

Article

A Semantic Modelling Framework-Based Method for Building Reconstruction from Point Clouds

Qingdong Wang ¹, Li Yan ^{1,*}, Li Zhang ^{2,*}, Haibin Ai ² and Xiangguo Lin ²

¹ School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; qd.wang@whu.edu.cn

² Institute of Photogrammetry and Remote Sensing, Chinese Academy of Surveying and Mapping, Beijing 100830, China; ahb32@163.com (H.A.); linxiangguo@gmail.com (X.L.)

* Correspondence: liyan@sgg.whu.edu.cn (L.Y.); zhangl@casm.ac.cn (L.Z.);
Tel.: +86-27-6877-8265 (L.Y.); +86-10-6388-0531 (L.Z.)

Academic Editors: Norman Kerle, Randolph Hamilton Wynne and Prasad S. Thenkabail

Received: 12 June 2016; Accepted: 30 August 2016; Published: 8 September 2016

Abstract: Over the past few years, there has been an increasing need for semantic information in automatic city modelling. However, due to the complexity of building structure, the semantic reconstruction of buildings is still a challenging task because it is difficult to extract architectural rules and semantic information from the data. To improve the insufficiencies, we present a semantic modelling framework-based approach for automated building reconstruction using the semantic information extracted from point clouds or images. In this approach, a semantic modelling framework is designed to describe and generate the building model, and a workflow is established for extracting the semantic information of buildings from an unorganized point cloud and converting the semantic information into the semantic modelling framework. The technical feasibility of our method is validated using three airborne laser scanning datasets, and the results are compared with other related works comprehensively, which indicate that our approach can simplify the reconstruction process from a point cloud and generate 3D building models with high accuracy and rich semantic information.

Keywords: semantic reconstruction; point cloud; shape grammar; building reconstruction; semantic modelling; city modelling

1. Introduction

3D building models are demanded for various fields, such as city planning, city emergency, urban computing, indoor positioning, and GIS services. However, due to the demand for city modelling from emerging industries, such as virtual reality (VR), augmented reality (AR), the self-driving industry, and building information modelling (BIM) industry, some considerably higher requirements of building models are needed, including entity models, enriched semantics, and more detailed geometries. Recently, there have been many methods for 3D building reconstruction, which can be classified into three types: data-driven methods, model-driven methods, and knowledge-based methods. For data-driven methods, aerial data and laser scanning data have been the primary data sources for 3D reconstruction, i.e., 3D information on buildings, roads and other facilities are extracted automatically or semi-automatically from these data, and a triangular mesh or outline is used to model the buildings [1–6]. Sui et al. [7] proposed a layer-wise floorplan-based method for building reconstruction, which can reconstruct the building composed of planar polygons and aligned with the gravity direction. The data-driven approach can obtain a high degree of automation but also has a strong dependence on the data source. In the occluded area or area with many noise points, for instance, the data-driven methods will result in irrational products. Thus, the quality, precision, and completeness of the models also depend on that of the data source. Furthermore, this type of

approach is unable to be used to design architecture, and there is no semantic information, such as façade, windows, doors, and roofs, that restricts the application of the city models.

Meanwhile, in the model-driven methods, several grammar-based methods are used for creating models, and most of the methods are inspired by the shape grammar [8], which is used to describe geometric shapes in 2D or 3D. Parish et al. [9] proposed a system based on the shape grammar and L-system for the procedural modelling of a large-scale city. Wonka [10] presented the split grammar, a parametric grammar, which was also inspired by the shape grammar. The split grammar is primarily used to generate the geometric details on the facades of buildings. Based on their works, Müller et al. presented a new grammar-computer generated architecture (CGA) shape [11], which can generate a hierarchical description of the building and the semantic information during the modelling process. Furthermore, these information can be reused to model the variance by adjusting the parameters. Larive et al. presented a wall grammar based on the split grammar [12] and, compared to other grammars, the abstraction and derivation from an object-oriented concept were introduced into the design of this grammar, which improved its readability and usability. To facilitate the description of a building, a number of systems for hierarchical and parametric modelling were proposed [13,14]. Although the grammar-based method can effectively model buildings with a detailed structure, manual interaction is still involved. To improve the automation, several methods combining data analysis and model-driven methods were proposed. Müller et al. combined the façade image analysis with the CGA shape to reconstruct the building façade [15], in which manual information was used to split the façade image into floors and tiles, which were then encoded as shape grammar rules [16] to reconstruct the façade. To reconstruct the whole building, several 3D point cloud-based methods were proposed [17–20]. Another similar approach was proposed [21], in which Manhattan grammar converts the reconstruction into processes of extracting the building contour. This type of method can automatically reconstruct buildings with semantic information, however, there are two challenges to be solved: the architectural rules detection and spatial relationship detection among building elements. Therefore, many existing methods only focus on the simple structural buildings.

To reconstruct the semantic labelled building models at a high level of automation, the knowledge-based approaches were proposed [22–26]. In [22,23], the knowledge about features' sizes, positions, orientations, and topology was introduced to recognize building elements. In [24], knowledge about local symmetry of the roof that is contained in the building structure is exploited to decompose the building into some semantic building primitives. In [25], a knowledge-based algorithm is designed to generate building models from point cloud and aerial images. In the algorithm, the knowledge about the geometry of building arrangement is used to calculate normal vectors to candidate roof segments. In [26], the facade knowledge was stored in a hierarchical tree to conduct the classification. In these types of methods, the number of geometric features needed for knowledge is less than that in data-driven methods, and these type of methods have advantages in segmentation of point clouds and semantic feature recognition. However, the knowledge-based method is weak in geometric modelling, as it still relies on polygon fitting, which is difficult to be used in complex structural buildings and can result in gaps existing among the facets; to obtain a watertight model, a gaps filling algorithm or external knowledge is needed. This problem does not exist in our method.

As described above, each type of method has different advantages. However, each of them has different limitations; thus, it is necessary to find a way to integrate their advantages and avoid the limitations. In this paper, we present a semantic modelling framework-based method for building reconstruction, which combines the advantages from the data-driven method, model-driven method, and knowledge-based method. Compared to the methods mentioned above, the main contributions of this paper are summarized as follows:

- A semantic modelling framework for building reconstruction is presented. The framework simplifies the reconstruction process. In the framework, several semantic building elements are provided which encapsulates a series of procedural modelling algorithms, using them to describe a building in reconstruction can avoid detecting architectural rules and spatial relationships. The framework can use the semantic information to naturally generate a watertight model without considering the polygon fitting and gaps among surfaces.

- Detailed semantic information extraction. In contrast to the previous work, instead of only detecting the roof and façade, we pay more attention to recognizing details from extracted semantic features, including eaves, parapet walls, floors, and openings on the airborne laser scanning data.
- Automatic recognition, evaluation of parametric building elements and geometric reconstruction. Based on our semantic modelling framework, a workflow for automated reconstruction is established. In the workflow, the main parts, including the semantic feature recognition, parameter evaluation of the building elements and semantic information conversation for geometric reconstruction, are fully automatic.

The remainder of the paper is organized as follows: the proposed semantic modelling framework-based method is explained in Section 2, including the semantic modelling framework, reconstruction workflow, data pre-processing, semantic feature recognition, and geometric reconstruction. The proposed method has been tested with three airborne laser datasets containing buildings. The experimental results are shown in Section 3. Some discussions about the proposed method are given in Section 4. Lastly, Section 5 gives the conclusions extracted from the work.

2. Methodology

This section presents the methodology which relies on the semantic modelling framework and reconstruction workflow. Section 2.1 briefly introduces our semantic modelling framework. Section 2.2 gives an overview of our reconstruction workflow. Based on the workflow, the main procedures and algorithms are described in Sections 2.3–2.5. Section 2.3 describes the pre-processing of the point cloud. Section 2.4 explains the semantic features recognition, including recognition of roof, façade, eaves, parapet wall, and windows. Section 2.5 explains how to integrate the semantic information into our semantic modelling framework for geometric modelling.

2.1. Semantic Modelling Framework

In the building reconstruction field, a large number of related studies demonstrate that the hierarchical description of a building is an effective approach, which is closer to the actual architectural structure and more suitable for parametric control and other applications, such as the BIM and GIS fields. For the hierarchical description, compared to the traditional grammar-based approach, XML is more suitable, and has a better readability. There are a number of existing techniques using XML to describe buildings, such as City Geographic Markup Language (citygml) [27], Industry Foundation Classes XML (IFCXML) [28], and Green Building XML (gbXML) [29]. Among them, citygml and gbXML are suitable for the data exchange because they record underlying geometry information, such as the vertex coordinates of a line or polygon, which make the citygml or gbXML file very large and make it difficult to maintain when describing a complex building. As a standard for data exchange in architectural engineering, the IFC covers the entire life cycle of the architectural project. However, the system and architecture of the IFC are too large and complex to control; hence, it is a heavyweight framework that is targeted at construction rather than reconstruction, as it is inconvenient for converting the semantic information into their framework.

Thus, we believe a lightweight, no underlying geometry information, such as the vertex coordinates for polygon, and extensible framework is necessary for the semantic reconstruction. To simplify the semantic expression and building descriptions, we presented a lightweight semantic modelling framework, which included two parts: the extensible building modeling language (XBML) for building description and building element class library that is responsible for geometric implementation in the 3D modelling engine. Differing from other analogous studies, this framework is inspired by the extensible application markup language (XAML) technology in WPF, which is used for user interface (UI) design [30–33]. Compared with other studies, our semantic modelling framework places emphasis on using the semantic information for direct modelling, similar to programming the building, instead of only using it for the data exchange.

2.1.1. Building Description Language-XBML

We can refer to the design pattern of XAML to develop a building description language-extensible building modeling language (XBML) and then predefine building element classes based on a 3D modelling platform with a supporting script language, e.g., MAXScript or Python. The algorithms of the geometry construction are encapsulated in the class (Figure 1).

