Free multi-floor indoor space extraction from complex 3D building models

Xiong, Qing; Zhu, Qing; Du, Zhiqiang; Zlatanova, Sisi; Zhang, Yeting; Zhou, Yan ; Li, Yun

DOI
10.1007/s12145-016-0279-x

Publication date
2016

Document Version
Accepted author manuscript

Published in
Earth Science Informatics

Citation (APA)
Xiong, Q., Zhu, Q., Du, Z., Zlatanova, S., Zhang, Y., Zhou, Y., & Li, Y. (2016). Free multi-floor indoor space extraction from complex 3D building models. *Earth Science Informatics*, *10*(1), 69-83. https://doi.org/10.1007/s12145-016-0279-x

Free Multi-floor Indoor Space Extraction from Complex 3D Building Models

Qing Xiong[1], Qing Zhu[1,2,3] , Zhiqiang Du[1,3], Sisi Zlatanova[4], Yeting Zhang[1,3], Yan Zhou[5] and Yun Li[2]

[1]State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei, China
[2] Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu, China
[3] Collaborative Innovation Center for Geospatial Technology, Wuhan, China
[4] 3D Geoinformation, Urbanism, Delft University of Technology, Delft, Netherlands
[5] School of Resources and Environment, University of Electric Science and Technology of China, Chengdu, China

Abstract: Intelligent navigation and facility management in complex indoor environments are issues at the forefront of geospatial information science. Indoor spaces with fine geometric and semantic descriptions provide a solid foundation for various indoor applications, but it is difficult to comprehensively extract free multi-floor indoor spaces from complex three-dimensional building models, such as those described using CityGML LoD4, with existing methods for the subdivision or extraction of indoor spaces based on vector topology processing. Therefore, this paper elaborates a new voxel-based approach for extracting free multi-floor indoor spaces from 3D building models. It transforms the complicated vector processing tasks into a simple raster process that consists of three steps: voxelization with semantic enhancement, voxel classification, and boundary extraction. Experiments illustrate that the proposed method can automatically and correctly extract free multi-floor indoor spaces, especially two typical kinds of open indoor spaces, namely, lobbies and staircases.

Keywords: free multi-floor indoor space, CityGML LoD4, indoor space extraction, voxel

1. Introduction

Free multi-floor indoor spaces are the areas in which human activities are most intensively concentrated. Therefore, the increasing requirements regarding indoor navigation and facility management for such areas are giving rise to more explicit and elaborate methods for the perception and understanding of indoor spaces (Zlatanova et al. 2013a). Thus, the Open Geospatial Consortium (OGC) has drafted a critical indoor navigation standard that is focused on the modeling of indoor spaces for navigation purposes (OGC IndoorGML 2014). Specifically, it contains a minimal set of geometric and semantic models of construction components, which can be used to localize indoor objects and to map spatial relationships between indoor subspaces. However, it ignores components of indoor spaces that are considered irrelevant, such as furniture space, but indispensable in spatial analyses related to location intelligence, indoor emergency response, architectural structure assessment, and navigation. Additionally, if no physical spatial object exists between two spaces in CityGML or IndoorGML, such as in the space between a kitchen and living room, a virtual space or boundary is needed to make each room a closed space (Kim et al. 2014; Xie et al. 2013). Therefore, it is important to improve the semantic definition in IndoorGML to encompass the necessary spaces for indoor applications before indoor space extraction. Existing approaches can be divided into two broad categories: those based on 2D floor plans and those based on 3D models. The first category typically includes three types of methodologies: incomplete subdivision algorithms that generate networks (Liu and Zlatanova 2011), complete subdivision algorithms that generate networks (Lamarche and Donikian 2004; Wallgrün 2005; Lorenz et al. 2006; Stoffel et al. 2007; Geraerts 2010), and complete subdivision algorithms that generate regular grids (Li et al. 2010; Girard et al. 2011; Afyouni et al. 2012). Methods of the first type strive to derive a network to be used for path computation, which is based on Poincaré duality, centerlines, visibility graphs, or some combination of these concepts. Algorithms of the second type concern the subdivision of 2D plans into small convex cells according to certain criteria, such as convexity and the distances between walls, which are then used to derive a network by applying Poincaré duality while considering the avoidance of collisions between moving objects. Methods of the third type produce a set of grid cells (e.g., rectangular, hexagonal, octagonal, etc.) that are associated with certain semantics according to the underlying 2D objects (rooms, doors) that require tracking and integration with continuing phenomena, such as smoke or fire.

However, fully elaborated semantic information is not maintained in most of the 2D approaches developed from these 2D subdivision methods, and the semantics, when applied, represent only construction elements that are critical for navigation, such as doors, rooms and stairs. Moreover, these algorithms predominantly consider geometric subdivisions to support path-finding methods; they lack integration of the functional and thematic meanings of indoor spaces.

Similar to the first category, algorithms in the second category can also be subdivided into three types of methods: incomplete subdivision algorithms based on networks (Schaap et al. 2011; Thill et al. 2011), complete subdivision algorithms based on networks (Lee 2004; Meijers et al. 2005; Becker et al. 2008; Brown et al. 2012; Liu and Zlatanova 2012; Isikdag et al. 2013; Krūminaitė and Zlatanova 2014), and complete subdivision algorithms based on grids (Yuan and Schneider 2010). Algorithms of the first type derive a network embedding of 2D floor plans into a 3D building model and can accommodate semantics that are specific to individual indoor movements, such as ingress, egress, or vertical movements via stairs and elevators, to account for personal disabilities. Algorithms of the second type consider indoor spaces as nodes and edges connecting those nodes to form surfaces that respect Poincaré duality. These algorithms can be linked and used for common analysis and navigation applications based on the concept of dual graphs. Algorithms of the third type compute the accessible parts of an indoor environment to represent the complete 3D indoor space using grid-based graphs, considering the constraints imposed by the widths and heights of pedestrians.

In short, methods based on 3D grids are slightly superior to network methods because they can identify the types of spaces (e.g., rooms or corridors) represented by different grid cells and embed rich semantics for each grid cell. Moreover, they can also calculate paths, either for Unmanned Aerial Vehicles (UAVs) or for pedestrians, by applying 3D shortest-path algorithms, such as A* (Krūminaitė and Zlatanova 2014). Nevertheless, existing approaches cannot generate correct results when subdividing free multi-floor indoor spaces with abundant details and fuzzy boundaries.

As mentioned above, for various applications related to indoor navigation and facility management, the existing 2D and 3D methods of indoor space subdivision face the problem of balancing the complexity and accuracy of coherent geometric, topological and semantic relationships in a free multi-floor indoor environment. Here, complexity refers to the presence of not only irregular structures but also specialized types of connecting spaces with specific semantics that are used for vertical and horizontal travel, such as lobbies and staircases. Furthermore, accuracy requires the appropriate subdivision of an indoor environment to accurately locate areas of interest, and the results do not necessarily overlap with the enclosed boundaries of the environment. Consequently, the key problem in free multi-floor indoor space extraction is to completely and efficiently calculate the correct enclosed boundaries with specific and proper semantics to support further static or dynamic analysis for applications such as indoor navigation and facility management.

This paper proposes a novel and general method for the automated extraction of free multi-floor indoor spaces of types that are commonly used in complex building models, with the correct structures and boundaries. The remainder of this paper is organized as follows. In Section 2, an expanded definition of indoor spaces is introduced to enrich the geometric descriptions and semantic relationships of indoor spaces in IndoorGML. Section 3 presents the principles of the proposed algorithm and describes its implementation. Experimental results and further analyses are outlined in Section 4, and concluding remarks are presented in Section 5.

2. The expanded definition of indoor space in IndoorGML

The key strategy adopted in the method proposed in this paper is to reorganize the independent thematic surface elements of a CityGML LoD4 model to extract correct enclosed boundaries using a voxel-based definition of indoor spaces as the basic units for geometric analysis, because CityGML LoD4 cannot represent the correct enclosed boundaries of free multi-floor indoor spaces such as that shown in Figure 1. The space between a 2nd floor and 3rd floor comprises a typical free space that does not have a clear boundary, as shown in Figure 1(a) and Figure 1(b). The space must be identified and distinguished to separate each free space. The red dotted line in Figure 1(c) shows the free space boundary between the 2nd floor and 3rd floor. Furthermore, a free indoor space represents a navigable space in which users can move freely. This paper introduces newly expanded semantic and geometric definitions of indoor spaces to appropriately address this issue.

