e-mail: (e-mail: ruishwang@ucalgary.ca).

J. Peethambaran is with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada e-mail: (e-mail: jiju.poovvancheri@ucalgary.ca).

\* This work was performed when all the authors were in the Department of Geomatics Engineering at the University of Calgary.

building reconstruction using various data sources has gained a lot of attention in diverse communities including computer vision, machine learning, computer graphics, photogrammetry and remote sensing. Input data sources can be monocular or stereo images, videos or point clouds collected using different platforms. Among different data sources, the emergence of 3D laser scanners which can acquire high density point clouds with decimeter accuracy, has certainly elevated the building reconstruction research to a new direction where visual realism, accuracy, automation, real time rendering and storage are of utmost concern.

Different modalities of laser scanning such as Terrestrial Laser Scanning (TLS), Mobile Laser Scanning (MLS) or Airborne Laser Scanning (ALS) are employed to capture the surface geometry. Compared to TLS which is performed from a static viewpoint on the ground, MLS and ALS have an additional advantage of mobility and hence, are capable of producing high quality of building and surface models over large geographical areas. However, MLS systems are often found to scan only street side façades, leaving out the rear part of the buildings. To some extent, this limitation is addressed by ALS by capturing the entire rooftops of the buildings, which can possibly represent the overall building rooftop structure without the façade details. Wide coverage from the aerial view and an obstruction free movement make ALS a widely chosen data source for large-scale building reconstruction.

Broadly, the ALS based building reconstruction techniques can be categorized into data-driven, model-driven and hybrid-driven. Data-driven methods [6]–[11] adopt a bottom-up strategy that starts with the extraction and analysis of primitive’s (e.g., planes, cylinders, cones, spheres or tori) topology in 2D or 3D space. The geometric elements of the primitives, such as lines and critical vertices of the rooftops, are then extracted and grouped to form building models. In contrast, the model-driven approaches [12]–[15] employ a top-down strategy that usually maintains a hypothetical model library composed of basic building shapes, e.g., flat, gable, hip, shed, saltbox, mansard, pyramidal and gambrel and uses rooftop point clouds to search optimal model composition and the associated parameters from the model library. A few other techniques follow hybrid modeling by combining the advantages of data and model-driven methods. Basically, the hybrid-modeling methods often rely on ground plan [16] or sets of non-overlapped and connected quadrilaterals [17] to divide the complex rooftop into subsets where each subset can be easily matched with the library models. An alternative to hybrid representation

is rooftop topology graph (RTG) matching [18]–[24]. RTG consists of rooftop primitives as its nodes and their adjacency relationship as its edges. Once the RTG is constructed, the problem of rooftop modeling can be naturally converted to a problem of rooftop graph matching with basic topology graph elements in a model library.

Despite significant efforts, generation of visually convincing models which accommodate various pressing needs of urban applications are still far from satisfactory. To delineate the possible reasons, we briefly go over each ALS based reconstruction method. The model-driven methods are often limited by the types of models available in the libraries and possible mismatch between the data and models. Even though, a “divide-and-conquer” strategy for generating complex rooftops or expanding the model libraries can partially solve this problem, it is still difficult to accurately match various building rooftops with insufficient number and types of hypothetical building models. In hybrid modeling, how and to what extent the features of data and model-driven methods can be combined is a long standing problem. Comparatively, data-driven methods are capable of creating complicated building models, and are not restricted to rooftop geometry. Theoretically, they can be used to model any type of rooftops, including curved and free-formed rooftops. Typically, data-driven methods are dependent on the quality of point clouds and consequently, are sensitive to the local point density, outliers and noise. However, it should be aware that the quality of ALS data has been greatly improved over the years, reaching up to 50 points/m<sup>2</sup> and even more [25]. The ALS point clouds with high point densities, multiple returns, and full-waveform recording allow, to a certain level, effective reconstruction of buildings without imposing additional assumptions.

**Contributions:** Given that our work is built on data-driven framework, we explicitly state our original contributions as follows:

- **Enhanced PDC:** we propose an enhancement to the probability density clustering (PDC) [26] algorithm to cluster the rooftop primitives without topological errors.
- **Voronoi Subgraph for Boundary Tracing:** we present an algorithm for tracing the primitive boundaries via Voronoi subgraph, which seamlessly trace the adjacent primitive boundaries, thus achieving water-tight building models.
- **Hybrid Representation of Boundaries:** a hybrid representation of primitive boundary is proposed to represent the reconstructed models which ensures the building compactness and regularity.
- **Multiple LoDs Modeling:** The proposed framework is capable of generating building models at five levels of details in real time, enhancing the flexibility of the rooftop modeling.

This paper is organized as follows. Section II reviews the data-driven methods in ALS based building reconstruction. Section III describes the detailed methodology including an enhanced rooftop clustering, representation of primitive boundaries, segmentation of primitive boundaries, and rooftop reconstruction. In Section IV, the experimental data set, the

performance evaluation results of rooftop clustering and reconstruction based on actual ALS data set are presented, analyzed, and discussed. Finally, Section V concludes the paper along with a few suggestions for future research topics.

## II. RELATED WORK

In this section, we review the relevant methods in building reconstruction literature which are closely related to our work, i.e., data-driven modeling.

To date, a plethora of work has been proposed for ALS based building reconstruction. For example, Zhang et al. [6] present a prismatic building modeling through extraction and regularization of the building outlines. To model relatively complicated, multi-level flat buildings, the general steps including rooftop segmentation, surface fitting, boundary simplification, boundary refinement, and 3D building model generation are frequently adopted in a series of publications [7], [8], [26]–[28]. Since each primitive boundary is processed separately, these methods cannot guarantee the generation of water-tight models. Chen et al. [29] address this limitation by introducing a multi-scale method which accurately segments the building roofs by combining two-level depth images, then constructs the building rooftops by simply stretching the patches to the ground. By transforming the point clouds into the corresponding depth images, this method ensures the topological consistency among components of rooftop models. Li et al. [30] use a set of axis-aligned boxes to approximate the geometry of Manhattan-world buildings. In their method, the problem of scene reconstruction has been successfully transformed into an optimal subset of the candidate boxes selection problem which is well solved based on a Markov Random Filed formulation.

The most commonly used data-driven rooftop modeling technique is the polyhedral modeling, which can be adapted for generating building models with both simple and complex topology of rooftops. In [31], a rooftop modeling approach based on ridge and jump edges is presented. Ridge and jump edges are identified through the intersection of the rooftop faces and analysis of the height discontinuities. To obtain the accurate jump edges, the orientation of the main building is first derived and then used as a constraint for the jump edge orientation. The rooftop topology is finally maintained by analyzing the relationship of linear features and bridging the gaps between the detected adjacent edges. Binary Space Partitioning (BSP) tree [32] is another tool to recover the rooftop topology between adjacent planar or linear segments. Using BSP and a controlling depth, the authors [32] demonstrate the generation of different representations for a polyhedral roof model at different levels of details. In [33], Dorninger and Pfeifer analyze the topological relationship among adjacent rooftop primitives to produce intersecting edges combined with regular building outlines for generating water-tight polyhedral rooftop models. Although detailed 3D building models with rooftop overhangs are generated, manual interventions are required in a post-processing step to eliminate possible artifacts from the building models. Sampath and Shan [9] address the reconstruction of polyhedral rooftop using rooftop primitive segmentation via

an improved fuzzy k-means algorithm. The interior vertices of the roofs are restored by using an adjacency matrix of the primitives. Even though, most of the rooftop vertices are obtained from the adjacency matrix, the vertical walls and building boundaries are also required as a priori knowledge to determine the vertices on the outlines.

Kim and Shan [34] put forward a method for polyhedral rooftop reconstruction based on the level set method, by determining the interior and boundary vertices through adjacency relationship of primitives and building footprints, respectively. To make the polyhedral model visually appealing, a variety of global regularities, including roof-roof regularity, roof-boundary regularity and boundary-boundary regularity have been introduced in [10]. The principle of this method is that global regularity reveals the topological relationships and similarities of the primitives for intrinsic structure of building models arising from architectural designs. To simplify the maintenance of rooftop's topology, the authors [11] introduce a 2D snake algorithm which provides a trade off between parallelism of the rooftop's primitive boundary and deviation from its original position. They transform the problem of complex rooftop reconstruction in 3D space into a problem of maintaining the topology of 2D primitives. Similarly, Chen et al. [35] use ESRI polygon Z data structure to maintain the final rooftop vertices and their topological relationships, which makes it easy for delineating and rendering 3D building rooftop models through triangulation.

To deal with complex free-form rooftop structures, many researchers employed quadtree simplification [36], tetrahedral mesh simplification [37] and octree 3D geometry simplification [38], all of which are capable of representing building models at different levels of details. To obtain the semantic information, the simplification procedure usually needs additional constraints by adding specific semantic components, e.g., planes, cylinders, cones and tori [39], which are usually detected by shape detection algorithms such as RANSAC [40] or Hough Transform [41]. In particular, Lafarge and Mallet [42] propose a complete, realistic and semantically rich description of urban scenes by simultaneously reconstructing buildings, trees and topologically complex ground surfaces. To realize the semantic labeling of urban scenes (tree, building and ground), they employ a 2.5D planimetric map and solve a non-convex energy minimization problem. By using a 2.5D planimetric map for the representation of complex urban scenes, the problem of maintaining complex topology of rooftop primitives is transformed to a labeling problem. Zhou and Neumann [36] propose a general building representation using a mesh simplification procedure based on a 2.5D dual-contouring. The point clouds are first converted into Hermite data of surface and boundary over a 2D grid. A simultaneous optimization over 3D surface and 2D boundary of roof layers is implemented by minimizing quadratic error functions to generate a hyper-point in each grid cell, containing a set of 3D points with the same x and y coordinates but different z values. These hyper-points are connected to meshes by generating roof and wall polygons. The reconstructed models preserve sharp features and are guaranteed to be crack free with small fitting errors. This method is further extended to

model building rooftops in residential areas where most of the rooftops are partially missing due to vegetation occlusions [43]. To maintain topology of the primitive boundary, Zhou and Neumann extend the 2.5D dual contouring algorithm into a 2.5D modeling method with topology control [44].

The reader can refer to the review papers [39], [45]–[47] published among photogrammetry, computer graphics, computer vision, and remote sensing communities to learn more details about building modeling. Essentially, data-driven methods focus on maintaining the rooftop topology of primitive, boundary or their combinations in 2D planimetric space and/or 3D space of the point clouds. As mentioned above, the adjacency matrix [9], Doubly Connected Edge List (DCEL) data structure [34], BSP tree [32], 2.5D planimetric map [42], ESRI Shape [11], [35], 2.5D dual contouring [36], a set of axis-aligned boxes [30] and the simplest triangular meshes<sup>1</sup> are frequently employed to maintain topology of the rooftop structure. To better maintain the topology for rooftop reconstruction, we propose a topologically-aware rooftop modeling algorithm which can produce water-tight, compact, and high-fidelity 2.5D polyhedral rooftop models. Here the concept of “topology-aware” is deeply rooted in the whole pipeline of the rooftop model reconstruction. More precisely, in the clustering step, the algorithm well restrains the primitive from spanning over other adjacent primitives through an optimization mechanism, thereby maintains the topological consistency among primitives. In primitive boundary tracing, the proposed subgraph of the Voronoi diagram seamlessly maintains the connection among adjacent primitives, preserving the topological consistency among the traced primitive boundaries. In model reconstruction, a series of primitive boundary assembly strategies collectively guarantee the topological consistency of the reconstructed models.

### III. METHODOLOGY

A detailed block diagram of the proposed method and main functions used to construct 2.5D building models using ALS point clouds is shown in Fig. 1. It consists of three stages of processing including rooftop primitive clustering (see Fig. 1(a)), primitive boundary representation (see Fig. 1(b)) and multilevel geometric model reconstruction (see Fig. 1(c)).

For the rooftop primitive clustering, we proposed an enhancement to the PDC algorithm [26]. The algorithm integrates the improved clustering criterion, optimization procedure and refinement to collectively ensure the topology correctness among rooftop primitives. For primitive boundary representation, we propose a novel subgraph of the Voronoi diagram algorithm under the constraint of building inner and outer boundaries to accurately trace all the primitive boundaries. Thereafter, these primitive boundaries are further divided into multiple linear segments by the proposed optimization algorithm. The final primitive boundaries are generated through a hybrid representation of key points. During the rooftop model reconstruction, multiple primitive boundary assembly

<sup>1</sup>The well-known commercial photogrammetric software packages such as Acute3D (<http://www.acute3d.com>), Pix4D (<http://pix4d.com>) and SURE (<http://www.nframes.com>) have extensively used the triangular meshes for modeling large-scale urban scenes.

strategies, namely the occlusion processing, inner hole representation and abstract processing are employed to jointly maintain the topological consistency of the reconstructed models.

#### A. Rooftop Primitive Clustering via an Enhanced PDC

Inspired by the work in [26], the enhanced PDC algorithm consists of three steps including initial PDC clustering, cluster optimization and unlabeled point refinement. The initial PDC clustering uses an improved clustering criterion, i.e., a simple Euclidean distance from the point clouds to their corresponding primitives to improve the efficiency and enhance the robustness of clustering. In [26] that rely on high-dimensional features, the multiple quantities in high-dimensional space do not have the same scale which makes it biased towards one or the other. The clustering optimization solves the issues of topological inconsistency among rooftop primitives and the follow-up unlabeled point refinement further guarantees the completeness and correctness of rooftop primitives.

1) **Initial PDC:** The four steps of the initial PDC algorithm are described below:

① **Choose reliable seed points to fit a seed plane:** The points of each building are sorted according to each point's flatness property calculated by the local principle component analysis. The point with the highest flatness value is selected as a seed point. The seed point and its  $N$  nearest neighborhood points are combined together to represent the seed plane.

② **Determine a clustering criterion:** The current candidate point's probability density is calculated by

$$P_r(d_p) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(d_p - \mu)^2}{2\sigma^2}\right\}, \quad (1)$$

where  $d_p$  is the Euclidean distance from the current seed point to the fitted primitive. The parameters  $\mu$  and  $\sigma$  are the current cluster's mean and variance, respectively. According to the Eq. 1, if the candidate point's probability density is greater than or equal to the current cluster's probability density defined by  $P_r(\mu + k\sigma)$  ( $k$  is the correlation coefficient which will be discussed in Section IV), the candidate point will be merged into the current cluster. Meanwhile, the current cluster's mean value  $\mu$  and variance  $\sigma$  are updated simultaneously.

③ **Search neighborhood of current cluster:** Based on K-D tree indexing and breadth-first search technique, the current cluster's adjacent neighborhood points within a sphere of radius  $\mathcal{R}$  are searched, then we execute the step 2 to determine whether or not these adjacent points should be merged into the current cluster. This procedure is executed recursively until the current primitive clustering is finished.

④ **Search other rooftop primitives:** The remaining unvisited points with highest flatness value is selected and step ① through step ③ are repeated until all the unvisited remaining points have been processed.

2) **Cluster Optimization:** Essentially, the initial Gaussian PDC algorithm is a type of region growing algorithm and consequently, the dynamically growing cluster usually spans over other adjacent primitives due to the selection of inappropriate neighborhood size  $\mathcal{R}$ , unreasonable clustering order and

an inappropriate threshold  $k$ . Fig. 2 shows the comparisons of primitive topology before and after the optimization. As illustrated in red rectangles in Fig. 2(a), the density of point clouds is distributed unevenly and the details can be viewed from the enlarged rectangles. In this case, many typological errors occur after initial clustering, which can be verified from the red enlarged rectangles in Fig. 2(b). To address topological inconsistency, we define the energy function  $E_{\mathcal{P}}$  in Eq. 2 to optimize the results of initial PDC.

Given an arbitrary building point set, we expect to obtain  $m$  rooftop primitives  $\mathcal{P} = \{P_1, P_2, \dots, P_m\}$  by using the labeling optimization process  $\mathcal{F}$ . An arbitrary primitive  $P_i$  is composed of the point set  $p_i^1, p_i^2, \dots, p_i^{n_i}$ . To minimize Eq. 2, the segmented primitive should obey the rooftop structure. Meanwhile, the rooftop details should be maintained and the cluster results should be homogeneous as much as possible. Based on the above principles, we define the final energy function as follows:

$$E_{\mathcal{P}} = E_{alignment} + \eta_1 E_{smooth} + \eta_2 E_{fidelity}, \quad (2)$$

where  $E_{alignment}$  controls the consistency between the labeled points and the observed points.  $E_{smooth}$  controls the homogeneity of the cluster. The primitive points in the local neighborhood areas tend to be identically labeled.  $E_{fidelity}$  determines the details of segmented primitives, which penalize the small size of clusters and prevent producing redundant primitives.  $E_{\mathcal{P}}$  is defined as a weighted sum of three individual energy terms and the detailed description of the energy terms are defined below.