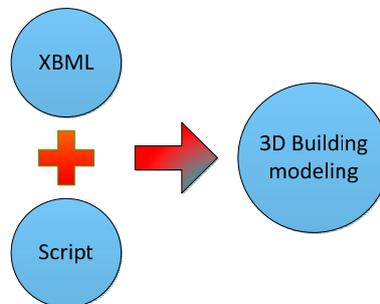


Figure 1. Pattern of the semantic modelling framework.

Compared to the CGA shape or split grammar, XBML has three advantages. First, XBML is more object-oriented, which makes it more convenient to describe the building. Secondly, it has an extensible building element class library, in which all of the classes are able to be inherited, and it is useful for the extension of elements, such as a roof and sloped roof, and window and sliding window. Lastly, the building description is separated from the geometric implementation; thus, users can focus only on describing the building semantically without considering the geometric rules description.

Compared to citygml or gbXML, the advantage of XBML is that there is no recording of underlying geometric information, such as the vertex coordinates of a line or polygon, because our XBML is 3D modelling engine-based. XBML places emphasis on parametric descriptions of building elements. Hence, XBML is more compact than citygml and gbXML, and it is more suitable for describing complex buildings.

In fact, the process of building tagging is a process of semantically describing the building; the semantic segmentation accuracy depends on the fineness of the building division. Currently, there are two primary split strategies (Figure 2):

- (1) Façade first split strategy: this strategy emphasizes the rules of each façade and splits the façade into tiles. Then, the tiles are subdivided according to shape, such as window or door. This strategy is used by the split grammar and the CGA shape grammar.
- (2) Floor first split strategy: this strategy is based on the building structure. It conforms to the process of building construction in reality, and the façades of each floor are then split. Compared with the former strategy, the building can be described more reasonably.

Considering the compatibility of the GIS or BIM area, we used the floor first split strategy in our framework.

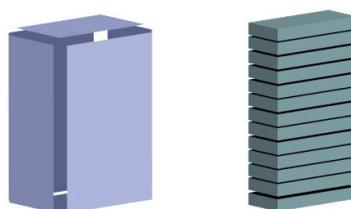


Figure 2. Split strategies: façade first split strategy and floor first split strategy.

2.1.2. Building Element Class Library

We predefined a few base element classes (Figure 3a). In the classes, we can consider the element as a container class; their relationships can be observed in Figure 3b. Remarkably, the façade class is a container for the wall class and its derived classes, and the wall and its derived classes (GridWall or StackWall) are layout management classes, which were designed for managing the layout of elements, such as windows or doors, on the façade. Figure 3b depict the relationship between the elements.

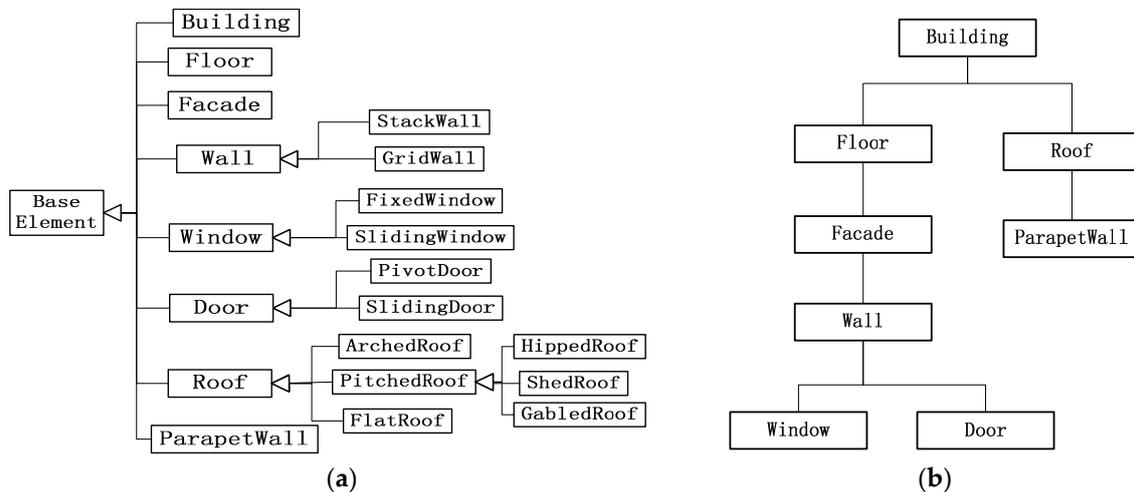


Figure 3. Building base element class and an example of building description: (a) building elements class in the framework; and (b) relationship between the building element classes.

The following example shows a simple building described by the XBML code (Figure 4).

```

1: <Building height="7.28" name="building1" type="#shape" points="135.117,...">
2:   <Floor height="2.65">
3:     <Facade maxpt="135.108,73.0606,114.67" minpt="92.6857,34.7895,112.02">
4:       <Stackwall length="*" repetition="15" margin="2.0">
5:         <window depth="0.3" height="1.5" pos="1.8,2.0,0" width="1.68357"/>
6:       </Stackwall>
7:     </Facade>
8:   </Floor>
9:   <Floor height="4.53">
10:    <Facade maxpt="135.108,73.0606,119.2" minpt="92.6857,34.7895,114.67">
11:      <Stackwall length="*" repetition="15" margin="2.0">
12:        <window depth="0.3" height="1.5" pos="1.8,2.0,0" width="1.68357"/>
13:      </Stackwall>
14:    </Facade>
15:   </Floor>
16:   <PitchedRoof cycles="2,3,1 2,1,4" planes="0.08453,0.09363,0.992,..."/>
17: </Building>

```

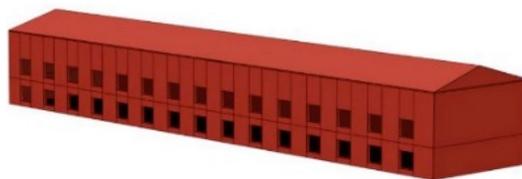


Figure 4. A two-storied building described by XBML code.

2.2. Workflow of Reconstruction

As illustrated in Figure 5, in the workflow, using the point cloud pre-processing, the ground points are first eliminated. Then, to preserve boundary information on building, we used the region-growing

algorithm to segment the point cloud. Secondly, based on the results, we used the knowledge on size, position, and orientation to recognize the semantic features, such as the roof and façade. Furthermore, other related building elements (doors, windows, and walls) and their parameters were evaluated with the corresponding algorithm. Then, we designed a converter for converting the semantic information and corresponding parameters into the XBML code. Lastly, we developed a parser for parsing the XBML code into the 3D modelling engine to generate the 3D model.

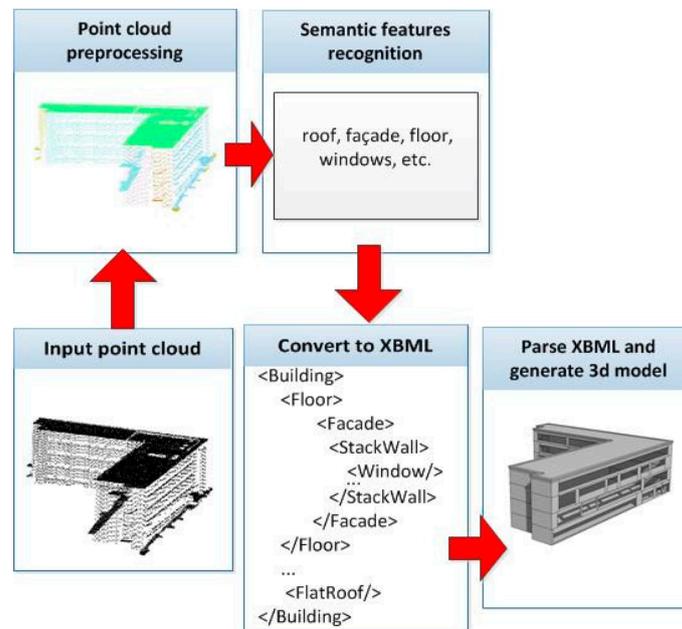


Figure 5. Workflow of the proposed semantic reconstruction.

2.3. Point Cloud Pre-Processing

One of the most important geometric features of a building is the large number of planar features. The different geometric features and implied information, such as curvature and normal, are the foundation for recognizing semantic building features. To facilitate the semantic feature recognition, we use the RANSAC (RANDOM SAMPLE CONSENSUS) [34] algorithm to filter out the ground point cloud because the RANSAC algorithm is insensitive to point noise, and then the region-growing algorithm is selected to roughly separate the roof point cloud from the façade point cloud (Figure 6), while preserving the boundary information.

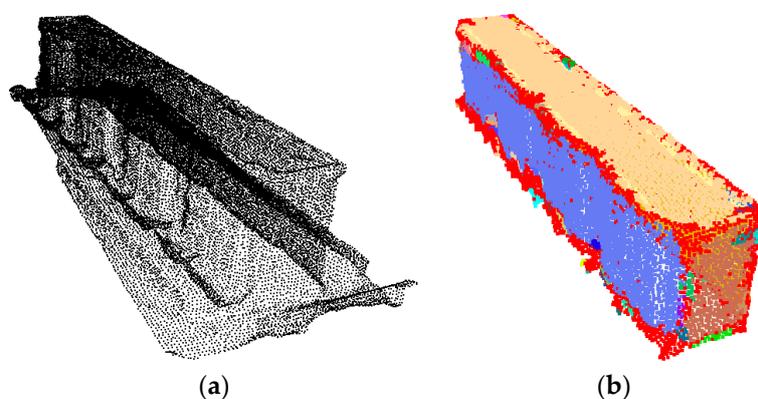


Figure 6. Results of pre-processing and initial segmentation: (a) original point cloud of a building; and (b) point cloud after filtering the ground points and initial segmentation.

