Optimal Model Fitting for Building Reconstruction From Point Clouds

Wenyuan Zhang, Zhixin Li, and Jie Shan, *Senior Member, IEEE*

***Abstract*—Geometric-semantic coherent building models are demanding in many geoscience applications. Conventional building modeling methods often rely on successive roof plane segmentation and fitting. The subsequent reconstruction procedure is difficult to assure topologic consistency and geometric accuracy. This article starts with a library of predefined building models or primitives, including pyramid, gable, hip, etc. We propose an optimal model fitting approach that holistically determines all of its parameters from segmented point cloud data. The approach is formulated as an optimization problem that minimizes the point-to-mesh distance between the point cloud and the meshed primitive model. Necessary constraints in the form of inequality equations are introduced to assure correct and reliable solution. For complex roofs consisting of several predefined primitive models, a hierarchical procedure is presented to reconstruct the major roof model and its superstructures sequentially. The CityGML LoD2 model is created from the parameterized primitives. The quality and performance of this approach are evaluated with airborne lidar and photogrammetric point clouds. Based on the experiments with 910 buildings, the primitive fitting accuracy is 7.8 cm and the corner uncertainty is 0.36 m or 0.78 times the ground point spacing; the building boundary consistency is 89.6%. The study demonstrates a piecewise continuous polyhedral building model can be determined through a holistic parameter optimization process. The resultant building models intrinsically best fit to the input point cloud with topologic integrity. The approach not only qualitatively generates semantic building models but also exhibits the potential for building reconstruction over large areas.**

***Index Terms*—Building model, CityGML, hierarchical reconstruction, point cloud, primitive fitting.**

I. Introduction

THREE dimensional (3D) building models with detailed roofs are essential for a large number of geospatial applications, such as 3D geographic information systems (GIS), urban planning, environmental simulation, energy consumption assessment, heritage preservation [1], and change detection [2]. With the rapid developments in aerial laser scanning and aerial oblique photogrammetry, building reconstruction from point clouds has been an active and ongoing research topic in photogrammetry and computer vision. A variety of algorithms have been proposed to rapidly generate polyhedral building models. Nevertheless, generating accurate semantic building models with certain levels of detail (LoD) remains to be challenging due to the complexity and diversity of building structures, as well as the sparsity and noise of the point clouds [3]–[6].

Amount of researches indicate that two categories, data-driven and model-driven, can be used to classify conventional building reconstruction methods. Data-driven building reconstruction is a prevailing approach in the past decades, which usually starts with a segmentation of the individual building points into planar roof patches, followed by intersecting them into polygonal meshes. Technically, this type of approach is based on individual plane fitting by using such algorithms as random sample consensus (RANSAC) [7], clustering [8], region growing [9], [10], or cost function-based method [11], which achieve locally fitting optimal. Although such solution has the advantage of reconstructing polyhedral building models with complex shapes, it is lack of topological integrity and geometric rigor, and sensitive to the incompleteness of input data caused by tree clutter, multipath reflectance, object overlap, or occlusion [12]. To overcome these limitations, researchers presented model-driven strategy by estimating the parameters of predefined primitives. This kind of method could achieve regularized, topologically correct, and completed building models. Some of the typical building primitive models include plane, sphere, cone, and cylinder, all of which can be mathematically represented by one smooth surface. However, this is not true for many common buildings, such as gable, pyramid, hip, or cuboid, which are all piecewise smooth and need a set of equations and boundary constraints to represent. Current reported works pay less attention to handle such mathematically piecewise smooth surfaces to determine their parameters under one framework, especially for plane fitting of piecewise surfaces simultaneously in 3D space. Consequently, intrinsic relations in the building primitives such as boundary, regularity, symmetry and semantics could not be used in an integral manner.

This article presents a hierarchical holistic primitive fitting framework to optimally reconstruct geometrically and topologically correct CityGML LoD2 building models from point cloud data [13], [14]. The key idea is to simultaneously determine all of the parameters of a building model under one unified framework. In this work, we extend the holistic optimization method [15] with the following contributions: 1) Instead of combining point-surface-distance (PSD) and intersection over union (IoU) as cost metrics in [15], a new metric, named point-mesh-distance (PMD), is employed for parameter optimization. The PMD-based cost function not only has higher penalty than PSD but also eliminates the problem of setting optimal weight parameter between PSD and IoU. 2) To make the optimization procedure feasible and reliable, we design 3D solid building primitives

Manuscript received May 15, 2021; revised July 6, 2021 and August 7, 2021; accepted August 31, 2021. *(Corresponding author: Jie Shan.)*

Wenyuan Zhang is with the National Research Center of Cultural Industries, Central China Normal University, Wuhan 430079, China (e-mail: zhangwy@ccnu.edu.cn).

Zhixin Li is with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: li2887@purdue.edu).

Jie Shan is with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: jshan@purdue.edu).

Digital Object Identifier 10.1109/JSTARS.2021.3110429

and introduce a set of inequality constraints to the optimization procedure based on the boundary, regularity, symmetry, and topological relations imbedded in the building primitives. 3) We apply the holistic optimization approach to recursively estimate the parameters of a building primitive and its superstructures, so that the primary structure with different roof shapes and the small objects on top of the roof can be hierarchically reconstructed. 4) Asymmetric roof primitives are designed to model those buildings with nonregular roof structures. Moreover, besides triangulated polygonal mesh representation, we create CityGML LoD2 models with roof details from the optimized parameters as the final representation of the reconstructed buildings. This allows us to retain geometric, topological, and semantic features of the building model.

The rest of this article is organized as follows. Section II provides a review of relevant works for building reconstruction from point clouds. Section III describes the predefined 3D building primitives with regard to different roof types and superstructures. Section IV presents our holistic parameter optimization approach for primitive fitting as well as the hierarchical reconstruction of roof details. Section V describes the procedure of generating semantic CityGML LoD2 building models. Section VI presents the experimental results from lidar point clouds and photogrammetric point clouds, while Section VII assesses and discusses these results. Section VIII summarizes our findings and perspectives for future work.

II. Related Works

Building reconstruction from point cloud has been a topic of intensive research in the past two decades. The existing building reconstruction techniques differ significantly based on the input data sources, procedures used, and the complexity of buildings. Based on our research scope, the following literature review covers roof structure recognition, parametric roof reconstruction, and CityGML building model generation.

A. Roof Structure Recognition

Roof primitive identification from point cloud is an essential prerequisite for rooftop reconstruction, especially for primitive-based reconstruction approach. A wealth of research exists for recognizing simple or complex roof shapes. A common way to detect roof structures is known as parametric primitive fitting. This kind of method iteratively recognizes potential and regular shapes which could be parameterized, such as planes, cylinders, and spheres, and finally assembles these detected primitives to a compact roof model. RANSAC [7] and Hough transform [16] are two widely employed approaches for planar primitive fitting. Theoretically, such solution is capable of recognizing various roof structures, while the outcomes of these methods are usually not satisfactory in practice when the roof structure is sophisticated and the point density is nonuniform. Moreover, some structures cannot be fully reconstructed due to occlusions or missing data. Finding optimal thresholds that are adaptive to input points is also difficult, which relies on careful and laborious parameter tuning [17]. The author in [18] presented an improved RANSAC algorithm to avoid abnormal and/or infinite solutions which are typically encountered in previously published methods that use the rooftop primitive adjacency matrix to solve the critical rooftop vertices. To assemble a number of detected planar shapes from point clouds, Bauchet and Lafarge [19] presented an efficient shape assembling mechanism with kinetic data structure to covert point cloud into watertight polygon meshes, and demonstrated its efficiency on large-scale downtown buildings reconstruction, while it relied on standard planar shape detection algorithms and was limited to roof reconstruction with planar shapes. To overcome the problems of outliers and/or noise in RANSAC, the authors in [20] presented a PCA-based robust segmentation algorithm for laser scanning 3D point cloud.

Since a complex building roof can be represented as a group of assembled simpler geometric primitives by analyzing the roof topology graphs (RTG) [21], some algorithms make use of RTG to identify certain roof shapes [6], [22]. This kind of method usually starts with roof plane detection by segmenting the building point clouds, followed by topology identification and recognition of roof structures based on a predefined primitive library. A complex roof could be finally reconstructed by assembling these basic primitives. Xiong *et al.* [6] defined several kinds of building primitives with loose nodes, edges, and minimum cycles, and constructed corresponding RTG for each primitive. In this approach, only plane normal vector associated with individual surface is designed as the primitive parameters. A constrained least-squares fitting method was exploited to perform roof plane fitting to simultaneously adjust all roof parameters. After obtaining optimal roof planes, the boundary of each roof polygon needs to be inferred by sequentially operating all inner ridges, outer boundaries, inner and outer corners for each roof plane. Since the automatic detection of RTG from point cloud is challenging and errors in the topology graph are inevitable, lots of manual correction of RTG is needed. In order to recognize the roof structure of a complex building from point cloud, Li *et al.* [23] first partitioned points into a triangulated irregular network (TIN) model and linked the TIN model to a label map of roof regions. The graph cut method was then applied to that label map to optimize the roof structures, and a filtering method was presented to regularize the boundary of extracted roof regions. Song *et al.* [24] presented an embedded deformation graph-based (ED) algorithm to reconstruct curved buildings by down-sampling the curved primitives identified from point cloud. The authors in [25] presented a clustering-based workflow to reconstruct building roof structures and the algorithm guarantees the production of geometric models without crack defects among adjacent primitives.

Recently, deep learning approaches produce impressive results by learning priors directly from the data. Especially, a range of deep neural networks have been developed for the task of segmentation and classification of point clouds, such as PointNet [26], PointNet++ [27], VoxelNet [28], and VoteNet [29]. Zhang *et al.* [30] proposed a deep reinforcement learning framework that integrates a 3D CNN, a deep Q-network, and a residual RNN for semantic segmenting 3D point clouds. The segmented building point clouds are reconstructed using 2.5D dual contouring method. These methods demonstrate remarkable performance in 3D semantic segmentation, classification, and object detection. However, using such deep learning methods to reconstruct various complete building models has not been possible, due to the lack of suitable point cloud training datasets. To the best of our knowledge, RoofN3D is the only publicly available 3D point cloud training dataset that can be used to train deep neural networks for the purpose of recognizing different roof types, and there are only three roof types in this dataset.