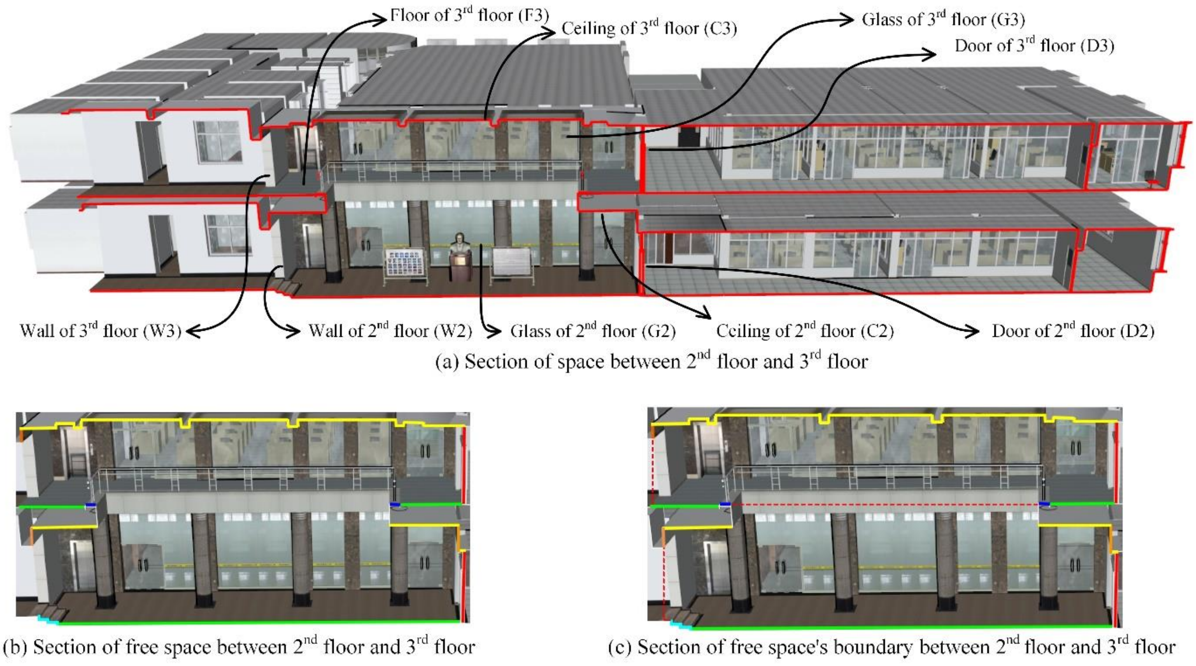


Figure 1. Example of a free multi-floor indoor space.

2.1 Semantic definition of indoor space

In IndoorGML, an indoor space is defined as a space within one or multiple buildings consisting of architectural components, such as entrances, corridors, rooms, doors and staircases (Li 2008; Jensen et al. 2010). For the purpose of indoor navigation, IndoorGML (OGC 2014) distinguishes four types of indoor spaces, as shown in Figure 2: 1) a general space is identified as any navigable space, such as a room, corridor or foyer; 2) a connection space represents an open space that provides passage between two indoor spaces; 3) an anchor space represents a special type of open space that provides a connection between an indoor space and an outdoor space; and 4) a transition space represents a real-world space that provides passage between two indoor spaces. IndoorGML is concerned with the spaces (e.g., rooms, corridors and stairs) formed by architectural components and the relationships amongst them. However, components that are not relevant to the presentation of these spaces, such as furniture and facilities, are not represented in IndoorGML.

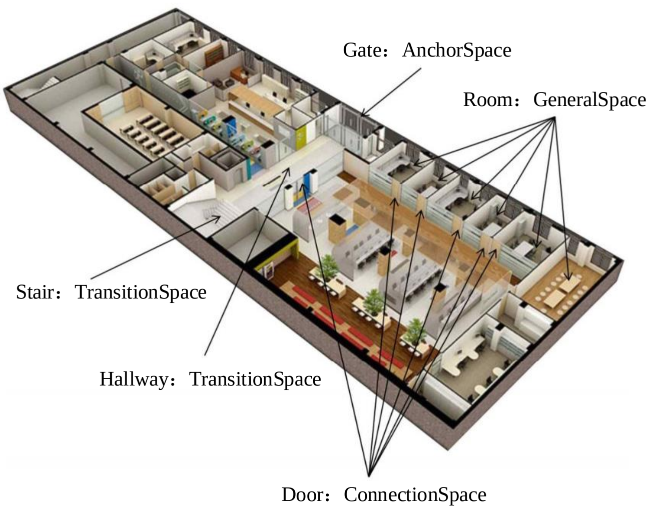


Figure 2 Classification of indoor spaces in IndoorGML.

To improve and enhance the network representation of IndoorGML, it is necessary to define a type of virtual space that provides a connection between two general spaces. Such a virtual space is usually located at the edge of a general space and represents nothing in the physical world. It can serve as an additional navigation node between general spaces to enrich the topological relationships represented in a navigation network. In Figure 3, the red ovals indicate the borders between two corridors in the 3D data, 2D data and network representation of the indoor space of a building. Based on an improved network representation that includes virtual spaces representing these border regions, an indoor navigation path can naturally transition from one corridor to another while avoiding an incorrect path that would pass through the wall between two such general spaces. Because of the inherent characteristics of virtual spaces, they are not easy to extract. The method presented in this paper can be used to obtain the correct 3D boundaries of virtual spaces by computing and identifying the edges between two general spaces.

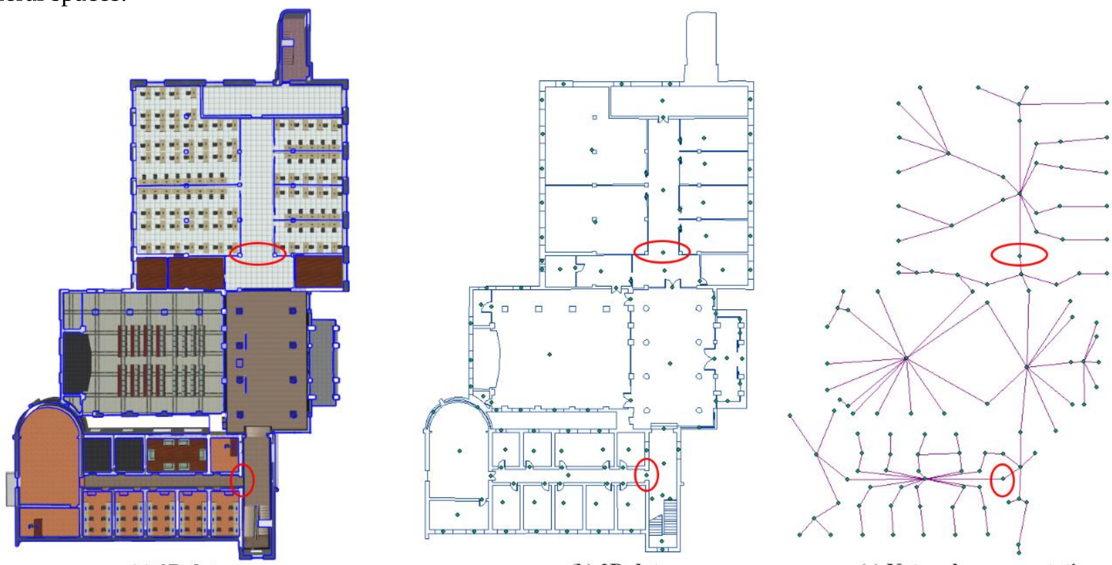


Figure 3. Example of virtual spaces.

As an extension of the representation of indoor spaces in IndoorGML, this paper defines a hierarchy of semantic information for indoor spaces (as shown in Figure 4), which refers to a previously developed conceptual framework of the subdivision of spaces for indoor navigation (Xiong et al. 2013; Zlatanova et al. 2013b). An indoor space can be classified as either navigable or non-navigable. Navigation analysis can be applied to navigable spaces, including general spaces and transition spaces. Facility management can be applied to non-navigable spaces, such as obstacle spaces. General spaces include corridors, rooms and foyers, which are basic types of navigable spaces. Same-floor connection spaces, between-floor connection spaces, anchor spaces, and virtual spaces can be classified as transition spaces. For a same-floor connection space, the navigable space can be defined as a portal in the same plane; such spaces include doors and windows. By contrast, a between-floor connection space represents a vertically navigable relationship, such as that provided by a staircase, elevator, or escalator. Anchor spaces include gates and fire stairs. Obstacle spaces represent the locations and boundaries of furniture. In summary, the proposed semantic hierarchy is used for two purposes: to provide classification and identification of indoor spaces and to determine the connectivity among spaces.