2.4. Semantic Feature Recognition

The recognition of semantic features is a process of analyzing the geometric features comprehensively, and each semantic feature can be considered as a combination of several geometric features. In the conventional method, only the semantic features were recognized with prior knowledge. In our method, apart from the semantic features, the attributes of the semantic features will be evaluated further.

In this section, the semantic features that will be recognized consist of a building, floor, façade, roof, door, and window. For certain features, such as the roof, the specific type (flat roof or sloped roof) will be further recognized. There are two steps in recognition: recognizing the semantic features from the segmentation and evaluating the attribute value based on the extracted geometric feature.

2.4.1. Roof and Façade Recognition

We can use the approach from [22] to build the knowledge (orientation, size, position) and then use the knowledge to recognize the semantic features. Three types of geometric information are used for determining the roof and façade:

- Size, which is the number of points in the segments. When the difference of the point density between roof and facade is not too much, the size information can be used to recognize the roof and façade from other segmentations;
- Orientation, which is the angle between the normal of the plane and the vertical direction. Generally, flat roof, sloped roof, and façade have different orientations;
- Position, which is the height of the lowest point above the ground in the point cloud segmentation, although most façades and roofs can be distinguished using size and orientation, in some special cases, such as the hip roof of church, position information is used; generally, the roof is always on the top of the facade (Table 1).

Table 1. Geometric features of flat roof, sloped roof, and façade.

	Flat Roof	Sloped Roof	Façade
#Size	A large number of points	A large number of points	A large number of points
Orientation (degree)	<10	>10 and <80	>80
Position (m)	>5 or 10 m higher than the ground or above the façade	>5 or 10 m higher than the ground or above the façade	<5

Using the defined knowledge, the roof and façade are distinguished from each other, and the roof type, i.e., sloped roof or flat roof, is also recognized. After recognizing the roof and façade, because there are several parameters that need to be calculated, a further segmentation based on the current results will be performed. Normally, the area of a house is greater than 30 m², which was obtained from our survey to the buildings; thus, the size should be greater than 30 × D at least, where D is point cloud density.

2.4.2. Outline Extraction of the Roof

An outline is the most basic feature for describing the building, and most modelers select the outline to build models because this way they can avoid the problems caused by occlusion or incomplete scanning. Generally, to obtain an accurate outline, a roof point cloud is often used for its extraction. However, the shape of the roof varies. To obtain a horizontal outline, the roof points are projected onto the XOY plane, and the α -shape algorithm [35] is then applied to obtain the sorted boundary points. Essentially, the roof corner needs to be detected. Figure 7 illustrates the process of outline extraction.

In the corner detection, the traditional methods are sensitive to noise. To overcome this, an improved adaptive corner detection algorithm is presented in this paper. We first use Gaussian smoothing to reduce the noise in the boundary points. After the Gaussian smoothing, an adaptive method proposed in [36] is used to determine the supporting region and calculate the curvature angle of each point, and then the corners are detected based on the curvature angle.

To smooth the boundary curve, we express the curve with two functions $x(t)$ and $y(t)$:

$$L = \{x(t), y(t)\} \quad t \in (0, S) \tag{1}$$

where t is the path length variable along the curve L , $x(t)$ and $y(t)$ are the x -coordinate and y -coordinate of the curve, respectively, and S is the length of the curve L .

Let $G(t, \sigma)$ be the Gaussian function:

$$G(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} \tag{2}$$

The curve smoothed by the Gaussian function:

$$X(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} x(v) e^{-\frac{(t-v)^2}{2\sigma^2}} dv \tag{3}$$

$$Y(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} y(v) e^{-\frac{(t-v)^2}{2\sigma^2}} dv \tag{4}$$

where $\sigma = 1$ is used in our paper.

In the determination of region of support (ROS), we define the length of chord p_{i-k} and p_{i+k} as follow:

$$L_{ik} = |\overline{p_{i-k}p_{i+k}}| \tag{5}$$

Let D_{ik} be the distance from p_i to chord $p_{i-k}p_{i+k}$, then start computing the L_{ik} and D_{ik} with $k = 1$ until satisfying the Equation (6) or Equation (7):

$$L_{ik} \geq L_{i,k+1} \tag{6}$$

$$\begin{cases} \frac{D_{ik}}{L_{ik}} \geq \frac{D_{i,k+1}}{L_{i,k+1}} & D_{ik} > 0 \\ \frac{D_{ik}}{L_{ik}} \leq \frac{D_{i,k+1}}{L_{i,k+1}} & D_{ik} < 0 \end{cases} \tag{7}$$

After finishing the determination of supporting region, we use k -cosine method to calculate the curvature of each point on the curve. Define vectors at p_i as follows:

$$\vec{a}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k}) \tag{8}$$

$$\vec{b}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k}) \tag{9}$$

and k -cosine at p_i as:

$$\cos_{ik} = \frac{\vec{a}_{ik} \cdot \vec{b}_{ik}}{|\vec{a}_{ik}| |\vec{b}_{ik}|} \tag{10}$$

$$\theta_{ik} = \cos^{-1} \frac{\vec{a}_{ik} \cdot \vec{b}_{ik}}{|\vec{a}_{ik}| |\vec{b}_{ik}|} \tag{11}$$

The angle θ_{ik} between vector \vec{a}_{ik} and \vec{b}_{ik} is proportional to the curvature, hence we define that the point is a corner if the $|\theta_{ik}| > T$, T is the angle threshold.

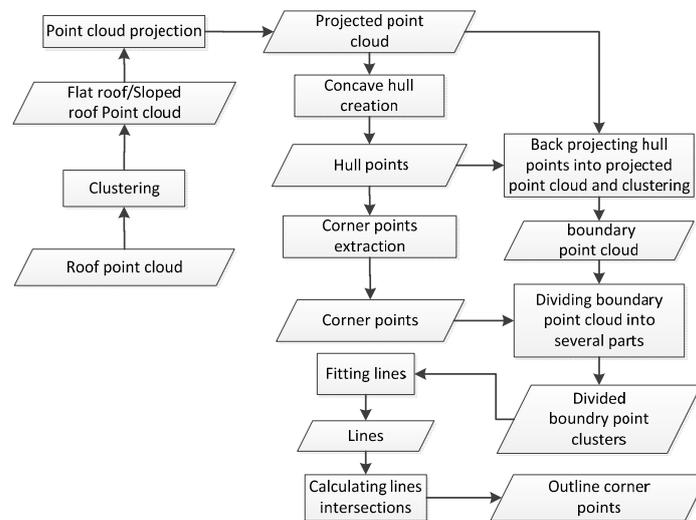


Figure 7. Process of outline extraction.

The outline extraction algorithm can be described as follows:

- (1) We can project the roof points onto a fitting surface that is parallel to the XOY plane.
- (2) To obtain the sorted boundary points, we can use the α -shape algorithm to create a concave hull, and the alpha value is 0.8 (Figure 8a). Generally, to preserve the sharp features on the roof, the alpha value is set between one time and two times the average distance of points.
- (3) Corner detection. We use our improved algorithm mentioned above to detect the corners. An angle threshold $T = 35^\circ \times 180/\pi$ is given to detect the corners. After detection, the set of boundary points is divided into several subsets using the corners (Figure 8b).
- (4) Boundary accurate extraction. Since the set of boundary points is too sparse, it is insufficient to accurately fit an edge; thus, we can back-project each subset of points into the original point cloud and obtain the k nearest neighboring points, and use these new subsets of points to fit the line. Lastly, we can obtain a set of ordered lines $L = (l_1, l_2, l_3, \dots, l_4)$ (Figure 8c).
- (5) Accurate corner extraction. Each corner is obtained by calculating the intersection of adjacent lines in a set of lines L (Figure 8d).

The final result is shown as following (Figure 8).

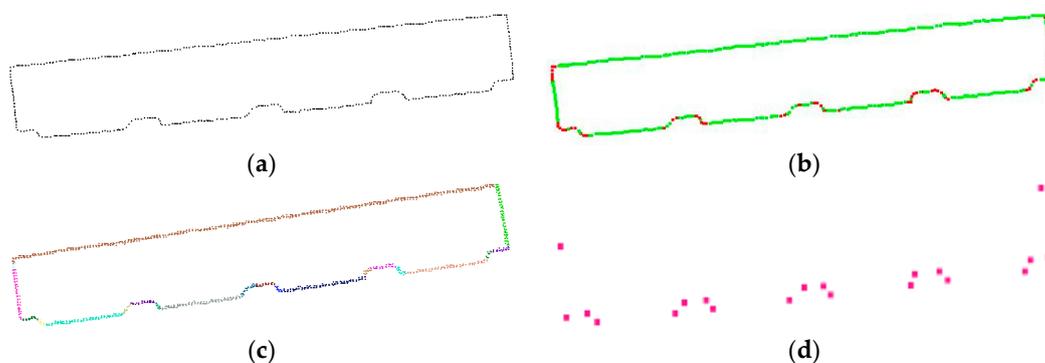


Figure 8. Results of each step: (a) sorted boundary points; (b) result after corner detection (red areas); (c) subsets divided by the corners; and (d) accurate corner points.

2.4.3. Topology Analysis of Pitched Roof

Roof reconstruction is an important part in building reconstruction, for complex types of pitched roofs, the information about roof outline and roof planes are not enough. To reconstruct complex pitched roofs, we use a roof topology graph (RTG)-based algorithm for its construction [2,6,24,37]; thus, the roof planes and the adjacencies among the planes are needed. The roof planes are available after the roof extraction. For the adjacency, we obtained the convex of each roof plane and then determined whether two planes are adjacent or not by checking the number of points within a neighborhood ($r < 0.6$) along the boundaries of the two convex hulls [2].