$E_{alignment}$  term penalizes the inconsistency between the point clouds and the potential primitive. A larger distance incurs a higher penalty. This term is given by

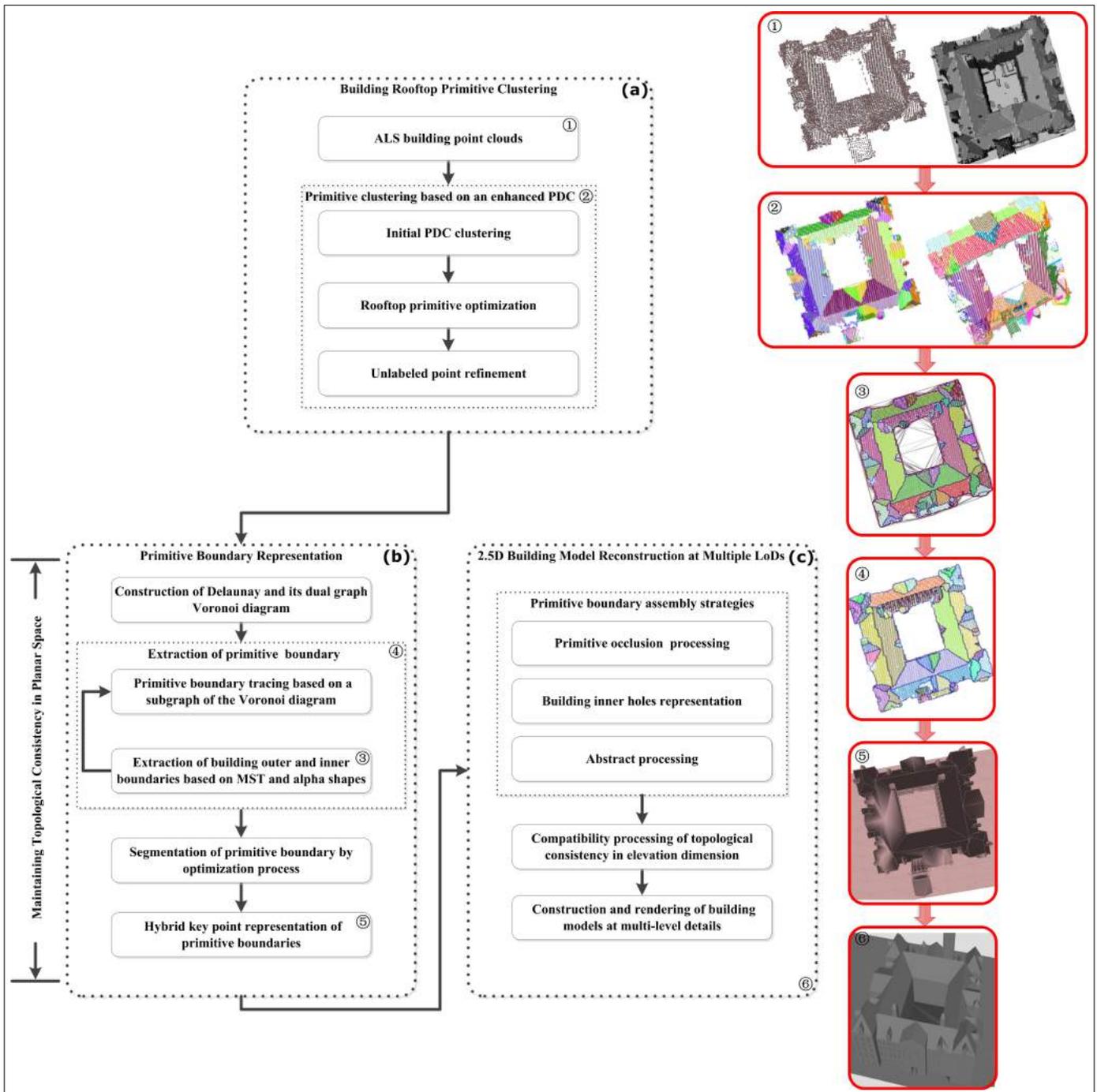
$$E_{alignment} = \sum_{i=1}^m \sum_{j=1}^{n_i} \left\| -\ln(P_r(p_i^j \in P_i)) \right\|^2, \quad (3)$$

where  $P_r(p_i^j \in P_i) \propto \frac{1}{\sqrt{2\pi}\sigma_i^j} \cdot \exp\left\{-\frac{dist(p_i^j, P_i)^2}{2(\sigma_i^j)^2}\right\}$ .  $dist(p_i^j, P_i)$  measures the Euclidean distance from a point  $p_i^j$  to the segmented primitive  $P_i$ . Obviously, the term  $E_{alignment}$  severely penalizes the current point  $p_i^j$  if it has a low probability density to  $P_i$  but favors an opposite situation.

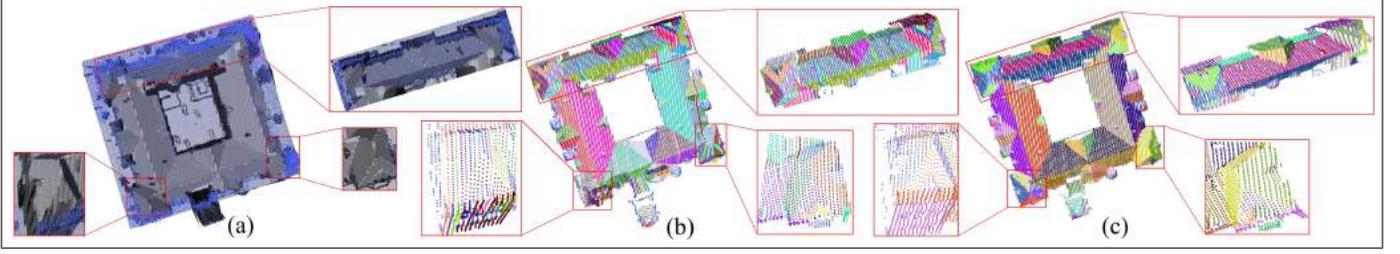
$E_{smooth}$  term penalizes the inconsistency between neighboring points. When the neighboring points have different labels, the longer distance is, the higher penalty it should be imposed. We define this term as follows:

$$E_{smooth} = \sum_{i=1}^m \sum_{j=1}^{n_i} \left\| \exp\{-p_i^j - q\} \cdot \delta(\mathcal{F}(p_i^j) \neq \mathcal{F}(q)) \right\|^2, \quad \forall q \in N_R, \quad (4)$$

where  $\mathcal{F}(p_i^j)$  is a labeling optimization process for each point  $p_i^j$  and  $\delta(\cdot)$  is an indicator function.  $q$  is the a neighborhood point set of  $p_i^j$  within a sphere of radius  $\mathcal{R}$ . Note that small value of  $\mathcal{R}$  can weaken the smoothing effect, which may result in inhomogeneous rooftop primitive. On the other hand, large  $\mathcal{R}$  can lead to over-smoothed segmentation, resulting in somewhat under-segmented primitives.



**Fig. 1. Block diagram showing the proposed methodology.** (a) Rooftop primitive clustering. (b) Representation of primitive boundary. (c) Reconstruction of 2.5D building models at multiple LoDs. Note that some module results are vividly shown in the red rectangles with identical index numbers.



**Fig. 2. Comparisons of primitive topology before and after optimization.** (a) The raw point clouds are overlaid on the digital surface relief map. (b) The initial results of PDC clustering. (c) The refined segmentation results after optimization. The blue points present the raw data and the different colors represent different segmented primitives. A few colors have been reused and as a result, different disjoint primitives may share same color.

Intuitively, the term  $E_{fidelity}$  penalizes the redundant primitives and encourages merger of the small primitives into large primitives to avoid the over-segmentation effectively and is given by

$$E_{fidelity} = \sum_{\substack{i=1 \\ P_i \in \mathcal{P}}}^m \left\| \exp\left\{-\frac{|P_i|}{\max(|P_i|)}\right\} \right\|^2 \cdot \delta(\mathcal{F}), \quad (5)$$

$$\delta(\mathcal{F}) \stackrel{def}{=} \begin{cases} 1, & \exists p_i^j : \mathcal{F}(p_i^j) = P_i \\ 0, & \text{otherwise,} \end{cases}$$

where  $|P_i|$  represents the number of points in the segmented primitive  $P_i$ .  $\max(|P_i|)$  refers to the maximum number of the primitive points in set  $\mathcal{P}$ .

To solve the Eq. 2, we use the graph optimization technique, i.e., alpha-expansion algorithm to obtain the optimized primitives. The initial labels are provided by the initial PDC. Compared with the initial results illustrated in Fig. 2(b), after optimization, the rooftop primitives have been accurately clustered even for the point clouds with high uneven density as shown in Fig. 2(c).

3) **Unlabeled Point Refinement:** To further improve the integrity of primitives, the remaining unlabeled points (their cluster sizes are less than the minimum size of primitive  $M_p$  as defined in Table I after initial PDC clustering) should be projected onto their associated primitives. The statistical analysis algorithm is used to map some qualified unlabeled points onto their corresponding segmented primitives. More specifically, we first search the neighborhood of each unlabeled point with a sphere of radius  $\mathcal{R}$  and apply a statistical analysis according to the frequency of label occurrence. The label with highest frequency is assigned to the current unlabeled point. If none of the points has been labeled, the current point will remain unlabeled.

### B. Rooftop Primitive Boundary Representation

After primitive clustering, next step is to accurately trace the primitive boundaries. To this end, we employ a novel algorithm based on the subgraphs of Voronoi diagram. To further make the model compact and regular, the primitive boundaries are segmented into multiple and linear segments by the proposed optimization algorithm. The primitive boundaries are represented by the key points, which combines the original boundary points with the projected points to jointly achieve a hybrid representation of the primitive boundary.

#### 1) Primitive Boundary Tracing via Voronoi Subgraph:

The detailed algorithm involves the following four steps:

① 2D Delaunay triangulation and its dual graph Voronoi diagram are constructed from the labeled roof points (see Section III-A). The triangular meshes are classified into three categories according to point cloud labels as follows:

- a) **Identically labeled:** all the triangle vertices have the same label implying that all the three vertices are from the same primitive.
- b) **Mixedly labeled:** any two vertices of the triangle have the same label and the third one is from another primitive.
- c) **Distinctly labeled:** triangles with distinctly labeled vertices, which means that each of vertex is from a different primitive.

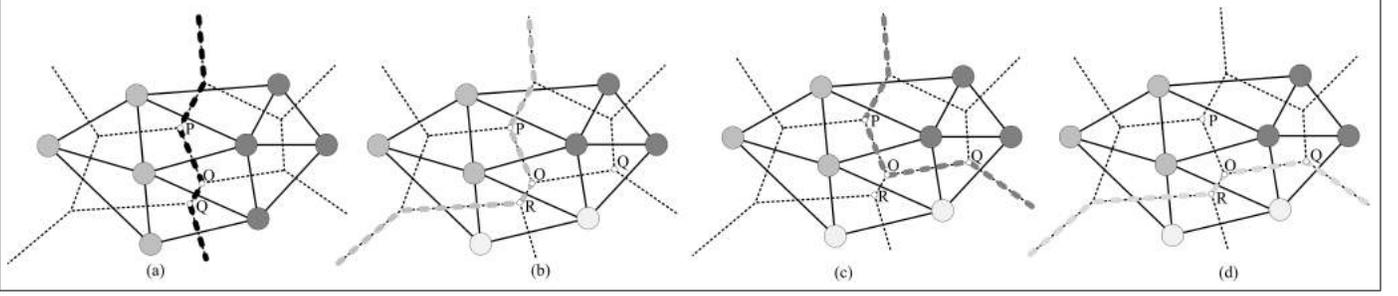
Generally, type a) triangles are usually located within the interior of the primitive. Type b) triangles generally lie in the intersection areas of two primitives (see Fig. 3(a)), however type c) triangles are normally distributed in the regions where three or more primitives meet together (see Figs. 3(b)-3(d)).

② Search an arbitrary type b) or c) triangle. If the triangle meets type b), two auxiliary directed edges  $OP$  and  $OQ$  are inserted, where,  $O$  is the circumcenter of current processing triangle and  $P$  and  $Q$  are the circumcenters of neighborhood triangles from type b) or c), as shown in Fig. 3(a). If the triangle meets type c), as shown in Fig. 3(b)-3(d) three auxiliary directed edges  $OP$ ,  $OQ$  and  $OR$ , where  $O$  is the circumcenter of current processing triangle and  $P$ ,  $Q$  and  $R$  are the circumcenter of neighborhood triangles are inserted. Meanwhile, we also insert the reversed edge of the same coordinates but reverse direction of each interpolated edge.

③ Based on breadth-first search algorithm on Delaunay triangular meshes, the current processing triangle's adjacent triangular meshes are retrieved. We then process these adjacent triangular meshes according to step ②.

④ The steps from ② to ③ are executed repeatedly until all of the type b) or c) triangles have been processed.

Actually, the graph constituted by the interpolated edges is a subgraph of the Voronoi diagram of the building point clouds. As the interpolated edge of the Voronoi diagram is equidistant from its two nearest neighboring vertices, theoretically the subgraph of the Voronoi diagram is the best partition to divide the adjacent primitives. Although the subgraph of Voronoi diagram has such an excellent property, there exists a degenerate



**Fig. 3. Generation of primitive boundaries based on the subgraph of the Voronoi diagram.** Black Solid lines indicate the Delaunay triangulation, black dashed lines indicate the Voronoi diagram of the Delaunay triangulation and the grayscale bold dashed lines represent the subgraph of the Voronoi diagram which constitutes the result of the extracted primitive boundaries. The nodes in different grayscale colors represent different primitives. The points  $O$ ,  $P$ ,  $Q$  and  $R$  are the circumcenters of the processing triangle and its neighboring triangles. (a) The traced boundaries for two different types of rooftop primitives. (b), (c) and (d) are the traced boundaries for three different rooftop primitives.

case when the primitives adjoin to the outer and/or inner hole building boundaries as illustrated in Fig. 4. As shown in Fig. 4(a), part of the blue interpolated edges of the primitives which adjoin the outer or inner hole building boundaries are missing due to lack of adjacent primitive constraints. In addition, even though there exist part of interpolated edges for the primitives adjoining the outer or inner hole building boundaries, these edges have large deviations from the actual primitive boundary, as many long and narrow triangles exist near outer and inner building boundaries (see the enlarged rectangles in Figs. 4(a) and 4(b)). To solve these problems, the only one outer building boundary and the possible one or multiple inner building boundaries will have to be extracted in advance (described in Section III-B2), which are used as the auxiliary primitives before the following two step processes:

- **Densification of boundary points:** More specifically, for each boundary, we verify the Euclidean distance between arbitrary two adjacent points. If their distance is less than the predefined threshold  $\delta$ , additional middle point is inserted into these two adjacent points. This procedure is processed until the distance between any two consecutive points is less than  $\delta$ .

- **Zoomed process:** The densified outer and inner hole boundary points are expanded outward and contracted inward respectively according to the direction of each point normal. We set the outward and inward amplitude value to  $\rho/2$  ( $\rho$  is the mean density of point clouds) to avoid the partial overlaps with the original rooftop point clouds.

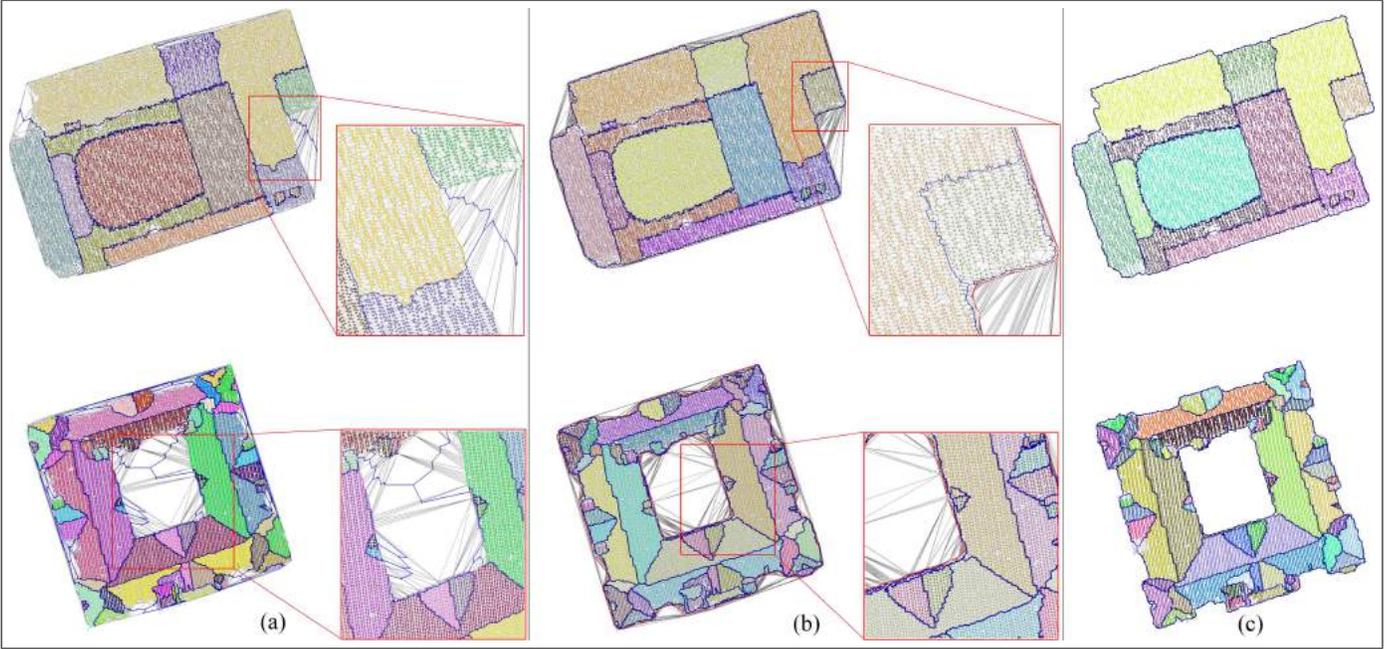
After the two above-mentioned processes, the outer and inner hole boundaries can be used as auxiliary primitives which are added to the segmented primitives to construct the Delaunay triangulation and its dual graph of the Voronoi diagram. As shown in Fig. 4(b), by adding the auxiliary primitives indicated with the red points, the question for primitive boundary tracing has been completely transformed into a problem of tracing boundaries of the “interior” rooftop primitives. We can accurately and completely trace the primitive boundaries (see Fig. 4(c)) no matter where a primitive is located on the rooftop.