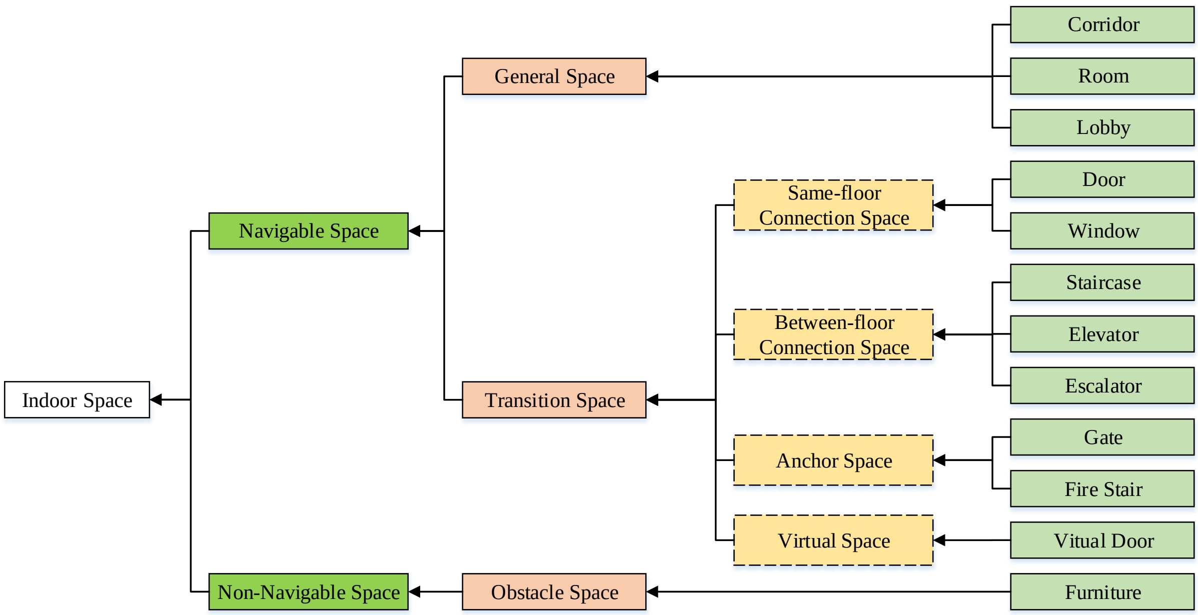


Figure 4 Hierarchy of semantic information for indoor spaces.

The expanded semantic definition of indoor spaces improves the hierarchy of indoor spaces by extending the structures that can be described from basic spaces to abstract indoor spaces with 4 levels of specification. It adds representations of virtual spaces and obstacles space to enrich the description of indoor spaces. It also enhances the network representation of indoor spaces to provide more reliable node-edge graphs.

2.2 Geometric definition of indoor space

To simplify the computations required for the extraction of complex indoor spaces, voxels are used in this paper to represent a cellular notion of indoor space. Voxels have four important properties. First, it is easy to improve the computational efficiency of voxel generation. Second, every voxel has an identifier that can easily be mapped to a room number or some other feature. Third, any given position in cellular space can be specified by a voxel identifier, although (x, y, z) coordinates may also be used for more precise position specification. Fourth, each voxel may have common boundaries with other voxels but may not overlap with them. Moreover, the proposed system of semantics allows the definition of voxels, which can be important for navigation.

A 3D indoor space is an enclosed functional interval (such as a room, corridor, or staircase) composed of basic building elements (such as a ceiling, walls, and a floor). To quantitatively express its geometric dimensions and details, each indoor space can be represented by a set of voxels. The minimum voxel size is determined by the geometric size of the objects in the indoor space. An indoor space can be defined as follows:

$$IS = \{Function(s), \cup R_i\}$$

A complete definition of an indoor space (*IS*) includes its functionality (*Function*) and a set of geometric locations ($\cup R_i$). *Function* refers to the semantic information *s* associated with the space. $\cup R_i$ includes a set of voxel locations that represent a list of subspaces (R_i). R_i is a normative cuboid space and can be defined as follows:

$$R_i = \begin{Bmatrix} origin\_x + size \times Index_x(i) \le x \le origin\_x + size \times (Index_x(i)+1) \\ origin\_y + size \times Index_y(i) \le y \le origin\_y + size \times (Index_y(i)+1) \\ origin\_z + size \times Index_z(i) \le z \le origin\_z + size \times (Index_z(i)+1) \end{Bmatrix}$$

Here, *origin_x*, *origin_y* and *origin_z* represent the coordinates of the origin in the target indoor space. *size* is the minimum voxel size. $Index_x(i)$ is the x-axis value assigned to subspace R_i. $Index_y(i)$ is the y-axis value assigned to subspace R_i. $Index_z(i)$ is the z-axis value assigned to subspace R_i. $Index_x(i)$, $Index_y(i)$ and $Index_z(i)$ cannot be computed using a universal formula because they refer to different dimensions, which increases the difficulty of automated indoor space extraction. However, because R_i is a typical description in a grid data format, such normative cuboid spaces can be useful and efficient for indoor computations.

Furthermore, the boundaries of each indoor space can be established by assessing the number of subspace neighbors. The marginal voxel of a subspace has at least one face with no adjacent voxel; however, each face of the voxels of an internal subspace is associated with an adjacent object. Based on this principle, the boundary of each indoor space can be defined as follows:

$$Space\_Boundary = \{R_j(adjacent\_number < 6)\}$$

where R_j *(adjacent_number < 6)* is the set of the marginal subspace in which less than 6 objects are adjacent to a voxel. Finally, each indoor space comprises a set of continuous voxels in a grid-based data format. In this format, each boundary of the indoor space is enclosed and continuously distributed in space to prevent 3D fuzzy boundaries.

3. Algorithm

3.1 Principle

To overcome the drawbacks of existing vector approaches, this paper presents a coarse-to-fine method for the extraction of indoor spaces from complex LoD4 building models. The proposed method considers the semantic relationships among the architectural components and transforms the analyzed model into a set of voxels in 3D space to perform efficient computations for the extraction of indoor spaces. A flowchart of the algorithm is shown in Figure 5, and the steps of the algorithm are further described below.

Step 1: Based on the voxelization approach, the original LoD4 models are transformed into a set of voxels that inherit the semantic relationships of the input data.

Step 2: Rough extraction is performed to eliminate invalid spaces while retaining the effective set of voxels extracted for indoor spaces with indistinct boundaries.

Step 3: The semantic relationships of the voxels are used as the constraints for performing an accurate cluster analysis of indoor spaces with distinct boundaries based on accurate extraction.

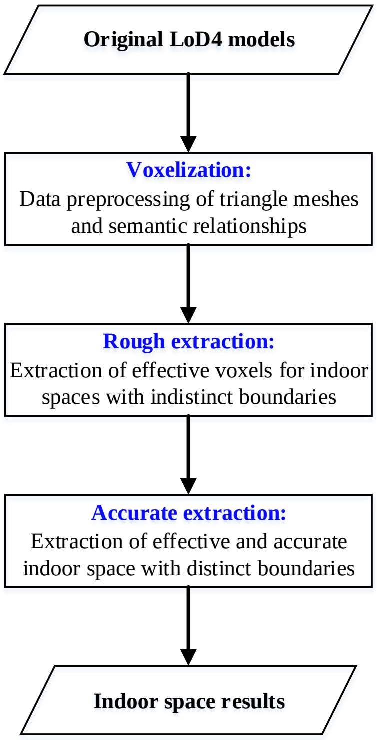


Figure 5. Flowchart of the proposed automated indoor space extraction algorithm.

3.2 Voxelisation: pre-processing of LoD4 model

Voxelization produces volume datasets that contain the surface information and interior attributes of the original 3D models. In addition, pre-processing of the LoD4 models can avoid complicated vector and topology calculations and support the partitioning and classification of complex indoor spaces. Based on scale-space theory (Oomes et al. 1997) as well as the distance field or distance transform method (Jones and Satherley 2000), this paper presents improvements that result in a faster voxelization algorithm whose efficiency does not depend on the original pose of the input data.

The advantages of the voxelization process are the introduction of coding rules and the binding of semantics to the six directions of each voxel for efficient querying and computation. First, the minimum bounding box of the 3D model is calculated. Based on the minimum bounding box, the algorithm determines the local voxel space (the origin point is the minimum point of the bounding box, for instance, point *a* in Figure 6, and the coordinate axis is the same as in Euclidean space). To merge adjacent voxels, this paper introduces an index structure based on the classic neighborhood search algorithm (Meagher 1982), as shown in Figure 6.