We define the roof corner into two types: outer corners and inner corners. The roof outer corners are obtained from the outline extracted in previous section, the inner corners is determined by the topology graph analysis. In our RTG-based algorithm, three rules are followed: (1) a cycle ($n \geq 3$) in the RTG represents an inner roof corner; (2) a pair of adjacent cycles represent a ridge line on roof; and (3) each cycle is the shortest cycle, which guarantee there are no cycles contained in another cycle.

Contrasting to other RTG-based algorithms [2,37], our algorithm simplifies the process of decomposition of the RTG because we abandoned the using of the step edge, and the perpendicular plane is introduced if a ridge line was intersected with the outline, which can guarantee the node number of each cycle is greater than or equal to three. In our method, to guarantee each cycle obtained is the shortest cycle, we use the Dijkstra algorithm to extract cycles. In Figure 9, for instance, the RTG is decomposed into six cycles: $\{2, 1, 11\}$, $\{2, 11, 6, 5, 3\}$, $\{5, 4, 3\}$, $\{6, 11, 7\}$, $\{11, 10, 8\}$, $\{10, 9, 8\}$, the blue lines 7 and 9 denote the perpendicular planes which are invisible in the perspective.

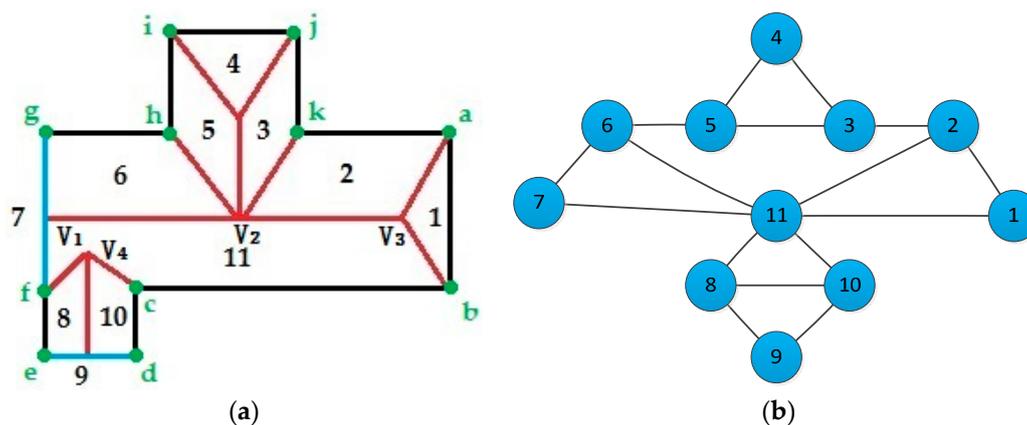


Figure 9. A complex roof and corresponding RTG: (a) a complex roof, green points donate the vertices on outline, blue lines denote the perpendicular planes; and (b) the corresponding RTG.

To obtain each roof facet, we categorize the facets into two types: inner facets and outer facets. The vertices of an inner facet can be determined by the inner corners around the facet, and the vertices' order depends on the adjacency of the cycles, while the vertices of an outer facet are determined by the inner corners and outer corners. To determine the vertices order, the connection between the corresponding outer corners and inner corners for the outer facet is needed.

We determine the connection between outer corners and inner corners by searching the common vertex among the linked facets. For instance, for facet 11, its inner corners include $V_1 = \{6, 11, 7\}$, $V_2 = \{2, 11, 6, 5, 3\}$, $V_3 = \{2, 1, 11\}$ and $V_4 = \{11, 10, 8\}$, its outer corners include (b, c, f), based on the adjacency of the cycles (V_1, V_2, V_3), their connections can be obtained: V_1 is connected with V_2 , V_2 is connected with V_1 and V_3 , V_3 is connected with V_2 . The outer corners of the linked facets, including facet 7, facet 8, facet 10 and facet 1, are (g, f), (f, e), (d, c) and (c, b). The common vertices between the facet 11 and them are f, f, c, b, respectively, that means, the vertex $V_1 = \{6, 11, 7\}$ is connected with outer corner f, vertex $V_4 = \{11, 10, 8\}$ is connected with corner f and c, vertex $V_3 = \{2, 1, 11\}$ is connected with corner b (Table 2).

For the vertices order of the facet, all the vertices can be connected in sequence by searching the connection vertex on the vertex connection list (Table 2). The searching start with arbitrary vertex (V_1) and next vertex should be one of its connection vertices which have not been traversed. The searching will be stopped when all of the connection vertices of a vertex (f) have been traversed already. As a result, the vertices of facet 11 are $\{V_1, V_2, V_3, b, c, V_4, f\}$.

Table 2. Vertex connection list of facet 11.

Vertices	V_1	V_2	V_3	b	c	V_4	f
Connection vertices	f	V_1	V_2	V_3	b	c	V_4
	V_2	V_3	b	c	V_4	f	V_1

2.4.4. Eaves and Parapet Wall Recognition

There are several detailed structures contained in the building main structure, e.g., eaves, which overhang the façade horizontally, and parapet walls, which extend from the roof vertically along the roof boundary. Both of these detailed structures need to be recognized, and their attributes, such as the overhang distance, the thickness of the eaves or parapets, topology of the pitched roof surface, height, and thickness of the parapet wall, also need to be evaluated because they are indispensable parts of the building description.

As important components, the eaves and parapet wall have clear geometric features, which are introduced into the knowledge library for recognition (Table 3).

Table 3. Geometric features of the eaves and parapet wall.

	Eave	Parapet Wall
Position	Near roof boundary	Near roof boundary
Horizontal distance (m)	>0.2	~
Vertical distance (m)	~	>0.3
#Size	Within certain range	Within certain range

In Table 3, position denotes whether the horizontal distance from the eave and parapet wall to the roof boundary points is less than a certain threshold (0.5), which is normally the mean distance between points, and we should detect them along the roof boundary points. Horizontal distance and vertical distance denote the overhang distance and height of the parapet wall, respectively. Size denotes the area of the eaves or parapet wall, and we can use the number of points to represent their size; furthermore, a threshold of size is provided for distinguishing them from noise. Through the survey of many buildings, we found the length of most of the eaves is greater than 2 m at least, the horizontal distance is between 0.2 m and 1.2 m. The thickness of the parapet wall is greater than 0.12 m, the length is greater than 2 m, and the height (vertical distance) is between 0.3 m and 1.2 m. Hence, the threshold of size should be between $2 \times 0.12 \times D$ and $2 \times 0.2 \times D$, D is the point cloud density.

After recognition through the above knowledge, the points are extracted for their attribute value estimation. We referenced most of the building types and established the following method for estimating these attributes.

- (1) Overhang distance of the eaves: we used the horizontal distance between the edge in the roof outline and the corresponding façade plane to evaluate this distance. However, the edge of the outline is not completely parallel to the corresponding façade plane; hence, to make the estimation as accurate as possible, we used the longest edge from the outline and several façade points that were selected randomly to calculate the average horizontal distance from the point to the line (Figure 10a).

- (2) Thickness of the eaves: if the eaves is recognized, the thickness should be the vertical distance from its lowest point to the fitting plane of the flat roof (Figure 10b).
- (3) Height of the parapet wall: the vertical distance from the highest point to the fitting plane (Figure 10b).
- (4) Thickness of the parapet wall: the horizontal distance from the outside to the inside of the parapet wall (Figure 10b).

Remarkably, after the overhang distance is obtained, the outline of the roof should be zoomed out by the same distance, which is the outline of the building's body.

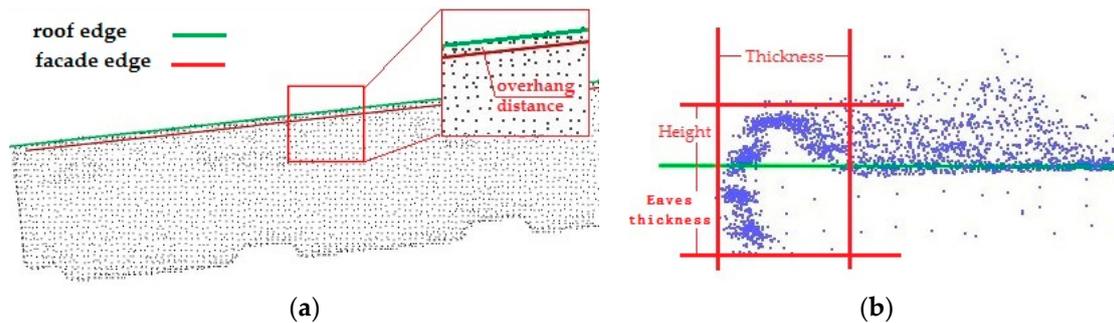


Figure 10. The height, thickness of the parapet wall, and distance of overhang: (a) the overhang distance, façade edge is intersection between roof and façade plane; and (b) the thickness and height of the parapet wall and the thickness of the eaves.

2.4.5. Floor Analysis and Extraction

The floor is a basic structural unit of the building in 3D space, however, in conventional methods, the information of floor was always ignored. To obtain the information, we designed a confidence-based floor splitting algorithm for floor analysis and extraction. Compared to other related works [15,38], our algorithm has a lower computational complexity.

The analysis on the floor is the foundation for the hierarchical description of the building. However, compared to the building elements, such as the windows or doors, the geometric features of the floor are not apparent if observed from outside; hence, the floor is primarily analyzed from the layout of the elements on the façade. Currently, a few related studies are available. An approach using imagery to analyse the floor structure was presented in [15]. In this method, mutual information was introduced for similarity detection, and the splitting lines were then computed by the following knowledge: horizontal splitting lines should be located where the vertical edges are rare and the horizontal edges are dense, and the vertical splitting lines are the opposite case. This method can work well for a multi-story building but it is sensitive to image noise and irregular elements. A new floor structure splitting algorithm based on the terrestrial point cloud was presented in [38], which incorporated the above knowledge into their algorithm, and the confidence of the boundary point and the accumulated effects of the vertical or horizontal lines were evaluated for a certain splitting plane. This method is more robust and more adaptive, and it also can be applied to image data.