The subgraph of the Voronoi diagram can well maintain the topological consistency among primitives by interpolating

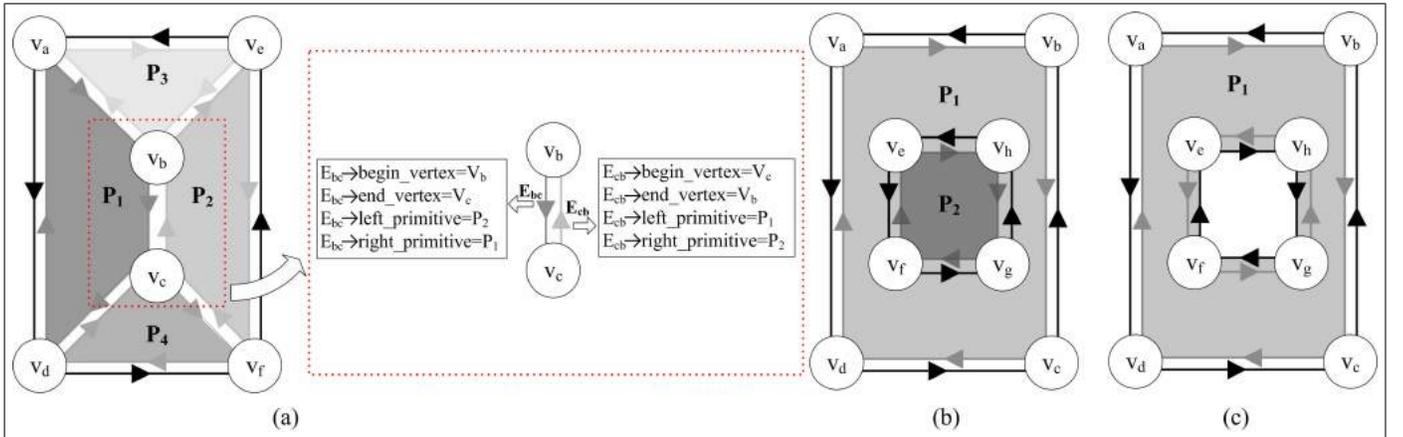
bidirectional edges. If an arbitrary primitive identity is given, the proposed algorithm can efficiently search the corresponding primitive edges in a certain order. Fig.5 gives the scheme of retrieving the interpolated edges of a specific primitive. For example in Fig. 5(a), each interpolated edge has a corresponding dual edge sharing same coordinates but with reverse direction. The data structure for each edge is clearly described in red dashed rectangles. If we want to retrieve the complete boundary of primitive  $P_1$  in Fig. 5(a), the edge set  $V_aV_b$ ,  $V_bV_c$ ,  $V_cV_d$  and  $V_dV_a$  in clockwise order can be easily found according to edge’s right primitive identity tag. If the primitive  $P_2$  is completely surrounded by the primitive  $P_1$  as shown in Fig. 5(b), we can retrieve two parts of boundaries of primitive  $P_1$ : part one ( $V_aV_b$ ,  $V_bV_c$ ,  $V_cV_d$  and  $V_dV_a$ ) and part two ( $V_eV_f$ ,  $V_fV_g$ ,  $V_gV_h$  and  $V_hV_e$ ). We directly discard the part two and only maintain the part one as the final boundary of primitive  $P_1$  because the part two is organized in the counter-clockwise order. The situation is different if the inner hole is included in the primitive  $P_1$  as illustrated in Fig. 5(c). In this case, parts one and two commonly constitute the final boundary of primitive  $P_1$ .

## 2) Building Outer and Inner Hole Boundary Extraction:

We propose a novel algorithm to accurately extract outer and inner hole boundaries by combining the traditional alpha shapes algorithm with Minimum Spanning Tree (MST). As previously mentioned in Section III-B1, the outer and inner hole building boundaries are used as the auxiliary primitives for assisting primitive boundary extraction. Fig. 6 shows the entire process and the intermediate results of the outer and inner hole extraction. Through the alpha shapes algorithm, the initial outer and inner hole boundary points are obtained in red color in Fig. 6(a). Unfortunately, these initial points are unorganized without specific topological relationship, i.e., clockwise or counter-clockwise order among boundary points. In addition, the alpha shapes might produce some pseudo boundary points (see Fig. 6(a)) due to the unevenly distributed point clouds. Despite having a few drawbacks, the alpha shape algorithm can easily control the LoD of the boundary points by simply tuning the parameter  $\alpha$ . We combine the alpha shapes and MST to highlight their respective advantages and



**Fig. 4. Primitive boundary tracing without and within constraints of auxiliary primitives.** (a) and (b) are the rooftop primitive tracing results without and with considering the constraints of auxiliary primitives. (c) The final rooftop primitive boundaries overlaid onto the segmented rooftop primitives. The blue lines are the interpolated edges and the red points from (b) are the auxiliary primitives.



**Fig. 5. Primitive boundary organization and retrieval diagrams.** (a) Topological relation of the primitive boundary for a hip rooftop. (b) Topological relation of the primitive boundary for a two-level flat rooftop. (c) Topological organization of primitive boundary for a flat rooftop with a hole. Note that the different gray level primitives are composed by the corresponding same gray level edges and the edges with black arrows represent the redundant dual edges.

suppress their drawbacks when they are used independently. We use MST to reorganize the alpha shape's results, during which we can accurately extract actual boundary and eliminate the pseudo points as well. In addition, the boundary points' topological relationship is also restored during this process. It should be noted that when constructing MST we should consider two factors, including Euclidean distance and local tangent among the boundary points. As illustrated in Fig. 6(a), the green lines represent the constructed MST, from which the only one outer building boundary and the potential one or more inner hole boundaries are expected to be obtained. To

be specific, all of the MST vertices of degree 1 are extracted and regarded as the starting vertices. From these starting vertices, we traverse each edge in MST and retrieve the subtree according to the following condition:

$$B_i^* = \underset{B_i}{\operatorname{argmin}} \frac{\|B_i^1 - B_i^{|B_i|}\|}{\|B_i\|}, \quad (6)$$

$$s.t. \quad \|B_i^1 - B_i^{|B_i|}\| \leq 2\rho \wedge \|B_i\| \geq \rho M_l,$$

where  $B_i^1$  and  $B_i^{|B_i|}$  are the starting and ending points of the current boundary  $B_i$ .  $\|B_i^1 - B_i^{|B_i|}\|$  is the Euclidean distance

between starting and ending points of  $B_i$ .  $\|B_i\|$  represents the length of  $B_i$ .  $\rho$  represents the mean point density.  $M_l$  is the predefined minimum length of the extracted boundary. Intuitively, the potential boundary with longer length and shorter distance between its endpoints is easier to be extracted by minimizing the Eq. 6. The maximum depth of the subtree has been first retrieved which generally represents the only longest outer building boundary. Similarly, from the remaining MST vertices, the inner hole boundaries are recursively extracted from the starting points of degree 1 by Eq. 6. The final boundaries are shown in Figs. 6(b) and 6(c).

3) **Rooftop Primitive Boundary Segmentation:** After rooftop primitive tracing, if we directly use the interpolated primitive boundaries to construct the rooftop models, the reconstructed rooftop models will not be compact and lightweight, which is not beneficial for storage, web transmission and large-scale rendering. In addition, the interpolated boundaries often represent “zigzag” geometry, which does not conform to the geometric regularity requirement of the man-made architecture. To address this issue, the interpolated primitive boundaries have been further segmented into multiple linear segments according to the reality of primitive boundary. Based on these linear segments, the boundary key points are extracted, which will be described in Section III-B4.

More specifically, given an arbitrary primitive boundary  $B$ ,  $m$  linear segments  $\mathcal{L} = \{L_1, L_2, \dots, L_m\}$  are anticipated to be obtained by the optimal labeling process  $\mathcal{F}$ . An arbitrary linear segment is composed of the point set  $p_i^1, p_i^2, \dots, p_i^{n_i}$ . The goal of the primitive boundary labeling is to divide the primitive boundary into multiple linear segments, from which each segment is consistent with the primitive’s actual boundary expression. Moreover, the points from each segment must keep the identical labels. Similar to Section III-A, We employ an energy function to realize this task, where the energy function is given by Equation 7.

$$E_{\mathcal{L}} = E_{alignment} + \eta_1 E_{smooth} + \eta_2 E_{fidelity}. \quad (7)$$

Next, we will give a detailed explanation of each energy term.  $E_{alignment}$  is composed of the following two energy terms:

$$E_{alignment} = E_{distance} + \lambda E_{direction}. \quad (8)$$

$E_{distance}$  penalizes the inconsistency distance between a point and its associated linear segment and is defined by

$$E_{distance} = \sum_{i=1}^m \sum_{j=1}^{n_i} \left\| -\ln(P_r(p_i^j \in L_i)) \right\|^2, \quad (9)$$

where  $P_r(p_i^j \in L_i) \propto \frac{1}{\sqrt{2\pi}\sigma_i^j} \cdot \exp\left\{-\frac{dist(p_i^j, L_i)^2}{2(\sigma_i^j)^2}\right\}$ .  $dist(p_i^j, L_i)$  is the distance from the point to the linear segments. The point  $p_i^j$  with high probability density  $P_r(p_i^j \in L_i)$  is encouraged to represent the segment  $L_i$ .

$E_{direction}$  penalizes the inconsistency between the current boundary point’s local tangent and the normal vector of linear segment and is given by

$$E_{direction} = \sum_{i=1}^m \sum_{j=1}^{n_i} \left\| N_{p_i^j} \cdot N_{L_i} \right\|^2, \quad (10)$$

where  $N_{p_i^j}$  represents the local tangent at point  $p_i^j$ .  $N_{L_i}$  represents the normal vector of linear segment  $L_i$ . If they have large deviations, the heavy penalties will be imposed. “ $\cdot$ ” is the dot product of two vectors.

As the interpolated primitive boundaries explicitly store the topological relationships among neighboring points in clockwise or counter-clockwise order, we only use current processing point’s two consecutive points as neighborhood to design the following smoothness term:

$$E_{smooth} = \sum_{i=1}^m \sum_{j=1}^{n_i} \left\| \exp\{-\tau\} \right\|^2, \quad (11)$$

$$\tau = dist(p_i^{j-1}, p_i^j) \cdot \delta(\mathcal{F}(p_i^{j-1}) \neq \mathcal{F}(p_i^j))$$

$$+ dist(p_i^j, p_i^{j+1}) \cdot \delta(\mathcal{F}(p_i^j) \neq \mathcal{F}(p_i^{j+1})),$$

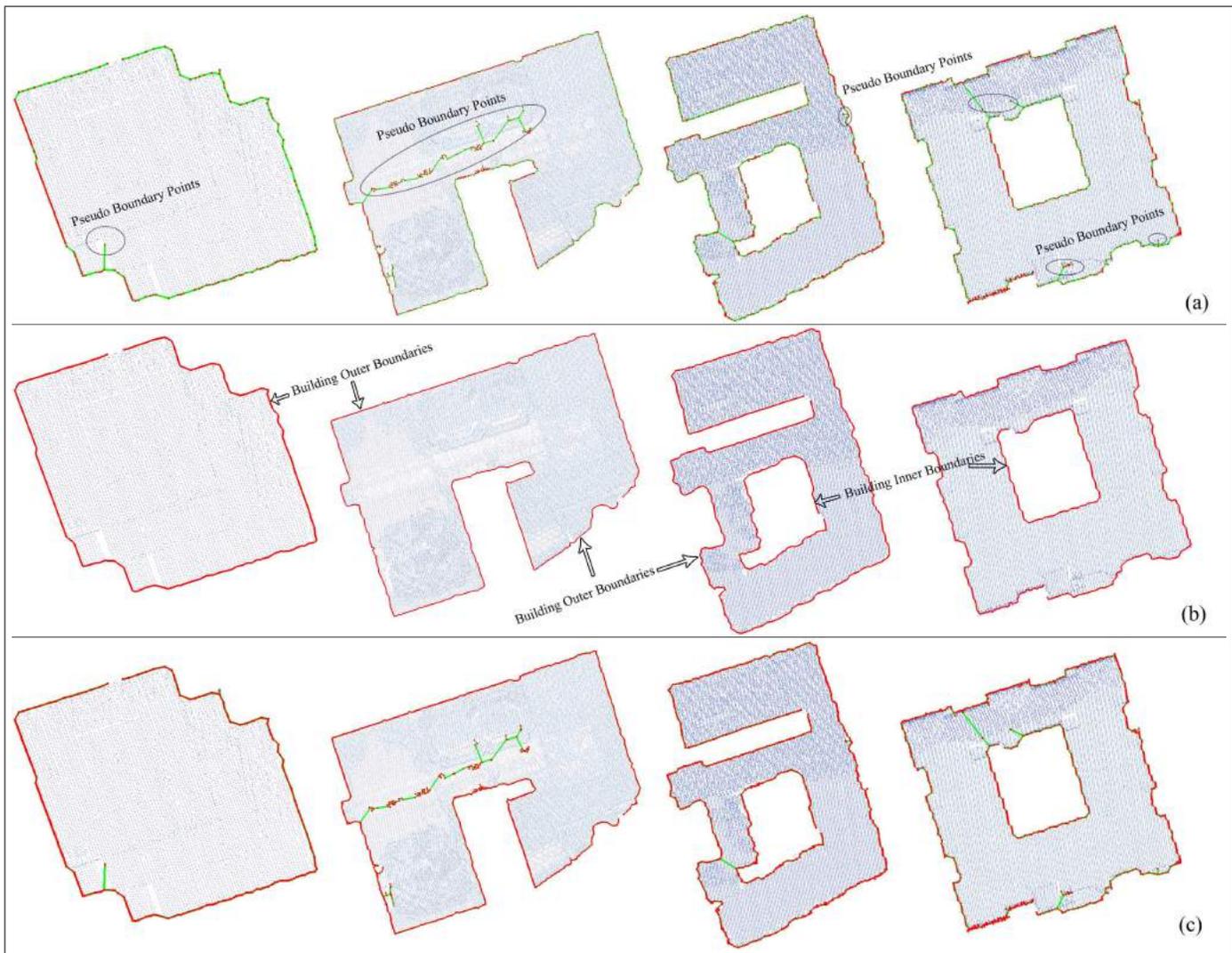
where  $\mathcal{F}(p_i^j)$  presents the labeling optimization process for point  $p_i^j$ . Intuitively, this term is inversely proportional to the distance between the neighboring points. To minimize Eq. 11, neighboring points are encouraged to have the similar labels. If we only consider the two above-mentioned terms for optimization, the linear segments tend to be over-segmented as illustrated in the regions highlighted by black rectangles in Fig. 7(a). To prevent producing the trivial, linear segments and maintain the linear segments consistency with the primitive’s actual boundary, an additional fidelity term (Eq. 12) is employed.

$$E_{fidelity} = \sum_{\substack{i=1 \\ L_i \in \mathcal{L}}}^m \left\| \exp\left\{-\frac{|L_i|}{\max(|L_i|)}\right\} \right\|^2 \cdot \delta(\mathcal{F}), \quad (12)$$

$$\delta(\mathcal{F}) \stackrel{def}{=} \begin{cases} 1, & \exists p_i^j : \mathcal{F}(p_i^j) = L_i \\ 0, & \text{otherwise,} \end{cases}$$

where  $|L_i|$  represents the point number in  $L_i$ .  $\max(|L_i|)$  represents the number of points from the maximum length of  $L_i$ .  $E_{fidelity}$  penalizes the linear segments of relatively small size, which tend to be merged with relatively large linear segments to eliminate the redundant labels (over-segmentation). Although the number of linear segments is reduced dramatically (see Fig. 7(b)) due to incorporating the fidelity term, some disconnected linear segments may be merged together. In Fig. 7(b), the separated linear segments with same label (a combination of ‘s’ with a number in red color) have been falsely merged. Ideally, we want to assure that linear segments are merged only when they are closely connected. Otherwise it will cause under-segmentation. To solve this under-segmentation issue, after optimization, we construct an undirected graph using the labeled points. Based on an undirected graph, the primitive boundary points of the same label are re-clustered using the principle of spatial continuity. The final linear segments are shown in Fig. 7(c). The initial labels of primitive boundary are provided by the Random Sample Consensus (RANSAC) algorithm. The alpha-expansion algorithm is used to solve the Eq. 7, which transforms the optimization into the max-flow min-cut problem based on graph data structure.