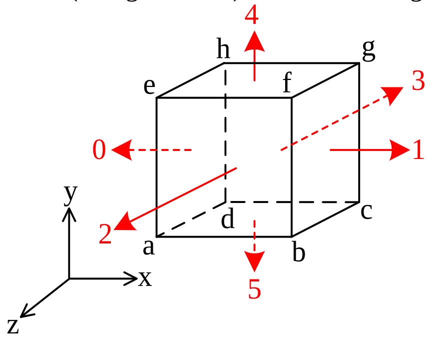


Figure 6 Voxel encoding rules.

Based on the voxel encoding rules illustrated in Figure 6, a table that represents each geometric unit and its encoding values can be constructed for neighborhood searching and voxel merging, as shown in Table 1.

Table 1 Geometry unit and its encoding value.

Geometry unit	Rectangular mesh				Edge				Vertex			
	adhe	bcgh	…	efgh	ab	bc	…	dh	a	b	…	h
code	0	1	…	5	25	15	…	03	025	125	…	034

Because the basic geometric unit of the input data is a triangular mesh, the voxelization process can be divided into two main stages: the voxelization of the edges and the voxelization of the triangular mesh. The voxelization of the edges serves as the basis for the voxelization of the triangular mesh.

3.2.1 The voxelisation of edges

As shown in Figure 7, a ray from the voxel at the starting point (P_0 (i0, k0, j0)) that points to the end of an edge (P_1 (i1, k1, j1)) will intersect the starting voxel at point P_{p0} (i0, k0, 0). During the point's voxelization, the location of the voxel that corresponds to point P_0 is calculated, and an index value is produced according to the voxel encoding rules. Based on these encoding rules, the code value of the neighbor of this point P_{p0} can also be calculated.

The node P_1 is then set as the new starting point, in accordance with the above method, to recursively find the intersections with all boundary voxels until the end of the edge is reached.

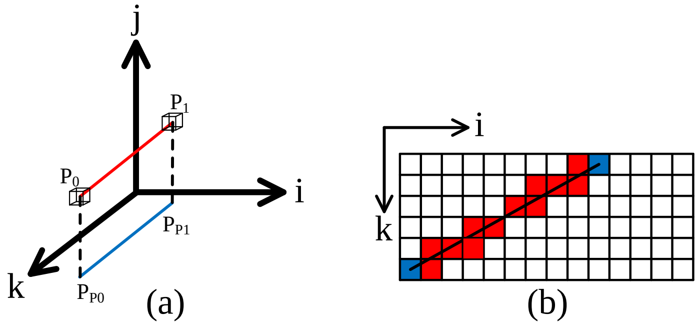


Figure 7. Voxelization of an edge.

3.2.2 The voxelisation of triangle mesh

After the process of edge voxelization, the voxels inside each triangular surface are determined, as shown in Figure 8. For clearer visual interpretation, the projected grid of voxels uses the four-connectivity in a 2D plane. In reality, the voxels use the six-connectivity in 3D space.

First, the triangular projection of $P_0P_1P_2$ (P_0 (i0, k0, j0), P_1 (i1, k1, j1), P_2 (i2, k2, j2)) onto the X-Y plane, $P_{P0}P_{P1}P_{P2}$ (P_{P0} (i0, k0, 0), P_{P1} (i1, k1, 0), P_{P2} (i2, k2, 0)), is generated, and then the voxels on the vertices and edges are determined. The grid coordinates of the projected zone are scanned over elements i to determine the component i0 that satisfies the grid voxel coordinate condition for the grid components (i0, k).

$$iu(k) \le i0 \le iv(k)$$

The linear intersection between i=i0 and k=k0, in the voxel space, j0, is then calculated. The resulting coordinates (i0, k0, j0) are the parameters needed to identify a voxel.

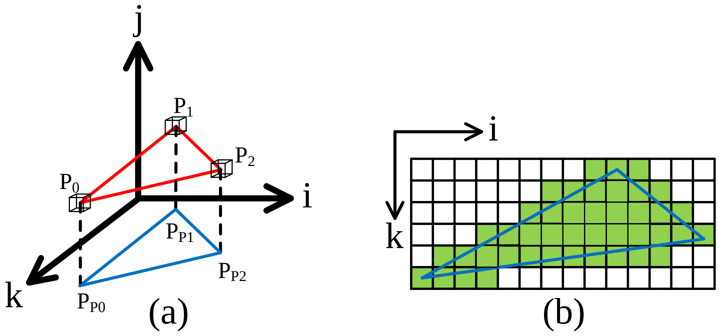


Figure 8 Voxelization of the triangular mesh.

During this step, the triangle's space (*TS*) is defined as follows.

$$TS = \{s_t, \bigcup R\}$$

The set of voxels $\bigcup R$ is tagged with the triangle's semantics s_t, representing the critical constraint conditions for the next step.

3.3 Rough extraction

We perform rough extraction by filtering out the invalid voxels and combining the valid voxels into a reliable indoor space with specific semantics. Figure 9 shows the flowchart for the extraction of valid voxels from the voxelization results based on their semantics. The algorithm for rough extraction is as follows.

Step 1: Because the voxel queue contains outdoor voxels as well as indoor voxels, traverse each voxel and determine whether its location is between the boundary box and the input model. Then, filter out the outdoor voxels and reserve the rest.

Step 2: Calculate the locations of any voxels between the internal and external walls, because these voxels are also invalid. Then, filter them out and reserve the rest.

Step 3: Traverse the remaining voxels without semantics and search the six directions (as shown in Figure 6) to identify the semantics that belong to the nearest voxel, as inherited from the input data, in each search direction.

Step 4: Verify the complete coverage of the remaining voxels according to step 2 to ensure the integrity of the indoor environment.

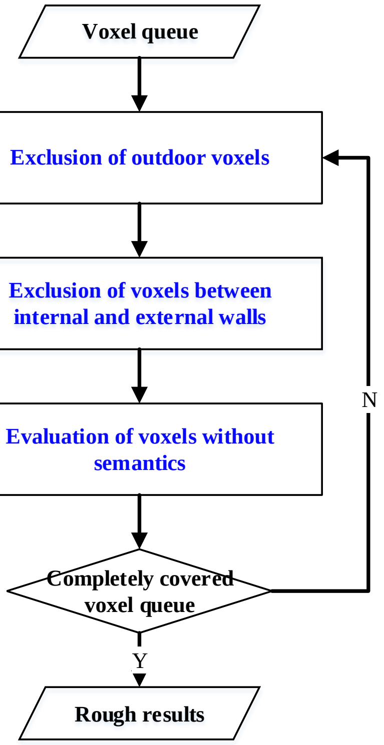


Figure 9 Flowchart of rough extraction.

After this step, we have obtained two types of voxels: a voxel of the first type is located in a specific indoor space and has only one semantic tag, whereas a voxel of the second type is located in an indeterminate space between two or more indoor spaces and has (potentially different) semantics in all six directions, as shown in Figure 10. A voxel of the first type, with semantics in only one direction, represents the semantics of a physical object, which is adjacent to the surface of the voxel that corresponds to the normal vector associated with the semantics. Voxels of the second type are always located in a free and open space, such as a lobby or staircase. For such voxels, we need to confirm the correct indoor space relationships via an accurate extraction step.

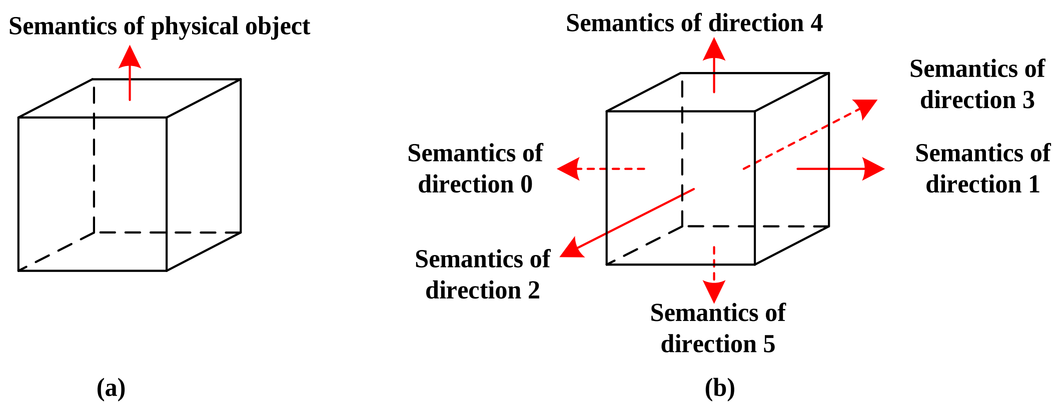


Figure 10 Examples of rough extraction results.