In our experiment, airborne laser scanning data of the façade is selected for floor analysis. Compared to the terrestrial point cloud, the density of the airborne point cloud is lower, especially for the façade point cloud, in which the point distribution is not uniform, i.e., the upper part is dense and lower part is sparse. Based on these differences with the terrestrial point cloud, we designed a splitting algorithm for the airborne data, which was still based on the prior knowledge: (1) points are rare on windows or doors and dense on walls; thus, the splitting lines should be located where the points are dense; and (2) as the distance to the splitting lines decreases, there is less likelihood that a point lies along a boundary of the window.

To evaluate the likelihood that a point lies along a boundary, we can introduce the confidence value and accumulated effect value from [38], but in [38], they used a covariance matrix to calculate the confidence value, which has a high computation complexity, to lower the computation complexity, in our algorithm, we use the method from [39] to evaluate the confidence value. In this method, four categories for the boundary point of window can be distinguished, including upper, lower, left, and right edge points. Take a query point for example: if no further facade point is found under the query point inside a given buffer, then the point is considered as an upper edge point. Using an upper edge point as an example, the confidence value can be computed as follows:

$$r_i = \frac{n_2}{n_1 + n_2} \quad (12)$$

$$\text{conf}(p_i) = 1 - \frac{|r_i - 0.5|}{0.5} \quad (13)$$

where n_1 denotes the number of points above the query point i ; n_2 denotes the number of points below point i ; and r_i denotes the ratio of n_1 to the total amount of points. As seen, the confidence value $\text{conf}(p_i)$ ranges from 0 to 1: $\text{conf}(p_i) = 1$ indicates that point i is a facade point, and $\text{conf}(p_i) = 0$ indicates that point i is an upper edge point.

The confidence value is considered as the weight of the point, and based on our prior knowledge, when the point is closer to the splitting line, the weight is closer to 1 and the points are denser; thus, the accumulated effect should be more notable. The accumulated effect value can be calculated as follows:

$$C_{hor}(P) = \sum_{i \in B} \text{conf}(p_i) \quad (14)$$

where B is a given buffer. Figure 11a depicts an analysis result while Figure 11b illustrates the final result.

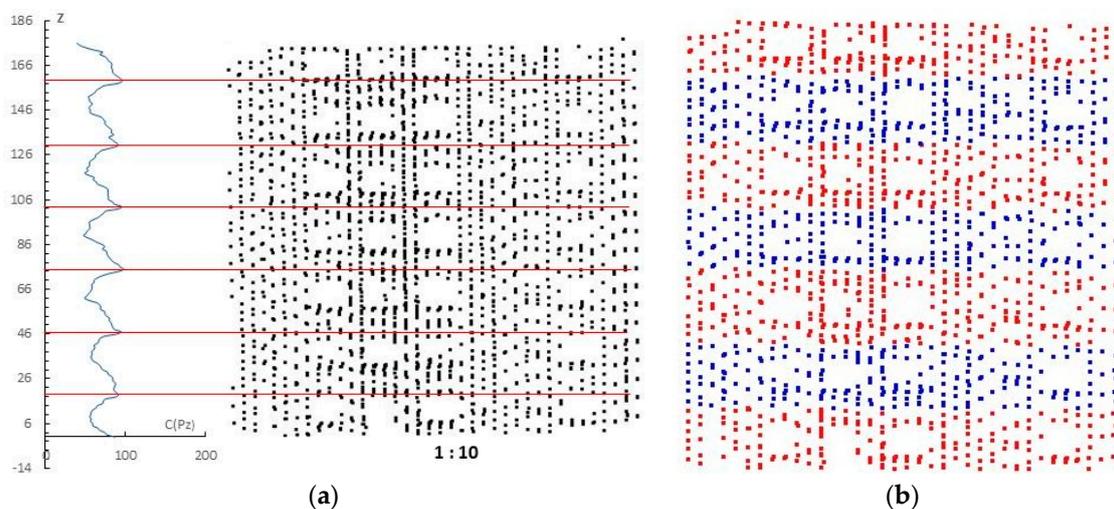


Figure 11. Floor analysis based on the airborne point cloud: (a) analysis result of horizontal splitting; and (b) final result after splitting.

2.4.6. Windows and Doors Extraction

After the floor analysis, the facade is split into several parts. In this section, the window and door components will be recognized for each part.

There have been some algorithms [40,41] for detecting the doors and windows on sparse point clouds, which mainly focus on the detection of holes caused by the windows or doors. In our method,

we designed a confidence-based algorithm for the airborne laser scanning data to identify the windows and doors and measure their size.

In the last section, we evaluated the likelihood that a point lies along a boundary of a window, that is to say, we can extract the boundary point base on the likelihood. In our algorithm, we first consider the likelihood r_i calculated by the Equation (12) as a confidence for the edge point, and use confidence to determine the edge point and label them as the upper, lower, left, or right edge points. On the horizontal direction, for instance, if the likelihood of a point greater than the threshold a ($0.5 + \partial \leq a \leq 1.0$), the point will be classified into right edge point; if the likelihood less than b ($0 \geq b \geq 0.5 - \partial$), then the point will be classified into left edge point. As shown in Figure 12, the red point donates the candidate of the left edge point or lower edge point, and the green point donates the candidate of the right edge point or upper edge point. The façade is split into pieces by the left edges and right edges, only the pieces with left edge and right edge, and left edge is on the left of right edge are the candidates for the detection of window or door. As shown in the Figure 13, the red line donates the left edge, and the green line donates the right edge. In the splitting pieces, only blocks A, B, C, and D satisfy the condition to be candidates for window or door detection. Then, the similar method is used to detect the upper and lower edge of the window in the candidate blocks.

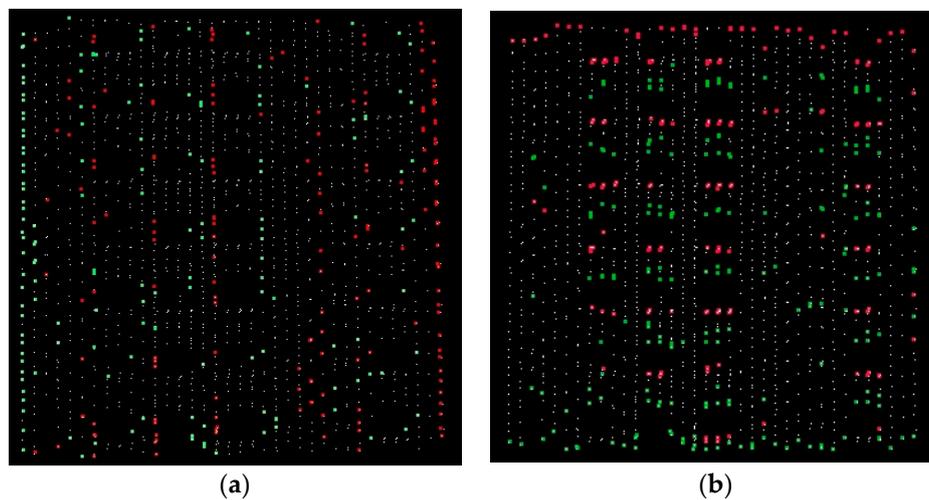


Figure 12. The window's edges detection: (a) the left and right edges detection; and (b) the upper and lower edges detection.



Figure 13. The process of window detection, (A–D) are the candidates for window or door detection.

The façade is then split into pieces by the left edges and right edges; only the pieces with left edge and right edge, and left edge is on the left of the right edge are the candidates for the detection of windows or doors. As shown in the Figure 13, the red line donates the left edge, and the green line donates the right edge. In the splitting pieces, only blocks A, B, C, and D satisfy the condition to be

candidates for window or door detection. Then, the similar method is used to detect the upper and lower edge of the window in the candidate blocks.

2.5. Semantic Modelling Framework-Based Geometric Reconstruction

In our workflow, after the semantic feature recognition, the extracted semantic information needs to be converted into the XBML code, and the XBML code is then parsed. In this section, we will describe the conversion and parsing of the semantic information.

2.5.1. Formatted Output of Semantic Information

Currently, many related studies only focus on the semantic information extraction without any relevance; hence, the extracted semantic information is unorganized. Only if the semantic information is organized hierarchically, then the information can be used for accurately describing the building and 3D reconstruction, which is our original intention of developing the semantic modelling framework.

To convert the semantic information into the XBML format, we need to first make a formatted output of the semantic information based on the hierarchical relationships among building elements. And the relationship can be recovered from the formatted output. The types and attributes or parameters of the elements are outputted as text files, and all of the information for the same category elements are saved in one file, such as building.txt, floor.txt, façade.txt, window.txt, and roof.txt. To record the information of the hierarchical relationship between the elements, we used the method inspired by the relational database management system (RDBMS). For each node recorded in the file, its file index is written at the end of its child node file.

Based on the formatted output, using recorded file index, the parent node can be searched. Then, according to the hierarchical relationship, we attempted to translate all of the semantic information into the XBML format.

2.5.2. Semantic Interpretation of XBML

A few characteristics of our semantic modelling framework are that the building description is separated from the geometric implementation, geometric implementation is encapsulated in the building element classes, which are defined on the 3D modelling platform, and XBML is primarily responsible for the description of the building. In this paper, the algorithm of the geometric construction is implemented based on the 3DS Max from Autodesk Inc., which provided a built-in scripting language for development, such as MAXScript.

Clearly, to model the building in a 3D modelling platform, such as 3DS Max, the XBML code needs to be parsed into the corresponding building element classes that was coded by the built-in script of the platform. In our design, the tags and attributes of XBML correspond to the objects and properties of the predefined classes in our framework. Therefore, we specifically designed an XBML parser (Figure 14).