4) **Hybrid Representation of Primitive Boundaries by Key Points:** Once the segments of primitive boundaries are available, the key points of primitive boundaries are needed. Using



**Fig. 6. The building outer and inner boundary extraction.** (a) The red point clouds indicate the results from alpha shapes. The green lines represent the MST edges and the blue point clouds are the raw data. (b) The building outer and inner boundaries are recursively extracted in red lines. (c) The intermediate and final results are overlaid together.

the key points, we can simplify and regularize each primitive boundary to generate compact and lightweight rooftop models in Section III-C2. The key point extraction algorithm is described in the following:

① Organize primitive boundary segments (see Section III-B3) by an undirected graph.

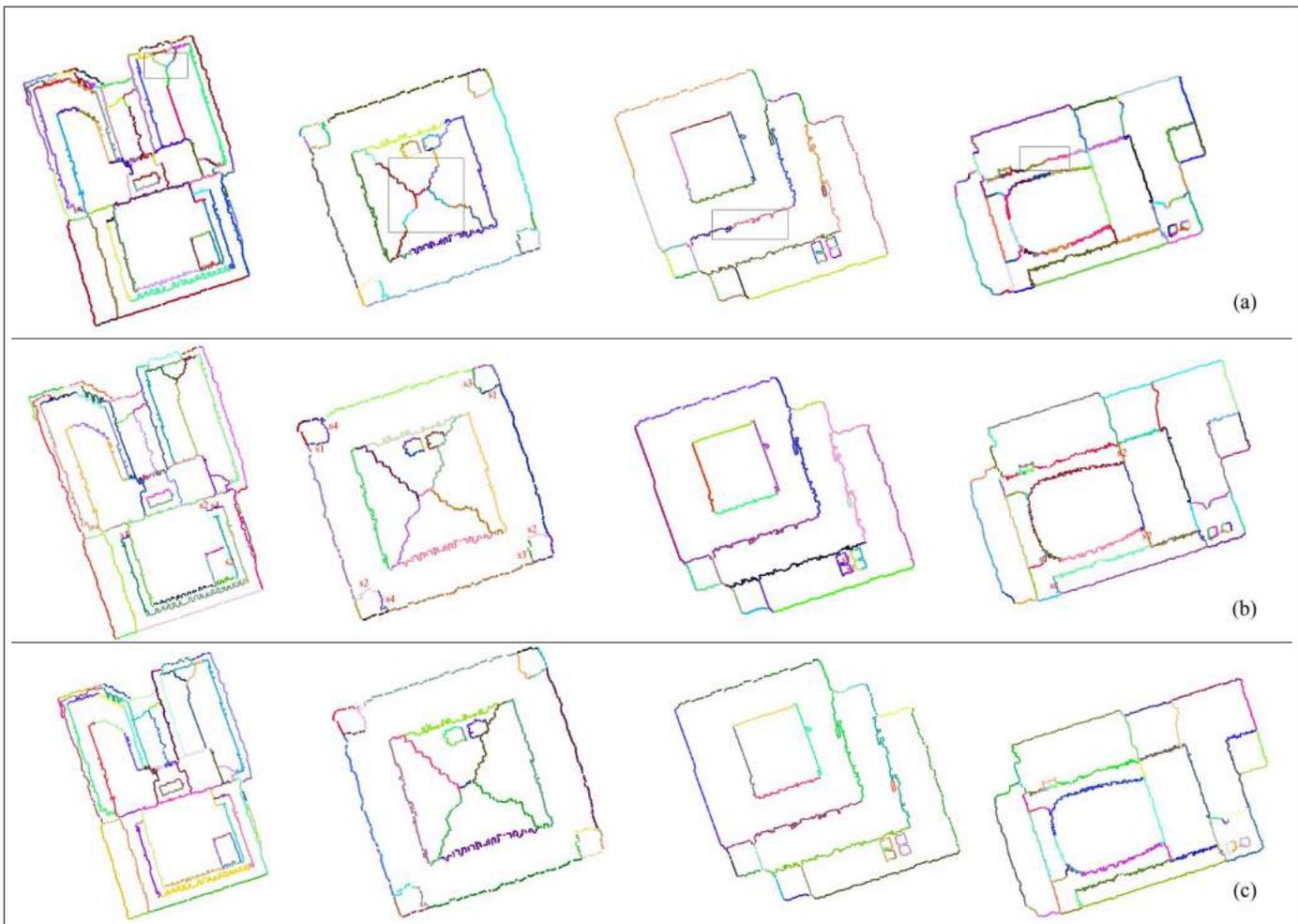
② Each segment is fitted by a linear least square regression whose linear parameters and the corresponding residual standard deviation (RSD) reflect the reliability of the fitted regression line. If the fitted RSD is less than the threshold of reliability ( $\epsilon$ ), the current segment's starting and ending points are projected onto the fitted line. Both the original primitive boundary points and their corresponding projected points are stored in a Look-up Table (LUT)<sup>1</sup>. On the contrary, if the RSD is greater than  $\epsilon$ , all of the current segment points are stored

<sup>1</sup>The LUT is defined as the specific mapping data structure, from which each element includes a pair of points, i.e., the original primitive boundary point and its projected point.

into the LUT. At this point, the original boundary points and their corresponding projected points in LUT should be kept identical.

③ The vertices in an undirected graph of degree 3 are the connection points that link at least three primitives and hence, these points are critically important for maintaining the multiple primitives topological consistency and need to be stored onto LUT. To acquire the projected points of degree 3, the labels of the direct connection points of degree 3 have been determined. Then the reliability of the linear regression equations associated with these labels is verified. The point of degree 3 is projected onto the fitted regression line with highest reliability. If none of fitted lines is reliable, we do not project point of degree 3 onto any fitted lines and directly store it into LUT. In this case, the points of degree 3 and their corresponding projected points in LUT should be kept identical.

Although each item in LUT corresponds to one key point,



**Fig. 7. Primitive boundary segmentation by the proposed optimization.** (a) The over-segmentation phenomena can be observed without considering fidelity term. (b) The under-segmentation phenomena occur when adding the fidelity term. (c) The final results can well balance between over-segmentation and under-segmentation. The black rectangles represent the place where the over-segmentation frequently occurs. The red codes in (b) that combined letter and number are linear segments with under-segmentation.

these LUT key points do not have the topological relationship. We can restore these key points topology by simply traversing the original primitive boundary once. Thereafter, each primitive boundary can be represented by the corresponding key points thereby achieving the goal of simplification and regularization of primitive boundary.

Essentially, the key point extraction algorithm is a type of hybrid representation of primitive boundary. That is to say, this hybrid representation combines part of original primitive boundary points with the projected points to achieve a more flexible representation. For example, the original key points can vividly and accurately delineate the primitive details when the fitted primitive boundary linear line is not reliable. On the other hand, the projected key points realize the generalized representation by using the high reliable fitted line segments, which can significantly reduce the number of generated triangles and ensure the regularity of the primitive boundary.

### C. Building Rooftop Geometric Modeling

In this section, we construct the 2.5D building geometric rooftop model by the primitive key points. To achieve this goal, we first assemble the primitive boundaries to generate the basic entities for triangulation. We then use Vertex Buffer Object (VBO) of Graphics Processing Unit (GPU) to render the large-scale building triangular meshes.

1) **Primitive Boundary Assembly:** The rooftop primitive boundary should be assembled to form the elementary primitive entity as the basic structure unit for triangulation. The following three types of topology assembly principles are considered while assembling the primitive boundaries.

- **Primitive occlusion processing:** As shown in Fig. 8(a), for simplicity, we use a two-level flat rooftop to explain the process. If current rooftop primitive  $P_1$ 's boundary  $B_1$  is entirely contained by another primitive  $P_2$ 's boundary  $B_2$  and  $H_{B_1} < H_{B_2}$ , where  $H_{B_1}$  and  $H_{B_2}$  are the mean heights of boundaries  $B_1$  and  $B_2$ , respectively, then  $B_2$  and  $\hat{B}_1$  should be combined to form the entity of primitive  $P_1$ .  $\hat{B}_1$  is a reverse

boundary of  $B_1$ , which is reorganized in counter-clockwise order. In most complicated cases, for example, a sunken roof window within the inclined rooftop as shown in Fig. 8(b), we compare the height of the contained primitive  $B_1$  with its projection onto  $B_2$ , i.e., a red transparent rectangle to assemble the primitive boundaries. Based on this principle, we can process not only multi-level flat rooftops but also inclined roof primitives or even the rooftops with the combinations of flat and inclined roofs. Using the proposed occlusion processing strategy, the contained primitives can be well maintained otherwise the rooftop details in reconstructed rooftop models will be inevitably occluded.

- **Inner hole representation:** If the current inner hole boundary is completely included by other primitive boundaries, the inner hole boundary should be assembled with the innermost contained primitive boundary. As shown in Fig. 8(c), inner hole  $B_0$  is completely contained by  $B_1$ , in this case  $B_0B_1$  should be assembled to form the entity of primitive  $P_1$  and the building inner hole can be represented as well. Otherwise, we simply ignore the current inner hole boundary because the building inner hole has already been implicitly expressed by multiple primitives  $P_1$  to  $P_4$  as shown in Fig. 8(d).

- **Abstract processing:** For some small and irregular primitives, for instance, the roof chimneys, air-conditioner or water storage tanks on rooftops, we use the bounding rectangle under the constraint of building dominant direction  $B_{rect}$  to substitute the actual primitive boundary  $B_{actual}$  as shown in Fig. 8(e). To obtain the building dominant directions, we observe that edges of building footprint have the minimum projection in building dominant and its orthogonal dominant directions, which means that the dominant edges of building footprint basically determine the orientation of the building. So we do the loop integral of the footprint edges, i.e., outer and inner building boundaries to obtain two mutually orthogonal dominant directions by minimizing the equation below:

$$\theta^* = \operatorname{argmin}_{\theta} \sum_{i=1}^{|B|} \sum_{j=1}^{|B_i|} \|S_{ij} \cdot \cos \theta + S_{ij} \cdot \sin \theta\|^2, \quad (13)$$

where  $|B|$  represents the total number of outer and inner boundaries of the building under processing.  $|B_i|$  denotes the total number of vertices of  $i$ th building boundary.  $S_{ij}$  indicates the length of  $j$ th edge in  $i$ th boundary and  $\theta^*$  is the expected variable of building dominant direction. To minimize Eq. 13, we employ the Newton's algorithm. We find that after ten iterations, Eq. 13 can reach an accuracy of  $\pi/360$ .

The rooftop primitive topology implies the relation among primitive boundaries, triangulation entities and primitive planes. Specifically, the triangulation entity is composed of one or more primitive boundaries. That is to say, many-to-one relation exists between primitive boundaries and the triangulation entity. The triangulation entity and the primitive plane assume a one-to-one correspondence. These topological relations lay a solid foundation for maintaining the model's topological consistency in the third (elevation) dimension. In fact, the essence of the rooftop reconstruction proposed in this paper revolves around the maintenance of rooftop primitive's

topology in two-dimensional space. We then carry out the following two-step topological compatibility processing to ensure the topological consistency of model's elevations: ① Adjust each primitive boundary point's elevation according to the corresponding primitive plane equation. ② For the key points of the primitive boundary from shared ridge or step edges, their elevations are equal to the mean of two projected elevations on their associated planes provided that their mean elevation residual deviation is within the specific predefined threshold ( $\eta$ ). The procedure ensures a complete topological consistency for the primitive boundary key points, in both, the two- and three-dimensional space.

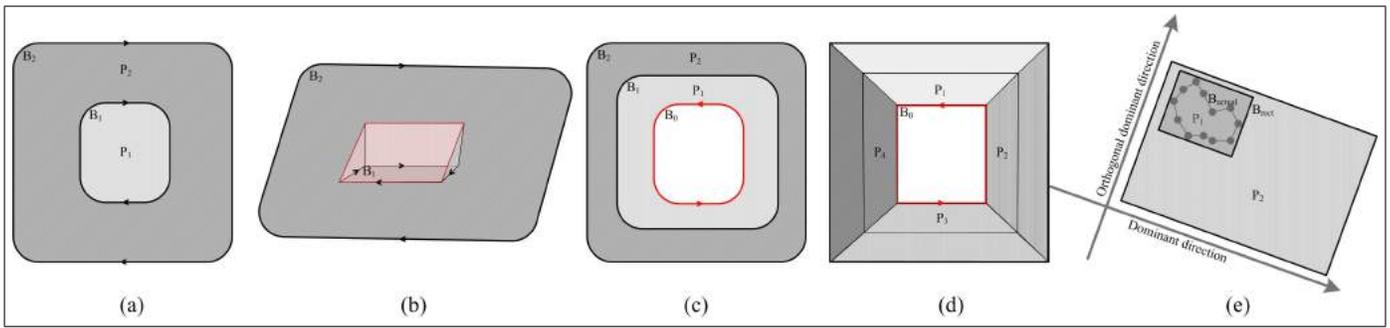
2) **Building Model Generation:** The entire 2.5D building models include rooftop and façade models. For reconstructing the rooftop models, the constrained triangulation algorithm is employed to triangulate the primitives of convex geometries, concave geometries and geometries with one or more inner holes. For reconstruction of building façades, we stretch the inner and outer boundaries to a place where the corresponding terrain exists.

#### IV. PERFORMANCE EVALUATION RESULTS AND DISCUSSIONS

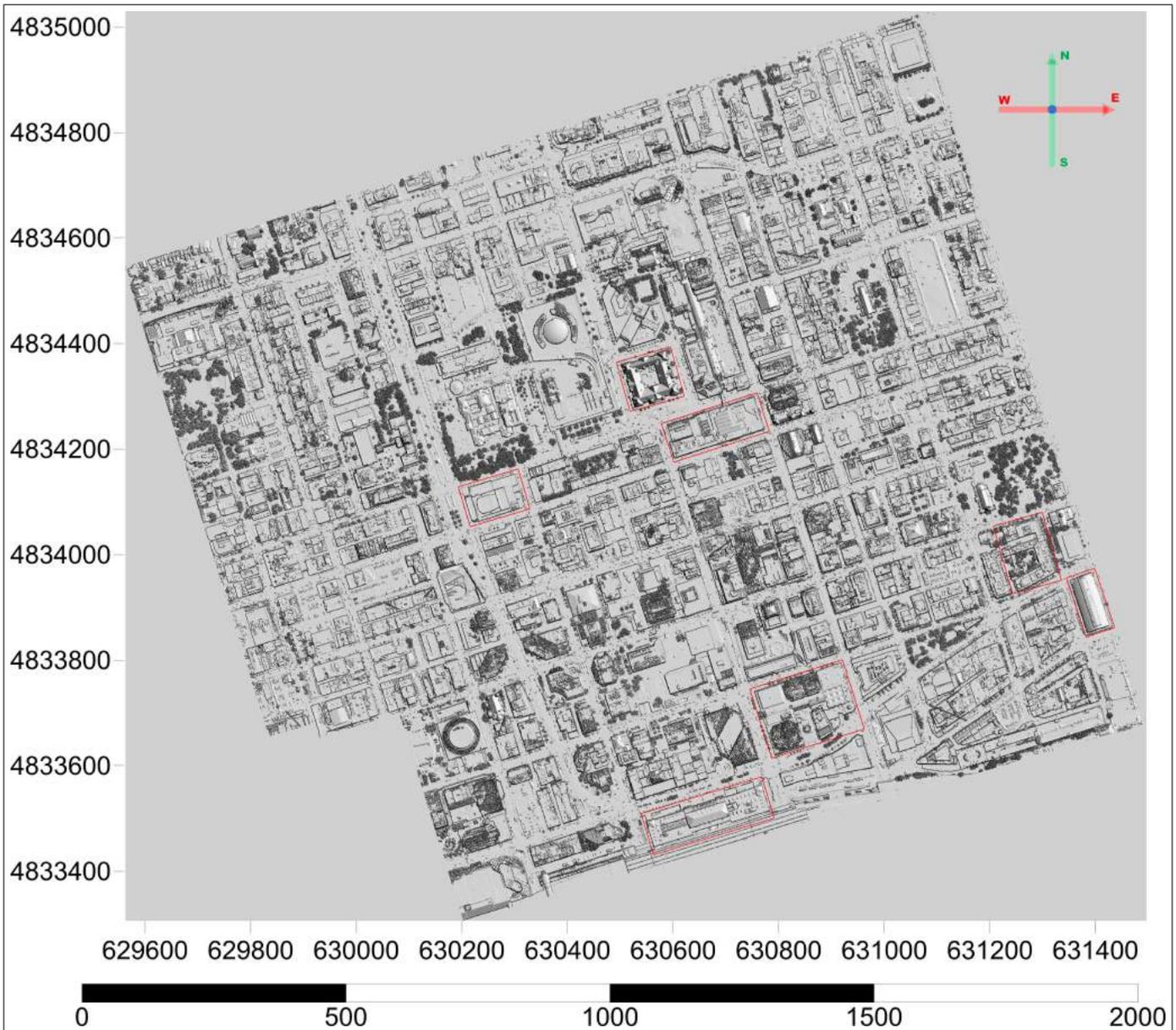
In this section, various performance results obtained through qualitative and quantitative evaluation procedures are presented and discussed. In particular, firstly the technical characteristics of the ALS data will be reviewed. Then, a qualitative and quantitative evaluation of reconstruction performance will be presented and analyzed in a systematic manner. This include discussions on total thresholds and the selection of their most appropriate values, as well as a detailed presentation of the numerous experiments which have been carried out in order to assess the performance of the proposed methodology.