3.4 Accurate extraction

To achieve the spatial demarcation of a free multi-floor indoor environment, we need to verify the reliable enclosure of each indoor space and create boundaries with coherent geometry when the extracted indoor spaces are not geometrically enclosed.

Because a voxel of the second type extracted in the rough results has six semantics in six directions, we must confirm its semantics, or the space to which it belongs. If the six semantics are the same, the decision is easy to make. However, if there are differences among the six semantics, we need to estimate the semantic weights in the six directions. Figure 11 shows the flowchart for updating the semantics of each such voxel from the rough results. The algorithm for this process is as follows:

Step 1: Set the weight of the semantics in Figure 4 equal to 1.

Step 2: Calculate the semantic weight for each unassigned voxel for each type of semantics as the sum of the weights in the directions associated with the same type of semantics. The weight (SW) can be defined as

$$SW = \left\{ \sum_{i=0}^{5} w_i , s_d \right\}$$

$\sum_{i=0}^{5} w_i$ indicates the summation of the semantic weight in direction d and s_d indicates the semantics in direction d.

Step 3: Determine whether the sum is 2 or greater. If not, mark this voxel as invalid and go to step 4. Then, determine whether the SW for the current direction d is equal to the maximum SW. If it is, assign the semantics of the voxel as s_d . Otherwise, either assign the semantics according to direction *d* if *d* is direction 4 or 5 or return the voxel to the rough results for further processing if *d* is one of the other directions. Finally, consider this voxel as ‘produced’ by the input data and update the semantics of the neighbors of this voxel in all six directions, as described in step 3 in Section 3.3.

Step 4: Verify complete coverage of the remaining voxels according to step 2 to ensure the integrity of the voxel assignments.

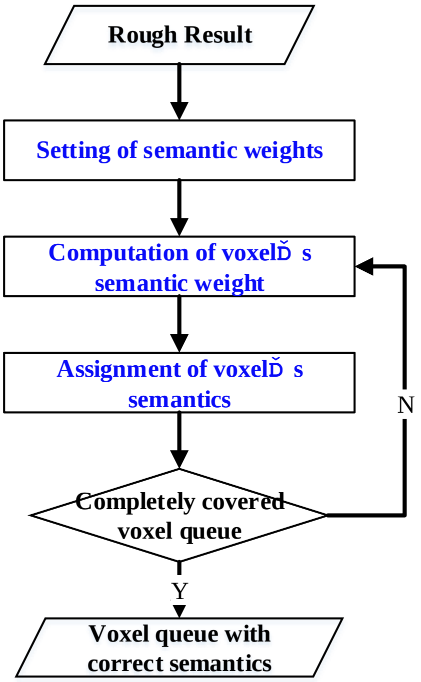


Figure 11 Flowchart of the update procedure for the voxels’ semantics.

Because each voxel in the rough results is an atomistic-level volumetric object, the geometry of the boundary of each indoor space can be treated as a whole or as a part of a coherent geometric border. Thus, we can verify the closure of each space's boundary and then automatically calculate the corresponding and matching geometric surface with its semantics using the flowchart shown in Figure 12.

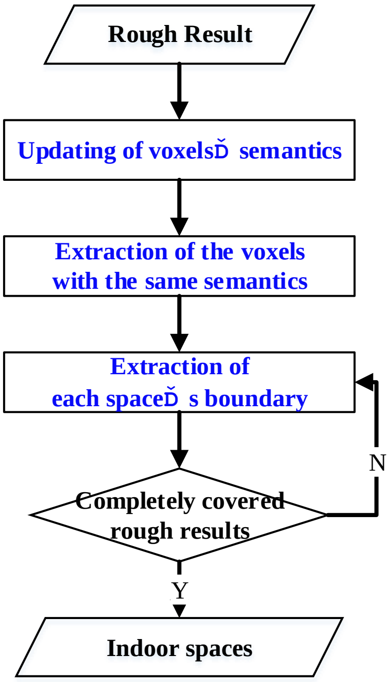


Figure 12 Flowchart of accurate extraction.

The algorithm for accurate extraction is as follows.

Step 1: Update the rough results for the voxels' semantics using the method described above.

Step 2: Extract the voxels with the same semantics from the valid queue to obtain a complete indoor space filled with voxels.

Step 3: Extract the edges of the boundary of each indoor space and search the closed contour until all of the extracted edges have been processed. The enclosed contours are then extracted as the geometric frontier of the indoor space.

Step 4: Verify complete coverage of the remaining indoor spaces according to step 3 to ensure the integrity of the indoor environment.

4. Experiments and analysis

4.1 Accuracy analysis

A voxel size of 0.3 m × 0.3 m × 0.2 m (x × y × z) allows the voxels inside and outside the building to be distinguished for parts of the building on the same floor without semantic constraints (i.e., in the rough extraction process, as shown in Figure 13a). However, the rough extraction results cannot distinguish voxels that lie between the interior and exterior walls because the voxels are too small to reject the cells between the inner and outer walls.

When the voxel size is increased to 0.8 m × 0.8 m × 0.6 m, the rough extraction results can successfully reject the voxels between the inner and outer walls (as shown in Figure 13b).

Another possibility involves adding a semantic constraint to the results obtained using the smaller voxels. By setting the semantic constraint condition that vertical boundary cells are not outer walls, the expected result is achieved (as shown in Figure 13c).

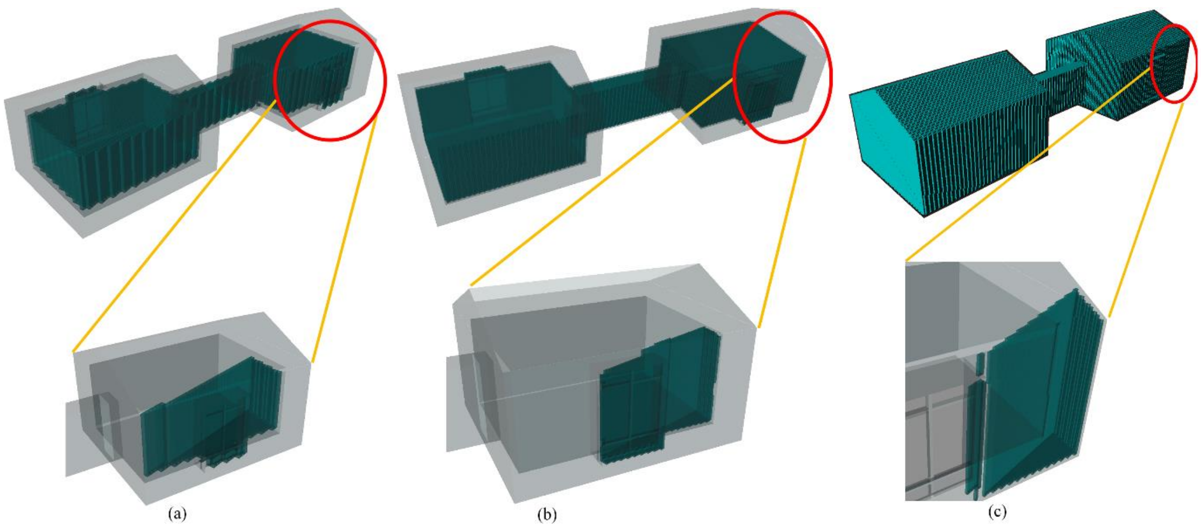


Figure 13 Rough extraction results.

As mentioned above, the extracted results offer different levels of precision depending on the size of the voxels. This might be useful for producing different types of indoor space data. For example, with the voxel size set to a scale of 1 m, the extracted indoor space results could be used for indoor pedestrian navigation (as shown in Figure 14a).

Alternatively, for a smaller voxel size, on the order of centimeters, the results could be used for the navigation of UAVs or robots (as shown in Figure 14b).