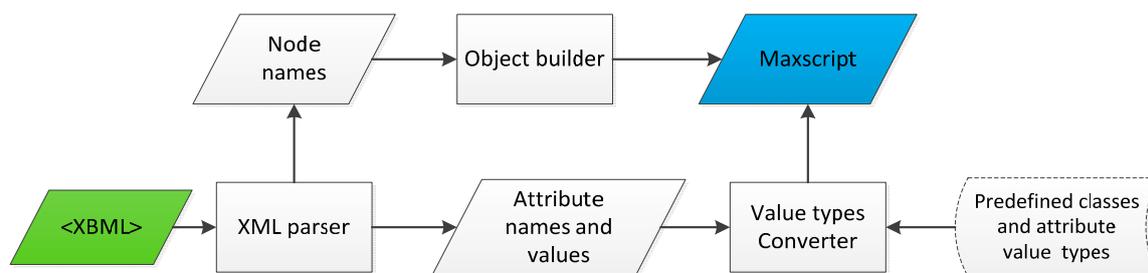


Figure 14. Flowchart of the XBML parser.

Since XBML is an XML-based markup language, we can use the XML parser to obtain the attribute values and tags, and convert the value type into the type supported in the built-in script using the developed value type converter. Lastly, based on the hierarchical relationship, all of the information is integrated together to form the built-in script.

3. Results

To validate our methodology and reconstruction workflow, three existing buildings were reconstructed. The three different buildings are different kinds of buildings; one is a building with a dual-pitched roof, one is a multi-storied building with a four-pitched roof, and the last one is a building with a flat roof. The datasets of the three building were scanned using an airborne laser scanner, and the average point density was 10 points per square meter.

In our reconstruction system, the algorithms used for recognizing the semantic features from the point cloud were developed in C/C++ on Windows, the semantic modelling framework was developed in MAXScript on the 3DS MAX 2015 platform of Autodesk Inc., and the XBML converter and parser were developed in Python.

The results of the three buildings are shown in Figure 15. Based on the knowledge we built, the semantic information and corresponding attributes value are extracted automatically, including the vertices of the outline, height of the floor, size of the openings, and topology of the pitched roof.

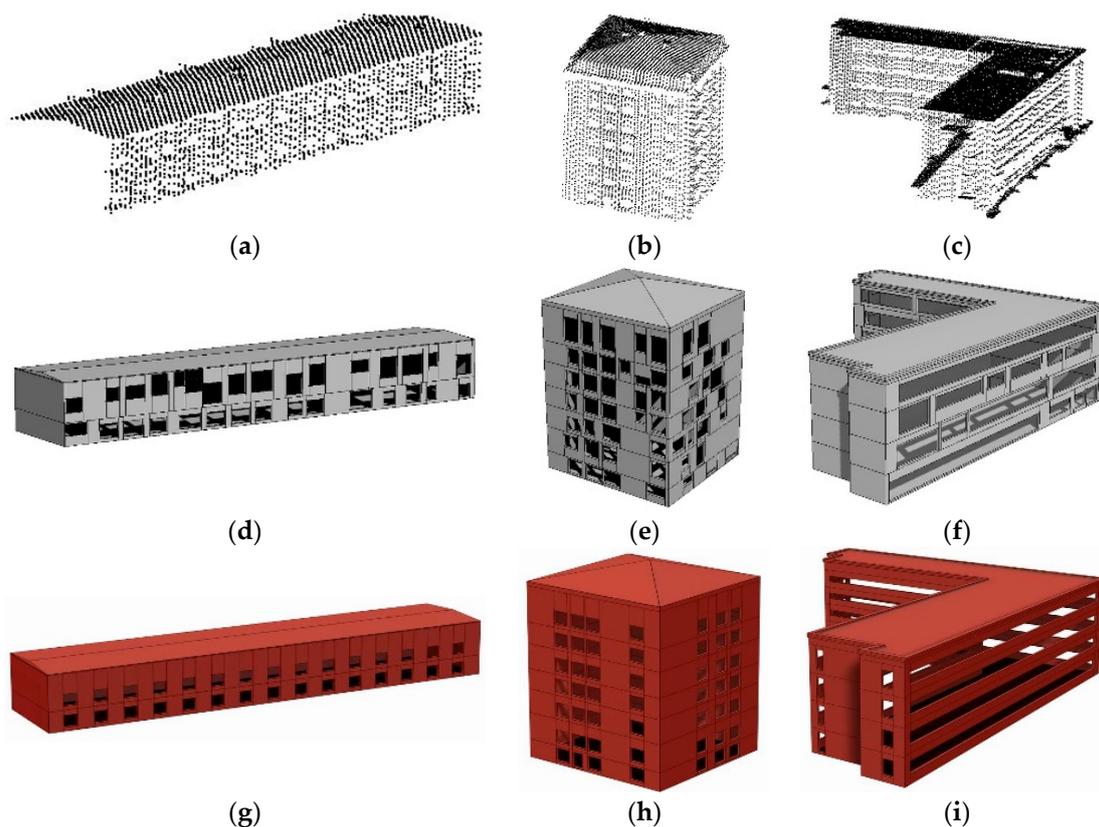


Figure 15. Reconstruction result: (a–c) three selected datasets; (d–f) reconstruction results of the three buildings; and (g–i) rReconstruction models after editing their XBML code.

As shown in Figure 15a,d,g the first building was a two-story building with a dual-pitched roof, and only one facade was scanned in the dataset. Figure 15a shows the origin dataset of the first building, Figure 15d shows the building with the roof extracted from the roof points and floors and windows recognized from the points of the façade. Figure 15g shows the final reconstruction result after editing the XBML code.

As shown in Figure 15b,e,h. The second building was a multi-storied building with a four-pitched roof, in which two façades were scanned. The one with the higher density was selected for the floor analysis (Figure 11), and the outline information was extracted from the roof point cloud. In Figure 15e, it is apparent that the details of the roof are slightly different from the original shape in the scanned data, four small faces on the roof were not recognized correctly, which was caused by the noise that influences the accuracy of the segmentation algorithm. In the window recognition process (Figure 15d,e), most of the windows were recognized correctly; however, on certain façades, the recognition was not precise, which was primarily due to the low density of the façade point clouds. Figure 15h is the reconstruction results of the second building after editing its XBML code.

As shown in Figure 15c,f,i. The third building was a flat roof building with four floors, and the building was located in an area with a slight slope. Thus, a few of the points on the ground were not filtered out completely; however, it did not affect the extraction of the semantic information. As shown in Figure 15f, the outline is more complicated than others, but it was still extracted correctly. Additionally, the overhang distance of the eaves and height of the parapet wall were evaluated using our method. However, the thickness of the parapet wall was not calculated correctly, and a default value of 0.25 was used (Figure 15f). Figure 15i shows the reconstruction results after editing the XBML code.

4. Discussion

4.1. Quality Evaluation

To evaluate the quality of reconstruction, we visualized the wireframe model and point cloud together to assess the quality (Figure 16). As seen, the wireframes coincide well with the point cloud, especially the façade, which is because the overhang distance was considered in our workflow and semantic framework.

To evaluate the completeness and geometric accuracy for the above building reconstruction results, two primary components, roof and façade, were selected for evaluation. For geometric accuracy, the distance from the roof plane and façade plane to the laser points were calculated for estimating the root mean square error (RMSE), and the points were selected manually (Tables 4 and 5).

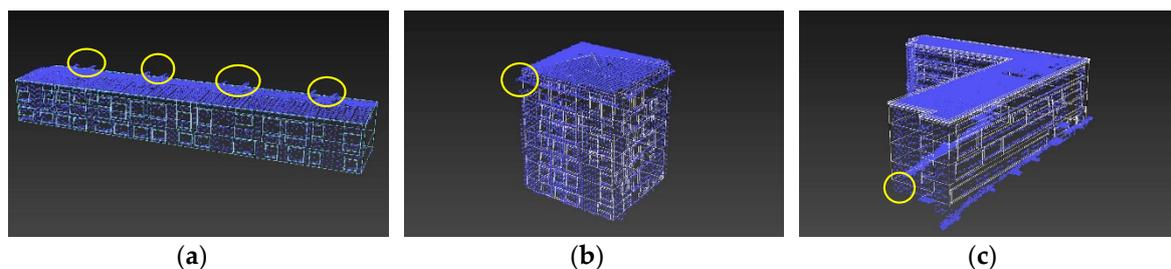


Figure 16. Model wireframes and the corresponding point clouds: (a) a two-storied building model with double pitched roof; (b) a seven-storied building with hip roof; and (c) a four-storied building with flat roof.

Table 4. Completeness and accuracy of roof segmentation.

#Building	#Segmented Roof Facets	#Roof Facets in Data	Completeness (%)	RMSE (m)
1	2	2	100	0.111
2	4	6	67.7	0.079
3	1	1	100	0.07

Table 5. Completeness and accuracy of facade segmentation.

#Building	#Segmented Façade Facets	#Façade Facets in Data	Completeness (%)	RMSE (m)
1	1	1	100	0.368
2	2	2	100	0.361
3	7	7	100	0.147

In Table 4, it should be noted that the completeness of building 2 is 67.7%, which is caused by the incorrect recognition in the roof, as two facets are not segmented.

The corresponding normal distribution graphs of distance from the roof plane and façade plane to the laser points for each building are shown as follows.

Figure 17 indicates that for the roof, the distance between the roof plane, and the majority of points are less than 0.2 m, i.e., the roof fitting plane coincides with the points well. For the façade, the RMSE value is slightly larger than that of the roof, and the distance is maintained between 0.1 m and 0.5 m, which primarily occurs because the windows and windowsills increase the distance from the points to the fitting plane of the façade; hence, the result is still acceptable.