The Toronto city data set (see Fig. 9) is acquired by using Microsoft Vexcel's UltraCam-D (UCD) camera and the Optech's Airborne Laser Terrain Mapping (ALTM) Orion M-Series scanner with a pulse repetition rate of 50 kHz and an average height of 850 m. The data set covers an area of approximately 2.0 km<sup>2</sup>, in which a total of 13,859,691 points were acquired resulting in an average density of 6.0 points/m<sup>2</sup>. Based on the sensor and flight specifications, the horizontal and vertical accuracies of 0.43 m and 0.15 m, respectively, are guaranteed. The overall elevation is almost flat, but the northern portion of the area is slightly higher than the southern portion. This particular scene is chosen since it contains a mixture of low- and high-storied buildings with a wide variety of rooftop structures in different orientations, making it an appropriate scene to assess the proposed reconstruction algorithm.

Before analyzing the performance of rooftop reconstruction, we first analyze all the relevant thresholds and explain how to select their appropriate values. For an enhanced PDC clustering algorithm, three parameters namely the correlation coefficient  $k$ , the minimum primitives  $M_p$  and the neighborhood size  $\mathcal{R}$  are used for rooftop clustering. The parameter  $k$  controls the strictness of the clustering criterion. Selecting a small or large value will cause over- or under-segmentation,



**Fig. 8. Topological combination scheme.** Occlusion adverse effect may occur in (a) and (b). Inner hole (a red line) should be assembled with other primitive boundary in (c). (d) The inner hole does not need to be considered when it is implicitly represented by other primitives. (e) Abstract processing of a trivial primitive boundary. The red lines represent the inner hole lines which are organized in the counter-clockwise order.



**Fig. 9. The shaded relief map of Toronto data set.** The shaded relief map of Toronto downtown areas. The representative buildings illustrated by the red rectangles are selected for the subsequent rooftop cluster (see IV-C1) and model accuracy (see IV-C2) evaluations.

respectively. Fortunately,  $k$  is less sensitive over a wide range of values. Actually, parameter  $k$  is encouraged to be set with a relatively small value because it tends to produce over-segmentation. In this case, the redundant primitives can be well merged or eliminated after the optimization. In contrast, with large value of  $k$ , once under-segmentation occurs, the optimization cannot produce more primitives than the initial number of clusters. Through trial and error experiments, the value of 1.0 for  $k$  can well achieve a balance between the accuracy and the topological consistency of primitives.  $M_p$  is selected via a statistical analysis of primitive size from the data set. Parameter  $\mathcal{R}$  will affect the primitive topological consistency, as shown in Fig. 2(b). In fact, most of the inconsistent topologies are due to selection of an inappropriate  $\mathcal{R}$ . This parameter is also used in smoothness term during primitive optimization, where a large or small value will probably lead to under- or over-segmentation. To choose an appropriate value, we use Eq. 14 proposed by Demantke et al. [48] which minimize the function  $\mathcal{R}^* = \operatorname{argmin}(E_f)$  to dynamically obtain current point's optimal neighborhood size  $\mathcal{R}^*$ , where

$$E_f = -\alpha_{1D}\ln(\alpha_{1D}) - \alpha_{2D}\ln(\alpha_{2D}) - \alpha_{3D}\ln(\alpha_{3D}), \quad (14)$$

parameters  $\alpha_{1D}$ ,  $\alpha_{2D}$  and  $\alpha_{3D}$  are dimensionality features that indicate whether the geometric shape of current point and its neighborhood points is linear, plane or spherical, respectively.

For rooftop primitive boundary extraction, the building outer and inner boundaries need densification processing before being used as the auxiliary primitives (Section III-B1). The densification parameter  $\delta$  controls the density of the produced auxiliary primitives to prevent producing long and narrow triangular meshes that can impair the accuracy of the traced primitive boundaries. We find that  $\delta < \rho/2$  is sufficient to maintain the compactness of the auxiliary primitives. For initial building inner and outer boundary extraction (Section III-B2), the  $\alpha$  is suggested to be chosen in the range  $[2\rho, 3\rho]$ , which can strike a balance between maintenance of relevant boundary points and suppression of pseudo boundary points. The minimum length of the extracted inner and outer boundaries  $M_l$  are restricted to at least 5 points. If the length of the extracted building boundary is less than  $M_l$ , we do not regard it as the actual building boundary. In Section III-B3, the RANSAC algorithm provides the initial labels for further primitive boundary optimization. The minimum boundary length in RANSAC algorithm is also set to  $M_l$ . For hybrid representation of primitive boundary in Section III-B4, we set standard deviation error  $\varepsilon$  to 0.4, which maintains a balance between compactness of the building models and redundant rooftop details. A smaller  $\varepsilon$  value enforces the model to be represented by more primitive boundary points, while a larger value makes the building model expressed by more fitted points. For modeling construction in Section III-C, only one parameter  $\eta$  is used to control the topological consistency of the elevation, which is set to the precision of the data set. According to specification of the Toronto data set, we directly set  $\eta = 0.15$  m. All the parameters are thoroughly listed in Table I.

For evaluation of building reconstruction, there are no standard procedures available to compare the performance and accuracy. There are many well-known reasons to explain the difficulty of not having a universally accepted methodology for making such comparisons. For example, the factors such as input data sources, different types of reconstructed models and level of human-machine interaction, possibly result in making unfair comparisons. Although difficult, numerous internal measures are investigated for approximately evaluating the reconstructed building models on the premise of lacking ground truth. To date, the frequently used quality measures are in the following: 1) root mean square (RMS) of the surface fitting error denoting the orthogonal distance from the ALS point clouds to the corresponding primitives [26], [49]; 2) computational time [26], [42]; 3) compactness of the obtained models [10], [26], [42]; 4) the shortest distance from the reconstructed vertices to the ALS point clouds on the corresponding primitives [49]; and 5) Type I or II errors [50].

Based on the above-mentioned measures, we propose the following three criteria to comprehensively evaluate the proposed reconstruction methodology: 1) whether the model can be represented at multiple levels of details (LoD); 2) whether the model is compact; and 3) whether the model is accurate and well fits the input point clouds. Please note that it is extremely difficult to make an explicit definition of computational time due to different data structures, hardware configurations and level of human-machine interaction. We do not compare with other related methods in terms of computational efficiency.

#### A. Level of Detail Model Representation

Different types of buildings can be modeled by different LoDs ranging from simple building blocks to detailed building façades and interior structures according to the actual building's appearances and functions. LoD of building models is determined by factors, including data acquisition cost, time and labor investment and specific applications. For example, when the building models are used in outdoor navigation, the simplest 3D block representations probably meet the requirements. When considering preservation, restoration or archive of historical heritage buildings, the most detailed building models should be considered. Moreover, in visualization and rendering fields, it needs building models at different LoDs rather than a single level model representation. In this case, a series of multilevel models should be generated on the fly to enhance the user's experience. So whether or not the reconstruction method has capability to model building in such a way that different LoD models can be derived automatically is a very important criterion for evaluating the flexibility of the proposed algorithm.

Fortunately, the proposed 2.5D rooftop reconstruction framework can automatically generate models at five levels of detail to realize the multi-scale (resolution) rooftop representation. Fig. 10 shows the five levels of reconstructed models. The first type of building models in Fig.10(a) with the lowest-level of details are constructed by stretching the building bounding rectangles, under the constraints of two

| Parameters              | Values           | Descriptions  |
|-------------------------|------------------|---|
| $\rho$                  | -                | The mean density of point clouds, which is closely related to the raw data.                       |
| $k$                     | 1.0              | Correlation coefficient which is used to control the strictness of clustering.                    |
| $M_p$                   | 15               | The minimum size of extracted rooftop primitive.  |
| $\mathcal{R}$           | -                | Neighborhood definition which has been dynamically calculated by Eq. 14.                          |
| $\delta$                | $(0, \rho/2]$    | Densification criterion parameter for generating auxiliary primitives .                           |
| $\alpha$                | $[2\rho, 3\rho]$ | Alpha shapes algorithm parameter for getting the initial building boundary points.                |
| $M_l$                   | 5                | The minimum length of the building boundary; The RANSAC algorithm parameter.                      |
| $\varepsilon$           | 0.4              | The reliability threshold of RSD.   |
| $\eta$                  | -                | The topological consistency of the elevation whose value is set to the precision of point clouds. |
| $\eta_1/\eta_2/\lambda$ | 1.0              | The weighted coefficients in Eqs. 2, 7 and 8  |

TABLE I: **Parameters for the proposed topologically-aware rooftop reconstruction algorithm.**The symbol “-” represents the value is determined the individual data set.

dominant directions to the mean height of the building point clouds. The second-level of abstraction in Fig.10(b) is also the flat rooftop models formed by stretching building inner and/or outer boundaries to the mean height of the building point clouds. Compared to the lowest-level models, the planar location of the second level model is more accurate. The third level models in Fig.10(c) are represented by multi-height flat roofs. It is obvious that the rooftop structure details have been dramatically enhanced. Each flat substructure has been constructed from a corresponding rooftop primitive boundary using the first level building model reconstruction method. The models with the fourth-level of details in Fig.10(d) are generated from the interpolated primitive boundaries. These types of models are the polyhedral rooftop models made up of flat roofs, multi-height flat roofs, inclined roofs and/or their combinations. Although the model’s accuracy is very high, these types of models are neither compact nor regular. Therefore, they are rarely used in the actual projects. The fourth-level models are refined by a hybrid representation to form the highest-level abstraction in Fig.10(e). These types of models inherit the advantage of polyhedral model representation, significantly reducing the number of triangular meshes and maintaining the model regularity to some extent.

The multi-level definition in our paper is different from the definition of CityGML [51] which is mainly used for the representation of a set of virtual city models. Our first three levels of definition are corresponding to the definition of CityGML LOD1, representing the building volume with flat roofs. However, the fourth and fifth levels of detailed models correspond to CityGML LOD2 which provides additional details with piecewise-planar roofs. Although the highest level of LOD3 rooftop provides fine details of roof superstructures (e.g., pipes, lift cages, chimneys, dormer windows and other decorations on the rooftop) and extrusions or intrusions of doors and windows on facades, it is very challenging to reconstruct LOD3 or even higher level detailed models from ALS data as the density of the point clouds is relatively sparse and the façade points are not available in most cases.

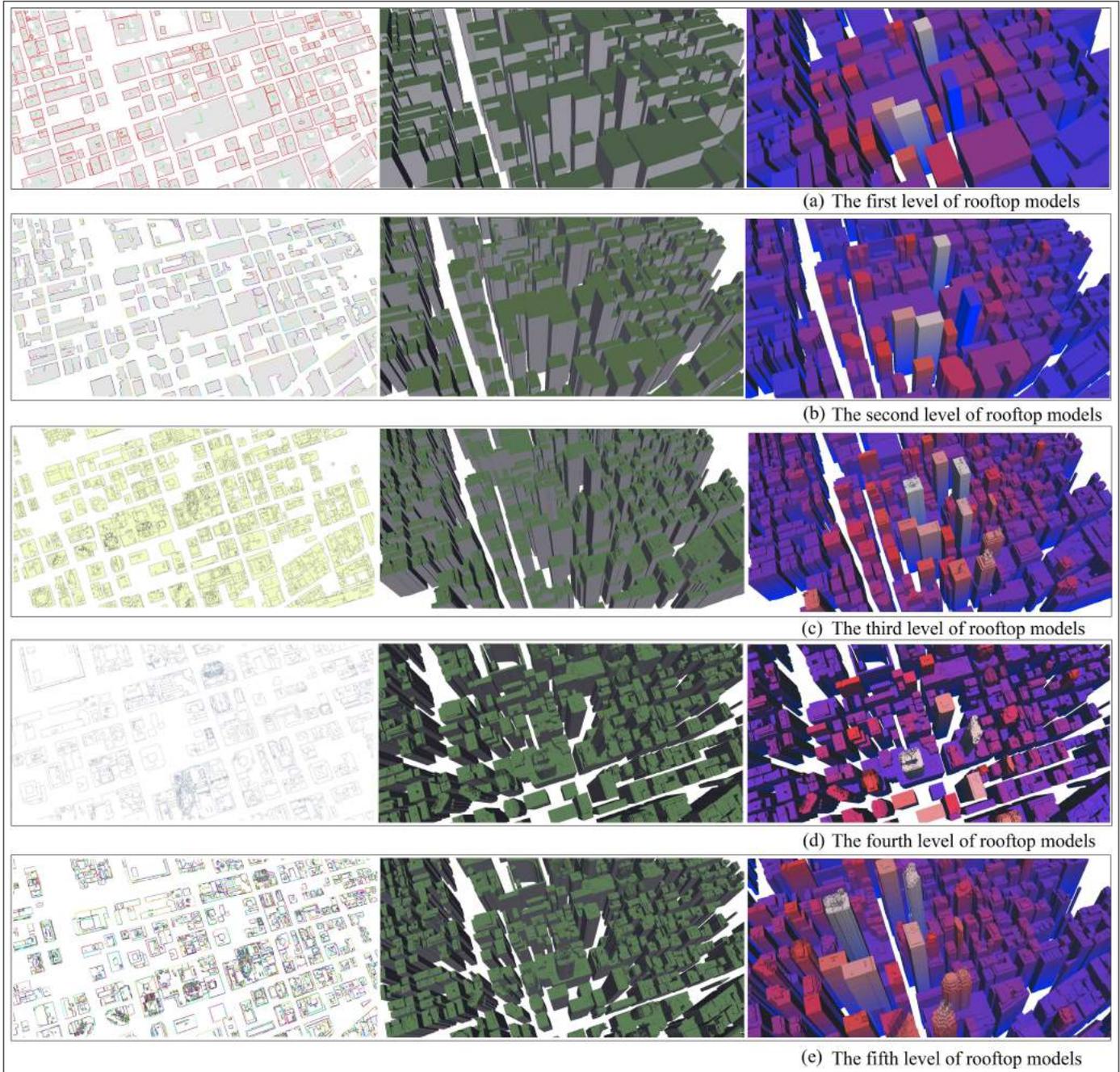
### B. Model Compactness Evaluation

The compactness measure is a crucial index to determine whether or not the reconstructed building model is lightweight. The reconstructed models with fewer triangles tend to be more compact. The lightweight models are suitable for storage,

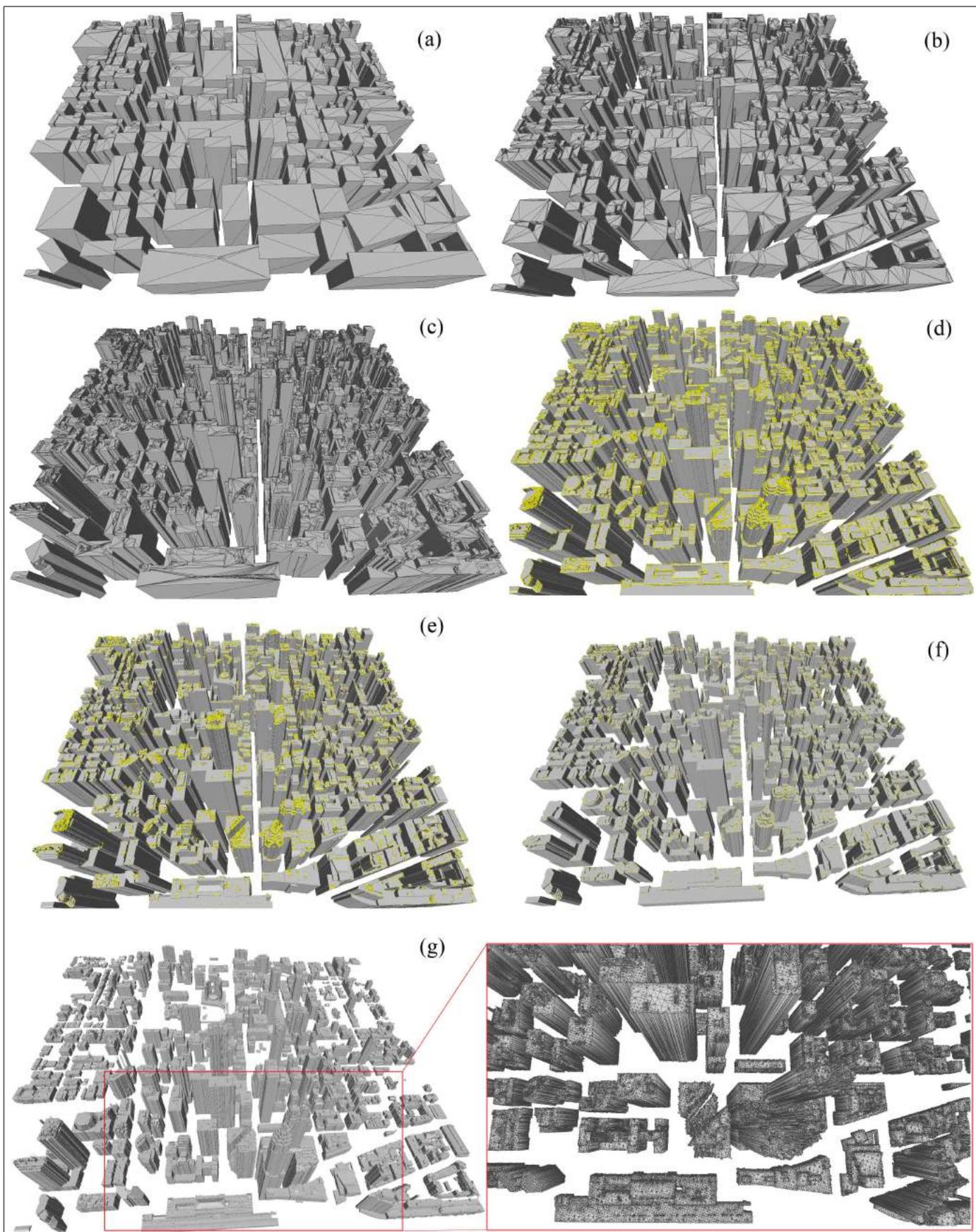
web transmission, acceleration of rendering and other relevant mobile applications.