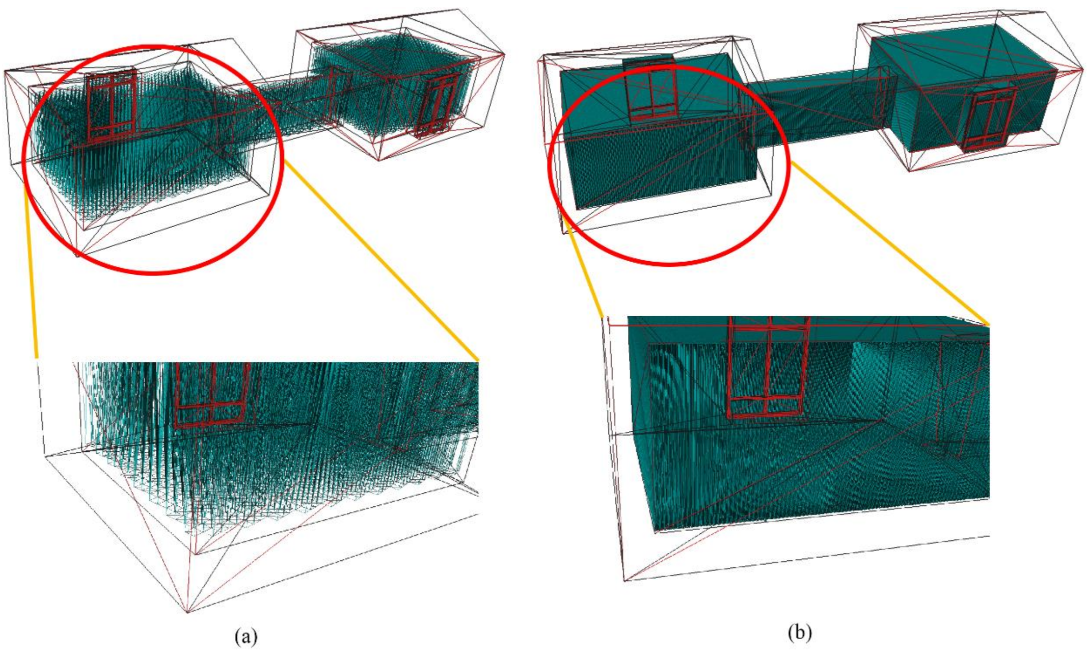


Figure 14 The accuracy of indoor space extraction.

4.2 Efficiency analysis

In this study, three LoD4 building models (as shown in Figure 15) were prepared to analyze the efficiency of indoor space extraction. Models 1 and 2 are hypothetical, whereas model 3 represents the building that houses the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing at Wuhan University. The detailed geometric information is shown in Table 2. In particular, the numbers of indoor spaces, which are the true values for the experiments, were obtained manually. The numbers of vertices, triangles, and spaces are the test factors for efficiency.

Table 2 The detailed geometric information of the experimental data.

Name	Number of Vertices	Number of Triangles	Number of General Spaces	Number of Transition Spaces	Number of Indoor Spaces
Model 1	359	570	3	4	7
Model 2	4497	6942	12	41	53
Model 3	147018	211177	114	261	375

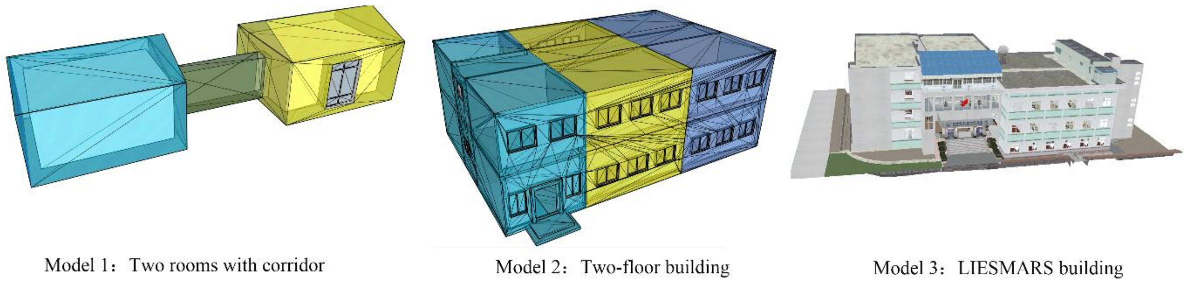


Figure 15 Experimental data.

The criteria for selecting the voxel size are specific to particular applications and users. In the experiments presented in this paper, the voxel size was set to 0.3 m × 0.3 m × 0.2 m (x × y × z) because this size can be used to distinguish objects that are relevant for pedestrians and navigation. For the voxelization algorithm described above in Section 3.1, the efficiencies of voxelization and indoor space extraction for the three building models are shown in Table 3.

Table 3 Voxelisation times of experimental data.

Model Name	Voxelisation Time (s)	Extraction Time (s)
Model 1	0.964	0.7
Model 2	1.864	0.936
Model 3	17.489	7.313

Based on the statistics of the input data and the time consumption data presented above, the line charts shown in Figure 16 offer a more visual representation that reveals that the efficiency of the algorithm is related to the structural complexity of the building, as indicated by the positive correlation between them.

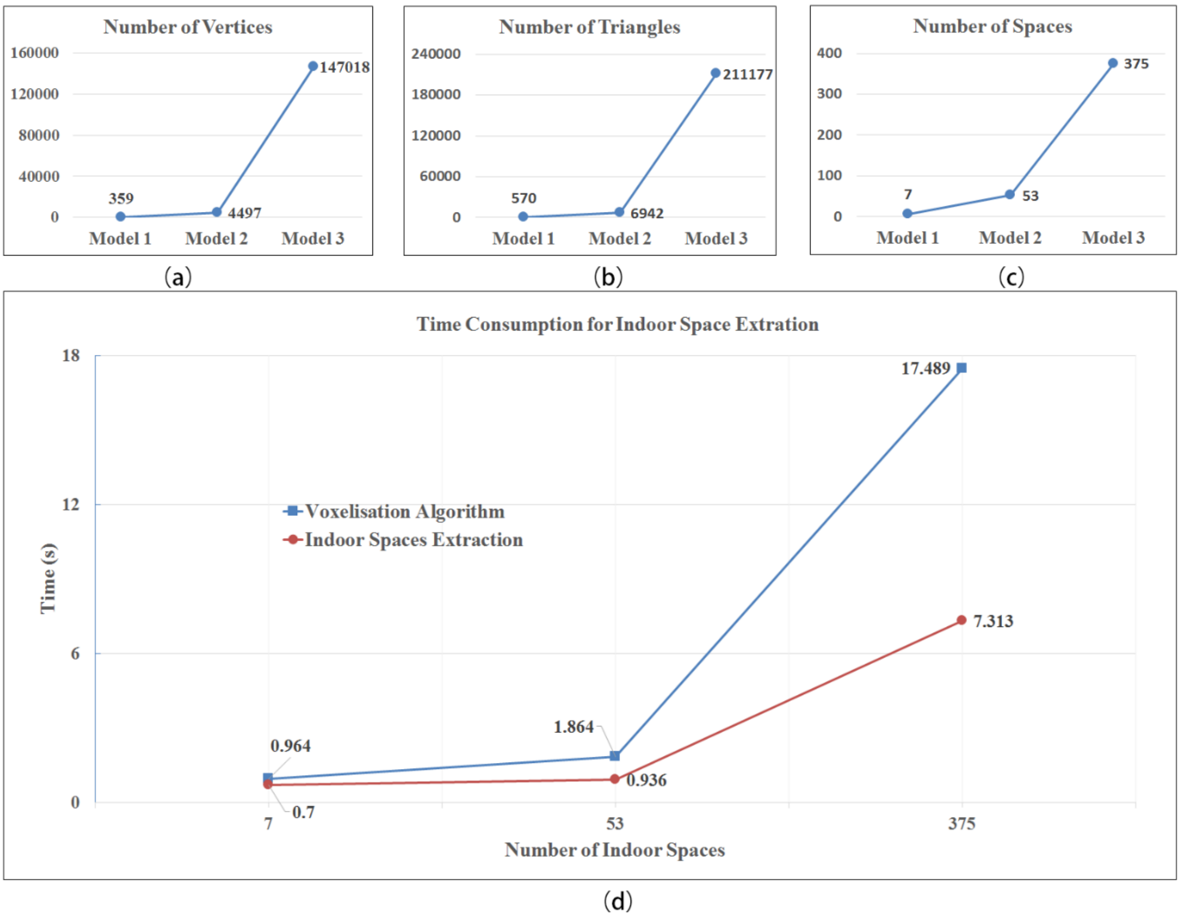


Figure 16 Experimental statistics.

In addition, a summary comparison of the efficiencies during the two main steps of the algorithm when applied to the experimental data is given in Figure 17. In particular, the processing speed during voxelization grows monotonically as the number of triangles increases, whereas the processing speed during extraction first increases and then declines. With respect to the computation time required for the entire algorithm, the proportion of time spent on voxelization is greater than that spent on extraction and shows an upward trend with increasing model complexity. However, compared with the growth in the spatial complexity of the input data, the rate of increase and the volatility of the method are negligible. High-performance computing techniques, such as parallel computing, can also be easily used to improve the computational efficiency by taking advantage of the unified organization offered by the approach. Consequently, the proposed algorithm is well suited for application to large-scale indoor scenes.