B. Parametric Roof Reconstruction

Due to possible incompleteness, sparse density and high noise of the point clouds, parameterized roof primitive fitting methods are developed to overcome these difficulties by applying prior knowledge and features of roofs. Roof primitives, e.g., gable-roof and hip-roof, embed useful and accurate geometric and topological information for building structures. Primitive-based method can not only yield high accuracy but also a best-fitted roof model with complete structure even for low quality point clouds.

A primitive-base building reconstruction method has been proposed by Tseng and Wang [31]. They first extract edges on multiple images. The optimal fitting was then performed by minimizing the perpendicular distances from the detected edge pixels to the projected edge lines of the model. The underlying mathematics is projective geometry for object-image relationship. Zhang *et al.* [32] proposed a primitive-based building reconstruction method by utilizing airborne lidar data and optical imagery, in which planes in point cloud and corner points in the imagery are extracted to optimize primitive parameters separately. However, what is lacking is quantitative evaluation of model accuracy and experiments for large-scale or multiple buildings with different roof types. In addition, the reconstruction process was not fully automatic, since the decomposition of building, recognition of primitives and measurement of initial parameters were done manually. Huang *et al.* [12] presented a statistical approach to automatically reconstruct building roofs from lidar point clouds. Based on a set of predefined 2D roof primitives, a variant of Markov chain Monte Carlo method with specified jump mechanism is explored to select proper roof primitives. Different simple primitives are searched one by one, and estimated simple primitives are finally assembled into the entire roof based on predefined rules of combination and merging. Wang *et al.* [33] proposed an algorithm to automatically decompose compound buildings of symmetric roofs into different primitives. The primitive parameters are estimated in 2D image space by a nonlinear least squares optimization with constraints from lidar data and aerial imagery.

To reconstruct complex and irregular roofs, Lafarge and Mallet [34] proposed a method for large-scale city modeling by the combination of geometric 3D primitives with meshes. In which 3D primitives such as planes, cones, and cylinders were used to describe regular roof sections, while mesh-patches are adopted to represent irregular roof components, a nonconvex energy optimization strategy was then explored to extract building points and find labels of different primitives. Efforts on complex roof reconstruction were also conducted on the combination of data-driven and model-driven methods by making use of their advantages [35], [36]. Jarzabek-Rychard and Borkowski [35] presented an unambiguous decomposition of complex objects into predefined simple structures, as well as a library of elementary building structures and their corresponding explicit topology graphs. A set of flexible reconstruction rules were predefined to generate complete polyhedral models from point clouds. Nevertheless, it still remains to be a challenge on primitive definition, superstructure modeling, and achieving accurate modeling.

In terms of model representation, the outputs of most existing methods are either polygonal meshes [19], [37] or parameterized building models [38], both are geometric models without semantic information, which limit the potential application of these models.

C. CityGML Model Reconstruction

Semantic building reconstruction is a new research topic since the last decade due to the rapid development of CityGML. Building models based on CityGML have been increasingly used for a large range of applications beyond visualization, such as urban planning, environmental simulation, disaster management, and 3D navigation [39]. However, automatically creating large-scale CityGML building models with high levels of detail is difficult, since most existing efforts focus on the transform and integration between building information modeling (BIM) and CityGML [40], [41]. Recently, a few researchers intended to generate CityGML building models with roof details. Henn *et al.* [42] introduced an automatic method for the reconstruction of CityGML LoD2 models with prototypical roofs from sparse lidar point cloud and building footprints. They used supervised machine learning to identify roof types and select the best model. Zheng *et al.* [35] proposed a hybrid method to generate CityGML LoD2 building models by using lidar point clouds, 2D building footprint, and high-resolution aerial images. Jayaraj and Ramiya [43] used commercial software (ArcGIS Pro) and open source packages to create LoD1 and LoD2 building models in CityGML format from airborne lidar point cloud. However, the generated CityGML LoD2 models do not explicitly separate various semantic surfaces, and the process requires a lot of human interaction.

In summary, most of reported approaches focus on geometric reconstruction of main roof structure and rely on multisource data. Very few researches achieved the reconstruction and representation of small objects (e.g., dormers and chimneys) on top of a roof. Although Salehi and Mohammadzadeh [44] reconstructed several building models with dormers and chimneys by utilizing a data-driven method, only hip roof was supported, and lidar point cloud, vector data, and images were combined to use. Automation and accurate building reconstruction with semantics from point clouds still pose great challenges to the existing algorithms. In comparison to previous approaches, this article can be regarded as a hierarchical and holistic parametric reconstruction of building models with roof details. Our input data are point clouds of segmented individual buildings, and the proposed approach consists of four major steps. As shown in Fig. 1, it includes 1) 3D building primitive definition, and recognition by utilizing a widely used deep learning network called PointNet++ [27], 2) primary building structure reconstruction by solid primitive fitting with holistic optimization strategy, 3) secondary building structure reconstruction through clustering and primitive fitting, and 4) superstructure refinement and CityGML LoD2 building model generation. In the following sections, we detail the methods and key steps of our approach with focus on formulating the primitive fitting as an optimization problem.

III. Building Primitives and Parameterization

A. Building Primitives

The method follows the idea that most complex buildings can be decomposed to one primary primitive and several secondary primitives or superstructures. Building models derived from

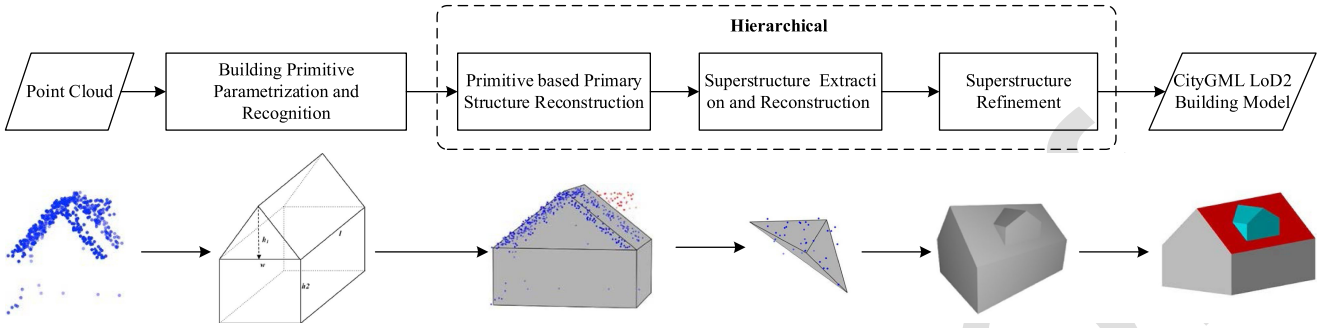


Fig. 1. Workflow of hierarchical reconstruction of building models with superstructure by holistic primitive fitting.

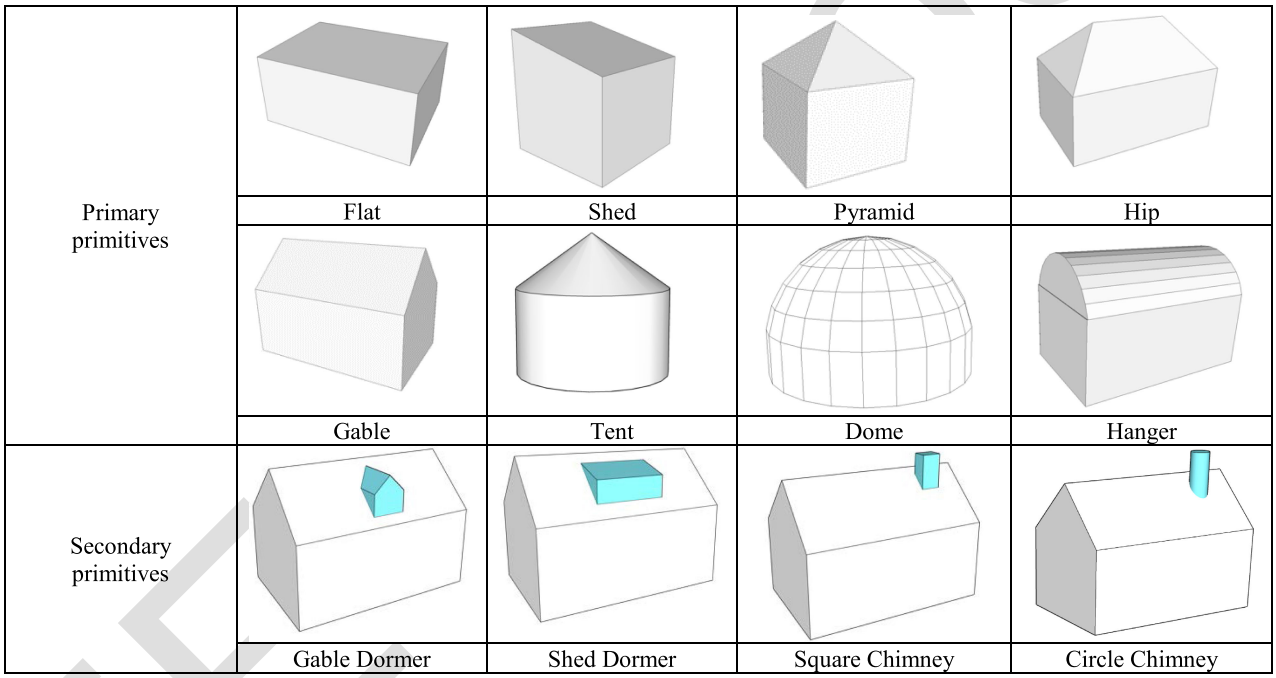


Fig. 2. Predefined building primitives and superstructures atop a gable roof.

these basic primitives and their combinations will cover majority of the buildings.

As shown in Fig. 2, we define several classical types of building as primary primitives according to the semantic-rich CityGML standard [13]. Each primary primitive in our library is a polyhedral solid model, which is composed of several planar surfaces or nonplanar surfaces representing roof, facades, and bottom (ground). The bottom boundary of each building primitive is a convex polygon. It should be noted the optimization framework requires that the primitives must be piecewise smooth, i.e., they need to be closed or a sloid. Besides these building models, Fig. 2 also illustrates a few secondary primitives of our building primitive set, which are used for superstructure reconstruction such as dormer and chimney. All primitives embed explicit or implicit geometric regularities, topological relationships between surfaces, or symmetries, thus allowing us to involve common knowledge and rules about the shape and topology in the process of parameter optimization. Meanwhile, these primitives can also be considered as a combination of semantic units, e.g., roof faces, wall faces, which facilitate effective CityGML LoD2 model generation from these parametric primitives.