To effectively evaluate the compactness of the reconstructed building models, we analyze the number of vertices, and triangular facets of rooftop and façade from the Toronto data set. Meanwhile, we compare with other two methods in terms of model compactness. Fig. 11 represents the compactness of the reconstructed models with different algorithms. Figs. 11(a)-11(e) features the models generated by the proposed method at five different levels of details. At the first three levels of the generated building models, the number of roof and façade triangular facets are very low. For instance, the absolute maximum value in the third level is only 140,650 (see Table II). The fourth level is the polyhedral building models constructed by all the interpolated primitive boundary edges. Therefore, the total number of the triangular facets increases to 4,791,058. As previously mentioned, the fourth-level models are neither compact nor regular, which are rarely used in actual applications. Therefore, we simplify and regularize these types of models by the hybrid key point representation. After refinement, the compactness and regularity of the reconstructed models have been significantly enhanced. The number of triangular facets is significantly reduced down to 693,815, which only accounts to 14% of the fourth level model.

We further compare the compactness of reconstructed building models with the methods proposed by Poullis [26] and Zhou [36]. As illustrated in Table II, although the number of triangular facets is gradually increasing for the first three levels models, triangular facets of the third-level model only take up to 14% of triangular facets in Zhou’s method and account for just 60% of triangular facets in Poullis’s method. In the highest level models, the number of triangular facets reaches up to 693,815, which only account for 68% of triangular facets in Zhou’s method. Although this number is nearly triple than Poullis’s triangular facets, our highest level models contain more details than the models created by Poullis. That is to say, we achieve a good balance between rooftop details and model compactness. When the model accuracy is not our main concern, this number can even be reduced by tuning the parameter  $\varepsilon$  in Table I. In our method and Poullis methods, we also notice that the number of vertices is equal to the number of façade triangular facets as these two algorithms reconstruct façades by stretching the boundary vertices into their ground surface.



**Fig. 10. The multi-level rooftop representation.** The leftmost column represents the most important 2D elements for producing each level of 2.5D rooftop models. The left column subfigures, from top to bottom, are the building bounding rectangles under constraints of two dominant directions, the segmented inner and outer building boundaries, the rooftop primitive bounding rectangles under constraints of building dominant directions, the primitive boundaries and the segmented primitive boundaries. The middle column is the reconstructed 2.5D rooftop models whose roofs and façades are rendered by green and gray colors, respectively. The rightmost column is the corresponding rooftop models whose colors are filled according to their elevation.



**Fig. 11. Comparisons of 2.5D rooftop model compactness with different methods.** Our five level model results are shown in (a)-(e). The reconstructed models in (f) are generated using the method proposed by Poullis [26]. The models in (g) are generated by the 2.5D Dual Contouring algorithm proposed by Zhou [36]. The 2.5D contouring algorithm is based on the principle of mesh simplification and the simplified triangular meshes are shown vividly in a red enlarged rectangle. The yellow points from (d)-(f) are the vertices of building models.

| Performances           | The Modeling Algorithms |              |             |              |             |              |           |
|------------------------|-------------------------|--------------|-------------|--------------|-------------|--------------|-----------|
|                        | First level             | Second level | Thrid level | Fourth level | Fifth level | Poullis [26] | Zhou [36] |
| #Vertices              | 2,920                   | 22,060       | 106,068     | 3,207,184    | 475,442     | 160,808      | 572,489   |
| #Roof facets           | 866                     | 10,450       | 34,582      | 1,583,874    | 218,373     | 74,438       | -         |
| #Façade facets         | 2,920                   | 22,060       | 106,068     | 3,207,184    | 475,442     | 160,808      | -         |
| #Rooftop+Façade facets | 3,786                   | 32,510       | 140,650     | 4,791,058    | 693,815     | 235,246      | 1,014,436 |

TABLE II: **Model compactness statistics of three rooftop reconstruction methods.** *First level - Fifth level*: represent the multi-level building models constructed by the proposed method. The symbol “-” represents the value is not available from the corresponding algorithm.

### C. Accuracy Evaluation

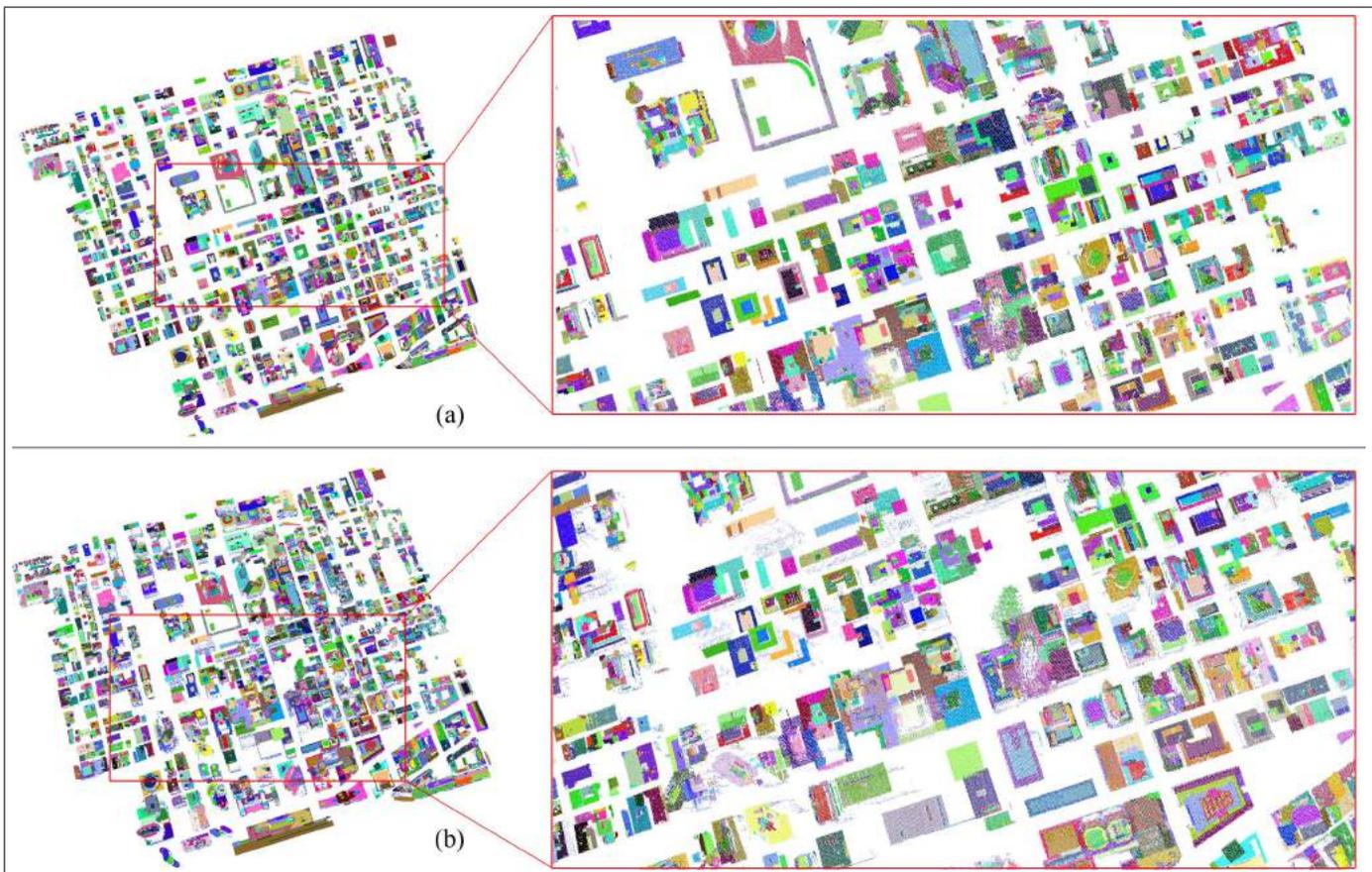
Theoretically, the accuracy of the reconstructed models is affected by each step of the reconstruction pipeline. However, since the proposed methodology is based on a data-driven framework, the accuracy of final models is mainly determined by the clustering algorithm and the modeling strategy. Therefore, in this section the accuracy of clustering results and reconstructed models are separately evaluated by qualitative and quantitative approaches.

1) **Clustering Accuracy Evaluation**: To qualitatively evaluate the rooftop clustering algorithm over the large-scale Toronto data set, 331 rooftops have been segmented using the enhanced PDC clustering algorithm. Fig. 12 shows the rooftop segmentation results from top-down view (see Fig. 12(a)) and side view (see Fig. 12(b)), respectively. Even the rooftop points contain too much noise, and façade points are very unevenly distributed, no obvious topological inconsistency can be found. Meanwhile, the unlabeled points are superimposed onto the segmented primitives to examine their spatial distribution. Fig. 12(b) clearly shows that numerous unlabeled points are from building façades, which proves that our clustering algorithm can distinguish building façades from rooftops. As the ALS point clouds contain less façade points, it is almost impractical to reconstruct the detailed building façades. We simply discard them to prevent the interference with the rooftop reconstruction.

To quantitatively evaluate the rooftop clustering results, seven representative buildings from Fig.9 indicated by red rectangles are chosen for further analysis. A summary of the various performance evaluation results can be found in Table III which reports the segmentation statistics of the tested rooftops. In particular, in this table the effective segmentation indices, e.g., completeness, mean, minimum and maximum primitive residual that may be necessary to evaluate the accuracy of rooftop segmentation, are presented. The average completeness value of 97% clearly indicates that our clustering algorithm can maintain the topology of rooftop primitives reasonably well. This average index is calculated by dividing the sum of numbers in the column  $SP$  by the sum of the numbers in the column  $RP$ . The values of  $Min_{pr}$ ,  $Max_{pr}$  and  $Mean_{pr}$  in Table III reflect the overall segmentation accuracy of the proposed clustering algorithm. It should be noted that the minimum Euclidean distance from each point to its corresponding primitive is determined by the precision of the ALS point clouds and the value of  $k$  during rooftop clustering. The minimum primitive residual with only 0.09 m shows that the segmented points fit well with their clusters. In addition,

the maximum primitive residual value of 0.60 m clearly shows that there are no large deviations between the point clouds and their corresponding primitives. Furthermore, the relatively small value of 0.40 m of the  $Mean_{pr}$  demonstrates that the corresponding points fit very well with their primitives. On the contrary, a large value of  $Mean_{pr}$  indicates that the obtained primitives are not accurate since the points most probably belong to different primitives and/or the assumption of the polyhedral nature of the rooftop is not valid. The proportion of  $U_c$  points is relatively large for building a, d and e ( $\sim 9.65\%$ ) because more building façade points, misclassified points and other noisy points have been included. In addition, the predefined primitive size  $M_p$  is also an important factor that contributes to  $U_c$ . If size of the rooftop primitive is smaller than  $M_p$ , these primitive points will be labeled as  $U_c$ , which commonly occurs at superstructures and other ornaments on rooftops. Note that this proportion is calculated by dividing the sum of the numbers in the column  $U_c$  by the sum of the numbers in the column  $Pts$ .

To further test the proposed rooftop clustering algorithm, we compare with three mainstream clustering algorithms, namely region-growing algorithm (RG), efficient RANSAC (ERANSAC) algorithm [40] and P2C clustering method [26]. The comparison results are shown in Fig. 13. Building *A* is an old city hall with Romanesque Revival style design whose rooftop is composed of many trivial and small sized primitives. From the results of RG and ERANSAC algorithms, we can observe that some primitives span over other adjacent primitives, causing topological inconsistency among these primitives (see the regions labeled in red ovals). Similarly, P2C algorithm causes serious segmentation errors over large areas. However, our method keeps topological consistency of the roof primitive well and even very small primitives indicated in red ovals can be successfully detected. Building *B* is a typical multi-level flat rooftop composed of many relatively large primitives. Although it is relatively simple to cluster these types of rooftops, in the region of primitive indicated by a red arrow, RG and P2C result in under-segmentation. However, ERANSAC causes over-segmentation. RG and ERANSAC fail to process some parts of jagged primitives (illustrated in red oval) of building *C*, leading to incorrect topology. Although P2C guarantees the topological consistency over these areas, many trivial clusters are observed, making the clustering result inhomogeneous. Building *D* and *E* include many unlabeled points depicted by the red arrows due to vegetation, façades or rooftop superstructures. Compared to other three methods, our method accurately recognizes the rooftop primitive from



**Fig. 12. Rooftop primitive segmentation of Toronto data set.** Primitive segmentation from top-down view (a) and side view (b). The unlabeled points are colored in blue from side view (b). It is to be noted that the different color represents the distinct primitives with the condition that the disjoint primitives may share the same color.

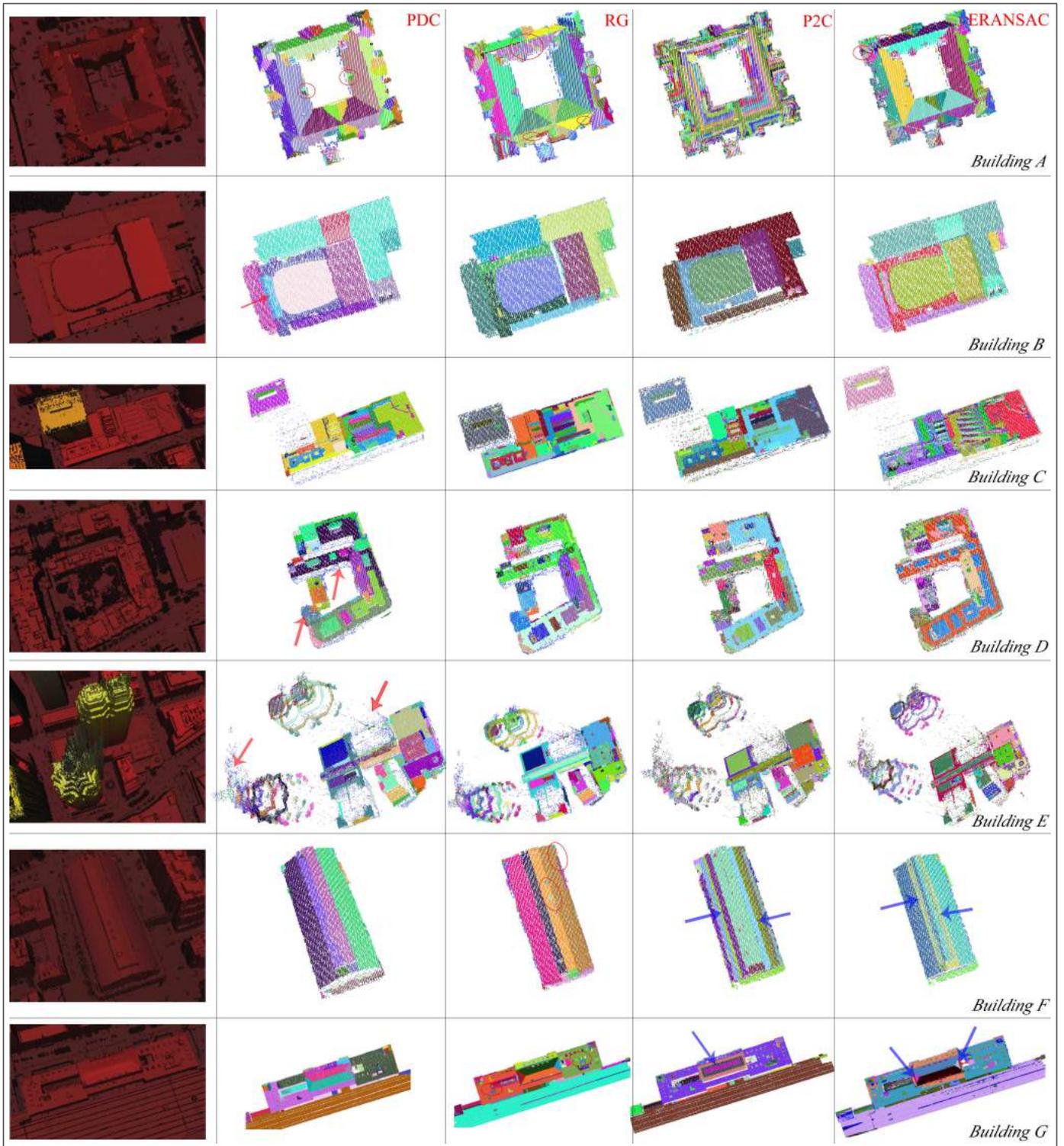
| $BN$ | $\#P_{ts}$ | $\#SP$ | $\#RP$ | $C_p$                 | $\#U_c$ | $Min_{pr}$ | $Max_{pr}$ | $Mean_{pr}$ |
|------|------------|--------|--------|-----------------------|---------|------------|------------|-------------|
| $A$  | 15,288     | 44     | 44     | 100% (correct)        | 1,070   | 0.30       | 0.58       | 0.46        |
| $B$  | 21,237     | 11     | 11     | 100% (correct)        | 302     | 0.25       | 0.49       | 0.35        |
| $C$  | 38,334     | 34     | 34     | 100% (correct)        | 1,653   | 0.09       | 0.61       | 0.35        |
| $D$  | 20,979     | 29     | 29     | 100% (correct)        | 2,025   | 0.10       | 0.58       | 0.42        |
| $E$  | 56,783     | 65     | 60     | 108% (over-segmented) | 4,835   | 0.21       | 0.80       | 0.43        |
| $F$  | 16,205     | 13     | 13     | 100% (correct)        | 293     | 0.19       | 0.54       | 0.37        |
| $G$  | 73,661     | 32     | 37     | 86% (under-segmented) | 925     | 0.15       | 0.63       | 0.44        |
|      | Mean       |        |        | 97%                   |         | 0.18       | 0.60       | 0.40        |