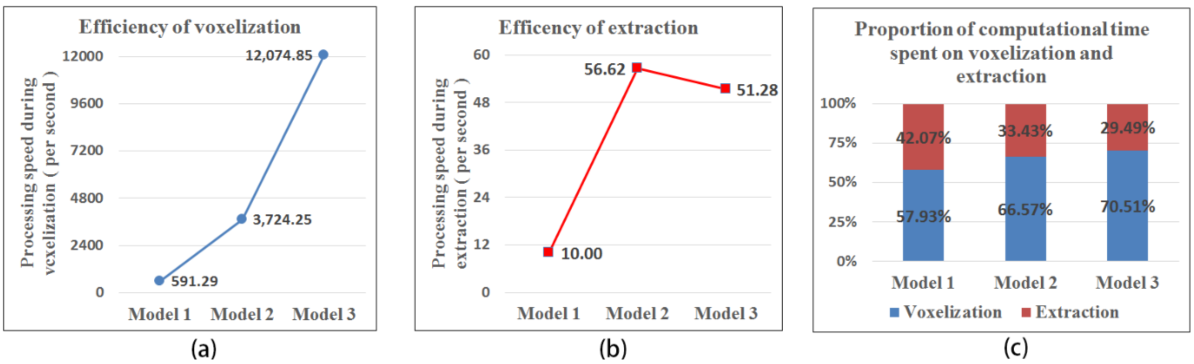


Figure 17 Efficiency comparisons.

4.3 Analysis of results

To validate the correctness of the proposed approach, we use model 3, described in Section 4.1, as an example (as shown in Figure 18). The model 3 is a typical complex 3D building model with interior structures that is based on a triangular mesh and is compliant with the CityGML standard.

The main section of this building contains various elements, such as gates, lobbies, corridors, staircases, doors, stents for show windows and suspended ceilings, forming 375 indoor spaces. In particular, the two-floor lobby space in the second layer is a typical free multi-floor space with a fuzzy boundary between the layers, and the left staircase is a classic multi-floor space with complex boundaries between the layers, as shown in Figure 18b.

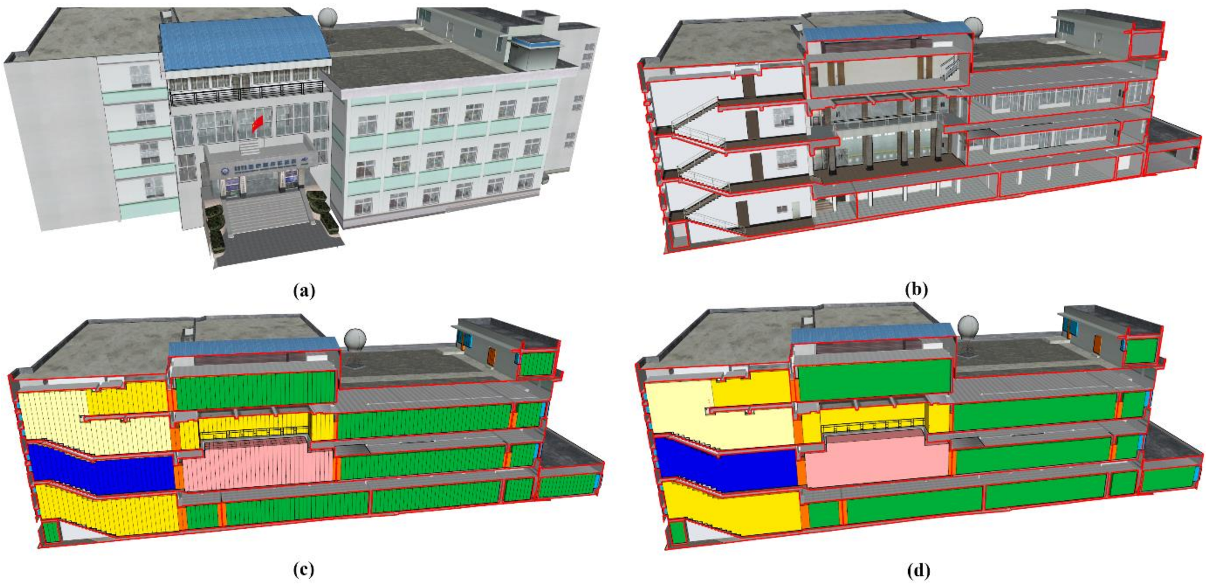


Figure 18 Detailed indoor information regarding the main section of model 3.

In applications such as indoor navigation, location awareness, and emergency response, a set of meaningful indoor spaces typically serves as the basic elements used to represent free multi-floor structures in an indoor environment. Therefore, we use a coherent geometric border to represent the lobby in the second layer of model 3, which is a typical free multi-floor space and a key example from which to perceive the complexity of the indoor environment. Figure 18c provides a cross-sectional view of the accurate extraction result with voxels, and Figure 18d presents a profile of the result with correct enclosed boundaries. The cost of updating the semantics of each indoor voxel and the geometric reconstruction increases exponentially with increasing complexity of the indoor environment. Furthermore, the free multi-floor indoor spaces between floors yield complete and correct enclosed boundaries; consequently, the 3D indoor spaces are divided by semantics.

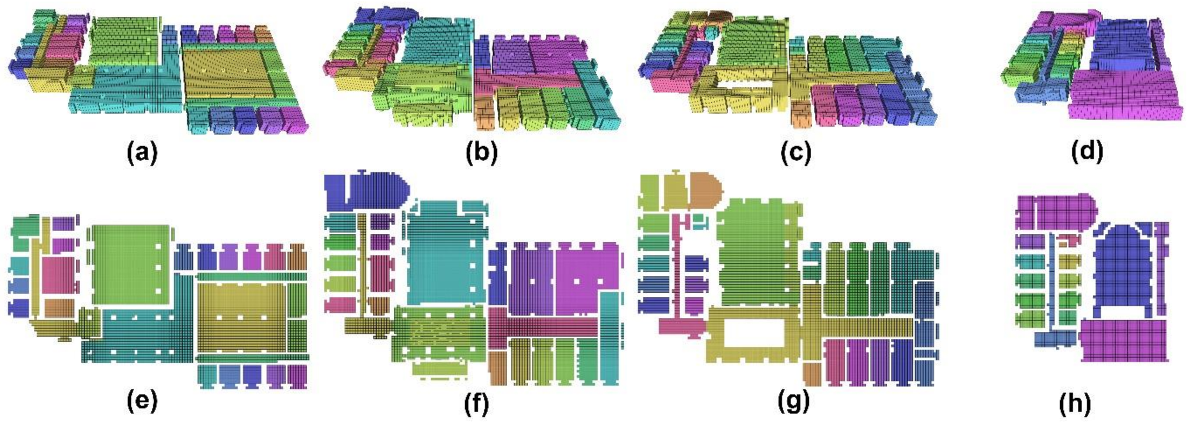
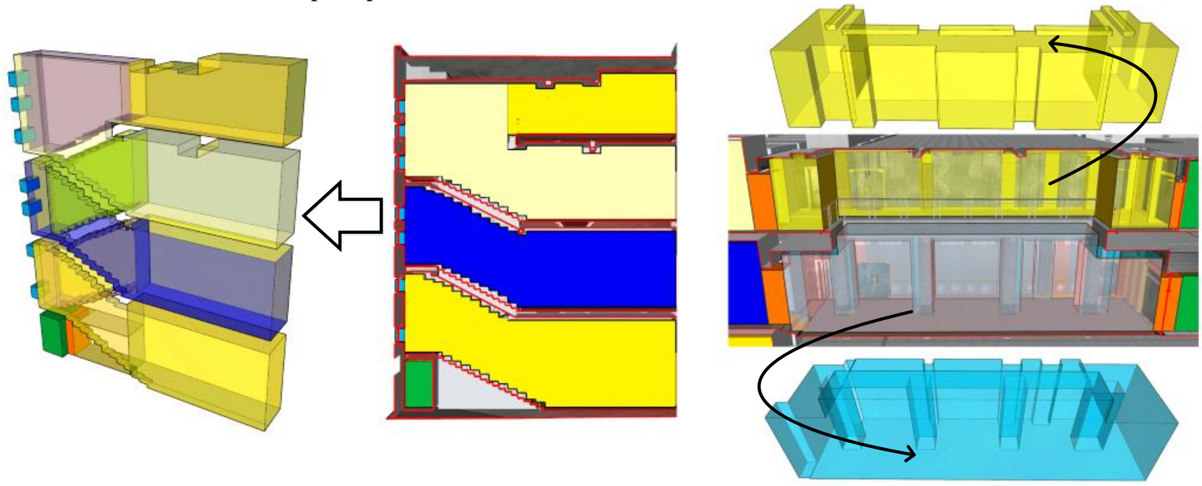


Figure 19 Extraction results for each floor.