B. Primitive Parametrization

Each building primitive in our library is encoded in terms of two kinds of parameters: Shape parameters and position parameters. The shape parameters describe the shape (including size) of the building, usually including width, length, roof height, wall height, etc. Consisting of a rotation angle (κ) along the z-axis and three translation parameters (tx, ty, tz), the position parameters define the exact orientation and location of the building. All the primitives in the library have the same set of position parameters, while the shape parameters vary for different primitives. This parametrization is beneficial to exploit the compositionality of primitives as well as the independence of "what" and "where" [45].

Take the hip roof building as an example. Shown in Fig. 3, there are five shape parameters: Width (w) of hip roof, bottom length of hip roof (l_b), ridge length (l_r), roof height (h_r), and

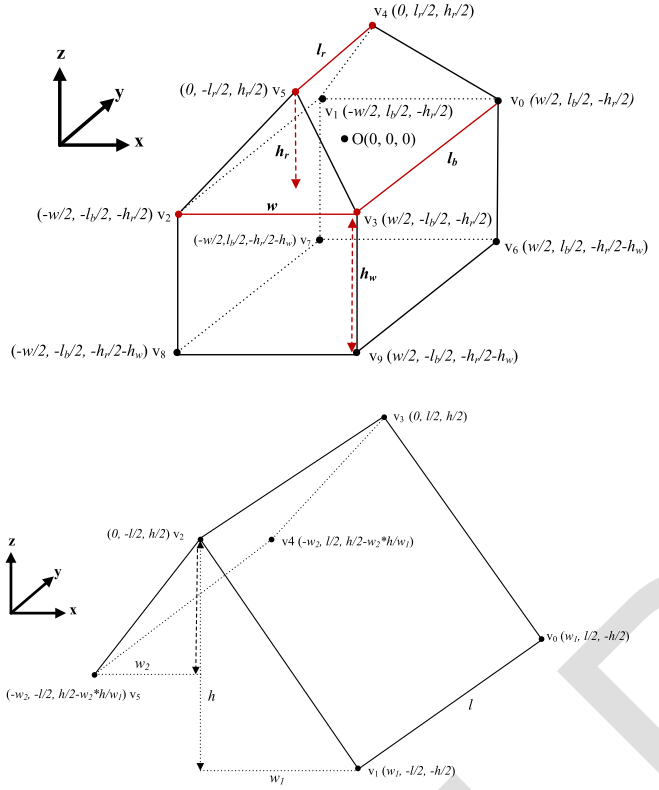


Fig. 3. Parameterization of a hip primitive (top) and asymmetric gable primitive (bottom).

wall height (h_w). By taking the center of the hip roof as the origin of the model coordinate system and assuming the roof ridge is parallel to the y-axis, the initial coordinates of all ten vertices (V) can be derived by these shape parameters, as annotated in Fig. 3 top. Fig. 3 bottom also shows the parameters of an asymmetric gable roof. When a primitive is rotated and translated according to its pose parameters, the new value of vertex coordinates can be calculated by the formula of a local transformation

$$V' = R_k\, V + [t_x, t_y, t_z]^T \tag{1}$$

where

$$R_k = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

Consequently, the vertex coordinates of a building primitive can be derived by its shape and pose parameters (θ). Vertices, edges, facets, and their relationships are all predefined and encapsulated in this parametric primitive model. The primitive also includes the necessary semantic information for CityGML model generation. The intrinsic geometric constraints among member of facets, such as rectangle, symmetry, parallel, or perpendicular relationships, etc., are embedded in this model to assure a regularized reconstruction. It should be noted our goal in primitive fitting is to determine all these parameters simultaneously under one optimization framework.

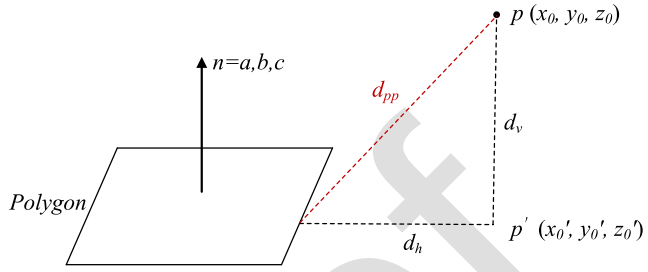


Fig. 4. Calculation of the point-to-polygon distance in $\mathbb{R}^3$ space.

IV. Holistic Primitive Fitting

Once the roof type of an individual building is identified from its point cloud, we will use the following holistic primitive fitting approach to determine its parameters.

A. Objective Function

To model the point cloud with a building primitive is essentially to estimate its parameters under certain criteria or an objective function. Our goal is to minimize the discrepancy between the estimated primitive and the point cloud so that the average of the squared distances of all points from the point cloud is taken into account in the following objective (cost) function:

$$J\ (\theta) = \min\left\{\frac{1}{N}\sum_{k=1}^{N}\mathrm{dist}(p_k, M)^2\right\} \tag{3}$$

where p_k is the kth point in a point cloud, N is the number of points in the point cloud, M is the 3D primitive mesh, and θ denotes the parameters of the primitive M. As an example, for building primitive with hip roof, $\theta = \{w, l_b, l_r, h_r, h_w, \kappa, tx, ty, tz\}$.

The key problem for this objective function is how to calculate the distance between a point in the point cloud and the (3D) primitive mesh. This article introduces the PMD in 3D space $\mathbb{R}^3$ as a metric to penalize the inconsistency between the point cloud and the primitive. The PMD is defined as the Euclidean distance from a point in 3D space to the closest point on the mesh surface. Consider a primitive which can be defined as a polygon mesh with multiple planar faces $\{F_i,\ i = 12, \ldots, n\}$, the PMD equals to the minimum distance of a point to all composing planar faces, i.e.,

$$\mathrm{dist}\ (p,\ M) = \min\left\{\mathrm{dist}\,(p,\ F_i)\,|i = 1, 2, \ldots, n\right\}. \tag{4}$$

Thus, the above problem is reduced to the calculation of point-to-polygon distance in $\mathbb{R}^3$. Illustrated in Fig. 4, for a given point $p(x_0,\ y_0,\ z_0)$ in a point cloud and a polygon (planar face F_i) of the primitive, the plane equation of F_i is $ax + by + cz + d = 0$, where the coefficients (a, b, c, d) could be inferred by any three vertices of this polygon. The desired 3D point-to-polygon distance d_{pp} can be divided into two components: Point-to-plane distance (d_v) and within-plane distance (d_h), i.e., $d_{pp}^2 = d_v^2 + d_h^2$.

The procedure of calculating the point-to-mesh distance in $\mathbb{R}^3$ is outlined below. First, computing the point-to-plane distance (dv) according to

$$d_v = \frac{|ax_0\ + by_0 + cz_0\ + d|}{\sqrt{a^2 + b^2 + c^2}}. \tag{5}$$

Second, calculating the projected point $p'(x_0^{'}, y_0^{'}, z_0^{'})$ of p onto the plane F_i as follows:

$$\begin{cases} x_0^{'} = x_0 - a \cdot t \\ y_0^{'} = y_0 - b \cdot t \\ z_0^{'} = z_0 - c \cdot t \end{cases} \tag{6}$$

where

$$t = \frac{ax_0 + by_0 + cz_0 + d}{a^2 + b^2 + c^2}. \tag{7}$$

Third, judging whether p' is inside, outside, or on the boundary of the polygon by the well-known ray casting algorithm [46] or winding number algorithm [47]. If p' lies inside or on the boundary of the polygon ($d_h = 0$), then d_v is the desired distance. Otherwise, we need to compute the within-plane distance. The within-plane distance in $\mathbb{R}^3$can be further reduced to the point-to-polygon distance problem in $\mathbb{R}^2$ by applying a rigid transformation that places the polygon and projected point onto a plane. The distance between a point and a polygon's boundary in $\mathbb{R}^2$ is computed by finding the minimum distance from the point to all the edges or vertices of this polygon.

It should be noted the vertices and planes of a building primitive is a function of its shape and pose parameters θ (see Fig. 3). They can be calculated through singular value decomposition (SVD) described by [15]. As such, the summary of such distances (PMD) is also a function of the primitive's parameters θ. This is the mechanism that they can be determined by using PMD as the objective function.

As the computation of the distance from a point to a single triangle in $\mathbb{R}^3$ is a standard operation in computational geometry [48], [49], the calculation of PMD could be efficiently performed by representing our building primitive as triangle meshes. On one hand, the exploitation of the minimum PMD as the cost allows us to simultaneously search for shape and position parameters. On the other hand, in contrast to simple plane fitting, all planar faces of the building primitive in our library can participate in fitting simultaneously to find the global optimal parameters under the metric of PMD.

It should be noted the introduction of PMD is an important extension to our previous work [15]. Compared with the perpendicular PSD utilized in Li [15], the PMD incurs a higher penalty for those points whose projections on the plane lie outside a polygon. Our experience shows this modification is essential and can lead to more robust and accurate fitting results.

B. Boundary Conditions

In order to achieve a robust and accurate solution for the primitive parameters, it is essential to introduce additional constraints or boundary conditions in conjunct with the objective function (3). In the case of hip building primitive fitting, the 2D boundary of the point cloud is first detected by the a-shape or convex hull algorithms. The area (S) of the 2D boundary is subsequently calculated. The inequality constraint with respect to 2D boundary is then defined as $w \cdot l_b \leq S$. Similarly, in terms of the vertical boundary $(h_{\min}, h_{\max})$ of the point cloud, another constraint is added by $h_r + h_w \leq h_{\max} - h_{\min}$. Considering that ridge length is usually less than bottom length of a hip roof, and roof height is usually lower than wall height, there will be additional constraints: $l_r \leq l_b,\ h_r \leq h_w$. As such, to achieve robust and accurate parameters of a building primitive with hip roof, the following inequality constraints will be applied to the aforementioned objective function (3)

$$\begin{array}{ll} w \cdot l_b & \leq S \\ h_r + h_w & \leq h_{\max} - h_{\min} \\ h_w - h_r & \geq 0 \\ l_b - l_r & \geq 0. \end{array} \tag{8}$$

Note that these constraints are varying for different primitives. In addition, even for the same type of primitive, some of the variables, e.g., area S and height boundary $h_{\max} - h_{\min}$, need to be calculated according to the input point cloud, while other constraints, such as $l_r \leq l_b$ and $h_r \leq h_w$ are enforced to the parameters of the predefined primitive.