**TABLE III: Statistics from seven representative rooftops depicted in Fig. 13.**  $BN$ : the building identifier.  $\#P_{ts}$ : the number of points in the building.  $\#SP$ : the segmented rooftop primitives.  $\#RP$ : the rooftop primitives in the data set.  $C_p$ : the completeness of the rooftop primitives.  $\#U_c$ : the points that are not labeled.  $Min_{pr}$ ,  $Max_{pr}$  and  $Mean_{pr}$  represent the minimum, maximum and mean residuals of primitives ( $m$ ) within each rooftop.

highly noisy point clouds. In buildings  $F$  and  $G$ , we can find the critical role played by the optimization fidelity term in reducing the trivial and very small redundant primitives. Other three methods cannot control the level of clustering granularity (see the red ovals) or may produce the topological inconsistent errors (see the blue arrows) for processing these two buildings.

2) **Model Accuracy Evaluation:** For assessment of building models accuracy, we overlay the 3D building models onto their digital surface models to qualitatively evaluate their fitness based on visual perception. Through overlapping with

DSM, if there exist obvious misalignments, it will be the place where the errors occur. These misalignment errors can be due to the reconstructed models' geometric or topological deficiencies. We select the third and fifth levels of the building models to inspect the model misalignments by superimposing on their DSM. Moreover, to make an easy comparison with other two methods, we also overlay the models from Poullis [26] and Zhou [36] onto their corresponding DSMs. The overlapped results by different reconstruction methods are shown in Fig. 14. Fig.14(a) is a model superposition result between the third level models and their corresponding DSM,



**Fig. 13. Typical rooftop segmentation results among different clustering algorithms.** Columns from left to right: shaded relief map of rooftops, the rooftop clustering results by the enhanced PDC, RG, P2C and ERANSAC algorithms, respectively.

from which we can see that the buildings with inclined or dome rooftops have not been correctly reconstructed as these regions are completely covered by the DSM. We overcome these drawbacks in the fifth level polyhedral building models as shown in Fig. 14(b). Although it appears more accurate and visually convincing, the rooftops with dome or cylinder shapes have discrepancy from DSM. The Poullis models in Fig. 14(c) can only generate the multi-level flat rooftops and hence, there exists obvious deviations between the building models and the DSM in the inclined and non-linear rooftop areas. Although the models from Zhou are well fitted by superposition of the DSM in Fig. 14(d), the mesh simplification method cannot maintain the geometric regularity, e.g., plane, sphere, cone, cylinder and torus. Another drawback of Zhou’s models is lack of semantic information, making it difficult to identify the specific types of triangular meshes of building components. However, in our multiple LoD rooftop model’s representation, we can easily switch the model representation to different levels of details as our algorithm can record the subordinate relationships among vertices, triangular facets and rooftop components.

As mentioned before, our model can generate building models at five levels of details. In each level, there exist some specific elements that affect the model quality. The first level model’s quality is dramatically restricted by the hypothesis that each building has two dominant directions, which means our optimization method can only obtain two dominant building directions. The results are shown in Fig. 15, from which we can see that most of the buildings can be successfully represented by the dominant and orthogonal dominant directions. However, if the actual building has more than two dominant directions indicated by the back arrows, the reconstructed models’ orientation will have significant discrepancy from the actual building.

The second level building model’s accuracy is influenced by the building inner and outer boundaries. Fig. 16 shows the extracted building inner and outer boundaries by the proposed algorithm. Although the point density is non-uniform due to multiple flight strips overlap and missing data, in most cases, the proposed algorithm can accurately trace the building boundaries, which can be verified in black enlarged rectangles. Although the results are very promising, a few errors occur on five rooftops that are clearly shown in the enlarged rectangles in Fig. 16. In particular, errors *a* and *b* occur at the multiple flight strips’ connection areas, where many pseudo boundary points are generated due to severe uneven distribution of the point clouds. For errors *c* and *d*, the partial inner boundary edges are shared with the outer boundary. When the outer boundary is extracted, the remaining inner boundary is separated into two parts and it will be never extracted as a whole. Similar situation appears on error *e* but with a minor difference, i.e., partial boundary edges are shared by two inner boundaries. These errors are expected to be solved in the future by incorporating the “sharing mechanism” during traversing the MST.

For the third level building models, the model accuracy is commonly determined by the primitive boundary and the building dominant direction. In Fig. 18, the quantitative eval-

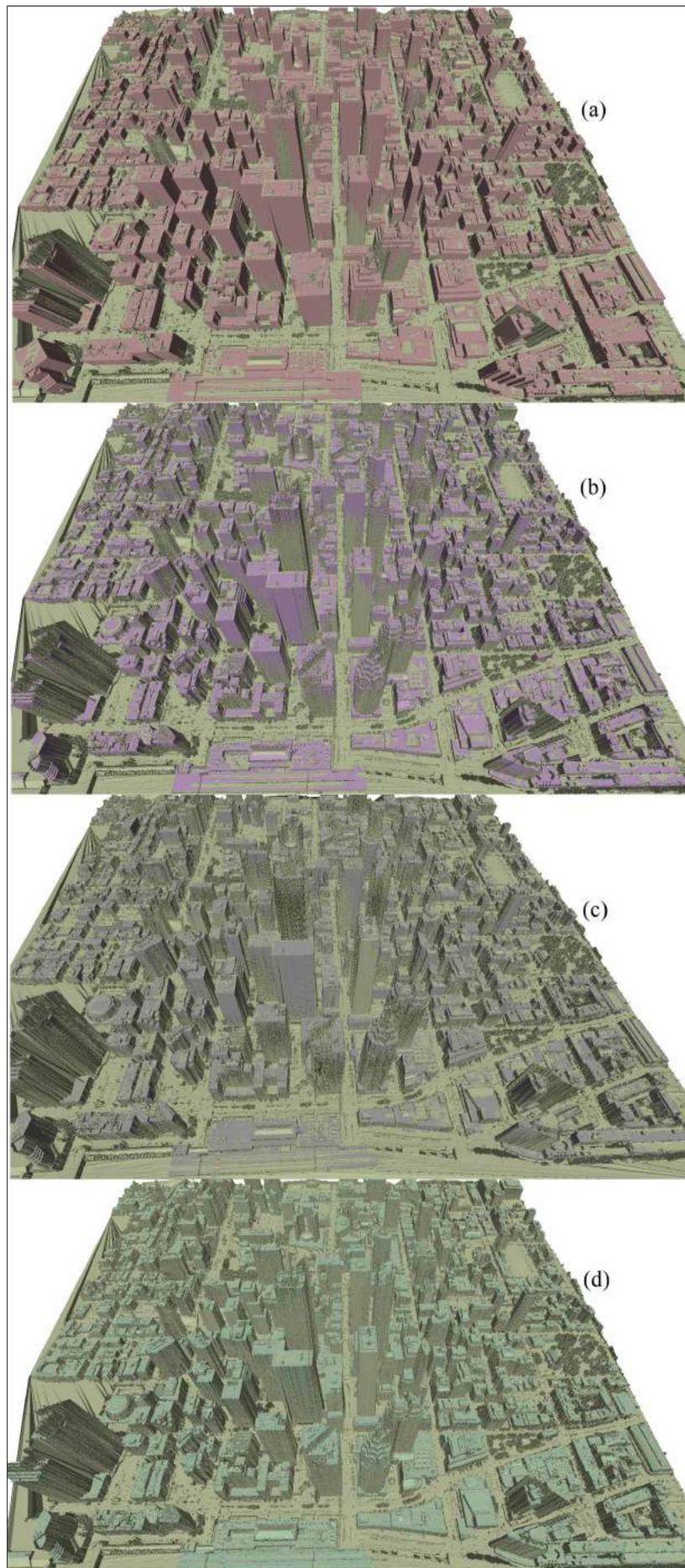
uation will be given for these types of models.

The fourth level building model accuracy is determined by the primitive boundary. Fig. 17 shows the rooftop primitive boundary results by the subgraph of Voronoi diagram algorithm described in Section III-B1. The rooftop primitive boundaries have been completely traced using the interpolated edges whose accuracy are well controlled within  $\rho/2$ . In addition, the accuracy of the traced boundary is also closely related to the enhanced PDC clustering algorithm. If the primitives are not correctly clustered, the corresponding primitive boundary will definitely not be correctly traced. Please be noted that the enhanced PDC clustering algorithm is based on 3D space. However, the subgraph Voronoi diagram tracing algorithm is operated in 2D space. When 3D rooftop point clouds are projected onto 2D space, some primitives may be overlapped, causing many trivial, spurious primitive edges indicated by red rectangles in Fig. 17. Fortunately in most cases the “mutual erosion of overlapped primitives” rarely happen on ALS point clouds due to their 2.5D characteristics. Actually, these spurious primitives are produced by the outliers because the rooftop materials allow the laser beam to penetrate the rooftop into the interior of the building. At this point, the point clouds are extremely messy, even human beings cannot distinguish each primitive accurately.

The fifth model’s accuracy is determined by all the above factors. To make qualitative evaluation of the third and fifth building model’s accuracy, we select seven representative buildings from Fig. 9 to compare the root mean square (RMS) distance from the point set to the reconstructed models. As shown in error map and RMS statistics in Fig. 18 and Table. IV respectively, the accuracy of the third level model is relatively low. However, our fifth level model’s accuracy is absolutely superior to the accuracy of third level and Poullis’s models but slightly below the accuracy of Zhou’s models. Zhou’s method is actually a remeshing method, which resamples or reorganizes the mesh vertices to meet specific quality requirements at the expense of semantic information and model compactness. Therefore, we conclude that our highest-level models achieve a very good balance between the accuracy, compactness and semantic representation.

## V. DISCUSSIONS AND CONCLUSIONS

In this paper, a novel topologically-aware rooftop modeling methodology has been proposed and its performance has been evaluated on the large-scale Toronto data set. The results have shown that the proposed enhanced PDC algorithm does not produce topological inconsistencies and fit well with their corresponding primitives. This is verified by considering not only visual inspection but also well accepted statistical criteria, i.e., quantitative evaluation. The accurate clustering result lays a solid foundation for the subsequent rooftop reconstruction procedure based on data-driven framework. The reconstructed models also clearly indicate that our algorithm can generate rooftop model at five levels of details which can possibly be used in a wide range of applications. The model performance evaluation shows that the created models are not only accurate, i.e., with small RMS value but also compact, i.e., meshes with



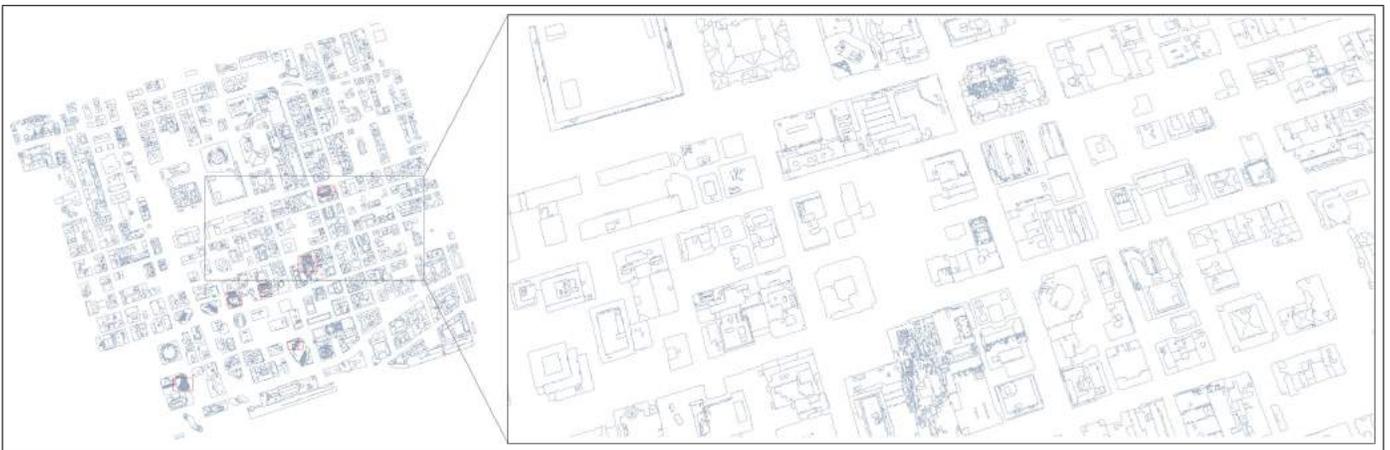
**Fig. 14.** A qualitative evaluation of the models by superimposing onto their DSM. Our third and fourth level models in (a) and (b), and the models from Poullis and Zhou in (c) and (d) are superimposed on their DSM, respectively.



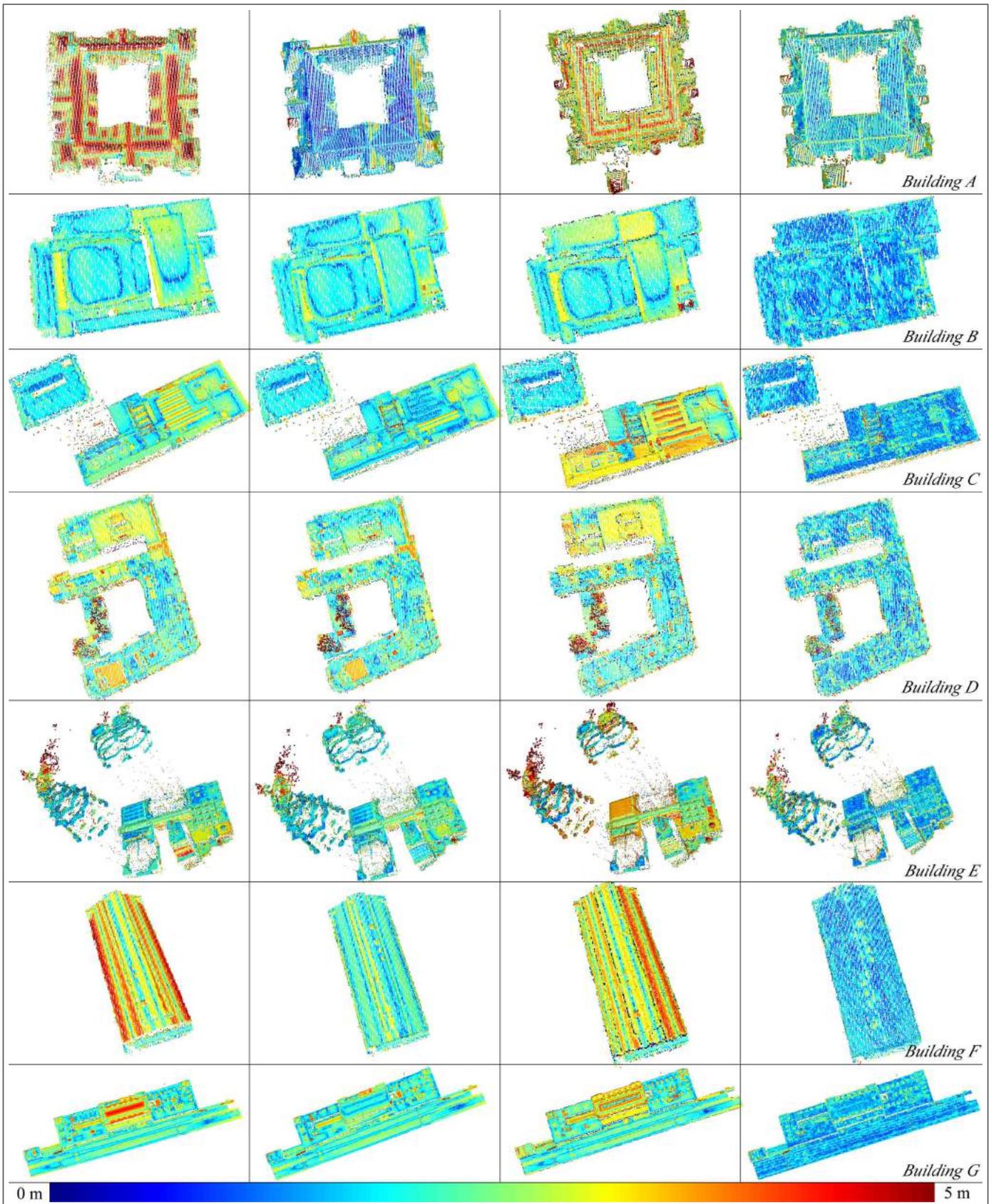
**Fig. 15. Building bounding boxes subject to the constraints of two dominant directions.** The red lines represent the building bounding boxes subject to the constraints of building dominant directions. The green lines represent the building's dominant and orthogonal dominant directions. The gray color represents the building location.