To evaluate the semantic recognition, precision, recall, and accuracy are used to evaluate our recognition algorithm. They are calculated as follows [42]:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (15)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (16)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

where TP = true positives, TN = true negatives, FP = false positives, FN = false negatives.

In case study 1 (Table 6), most of the building elements, except one window, were recognized correctly.

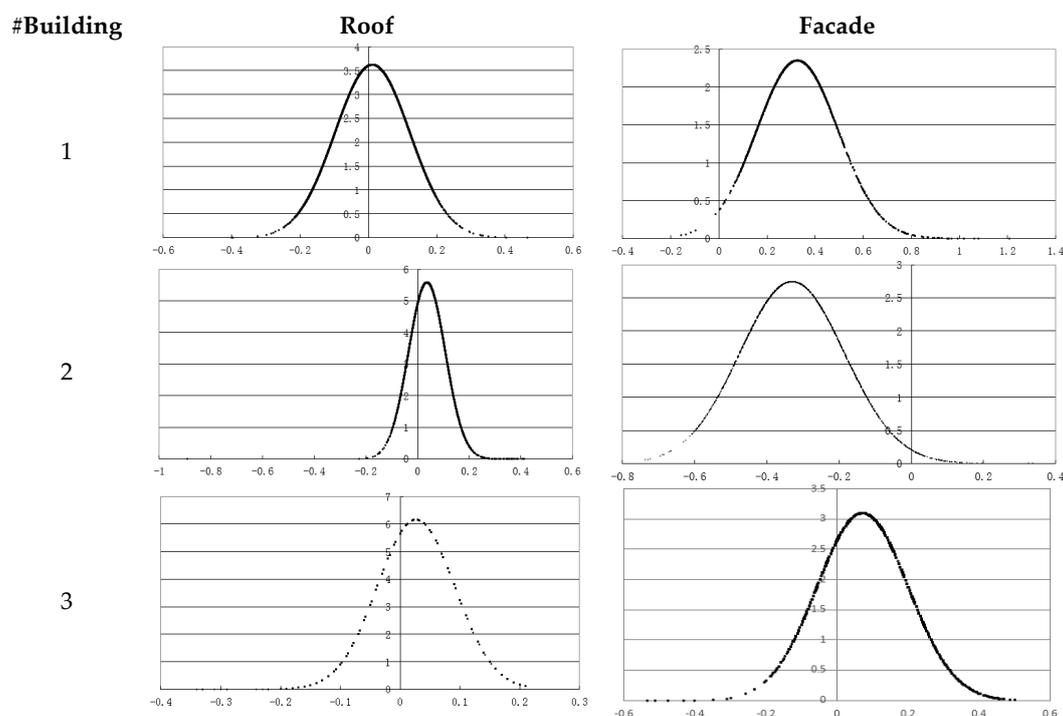
**Figure 17.** Normal distribution of distance from roof plane and façade plane to laser points.

Table 6. Evaluation of semantic recognition for case study 1.

	TP	FP	FN	TN	Precision (%)	Recall (%)	Accuracy (%)
Floor	2	0	0	33	100	100	100
Window	29	1	0	5	96.7	100	97.1
Roof	2	0	0	32	100	100	100

In case study 2 (Table 7), nine windows are not recognized correctly; this is caused by the low point density and incompleteness of the second façade. Two roof planes are not recognized because the two roof facets are narrow and the algorithm cannot detect them, which also effects the eaves recognition.

Table 7. Evaluation of semantic recognition for case study 2.

	TP	FP	FN	TN	Precision (%)	Recall (%)	Accuracy (%)
Floor	7	0	0	49	100	100	100
Window	44	9	8	12	83	84.6	76.7
Door	1	0	1	55	100	50	98.3
Roof	4	2	0	52	66.7	100	96.6
Eaves	0	1	0	56	0	0	98.2

In case study 3 (Table 8), the width of the windows in the data is too large, and our window recognition algorithm is not able to recognize this type of window very well, which results in the windows being split into several parts. Additionally, in the three cases, the precision of the door recognition is low; this is also caused by the completeness of the point cloud.

Table 8. Evaluation of semantic recognition for case study 3.

	TP	FP	FN	TN	Precision (%)	Recall (%)	Accuracy (%)
Floor	4	0	0	11	100	100	100
Window	8	0	0	7	100	100	100
Door	0	1	0	15	0	0	93.8
Roof	1	0	0	14	100	100	100
Eaves	1	0	0	14	100	100	100
Parapet wall	1	0	0	14	100	100	100

Tables 4–8 and Figure 17 indicate that the proposed method can maintain the high geometric accuracy of the data-driven method and flexibility of the model-driven method.

To further validate our approach, we select an area with a sufficient number of buildings which contain at least one façade in the dataset for reconstruction. The reconstructed result is shown in Figure 18.

From Tables 9 and 10, it can be seen that the proposed method can still maintain a high geometric accuracy and flexibility when reconstructing a large number of buildings. However, for the semantic recognition, there is room for improvement in recognition of the window and floor.

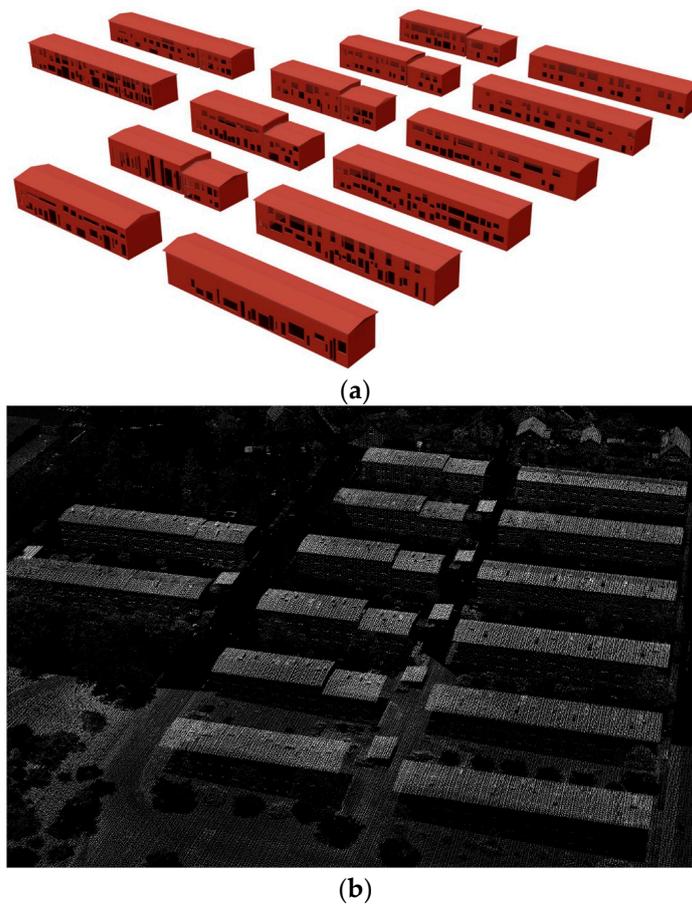


Figure 18. Reconstructed 3D building models: (a) 3D building models; and (b) point cloud with intensity.

The completeness and geometric accuracy for the above reconstruction results can be seen in Table 9:

Table 9. Completeness and accuracy of roof and façade segmentation.

	#Segmented Facets	#Facets in Data	Completeness (%)	RMSE (m)
Roof	2	2	100	0.21
Facade	1	1	100	0.37

The evaluation of semantic recognition of the result is shown in Table 10:

Table 10. Evaluation of semantic recognition for buildings.

	TP	FP	FN	TN	Precision (%)	Recall (%)	Accuracy (%)
Floor	34	0	6	323	100	85	98.4
Window	268	101	132	89	72.6	67	60.5
Roof	39	0	1	318	100	97.5	99.7
Eaves	16	0	4	341	100	80	98.9

4.2. Uncertainties and Limitations of the Proposed Method

Based on the results, we can see that despite not all parts of building being scanned, the building can still be reconstructed by using our method. Furthermore, all of the semantic information of the

building is preserved in the XBML format, which is convenient for reading and editing. Figure 15g–i illustrates the results after simple editing. However, there are still a number of limitations in our method:

- (1) The recognition of floor and window is not accurate enough. Although the structure splitting algorithm works well on the datasets, it cannot detect the internal architectural structure, i.e., the height is not a true value. Currently, we evaluate it just based on the number of floors by the manual observation from the point cloud. For the windows, most of the windows and doors are recognized correctly, however, it is clear that their size is not accurate, which is due to the non-uniformity of the local density of the façade point cloud, thus leading to inaccuracy in the boundary confidence value for the floor analysis.
- (2) During the roof reconstruction, a few results are different from those in the point cloud (Figure 16a–c), which is caused by the inaccurate segmentation of the point cloud and the threshold of the region growing algorithm. Another possibility is the algorithmic accuracy.
- (3) Limitations in the type of building reconstruction. Currently, we have developed several building base elements in our framework, and other elements, such as dormers, balconies, and windowsills, which limits the type of building reconstruction to simple structures. More building elements are under development for addressing complex building reconstruction.

Compared with the conventional method, the method proposed in this paper has several advantages: more detailed reconstruction, higher degree of automation, more compact building description, richer semantic information, and object-oriented design. In term of geometric accuracy, our method can achieve an accuracy that is comparable with related works.

However, a number of features still need to be improved. In the future, we will extend the building element class library so that more complicated buildings can be described. The current approach needs lots of prior knowledge; thus, in the next step, a more intelligent algorithm will be used for semantic feature recognition. Furthermore, due to the incompleteness of the building information in the airborne laser scanning data and bottlenecks in automatic registration, laser scanning data are limited in large-scale city modelling; hence, we plan to use a matching point cloud as one of our data sources in the future.