Another impact factor is the initial values for the optimization solution process. Poor initialization in some cases may lead to the divergence of a nonlinear optimization method [50]. The initial values of a primitive's shape parameters such as length, width, height could be estimated roughly according to the boundary of the point cloud, whereas the initial rotation parameter κ needs special care. Since the roof ridge indicates the primary orientation of a gable or hip roof, and ridge points are situated on top of the roof, we select certain number of highest points from the point cloud as candidate ridge points. We then project these points into the horizontal plane. The robust RANSAC algorithm [7] is subsequently applied to detect the ridge line from these projected points. The calculated angle between ridge line and y-axis is taken as the initial κ. It should be noted that the detected ridge line may not geometrically correspond to the exact roof ridge, but it is sufficient to be used as initial approximation for subsequent optimization calculation. In addition, good initial values will also reduce the search space for parameter κ and accelerate the convergence of the optimization process.

Finally, to solve the above objective function (3) under the constraints (8) for optimal primitive parameters, we use a constrained nonlinear optimization algorithm named sequential least squares programming (SLSQP [51], [52]). SLSQP uses the Han-Powell quasi-Newton method with a BFGS update of the B-matrix and an L1-test function in the step-length algorithm. Besides these constraints, the SLSQP method allows us to define bounds of parameters, i.e., $-\pi \leq \kappa < \pi$. Once the constraints and bounds are established, all the planar patches of the primitive would be adjusted simultaneously under the restriction of the constraints so as to make all the faces in the primitive to best fit the point cloud. Starting from the initial values of the shape and position parameters, every iteration the primitive will be resized in all dimensions, rotated, and shifted based on the updated parameters θ. The iteration stops until the PMD is minimal, which means the determined primitive fits best to the input point cloud. The optimal parameters of a building primitive are determined by using the constrained SLSQP method ([51], [52]). The process repeats building by building with the corresponding primitive type. The geometric models of all buildings are then reconstructed by these primitives with the optimal parameters.

C. Hierarchical Fitting

The above optimization process is first applied to the primary building primitive. Once this is done, we can then model the roof superstructures. To this end, we apply a threshold to filter those points with large PMDs after the primary fitting. This leads to the points that likely correspond to small objects on the roof, i.e., dormer and chimney. In order to recover these detail superstructures, we employ the density-based spatial clustering

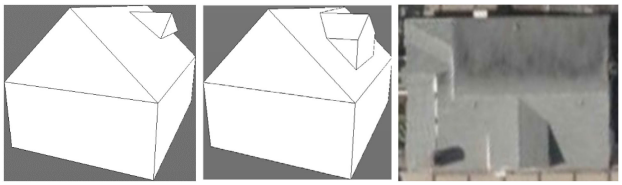


Fig. 5. Primitive fitted dormer (left), its topological integration (middle) with the primary roof model, and the reference image from Bing Map (RoofN3D_467).

of applications with noise (DBSCAN [53]) to classify residuals into different groups. Since the points of each group may not be sufficient to be classified using well-trained model, we calculate the PMDs between each point group and every superstructure primitive in the library iteratively. We then take the primitive with the minimum PMD as the candidate superstructure model. After that, each individual superstructure could be modeled by applying the aforementioned primitive fitting approach in the same way as the primary structure.

V. CityGML Model Generation

After the hierarchical fitting, we obtain optimal parameters for the primary structure and secondary structures of a building. To generate a complete and semantic building model from these parametric primitives, we need to create corresponding semantic objects of a building according to the CityGML boundary representation (B-rep) encoding standard.

To do so, we first need to integrate the primary and secondary primitives determined above. Inconsistence between these primitives exists due to the sequentially separate fitting steps. Although the primary roof and its superstructure have been geometrically modeled, their topological relationships are not established. A direct combination of these geometries may produce topological errors. As shown in Fig. 5, the models of dormer or chimney are usually disconnected from the primary roof model, since the boundary points connecting roof surface and superstructures are not included for superstructure reconstruction. To establish correct topologic relationships between these primitives, we need to refine the geometry of the reconstructed superstructures. This will then be followed by Boolean operations between the primary building geometry and the refined superstructures to generate a topologically consistent model. In the case of dormer with gable shape shown in Fig. 5, considering that the ridge direction of a dormer is usually perpendicular to the roof ridge orientation, the parameter of rotation angle of a dormer is first modified by applying this prior. Then we prolong the ridge line of the dormer to touch the nearest roof surface to get their intersection point. Similarly, we extend the vertical surface of a gabled dormer to intersect with the roof surface, so that we can get the exact height of the dormer. After that, a Boolean union is applied to the roof model and refined dormer model. A Boolean difference operation between the union result and original roof geometry is performed to get the new dormer, which is topologically touched with the roof surface. The parameters for the dormer primitive are then updated by the new length, height, rotation angle, and translation, leading to a refined dormer. To this end, the results from the new dormer together with the primary building structure are integrated to form a complete and topologically correct building model (Fig. 5).

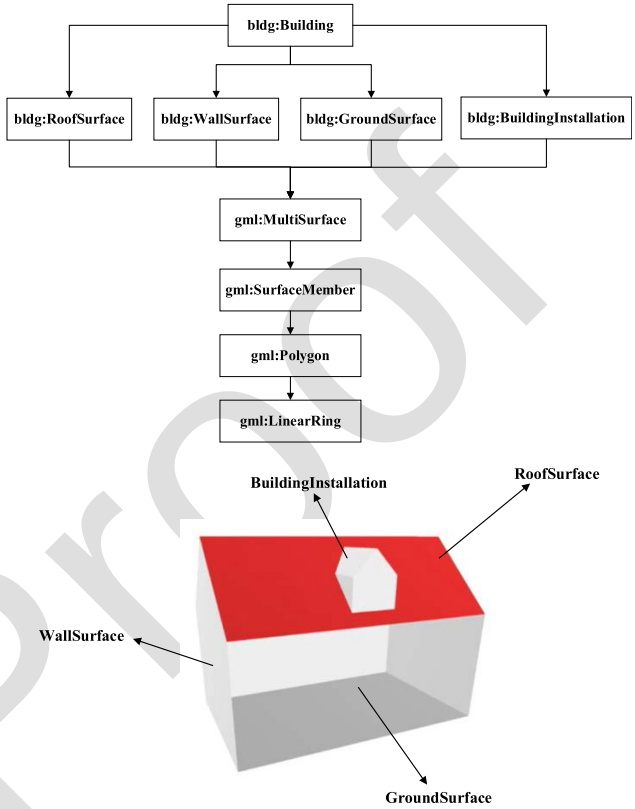


Fig. 6. Example of semantic-geometric model of the CityGML LoD2 building.

The next step is to represent the resultant integrated model to the CityGML LoD2 building model. Under CityGML standard, the exterior shell of a building is semantically composed of objects, such as RoofSurface, WallSurface, GroundSurface, and BuildingInstallation. Each object can be represented geometrically as a gml:MultiSurface geometry, the geometry of BuildingInstallation elements such as chimneys and dormers may be classified by BoundarySurface elements or gml:Solid geometry. A gml:MultiSurface geometry will be further decomposed to a few planar surfaces. Each planar surface is geometrically represented by a gml:Polygon element, which is eventually stored in the form of a series of coordinates. The key semantic objects and their geometric representation of a CityGML LoD2 building model are illustrated in Fig. 6.

Note that each semantic class can be finally defined as a polygon geometry with a series of coordinates. Therefore, the parametric primary structure may be represented by RoofSurface, WallSurface, and GroundSurface elements with corresponding vertices from primitives, while the superstructures could be stored as BuildingInstallation elements according to these vertices from the secondary primitives. Since the semantic classes and their coordinates could be easily extracted from parametric primitives, the LoD2 semantic model will be automatically created by aggregating corresponding vertices to form the semantic surfaces. The geographic coordinates of vertices are inferred through rotating and translating the primitive to correct pose and position, which can be expressed as follows:

$$V_g = R_\kappa \, V_\theta + T_L + T_G \tag{9}$$

where V_θ is the local coordinates of vertices with respect to optimal shape parameters θ_p, R_κ is the rotation matrix with

respect to optimal angle κ, T_L represents the estimated local translation from position parameters, and T_G denotes the given global translation which can be simply obtained from the point cloud.

Each surface of a B-rep model is oriented to distinguish between inside and outside. The surface normals of the outer shell of a CityGML building must point outward. Accordingly, the vertices of a polygon which represent a semantic boundary surface are carefully placed in counter clockwise order to ensure the appropriate orientation. All these polygons are then joined together in an implicit topological relationship. An additional geometric representation as volume model named lod2Solid is generated to explicitly reference the corresponding components of exterior boundary surfaces by using the XLink concept of GML.

VI. Experiments and Evaluation

This section will first introduce the two datasets used. We then brief the results of roof type recognition through a deep learning neural network. Focus is placed on presenting and evaluating the results of primitive fitting and building reconstruction. Discussion is made on their qualities and performance of the proposed framework.

A. Test Datasets

To evaluate the proposed approach, we conduct building reconstruction experiments using two different public point cloud datasets. The first one is a lidar point cloud dataset, RoofN3D [22], which covers a large area in New York City (NYC) and includes rich semantic information. The raw point clouds were acquired by airborne lidar from August 2013 to April 2014 for the high-resolution 3D elevation program (3DEP) product, and provided in LAS format by the U.S. Geological Survey (USGS). The average density of the point clouds is about 4.72 points/m^2. RoofN3D consists of lidar points of 118 074 separated individual buildings with three prelabeled roof shapes, i.e., pyramid, gable, and hip. It should be noted that most point clouds of the RoofN3D are from roofs with very few from facade or ground. To reconstruct complete building models, corresponding digital elevation model (DEM) with submeter resolution is also downloaded from USGS. Besides lidar point clouds, the corresponding footprints and boundary models of all buildings are also provided as reference in RoofN3D. A subset of 910 individual buildings located in NYC Queens residential area (including 81 pyramids, 630 gables, and 199 hips) is used for quality assessment in this study. In addition, two gable building point clouds with obvious roof details are selected to test hierarchical primitive fitting and model reconstruction.

The second dataset is a dense photogrammetric point cloud from the senseFly sample dataset of a small Switzerland village[1] The original point cloud was produced from eBee classic drone images by using the Pix4D software. We manually selected and labeled several individual buildings with detailed superstructures from the entire scene and removed obvious outliers by using the filter tool from the CloudCompare software. Besides the point cloud, a high-resolution ortho-mosaicked imagery is available for evaluation.