**Fig. 16. The building outer and inner boundary extraction.** The red lines are the inner and outer boundaries and the green lines are the constructed MST. The gray background color describes the building location.



**Fig. 17. The primitive boundary extraction.** The pseudo primitive boundaries appear in the regions illustrated by red rectangles.



**Fig. 18. Quantitative error map of representative buildings shown in Fig.9.** Columns from left to right: The model accuracy from the third level, fifth level, Poullis and Zhou models.

| Model Types             | The Representative Building Rooftops |                       |                       |                       |                       |                       |                       |
|-------------------------|--------------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|                         | Building A                           | Building B            | Building C            | Building D            | Building E            | Building F            | Building G            |
| The third level models  | $1.27 \times 10^{-2}$                | $2.28 \times 10^{-3}$ | $1.81 \times 10^{-3}$ | $5.64 \times 10^{-3}$ | $2.23 \times 10^{-3}$ | $9.41 \times 10^{-3}$ | $1.66 \times 10^{-3}$ |
| The fourth level models | $3.33 \times 10^{-3}$                | $1.51 \times 10^{-3}$ | $1.26 \times 10^{-3}$ | $3.35 \times 10^{-3}$ | $2.15 \times 10^{-3}$ | $1.76 \times 10^{-3}$ | $1.09 \times 10^{-3}$ |
| Poullis models [26]     | $6.62 \times 10^{-3}$                | $2.81 \times 10^{-3}$ | $3.08 \times 10^{-3}$ | $5.91 \times 10^{-3}$ | $7.61 \times 10^{-3}$ | $6.97 \times 10^{-3}$ | $1.65 \times 10^{-3}$ |
| Zhou models [36]        | $2.27 \times 10^{-3}$                | $1.11 \times 10^{-3}$ | $1.11 \times 10^{-3}$ | $2.69 \times 10^{-3}$ | $1.21 \times 10^{-3}$ | $9.19 \times 10^{-4}$ | $8.61 \times 10^{-4}$ |

TABLE IV: **Quantitative evaluation of representative buildings shown in Fig.9.** The table values represent the RMS distance from point clouds to the generated models. These error values are normalized w.r.t. the diagonal length of the original model’s bounding box.

a small number of triangles. In addition, there are no obvious artifacts such as misalignments between neighboring primitives and obvious discrepancy between the primitives and the original DSM data. This indirectly proves that our algorithm can well maintain the models’ topological consistency among rooftop primitives.

Although our algorithm achieves promising results on the large-scale Toronto data set, there are some interesting ideas which can be further explored to extend the research reported in this paper. A reasonable evaluation of the rooftop clustering accuracy clearly demands knowledge of proper usage of segmentation results. That is to say, the quality of rooftop primitive’s segmentation cannot be separately evaluated but should be taken into account in the entire pipeline. For example, for the proposed building reconstruction method we are expected to acquire the segmented results as homogeneous as possible. The “salt-and-pepper” clustering effect is not acceptable, which might impair the accuracy of primitive boundary extraction. In contrast, if the rooftop primitives that are mathematically the same but spatially separate, cannot be segmented correctly, our data-driven algorithm can cope with these under-segmentation results very well, obtaining error-free building models. If these under-segmentation results are used in model-driven reconstruction method, the error will be inevitable. In most of the cases, under-segmentation is more detrimental than over-segmentation. Thus, it is encouraged to obtain over-segmentation if the error-free segmentation cannot be guaranteed.

The proposed reconstruction method is based on 2D space. If we want to create the more complicated building models that conform to LoD 3 specification, the reconstruction algorithm based on 3D data structure should be considered.

For some rooftops composed of the special materials, e.g., glass, the laser beams can easily penetrate into the interior of the architectures. In this case, the point clouds are highly cluttered that even human beings are unable to correctly distinguish the individual primitives. This situation can be obviously observed from the roughness of the shaded relief map in Fig. 9. Using these extremely messy point clouds, most of the data-driven building reconstruction methods are incapable of successfully creating building models without artifacts. The concept behind multiple LoD representation can be used to solve this issue through abstract representation. To be specific, in the future we plan to design the “smart engine” to first evaluate the point clouds’ quality of each building and then adopt the abstraction modeling concept to construct building models with low-quality point clouds. This

simplification modeling concepts can also be used to process the relatively more complicated rooftops with high proportion of superstructure details, noise and outliers.

The proposed method is based on the assumptions that rooftop are composed of planar primitives and façades are vertical. The proposed polyhedral rooftop modeling method does not consider the superstructures or additional decorations on rooftops. In addition, as the eave exists, the accurate vertical walls usually have some distance to its original position. The façades with decoration of a wide variety of extrusion or intrusion components are also very commonly seen in metropolitan areas. Thus, the reconstruction method based on a series of assumptions may not be applicable for arbitrary rooftop and facades. We believe that the assumptions or constraints are nothing more than making the building model visually uniform or only reducing the complexity of rooftop modeling. How to make a trade-off between algorithm flexibility and photorealism of building models is a critical issue for us to ponder over.

With regard to the building model’s completeness, we believe that along with accurate geometry and correct topology, the building model should also contain semantic information. To enhance the semantics, the individual rooftop components of the reconstructed models should be labeled separately, which will facilitate the follow-up model editing and re-targeting. Moreover, taking a building model as a whole, the associated real images or synthetic textures should be mapped onto the corresponding models to enhance the visually appearance and semantic functions. In this sense, we may say that a complete building model should include detailed geometric, correct topology and abundant semantics.

From the above considerations, to further refine our work, we plan to, based on 3D data structure, integrate TLS or MLS point clouds to create more detailed rooftop models that conform LoD 3 specifications. To further validate the applicability of the proposed algorithm, we plan to employ the multiple and diverse data sources (point clouds from stereo image pairs) to evaluate the robustness of the proposed algorithm. In order to further enhance model semantics, we are expected to superimpose the aerial images onto their corresponding models.

## REFERENCES

- [1] M. Kada, A. Wichmann, Y. Filippovska, and T. Hermes, “Animation strategies for smooth transformations between discrete lods of 3d building models.” *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 413–420, 2016.

- [2] A. A. Diakité and S. Zlatanova, "Valid space description in bim for 3d indoor navigation," *International Journal of 3-D Information Modeling (IJ3DIM)*, vol. 5, no. 3, pp. 1–17, 2016.
- [3] C. F. Reinhart and C. C. Davila, "Urban building energy modeling—a review of a nascent field," *Building and Environment*, vol. 97, pp. 196–202, 2016.
- [4] N. Pelechano and A. Malkawi, "Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches," *Automation in construction*, vol. 17, no. 4, pp. 377–385, 2008.
- [5] H. J. Miller and K. Tolle, "Big data for healthy cities: Using location-aware technologies, open data and 3d urban models to design healthier built environments," *Built Environment*, vol. 42, no. 3, pp. 441–456, 2016.
- [6] K. Zhang, J. Yan, and S. C. Chen, "Automatic construction of building footprints from airborne LIDAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 9, pp. 2523–2533, sep 2006.
- [7] B. C. Matei, H. S. Sawhney, S. Samarasekera, J. Kim, and R. Kumar, "Building segmentation for densely built urban regions using aerial LIDAR data," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers (IEEE), jun 2008.
- [8] C. Poullis and S. You, "Automatic reconstruction of cities from remote sensor data," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 2775–2782.
- [9] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 3, pp. 1554–1567, March 2010.
- [10] Q. Y. Zhou and U. Neumann, "2.5d building modeling by discovering global regularities," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers (IEEE), jun 2012.
- [11] J. Yan, K. Zhang, C. Zhang, S.-C. Chen, and G. Narasimhan, "Automatic construction of 3-d building model from airborne LIDAR data through 2-d snake algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 3–14, jan 2015.
- [12] H. G. Maas and G. Vosselman, "Two algorithms for extracting building models from raw laser altimetry data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2-3, pp. 153–163, jul 1999.
- [13] C. Poullis and S. You, "Photorealistic large-scale urban city model reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 4, pp. 654–669, jul 2009.
- [14] H. Huang, "A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 79, pp. 29–43, 2013.
- [15] A. Henn, "Model driven reconstruction of roofs from sparse lidar point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 76, pp. 17–29, 2013.
- [16] G. Vosselman and S. Dijkman, "3d building model reconstruction from point clouds and ground plans," *International archives of photogrammetry remote sensing and spatial information sciences*, vol. 34, no. 3/W4, pp. 37–44, 2001.
- [17] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Structural approach for building reconstruction from a single DSM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 135–147, jan 2010.
- [18] B. Ameri and D. Fritsch, "Automatic 3d building reconstruction using plane-roof structures," *ASPRS, Washington DC*, 2000.
- [19] B. Xiong, S. O. Elberink, and G. Vosselman, "A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp. 227–242, jul 2014.
- [20] B. Xiong, M. Jancosek, S. O. Elberink, and G. Vosselman, "Flexible building primitives for 3d building modeling," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 101, pp. 275–290, mar 2015.
- [21] V. Verma, R. Kumar, and S. Hsu, "3d building detection and modeling from aerial LIDAR data," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*. Institute of Electrical and Electronics Engineers (IEEE), 2006.
- [22] S. Oude Elberink, "Target graph matching for building reconstruction," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 49–54, 2009.
- [23] S. Oude Elberink and G. Vosselman, "Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations," *Sensors*, vol. 9, no. 8, pp. 6101–6118, 2009.
- [24] H. Wang, W. Zhang, Y. Chen, M. Chen, and K. Yan, "Semantic decomposition and reconstruction of compound buildings with symmetric roofs from LiDAR data and aerial imagery," *Remote Sensing*, vol. 7, no. 10, pp. 13945–13974, oct 2015.
- [25] B. Höfle, M. Hollaus, and J. Hagenauer, "Urban vegetation detection using radiometrically calibrated small-footprint full-waveform airborne LiDAR data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 134–147, jan 2012.
- [26] C. Poullis, "A framework for automatic modeling from point cloud data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2563–2575, nov 2013.
- [27] Q. Y. Zhou and U. Neumann, "Fast and extensible building modeling from airborne LiDAR data," in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. Association for Computing Machinery (ACM), 2008.
- [28] Q. Y. Zhou and U. Neumann, "A streaming framework for seamless building reconstruction from large-scale aerial LiDAR data," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers (IEEE), jun 2009.
- [29] Y. Chen, L. Cheng, M. Li, J. Wang, L. Tong, and K. Yang, "Multiscale grid method for detection and reconstruction of building roofs from airborne lidar data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 10, pp. 4081–4094, Oct 2014.
- [30] M. Li, P. Wonka, and L. Nan, "Manhattan-world urban reconstruction from point clouds," in *European Conference on Computer Vision*. Springer, 2016, pp. 54–69.
- [31] G. Vosselman, "Building reconstruction using planar faces in very high density height data," *International archives of photogrammetry remote sensing and spatial information sciences*, vol. 32, no. Part 3/ 2W5, pp. 87–92, 1999.
- [32] G. Sohn, X. Huang, and V. Tao, "Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data," *Photogrammetric Engineering & Remote Sensing*, vol. 74, no. 11, pp. 1425–1438, nov 2008.
- [33] P. Dorninger, "A comprehensive automated 3d approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds," *Sensors*, vol. 8, no. 11, pp. 7323–7343, 2008.
- [34] K. Kim and J. Shan, "Building roof modeling from airborne laser scanning data based on level set approach," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 4, pp. 484–497, jul 2011.
- [35] D. Chen, L. Zhang, P. T. Mathiopoulos, and X. Huang, "A methodology for automated segmentation and reconstruction of urban 3-d buildings from ALS point clouds," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 10, pp. 4199–4217, oct 2014.
- [36] Q. Y. Zhou and U. Neumann, "2.5d dual contouring: A robust approach to creating building models from aerial lidar point clouds," in *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 115–128.
- [37] H. Vo, "Streaming simplification of tetrahedral meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 1, pp. 145–155, jan 2007.
- [38] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," *ACM Transactions on Graphics*, vol. 21, no. 3, jul 2002.
- [39] N. Haala and M. Kada, "An update on automatic 3d building reconstruction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 6, pp. 570–580, nov 2010.
- [40] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.
- [41] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nchter, "The 3d hough transform for plane detection in point clouds: A review and a new accumulator design," *3D Research*, vol. 2, no. 2, jun 2011.
- [42] F. Lafarge and C. Mallet, "Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation," *International Journal of Computer Vision*, vol. 99, no. 1, pp. 69–85, feb 2012.
- [43] Q. Y. Zhou and U. Neumann, "Complete residential urban area reconstruction from dense aerial LiDAR point clouds," *Graphical Models*, vol. 75, no. 3, pp. 118–125, may 2013.
- [44] Q. Y. Zhou and U. Neumann, "2.5d building modeling with topology control," in *CVPR 2011*. Institute of Electrical and Electronics Engineers (IEEE), jun 2011.
- [45] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer, "A survey of urban reconstruction," *Computer Graphics Forum*, vol. 32, no. 6, pp. 146–177, may 2013.

- [46] R. Wang, "3d building modeling using images and lidar: A review," *International Journal of Image and Data Fusion*, vol. 4, no. 4, pp. 273–292, 2013.
- [47] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," in *Computer Graphics Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 301–329.
- [48] J. Demantké, C. Mallet, N. David, and B. Vallet, "Dimensionality based scale selection in 3d lidar point clouds," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38-5/W12, pp. 97–102, sep 2012.
- [49] S. O. Elberink and G. Vosselman, "Quality analysis on 3d building models reconstructed from airborne laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 2, pp. 157–165, mar 2011.
- [50] R. J. You and B. C. Lin, "A quality prediction method for building model reconstruction using LiDAR data and topographic maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 9, pp. 3471–3480, sep 2011.
- [51] G. Gröger and L. Plümer, "Citygml–interoperable semantic 3d city models," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 71, pp. 12–33, 2012.



**Dong Chen** Dong Chen received the Ph.D. degree in Geographical Information Sciences from Beijing Normal University, Beijing, China, in 2013. He is an Assistant Professor with Nanjing Forestry University, Nanjing, China. Currently, he works as a postdoctoral fellow in the Department of Geomatics Engineering at the University of Calgary. His research interests include image- and LiDAR-based segmentation and reconstruction, full-waveform LiDAR data processing, and related remote sensing applications in the field of forest ecosystems.



**Ruisheng Wang** Ruisheng Wang received a B.Eng. degree from the Wuhan University, China, a M.Sc.E degree from the University of New Brunswick, Canada, and a Ph.D. degree from the McGill University, Canada. He is currently an associate professor in the Department of Geomatics Engineering, University of Calgary. His research interests include geomatics and computer vision, especially point cloud processing.



**Jiju Peethambaran** Jiju Peethambaran received the Bachelors in Information Technology from University of Calicut, India, the Masters in Computer Science from National Institute of Technology, Karnataka, India and the Ph.D. degree in Computational Geometry from Indian Institute of Technology, Madras, India. He is currently a postdoctoral researcher with the Department of Geomatics Engineering, University of Calgary. His research interests include computational geometry, computer graphics, visualization and related applications including

LiDAR-based urban modeling.