5. Conclusions

In this paper, we presented a semantic modelling framework-based method for reconstructing buildings from an unorganized point cloud. The semantic modelling framework inspired by the XAML in the WPF can use the semantic information to reconstruct the building without the needs of polygon fitting and gap filling. To extract the semantic information and convert it into the building description language, XBML, in our framework for modelling, a reconstruction workflow was established. We predefined several knowledge databases to assist in semantic features recognition. In the workflow, different algorithms for the roof, façade, floor, and windows were designed. The experimental results indicate that our approach can successfully extract semantic information and use it to reconstruct the building automatically. A high geometric accuracy and feasibility are maintained in our approach. Additionally, compared to conventional methods, our method can achieve a high level of geometric detail and rich semantic information.

The proposed semantic modelling framework-based reconstruction approach is aimed at combining the advantages from the data-driven, model-driven, and knowledge-based approaches. Currently, although our semantic modelling framework is still an experimental framework for the automated semantic reconstruction of buildings, it provides an interface for the semantic information extracted from various data. Additionally, in our reconstruction workflow, the semantic recognition of buildings will be an important part in the semantic reconstruction of a large-scale city, which is crucial for the quality of reconstruction. Hence, in our future studies, we will primarily concentrate on these two points.

Acknowledgments: This work is supported by the National Natural Science Foundations of China (NSFC) under Grant 41271374. The authors wish to thank all the numerous scientists who released their code under Open Source licenses and made our method possible.

Author Contributions: Qingdong Wang proposed the method and implemented it, all authors discussed the structure of the manuscript, Li Zhang and Li Yan revised the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Salman, N.; Yvinec, M.; Merigot, Q. Feature preserving mesh generation from 3D point clouds. *Comput. Graph. Forum* **2010**, *29*, 1623–1632. [[CrossRef](#)]
2. Perera, G.S.N.; Maas, H.-G. Cycle graph analysis for 3D roof structure modelling: Concepts and performance. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 213–226. [[CrossRef](#)]
3. Wang, R.S.; Hu, Y.; Wu, H.Y.; Wang, J. Automatic extraction of building boundaries using aerial Lidar data. *J. Appl. Remote Sens.* **2016**. [[CrossRef](#)]
4. Xiao, J.; Gerke, M. Building footprint extraction based on radiometric and geometric constraints in airborne oblique images. *Int. J. Image Data Fusion* **2015**, *6*, 270–287. [[CrossRef](#)]
5. Zhang, J.; Lin, X. Advances in fusion of optical imagery and lidar point cloud applied to photogrammetry and remote sensing. *Int. J. Image Data Fusion* **2016**. [[CrossRef](#)]
6. Xiong, B.; Oude Elberink, S.; Vosselman, G. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 227–242. [[CrossRef](#)]
7. Sui, W.; Wang, L.; Fan, B.; Xiao, H.; Wu, H.; Pan, C. Layer-wise floorplan extraction for automatic urban building reconstruction. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 1261–1277. [[CrossRef](#)] [[PubMed](#)]
8. Stiny, G.N. *Pictorial and Formal Aspects of Shape and Shape Grammars and Aesthetic Systems*; University of California: Los Angeles, CA, USA, 1975.
9. Parish, Y.I.H.; Müller, P. Procedural modeling of cities. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; pp. 301–308.
10. Wonka, P.; Wimmer, M.; Sillion, F.; Ribarsky, W. Instant architecture. *ACM Trans. Graph.* **2003**, *22*, 669–677. [[CrossRef](#)]
11. Müller, P.; Wonka, P.; Haegler, S.; Ulmer, A.; van Gool, L. Procedural modeling of buildings. *ACM Trans. Graph.* **2006**, *25*, 614–623. [[CrossRef](#)]
12. Larive, M.; Gaildrat, V. Wall grammar for building generation. In Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, Kuala Lumpur, Malaysia, 28 November–2 December 2006; ACM: New York, NY, USA, 2006; pp. 429–437.
13. Lipp, M.P.M. Interactive visual editing of grammars for procedural architecture. *ACM Trans. Graph.* **2008**, *27*, 10. [[CrossRef](#)]
14. Finkenzeller, D. Detailed building facades. *IEEE Comput. Graph. Appl.* **2008**, *28*, 58–66. [[CrossRef](#)]
15. Müller, P.; Zeng, G.; Wonka, P.; Van Gool, L. Image-based procedural modeling of facades. *ACM Trans. Graph.* **2007**. [[CrossRef](#)]
16. Bekins, D.; Aliaga, D.G. Build-by-number: Rearranging the real world to visualize novel architectural spaces. In Proceedings of 16th IEEE Visualization Conference (VIS 2005), Minneapolis, MN, USA, 23–28 October 2005; pp. 143–150.
17. Hohmann, B.; Krispel, U.; Riemenschneider, H.; Havemann, S.; Karner, K.; Bischof, H.; Fellner, D.D. Cityfit-High-Quality urban reconstructions by fitting shape grammars to images and derived textured point clouds. In Proceedings of the 3rd ISPRS International Workshop 3D-ARCH 2009, Trento, Italy, 25–28 February 2009.
18. Hohmann, B.; Havemann, S.; Krispela, U.; Fellner, D. A GML shape grammar for semantically enriched 3D building models. *Comput. Graph.* **2010**, *34*, 322–334. [[CrossRef](#)]
19. Mathias, M.; Martinovic, A.; Weissenberg, J.; Gool, L.V. Procedural 3D building reconstruction using shape grammars and detectors. In Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Hangzhou, China, 16–19 May 2011; pp. 304–311.
20. Vanegas, C.A.; Aliaga, D.G.; Benes, B. Automatic extraction of Manhattan-world building masses from 3D laser range scans. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1627–1637. [[CrossRef](#)] [[PubMed](#)]

21. Vanegas, C.A.; Aliaga, D.G.; Benes, B. Building reconstruction using Manhattan-world grammars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 358–365.
22. Pu, S.; Vosselman, G. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 575–584. [[CrossRef](#)]
23. Wang, C.; Cho, Y.K.; Kim, C. Automatic bim component extraction from point clouds of existing buildings for sustainability applications. *Autom. Constr.* **2015**, *56*, 1–13. [[CrossRef](#)]
24. Wang, H.; Zhang, W.; Chen, Y.; Chen, M.; Yan, K. Semantic decomposition and reconstruction of compound buildings with symmetric roofs from Lidar data and aerial imagery. *Remote Sens.* **2015**, *7*, 13945–13947. [[CrossRef](#)]
25. Susaki, J. Knowledge-based modeling of buildings in dense urban areas by combining airborne Lidar data and aerial images. *Remote Sens.* **2013**, *5*, 5944–5968. [[CrossRef](#)]
26. Luo, C.; Sohn, G. A knowledge based hierarchical classification tree for 3D facade modeling using terrestrial laser scanning data. In *Geomatics—Shaping Canada’s Competitive Landscape*, Proceedings of the Canadian Geomatics Conference 2010 and ISPRS Com I Symposium, Calgary, AB, Canada, 15–18 June 2010; Copernicus Gesellschaft Mbh: Gottingen, Germany, 2010; Volume 38.
27. Groger, G.K.; Nagel, T.H.; Hafele, C.; Hrsg, K.H. *OGC City Geography Markup Language (Citygml) Encoding Standard*; No 12-019; Open Geospatial Consortium: Wayland, MA, USA; 4 April 2012.
28. IFC Overview Summary. Available online: <http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary> (accessed on 31 August 2016).
29. Green Building XML. Available online: https://en.wikipedia.org/wiki/Green_Building_XML (accessed on 28 January 2016).
30. Solis, D.M. *Illustrated WPF*; Apress: New York, NY, USA, 2009.
31. Sells, C.; Griffiths, I. *Programming WPF*, 2nd ed.; O’Reilly Media: Sebastopol, CA, USA, 2007; p. 8.
32. MXML. Available online: <https://en.wikipedia.org/wiki/MXML> (accessed on 29 July 2016).
33. Extensible Application Markup Language. Available online: https://en.wikipedia.org/wiki/Extensible_Application_Markup_Language (accessed on 29 July 2016).
34. Fischler, M.A.B.; Robert, C.; Foley, J.D. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
35. Edelsbrunner, H.; Kirkpatrick, D.; Seidel, R. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory* **1983**, *29*, 551–559. [[CrossRef](#)]
36. Teh, C.; Chin, R.T. On the detection of dominant points on digital curves. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 859–872. [[CrossRef](#)]
37. Verma, V.; Kumar, R.; Hsu, S. 3D building detection and modeling from aerial lidar data. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006.
38. Shen, C.H.; Huang, S.S.; Fu, H.B.; Hu, S.M. Adaptive partitioning of urban facades. *ACM Trans. Graph.* **2011**, *30*, 1–9. [[CrossRef](#)]
39. Tutzauer, P.H.N. Facade reconstruction using geometric and radiometric point cloud information. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-3/W2*, 247–252. [[CrossRef](#)]
40. Tuttas, S.; Stilla, U. Window detection in sparse point clouds using indoor points. In Proceedings of the PIA11: Photogrammetric Image Analysis, Munich, Germany, 5–7 October 2011.
41. Tuttas, S.; Stilla, U. Reconstruction of facades in point clouds from multi-aspect oblique als. In *Cmrt13—City Models, Roads and Traffic 2013*; Rottensteiner, F., Stilla, U., Hinz, S., Eds.; Copernicus Gesellschaft Mbh: Gottingen, Germany, 2013; Volume II-3/W3, pp. 91–96.
42. Dahlke, D.; Linkiewicz, M.; Meissner, H. True 3D building reconstruction: Façade, roof and overhang modelling from oblique and vertical aerial imagery. *Int. J. Image Data Fusion* **2015**, *6*, 314–329. [[CrossRef](#)]