[1][Online]. Available: https://www.sensefly.com/education/datasets/?dataset=1419

TABLE I
Accuracy of Building Primitive Recognition by Using PointNet++

Dataset	Number of Buildings		Overall Accuracy
RoofN3D	Training	4,091	87.54%
	Testing	1,023	85.31%
senseFly	Testing	30	86.67%

B. Building Primitive Recognition

One prerequisite for primitive-based 3D building reconstruction is that we know the type of the roof structures and superstructures from the point clouds. Subsequent primitive fitting, superstructure fitting, and semantic model generation rely on this prior knowledge. Consequently, building structure inference based on the predefined primitive library is a necessary process for model generation.

Although not being the focus of this article, deep learning-based classifiers are explored to recognize different roof types in this study. Our study uses the well-known deep neural network PointNet++ as the classifier for building primitive recognition. We select 5114 labeled buildings (210 flats, 213 sheds, 250 pyramids, 1890 hips, and 2551 gables as shown in Fig. 2) from the RoofN3D dataset. We randomly use 80% (4091) of this dataset for training and use 20% (1023) for testing. The RoofN3D trained classifier is then directly used to the photogrammetric dataset for testing. Table I shows the accuracy of building primitive recognition. The testing results from both the RoofN3D lidar dataset and the photogrammetric dataset (30 buildings) are quite consistent, with an overall recognition accuracy of 85.3% and 86.7%, respectively. The recognition results are then used for subsequent primitive fitting.

C. Primitive Fitting

For ALS point clouds with only roof points, we extract the corresponding rooftop primitive from the building primitive library, and perform the same primitive fitting procedure. For dense photogrammetric point clouds with roof details, Fig. 7 illustrates the workflow of the primitive fitting and hierarchical reconstruction. The input point cloud [Fig. 7(a)] is from the lidar dataset and is first filtered into ridge candidate points by a height threshold. RANSAC-based line fitting is applied to these selected points to estimate the initial building primary orientation. The selection of height threshold is not critical; 20 points with the largest Z values is sufficient for ridge line fitting without including significant nonridge points, see Fig. 7(b). Once we calculated the approximate building orientation, parameter optimization with SLSQP is utilized to estimate the eight parameters $(w, l, h_r, h_w, \kappa, tx, ty, tz)$ of the gable building primitive [Fig. 7(c)]. The resultant average PMD is 0.2608 m, while the primary building primitive fits tightly to the points except those on superstructures. After fitting the point cloud onto a gable building primitive, residuals (i.e., residual points) are extracted according to the distance between a point to the mesh surface of the optimal parametric building primitive. Fig. 7(d) shows the points whose distances to the primary primitive mesh are greater than 0.10 m and whose Z values are larger than the minimum elevation of the roof points. These residuals are further clustered by DBSCAN under radius distance (eps $= 0.8$ m) and

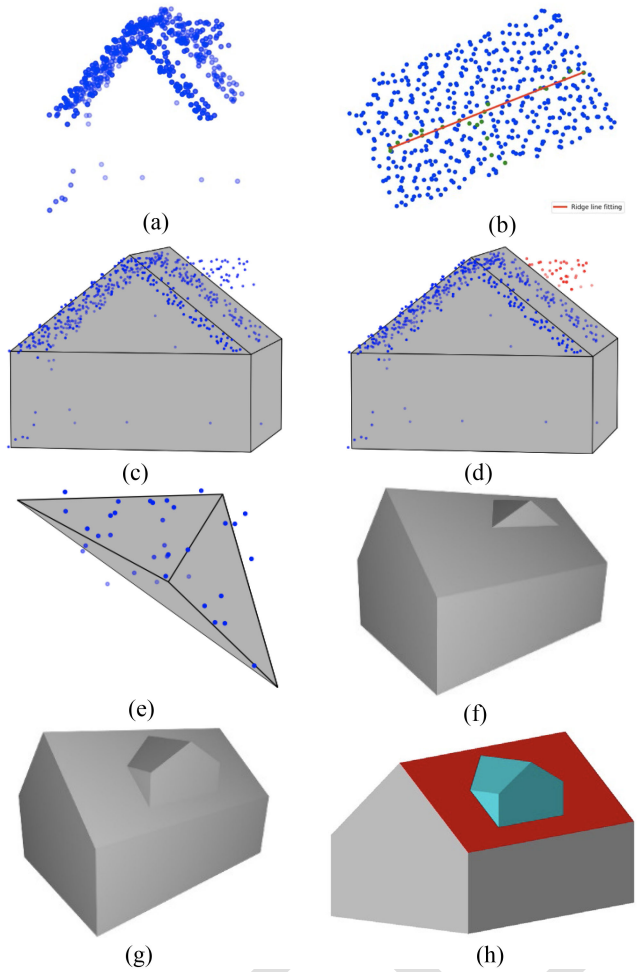


Fig. 7. Reconstruction procedure of a building with detailed roof structure from lidar point cloud (RoofN3D_467).

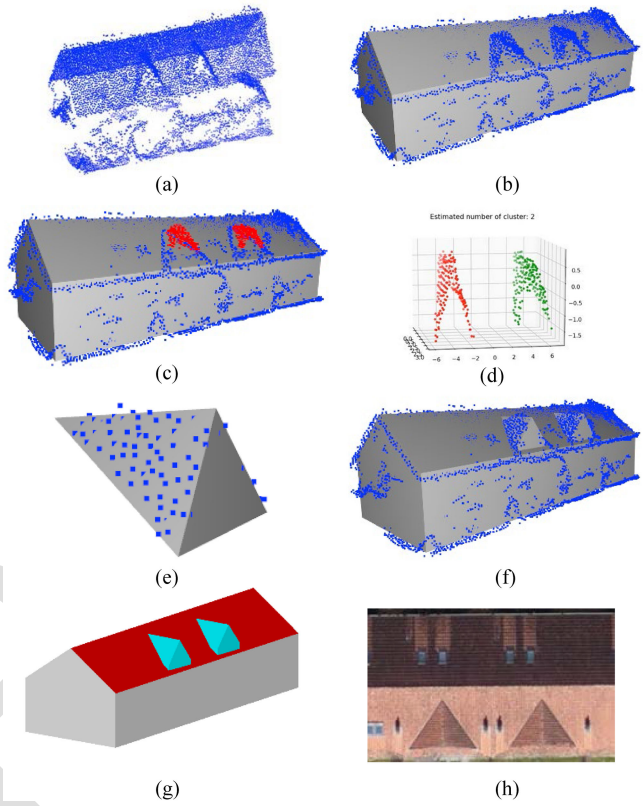


Fig. 8. Reconstruction procedure of a building with detailed roof structure from photogrammetric point cloud (senseFly_7).

minimum number of points (minPts = 5), which means that the required minimum number of points assigned as a cluster within 0.8 m from the seed points is five (5) points. The clustering result is regarded as a superstructure atop the roof. Fig. 7(e) shows that the dormer is fitted to the extracted points with a 0.0850-m PMD by using the same techniques as the one fitting the primary roof structure. Most points find a good fit to this dormer primitive except a few outliers. Fig. 7(f) demonstrates the simple combination of the primary and secondary primitives. To reconstruct a topologically correct model, the parametric dormer is further refined by prolonging to the nearest roof surface and by applying Boolean operation with the primary building model, whose result is illustrated by Fig. 7(g). As shown in Fig. 7(h), all the reconstructed geometries are finally integrated into one CityGML LoD2 building model with semantic surfaces and detailed roof structure.

Fig. 8 illustrates the similar reconstruction procedure for a gable roof building with two parallel dormers from the senseFly photogrammetric dataset. The input point cloud consists of 111 27 points. The average PMD of parameter optimization for primary structure is 0.2340 m. A total of 545 superstructure points are extracted according to the primary primitive fitting result. Fig. 8(d) shows two clusters representing two dormers obtained by DBSCAN. The average PMD values with respect to the two dormers are 0.1026 and 0.1507 m, respectively. The fitting result of one dormer primitive is shown in Fig. 8(e). The final CityGML model after hierarchical reconstruction is shown in Fig. 8(g). A UAV image clip is shown in Fig. 8(h) for reference.

The proposed approach is further validated on a number of buildings from the two datasets. Some of the reconstruction results and their corresponding images are depicted in Fig. 9. Table II summarizes the input point count, roof type, distance costs of the primary and secondary primitive fitting, and clusters of DBSCAN for the above seven buildings (1 in Fig. 7, 1 in Fig. 8, and 5 in Fig. 9). We achieve good parameter estimations for all of the primary structures. All the PMD values for primary building primitive fitting are better than 0.30 m, partially because the residuals belonging to superstructures also contribute to the costs. The PMD for subsequent dormer primitive fitting is in the range of 0.0562 to 0.1285 m. This is much smaller than the primary fitting since the input for dormer fitting has only one individual cluster and there is no modeling bias involved. The above experimental results demonstrate that the hierarchical reconstruction approach is effective and capable of generating building models with detail roof structures. Moreover, the optimization method with PMD as a cost metric is robust to noise and outliers.

D. Reconstructed Buildings

The independent quantitative evaluation of the reconstructed building models is a challenging task due to the absence of benchmark or ground truth models. Since the RoofN3D dataset provides reference boundary models and footprints, we use them

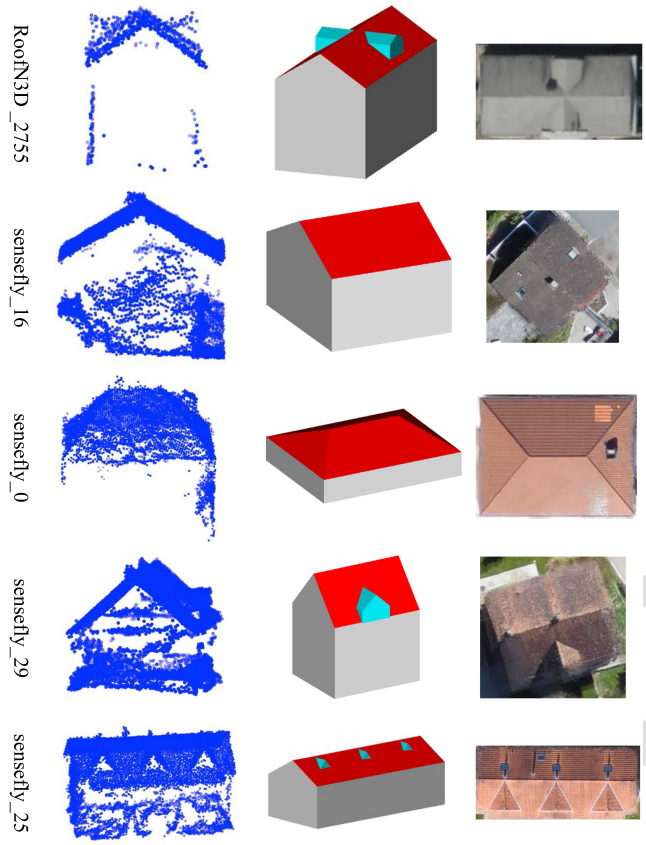


Fig. 9. Reconstruction results of selected buildings (left: point cloud, middle: CityGML model, right: Reference images).

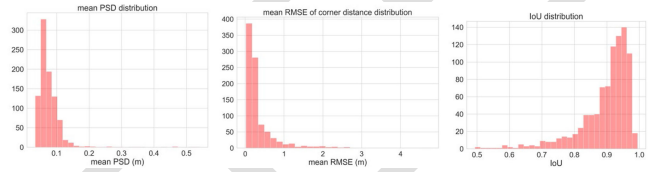


Fig. 10. Distribution of quality metrics (PSD, RMSE, IoU from left to right) for the reconstructed 910 buildings in the RoofN3D dataset.

to evaluate the geometric quality of the reconstructed building models. The geometric quality consists of the following three criteria. 1) Point-surface-distance (PSD): The average orthogonal distance between the input points and their nearest roof planar surface of the reconstructed model [54]. 2) The root mean square error (RMSE) of corner distances: The distances between a model vertex and the nearest roof corer point in the reference model. 3) 2D IoU between the bottom of the reconstructed building model and the corresponding building footprint of the reference model.

For a subset with a total of 910 buildings from the RoofN3D dataset, the above three metrics are calculated to evaluate the geometric quality of the reconstructed buildings. The distributions of these quantitative metrics across all 910 buildings are depicted in Fig. 10. The averages of each quality metric for all buildings are taken to determine the overall quality.

The three evaluation criteria address different aspects of the reconstructed building models. The PSD quantifies the discrepancies between the point cloud and the reconstructed model,

TABLE II
STATISTICS (POINT TO MESH DISTANCE) OF PRIMITIVE FITTING ON SELECTED BUILDINGS

Building ID	#Points	Roof type	Primary structure	Secondary structures	
			PMD (m)	#Clusters	PMD (m)
RoofN3D_467	489	Gable	0.2608	1	0.0850
RoofN3D_2755	709	Gable	0.2087	2	0.0902 0.1285
senseFly_16	7654	Gable	0.2637	0	NA
senseFly_0	7878	Hip	0.1190	0	NA
senseFly_29	6557	Gable	0.2923	1	0.0562
senseFly_25	10074	Gable	0.2549	3	0.1250 0.1129 0.1063
senseFly_7	11127	Gable	0.2340	2	0.1026

TABLE III
QUALITY METRICS OF THE RECONSTRUCTED BUILDINGS IN THE ROOFN3D DATASET

	Pyramid	Gable	Hip	Overall
# Buildings	81	630	199	910
Mean PSD (m)	0.0665	0.0834	0.0648	0.0778
Mean RMSE(m)	0.2191	0.4021	0.2740	0.3578
Mean IoU (%)	90.69	88.27	93.20	89.57

while the RMSE of corner points indicates the discrepancies at key locations between the reconstructed model and the reference model. The IoU evaluates the horizontal boundary accuracy between the model and reference. Fig. 10 shows for the more than 84.7% of the reconstructed buildings their perpendicular distance between point clouds and their closest reconstructed roof surfaces is smaller than 0.1 m, and the overall PSD is 0.0778 m. The overall RMSE of corner distances for all buildings is 0.3578 m (0.78 times ground point spacing), with a few percentages (7.03%) worse than 1.0 m. The IoU for most (89.34%) building models are higher than 80%, which mostly demonstrates the accuracy of the derived building boundary.

Primitive-based building modeling allows us to evaluate the reconstructed buildings in terms of roof shapes. Table III lists the distribution of quality metrics among different roof shapes. The overall average PSD and RMSE of the proposed approach are small, which indicates that we achieved a good model fitting with small residuals. It should be noted that the average PSD is computed from all the points of the input point cloud without removing any outliers (points from superstructures), which have boosted the error metric. It should be noted that the overall PSD of 910 buildings is smaller than the result of our previous work by using PSD and IoU as cost metrics. There seems to be a trend that the building group that has a greater number of buildings gets slightly larger errors. For instance, the RMSE for 630 gable buildings is 0.4021 m (0.87 times ground spacing distance), whereas it is 0.2740 m (0.60 times ground spacing

Fig. 11. Footprints (top) of reconstructed models (red) and ground truth (green), and a 3D view of the reconstructed building models (bottom).

distance) for 199 hip buildings, and 0.2191 m (0.48 times ground point spacing) for 81 pyramid buildings. This is likely that more buildings would have more diversities in their actual shapes that deviate from the standard primitives we defined. Such results reveal the primitive fitting accuracy in terms of corner point differences is in the range of one half (0.5) to one (1) ground point spacing. As for the accuracy of plane fitting, the metrics of PSD cannot be more optimized in the range of 0.06–0.08 m. Such a scenario is largely due to the fact that there are more than enough points for obtaining reliable primitive through primitive fitting.

Finally, the roof boundary discrepancies between point cloud and model can be visualized by overlapping the building footprints derived from the reconstructed models and the ground truth provided by RoofN3D. Part of the test area is presented in Fig. 11 in both 2D and 3D views. Most of the ground polygons from reconstructed models almost completely coincide with the referenced footprints, which is consistent with the quantitative measure IoU (89.6%) in Table III. Only a few buildings have a size or shape noticeably different from the ground truth because the ground polygons of our predefined building primitives are rigid rectangles, which is not true for some buildings. It is also noticed that the reconstructed models tend to overestimate the footprints of the ground truth. This is because any single artifact or noise lidar point in the building segmentation results would lead to an extension of the actual building boundary, whereas an

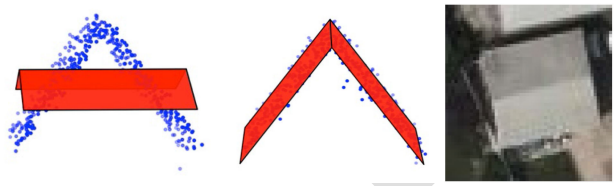


Fig. 12. Primitive fitting results without (left, PMD = 0.5198 m) and with (middle, PMD = 0.0745 m) initial estimation of k. The right figure is a reference image from Bing Map. The building ID is RoofN3D_224440.

underestimation of the building boundary would need a complete missing of the lidar points at the building boundary, which is unlikely to happen.

It should be stated that RoofN3D mainly provides roof points of each building. There may not be enough points on facades or ground. This situation makes the model fitting difficult for a complete building primitive with wall surfaces. To solve this problem, a rooftop primitive model simplified from the complete building primitive is used for the same optimization framework, i.e., all parameters except wall height are estimated. After that, the ground elevation with respect to each roof vertex is calculated by combining the DEM and its footprint coordinates on the ground, and the average height of all facades can be finally achieved. Therefore, combing point clouds and corresponding DEM data is a feasible solution to reconstruct complete building models from point clouds acquired by airborne LiDAR and aerial photogrammetry.

VII. Discussion

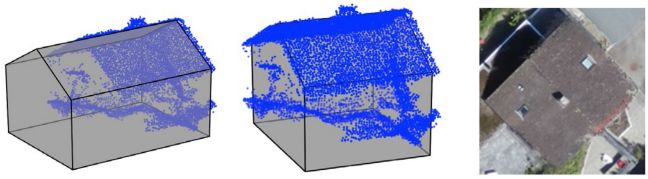
A. Initialization of Primitive Parameters

The convergence of the holistic optimization process for primitive fitting can be affected by the initials of the parameters to be estimated. We found that the initial rotation angle κ needs certain care since it affects the convergence when fitting the hip and gable primitives. A comparison of primitive fitting results from RoofN3D dataset with different initial rotation κ is illustrated in Fig. 12. Comparing the two primitive fitting results, the default $\kappa = 0$ produces a wrong set of parameters, whereas the initial angle derived by RANSAC line fitting leads to a correct fit to the input point cloud. We applied this initialization approach to all gable roofs and hip roofs in the two test datasets. All lead to stable and correct convergency. Without proper initialization of the building orientation, the optimization solution could be terminated at a local minimal. In contrast to building orientation κ, other primitive parameters, such as length, width, roof height, are less critical to the convergency. Their initials can be determined based on the bounding box of the point cloud.

B. Boundary Conditions

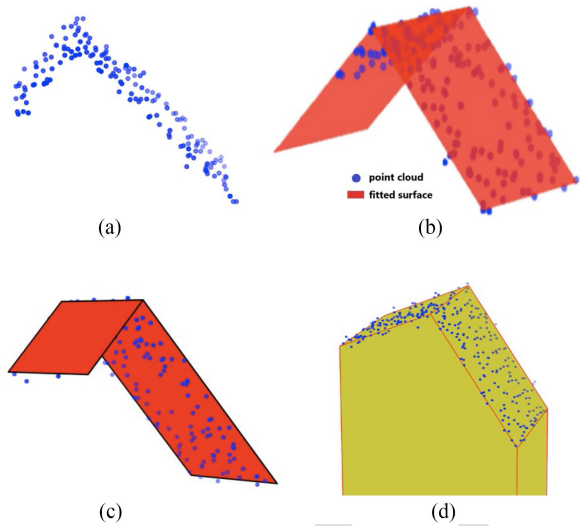
Boundary conditions are essential for the correct convergence of the optimization. During parameter estimation with SLSQP method, we found that some primitive fitting results are not good by only considering the point-mesh-distance constraint and a proper parameter initialization. Fig. 13(a) presents one example of such results.

Although the two roof surfaces, three wall surfaces, and the ground surface are well fitted with the point cloud, the rest one facade with triangle can be far from the corresponding input



(a) (b) (c)

Fig. 13. Comparison of primitive fitting results without (left, PMD = 0.3233 m) and with (middle, PMD = 0.2637 m) the area constraint. The right figure is the reference image of the building sensefly_16.



(a) (b) (c) (d)

Fig. 14. Asymmetric roof reconstruction with modified gable roof primitive (RoofN3D).

point cloud, which leads to a large distance cost. The area constraint of the building boundary, which considers both width and length parameters, can yield an ideal fitting result Fig. 13(b). Its average PMD distance is reduced from 0.3233 to 0.2637 m, and the estimated length parameter is changed from 15.3285 to 9.9788 m.

C. Bias of Symmetric Primitives

As stated earlier, symmetries are implicitly enforced in our predefined primitives (i.e., gable, hip). This is beneficial to achieve integrity for building models in most cases. However, there are cases where roof patches are not symmetric in the RoofN3D dataset. Fig. 14(a) is one such example, whose fitting result by using a symmetric gable roof primitive is shown in Fig. 14(b). In this study, the aforementioned quality evaluation is helpful to detect this situation since its PSD, RMSE, and IoU are usually much worse than the other buildings. A few cases with higher errors can be found from Fig. 10, e.g., there are a total of 9 building models (including 7 gable roofs and 2 hip roofs) whose mean PSD are larger than 0.3 m. As we visually check the point clouds and the reference models of these cases, most of them exhibit asymmetric roof structures. To solve this problem, asymmetric gable roof structure is defined as shown in Fig. 3(b). In contrast to symmetric gable roof primitive, the asymmetric gable roof surface has two widths and two heights. Based on this new parametric primitive, the same optimization framework is applied and a corresponding roof model with much better quality is achieved (Fig. 14c). The result is more consistent with the provided reference model shown in Fig. 14(d).

TABLE IV
FITTING ERRORS OF BUILDING RECONSTRUCTION METHODS FOR THE ROOFN3D DATASET

	Wichmann [22]	Li [15]	Proposed Method
# Buildings	910	910	910
Mean PSD (m)	0.0993	0.0805	0.0778

D. Computational Performance

All algorithms in this article are implemented in Python 3.7.4. The program is carried on a PC with Intel Core i5 CPU @ 2.40 GHz and 8.0-GB RAM. Parameter estimation is the most computationally expensive part of the entire reconstruction process. The total time taken for parameter optimization of all the 910 buildings in the RoofN3D dataset is 57.28 min, with an average of 3.78 s per building. In order to reduce the computing time for large-scale building reconstruction, a parallelization scheme is adopted on a multicore computer. The total time is reduced to 23.85 min, or 1.57 s per building. Since the parameter estimation of each building is independent, the parallelization is easy to implement. This allows us to reach a good optimization of the primitive parameters while significantly reducing the computational cost on a multicore computer. In general, the proposed approach is efficient for large-scale building reconstruction.

E. Comparison With Other Methods

To demonstrate the performance of the proposed method, we make a comparison with the recent work from Wichmann [22] and the popular RANSAC-based method by using the same test datasets.

First, our method is quantitatively compared with Li [15] and Wichmann [22] by using the RoofN3D dataset. Wichmann [22] produced reconstructed building models by using reassignment-based subsurface growing, half-space modeling, and divisive clustering techniques, while Li [15] reconstructed building models by using roof primitive fitting method. For a subset of 910 buildings in the RoofN3D dataset, we calculated the PSD between each point in the point cloud data and the corresponding building models produced by the three methods. A quantitative comparison between these methods is listed in Table IV.

It is seen that the overall average PSD of our models is smaller than the other two methods, which indicates that the proposed method achieved the fitting result. The proposed method is the extension of Li [15]. Although the modified objective function in our method has only one cost metric, it performs better than the combination of PSD and IoU metrics. Moreover, the problem of weight parameter tuning is avoided in the extended algorithm.

In addition, detailed comparison is made for individual buildings. Fig. 15 illustrates the reconstruction results of three buildings from the RoofN3D dataset by using Wichmann method [22], RANSAC-based method [7], and our method. The RANSAC method in CloudCompare[2] uses a fitting threshold of 0.2 m and a minimum supporting point number of 50. Both Wichmann method and RANSAC approach did not produce superstructures atop of the building roof. PMD RMSE is employed

[2][Online]. Available: https://www.cloudcompare.org/

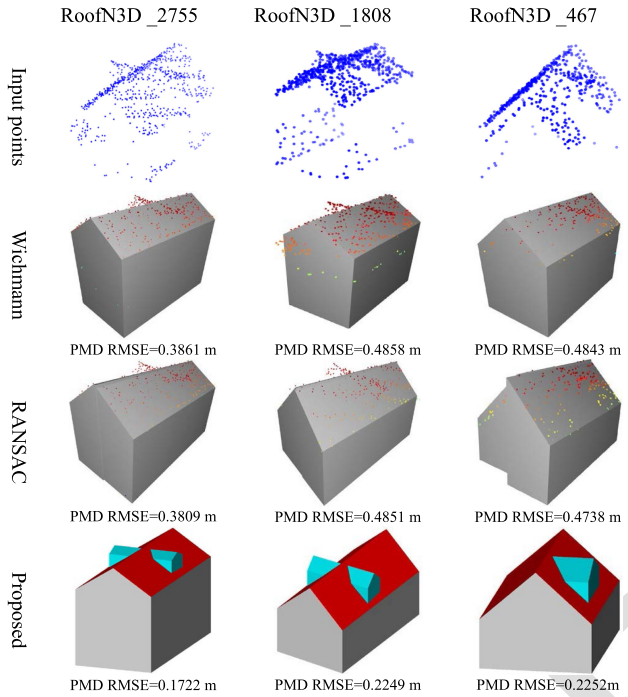


Fig. 15. Reconstruction results of three buildings from Wichmann, RANSAC, and the proposed method by using the RoofN3D dataset.

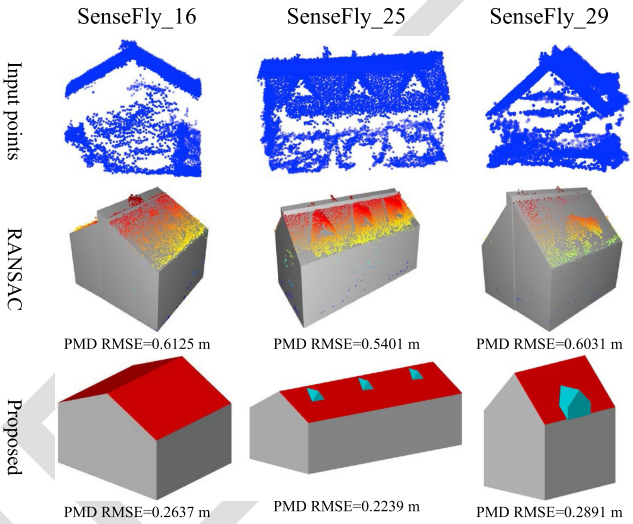


Fig. 16. Reconstruction results of three buildings from the RANSAC-based method and the proposed optimal model fitting method by using the SenseFly dataset.

to quantitative evaluate the performance of the three methods. Fig. 15 shows that our method achieves the smallest PMD RMSE than the other two approaches.

Since there are no reference models from Wichmann for the SenseFly dataset, we only compare our algorithm with the RANSAC-based method. As shown in Fig. 16, detail roof structures of the three buildings are not correctly recovered by the RANSAC-based method. Furthermore, since the RANSAC-based primitive fitting is based on sequential local fitting, its results are lack of topological integrity and geometric rigor. In contrast, our approach successfully reconstructs complete building models with its superstructures. The use of primitive models is capable to retain the desired topology, symmetry, and semantics among the building components. All the PMD RMSE of our models are smaller than the ones from the RANSAC-based method, meaning the holistic primitive fitting approach has a superior reconstruction performance.

VIII. Conclusion

We present a hierarchical holistic primitive fitting approach for building reconstruction with roof details from point cloud in an automated manner. It is under a mathematic framework of primitive fitting that simultaneously determines all of the parameters of a selected building primitive. Based on a set of predefined building primitives with piecewise smooth surfaces, the primary structure of a building is first reconstructed by using the proposed holistic primitive fitting algorithm. After analysis and clustering of the residuals from this fitting step, secondary roof structures or superstructures are subsequently reconstructed in a similar manner. The hierarchically reconstructed primitives are further refined and aggregated to form a complete CityGML LoD2 building model with roof details. Experiments conducted on airborne lidar point clouds and UAV photogrammetric point clouds with different roof shapes and superstructures demonstrate the effectiveness and accuracy of this approach.

Under this novel optimization framework, one of our contributions is the introduction of the point-to-mesh distance for calculating the objective function. Such distance is explored as the cost metric of the constraint optimization method, while RANSAC-based ridge line detection provides proper initialization of the building orientation parameter. Compared with our previous work, the PMD is more robust and accurate for parameter estimation than the PSD metric. Moreover, PMD is the only cost metric for the new objective function, so there is no human adjustment of weight parameter to balance two or more cost metrics. Another contribution is the instruction of a set of inequality constraints into the optimization framework. This formulation is essential to lead to a determined solution for the fitting problem. All these developments assure a robust and accurate fitting results than our previous work. Experimental results with a subset of 910 buildings from RoofN3D showed the proposed method can accurately and effectively reconstruct building models with three different roof structures. Additionally, the final reconstruction results of our approach are semantic-geometric coherent representation of buildings in CityGML format. The introduction of semantic knowledge in explicit representations provides an intuitive understanding of building structure and potential applications. Beside rich semantic information, the building models generated by our approach are intrinsically consistent and regularized, since the geometric and topological constraints are also embedded in the definition of the 3D primitives. Although the primitive shapes defined in our library are still limited, and only relatively simple buildings with regular dormers or chimney are represented, the developed framework for constrained optimization can be easily adapted for other types of buildings. Even for buildings that are not strictly regular, some complicated primitives could be further added into this library, and the calculation of PMD is still easy to perform. Furthermore, such 3D solid primitive-based reconstruction is less vulnerable for sparse point clouds and therefore can tolerate incomplete data samples. As such it can be a promising new approach for effective building modeling over large areas.

In spite of the above progress, it is still a challenging task to reconstruct building models with geometric complexity and semantic ambiguity, such as a townhouse with several adjacent units. First, it is difficult to automatically segment point clouds with complex structures into simple primitives. The interpretation of roof type with complicated shapes becomes less reliable and robust. In addition, the integration of different primitives into an accurate building model is not easy to achieve by simple nesting or Boolean union of the building components. This is because the topology of a complex building is complicated and the geometry of a building component can be influenced by adjacent components. In future work, we will extend the primitive library by including more types of complex structures, and to investigate effective approach to decomposing a complex building into secondary primitives using learning-based method.

REFERENCES

[1] X. Gao, S. Shen, Y. Zhou, H. Cui, L. Zhu, and Z. Hu, "Ancient Chinese architecture 3D preservation by merging ground and aerial point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 143, pp. 72–84, Mar. 2018.

[2] M. Awrangjeb, S. A. N. Gilani, and F. U. Siddiqui, "An effective data-driven method for 3D building roof reconstruction and robust change detection," *Remote Sens.,* vol. 10, no. 10, 2018, Art. no. 1512.

[3] R. Cao, Y. Zhang, X. Liu, and Z. Zhao, "3D building roof reconstruction from airborne LiDAR point clouds: A framework based on a spatial database," *Int. J. Geographical Inf. Sci.*, vol. 31, no. 7, pp. 1359–1380, 2017.

[4] N. Haala and M. Kada, "An update on automatic 3D building reconstruction," *ISPRS J. Photogrammetry Remote Sens.*, vol. 65, no. 6, pp. 570–580, 2010.

[5] S. M. Iman Zolanvari and D. F. Laefer, "Slicing method for curved façade and window extraction from point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 119, Sep. 2016, pp. 334–346, 2016.

[6] B. Xiong, M. Jancosek, S. O. Elberink, and G. Vosselman, "Flexible building primitives for 3D building modeling," *ISPRS J. Photogrammetry Remote Sens.*, vol. 101, pp. 275–290, 2015.

[7] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[8] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1554–1567, Mar. 2010.

[9] G. Vosselman, B. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," *Proc. Int. Conf. NATSCAN, ISPRS*, vol. XXXVI-8-W2, 2004, pp. 33–38.

[10] L. Zahra and H. Ayman, "An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 93, no. 7, pp. 192–212, 2014.

[11] L. Nan and P. Wonka, "PolyFit: Polygonal surface reconstruction from point clouds," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2372–2380.

[12] H. Huang, C. Brenner, and M. Sester, "A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 79, pp. 29–43, 2013.

[13] G. Gröger, T. H. Kolbe, C. Nagel, and K.-H. Häfele, "Open geospatial consortium OGC city geography markup language (CityGML) encoding standard," OGC, Wayland, MA, USA, Version 2.0.0, 2012. [Online]. Available: http://www.opengeospatial.org/legal/

[14] F. Rottensteiner, G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf, and J. Jung, "Results of the ISPRS benchmark on urban object detection and 3D building reconstruction," *ISPRS J. Photogrammetry Remote Sens.*, vol. 93, pp. 256–271, 2014.

[15] Z. Li, W. Zhang, and J. Shan, "Holistic parameteric reconstruction of building models from point clouds," *Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. XLIII-B2-2020, pp. 689–695, no. 2020.

[16] J. Overby, L. Bodum, E. Kjems, and P. M. Iisoe, "Automatic 3D building reconstruction from airborne laser scanning and cadastral data using hough transform," *Int. Arch. Photogrammetry Remote Sens.*, vol. 35-B3, pp. 296–301, 2004.

[17] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. J. Guibas, "Supervised fitting of geometric primitives to 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 2652–2660.

[18] 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 J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 10, pp. 4199–4217, Oct. 2014.

[19] J. P. Bauchet and F. Lafarge, "Kinetic shape reconstruction," *ACM Trans. Graph.*, vol. 39, no. 5, 2020, Art. no. 156.

[20] A. Nurunnabi, D. Belton, and G. West, "Robust segmentation for large volumes of laser scanning three-dimensional point cloud data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4790–4805, Aug. 2016.

[21] V. Verma, R. Kumar, and S. Hsu, "3D Building detection and modeling from aerial LIDAR data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2006, pp. 2213–2220.

[22] A. Wichmann, A. Agoub, and M. Kada, "ROOFN3D: Deep learning training data for 3D building reconstruction," *Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 42, no. 2, pp. 1191–1198, 2018.

[23] M. Li, F. Rottensteiner, and C. Heipke, "Modelling of buildings from aerial LiDAR point clouds using TINs and label maps," *ISPRS J. Photogrammetry Remote Sens.*, vol. 154, pp. 127–138, Feb. 2019.

[24] J. Song, S. Xia, J. Wang, and D. Chen, "Curved buildings reconstruction from airborne LiDAR data by matching and deforming geometric primitives," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1660–1674, Feb. 2021.

[25] D. Chen, R. Wang, and J. Peethambaran, "Topologically aware building rooftop reconstruction from airborne laser scanning point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7032–7052, Dec. 2017.

[26] C. R. Qi, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 77–85.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *NeurIPS 2017*, 2017. [Online]. Available: https://arxiv.org/abs/1706.02413

[28] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 4490–4499.

[29] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vision*, 2019, pp. 9276–9285.

[30] L. Zhang and L. Zhang, "Deep learning-based classification and reconstruction of residential scenes from large-scale point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1887–1897, Apr. 2018.

[31] Y. H. Tseng and S. Wang, "Semiautomated building extraction based on CSG model-image fitting," *Photogrammetry Eng. Remote Sens.*, vol. 69, no. 2, pp. 171–180, 2003.

[32] W. Zhang, P. Grussenmeyer, G. Yan, and M. Mohamed, "Primitive-based building reconstruction by integration of lidar data and optical imagery," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XXXVIII-5, no. 5, pp. 7–12, 2012.

[33] 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 Sens*, vol. 7, no. 10, pp. 13945–13974, 2015.

[34] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-Point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vision*, vol. 99, no. 1, pp. 69–85, 2012.

[35] M. Jarzabek-Rychard and A. Borkowski, "3D building reconstruction from ALS data using unambiguous decomposition into elementary structures," *ISPRS J. Photogramm. Remote Sens.*, vol. 118, no. C, pp. 1–12, 2016.

[36] Y. Zheng, Q. Weng, and Y. Zheng, "A hybrid approach for three-dimensional building reconstruction in Indianapolis from LiDAR data," *Remote Sens,* vol. 9, no. 4, 2017, Art. no. 310.

[37] M. Berger *et al.*, "A survey of surface reconstruction from point clouds," *Comput. Graph. Forum*, vol. 36, no. 1, pp. 301–329, 2017.

[38] L. Zhang, Z. Li, A. Li, and F. Liu, "Large-scale urban point cloud labeling and reconstruction," *ISPRS J. Photogrammetry Remote Sens.*, vol. 138, pp. 86–100, 2018.

[39] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, "Applications of 3D city models: State of the art review," *ISPRS Int. J. Geo-Inf.*, vol. 4, no. 4, pp. 2842–2889, 2015.

[40] S. Donkers, H. Ledoux, J. Zhao, and J. Stoter, "Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings," *Trans. GIS.*, vol. 20, no. 4, pp. 547–569, 2016.

[41] X. Liu, X. Wang, G. Wright, J. C. P. Cheng, X. Li, and R. Liu, "A state-of-the-art review on the integration of building information modeling (BIM) and geographic information system (GIS)," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 2, pp. 1–21, 2017.
[42] A. Henn, G. Gröger, V. Stroh, and L. Plümer, "Model driven reconstruction of roofs from sparse LiDAR point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 76, pp. 17–29, 2013.
[43] P. Jayaraj and A. M. Ramiya, "3D CityGML building modelling from lidar point cloud data. International archives of the photogrammetry," *Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. XLII–5, pp. 175–180, 2018.
[44] A. Salehi and A. Mohammadzadeh, "Building roof reconstruction based on residue anomaly analysis and shape descriptors from LiDAR and optical data," *Photogrammetry Eng. Remote Sens.*, vol. 83, no. 4, pp. 281–291, 2017.
[45] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros and J. Malik, "Learning shape abstractions by assembling volumetric primitives," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1466–1474.
[46] S. D. Roth, "Ray casting for modeling solids," *Comput. Graph. Image Process.*, vol. 18, no. 2, pp. 109–144, 1982.
[47] K. Hormann and A. Agathos, "The point in polygon problem for arbitrary polygons," *Comput. Geometry*, vol. 20, no. 3, pp. 131–144, 2001.
[48] M. W. Jones, J. A. Bærentzen, and M. Sramek, "3D distance fields: A survey of techniques and applications," *IEEE Trans. Vision Comput. Graph.*, vol. 12, no. 4, pp. 581–599, Jul./Aug. 2006.
[49] B. A. Payne and A. W. Toga, "Distance field manipulation of surface models," *IEEE Comput. Graph. Appl.*, vol. 12, no. 1, pp. 65–71, Jan. 1992.
[50] F. Lafarge, R. Keriven, and M. Brédif, "Insertion of 3-D-primitives in mesh-based representations: Towards compact models preserving the details," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1683–1694, Jul. 2010.
[51] P. Boggs and J. Tolle, "Sequential quadratic programming," *Acta Numerica,* vol. 4, pp. 1–51, 1995.
[52] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.
[53] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
[54] S. O. Elberink and G. Vosselman, "Quality analysis on 3D building models reconstructed from airborne laser scanning data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 66, no. 2, pp. 157–165, 2011.

Wenyuan Zhang received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2011.

He was a Visiting Scholar with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, USA, during September 2019 and August 2020. He is working as an Associate Professor with Central China Normal University, Wuhan, China. His research interests include 3D building modeling and spatial data mining.

Zhixin Li received the M.S. degree in geomatics engineering, in 2019, from Purdue University, West Lafayette, IN, USA, where he is working on building reconstruction from point clouds for his Ph.D. degree with the Lyles School of Civil Engineering.

His research interests include objects recognition with remote sensing, urban 3D building modeling, UAV photogrammetry, and 3D deep neural network.

Mr. Li was the recipient of the Best Paper Award at the IEEE/ISPRS Workshop in Conference on Computer Vision and Pattern Recognition 2019 Earth Vision as a coauthor.

Jie Shan (Senior Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1989.

He has worked with universities in China, Germany, and Sweden. He is a Professor with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, USA. His research interests include sensor geometry and positioning, object extraction and reconstruction from images and point clouds, urban remote sensing, pattern recognition, and data mining of spatial, temporal, and semantic data.

Dr. Shan was elected as ASPRS Fellow. He was the recipient of multiple best paper awards and recognitions. He serves on the editorial boards of several remote sensing journals.