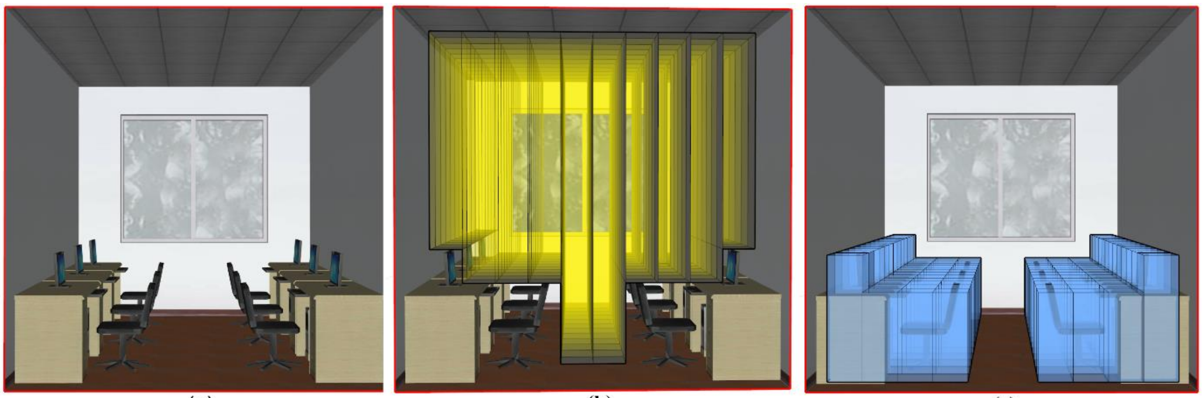
In Figure 19, the indoor spaces on each floor are distinguished by different colors; the graphics presented in the top row are viewed at a 45° angle, whereas those in the bottom row are plan forms. The extraction results show that the method proposed in this paper can effectively handle 3D model data on the same floor. The geometric representation of regular indoor spaces makes the production of horizontal networks for indoor navigation easy and accurate. Additionally, beyond these horizontal networks, indoor navigation analyses can further benefit from the extraction of staircases, open spaces, and obstacles.



(a) Staircase extraction results **(b) Lobby extraction results**

Figure 20 Extraction performance for free multi-floor indoor spaces.

A staircase is an important component of a building that connects multiple floors. As shown in Figure 20a, the multi-floor extraction results well preserve the volumetric features of the staircase in the model and can effectively support subsequent analysis for multi-floor positioning and navigation. In addition, the open lobby space contains several portals and crosses floors, such that it is difficult to correctly extract its boundaries. The extraction results for the two-floor lobby show that the extracted geometric features approximately represent the boundaries of the lobby, as shown in Figure 20b. Finally, ubiquitous obstacle spaces should be distinguished because of their non-navigability. As shown in Figure 21, a comparison of the extracted obstacle spaces for a room with furniture reveals a correct distinction between navigable spaces and non-navigable spaces. Figure 21a shows the profile of the room considered in the experiment, Figure 21b presents the navigable space in the room in the form of voxels, and Figure 21c shows the distribution of the obstacle spaces. The shaded boundaries were extracted via the accurate extraction process based on the rough extraction results and were created through the assignment of semantics to the voxels.



(a) **(b)** **(c)**

Figure 21 Extraction performance for obstacle spaces.

5. Conclusions and outlook

The correct modeling of free multi-floor indoor spaces provides fundamental indoor environmental information for indoor navigation and facility management. To overcome the inherent complexity and accuracy of the extraction of indoor spaces from 3D building models described using CityGML LoD4, this paper presents a novel algorithm for free multi-floor indoor space extraction. This algorithm utilizes geometric and semantic relationships to transform the complete results of indoor space extraction into voxels and non-overlapping boundaries. In conclusion, the proposed approach effectively reduces the computational burden of free multi-floor indoor space extraction and increases the reliability of the calculated results. Experimental results obtained for a typical 3D building model give evidence of the effectiveness and accuracy of the algorithm in classifying indoor spaces based on their semantic relationships. However, our solution also has limitations: the accuracy and reliability of this algorithm depend upon the quality of the input 3D building model, which is required to satisfy the strict specification of the CityGML LoD4 standard. Further indoor analysis, such as multi-floor indoor navigation and facility management, can be applied based on the indoor space extraction results. Thus, future work will include the consideration of non-navigable spaces (e.g., shafts for cables and pipes) bounded by non-navigable spaces (walls) to rebuild the indoor topological relationships and efficient indoor data organization to support dynamic indoor analyses for various purposes (e.g., for responses to fires, earthquakes and terrorist attacks and for facility management).

Acknowledgements

This paper was supported by the National Nature Science Foundation of China (No. 41571390, 41471320, and 41471332) and the National High Technology Research and Development Program of China (2015AA123901).

References:

Afyouni I, Cyril R, Christophe C (2012) Spatial models for context-aware indoor navigation systems: A survey. J Spat Inform Sci 1:85-123

Becker T, Nagel C, Kolbe TH (2008) A multilayered space-event model for navigation in indoor spaces. In: Lee J, Zlatanova S (eds) 3D Geo-information sciences. Springer, Heidelberg, Berlin, pp 61-77

Brown G, Nagel C, Zlatanova S, Kolbe TH (2012) Modelling 3D topographic space against indoor navigation requirements. In: Pouliot J, Daniel S, Hubert F, Zamyadi A (eds) Progress and new trends in 3D geoinformation sciences. Springer, Heidelberg, Berlin, pp 1-22

Geraerts R (2010) Planning short paths with clearance using explicit corridors. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), May 3-8, 2010. Anchorage, Alaska, pp 1997-2004

Girard G, Côté S, Zlatanova S, Barette Y, St-Pierre J, van Oosterom P (2011) Indoor pedestrian navigation using foot-mounted IMU and portable ultrasound range sensors. Sensors 11:7606-7624. doi: 10.3390/s110807606

Isikdag U, Zlatanova S, Underwood J (2013) A BIM-oriented model for supporting indoor navigation requirements. Comput Environ Urban Syst 41:112-123. doi: 10.1016/j.compenvurbsys.2013.05.001

Jensen CS, Lu H, Yang B (2010) Indoor-a new data management frontier. IEEE Data Eng Bull 33:12-17

Jones MW, Satherley R (2000) Voxelisation: Modelling for volume graphics. In: Girod B, Greiner G, Niemann H, Seidel H-P (eds) Vision modeling and visualization, November 22-24. Saarbrücken, Germany, pp 319-326

Kim J, Yoo S, Li K (2014) Integrating IndoorGML and CityGML for Indoor Space. In: Dieter P, Ki-Joune L (eds) Web and wireless geographical information systems. Springer, Heidelberg, Berlin, pp 184-196

Krūminaitė M, Zlatanova S (2014) Indoor space subdivision for indoor navigation. In: The Sixth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, November 04-07, 2014. Dallas/Fort Worth, TX, pp 25-31

Lamarche F, Donikian S (2004) Crowd of virtual humans: A new approach for real time navigation in complex and structured environments. Comput Graph Forum 23:509-518. doi: 10.1111/j.1467-8659.2004.00782.x

Lee J (2004) A spatial access-oriented implementation of a 3-D GIS topological data model for Urban Entities. GeoInformatica 8:237-264. doi: 10.1023/b:gein.0000034820.93914.d0

Li K (2008) Indoor space: A new notion of space. In: Bertolotto M, Ray C, Li X (eds) Web and wireless geographical information systems. Springer, Berlin, pp 1-3

Li X, Claramunt C, Ray C (2010) A grid graph-based model for the analysis of 2D indoor spaces. Comput Environ Urban Syst 34:532-540. doi: 10.1016/j.compenvurbsys.2010.07.006

Liu L, Zlatanova S (2011) A "door-to-door" path-finding approach for indoor navigation. In: Gi4DM 2011: GeoInformation for Disaster Management, May 3-8. Antalya, Turkey

Liu L, Zlatanova S (2012) A semantic data model for indoor navigation. In: The Fourth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, November 07-09, 2012. Redondo Beach, CA, pp 1-8

Lorenz B, Ohlbach HJ, Stoffel E (2006) A hybrid spatial model for representing indoor environments. In: James DC, Taro T (eds) Web and wireless geographical information systems. Springer, Heidelberg, Berlin, pp 102-112

Meagher D (1982) Geometric modeling using octree encoding. Comp Grap Ima Proc 19:129-147. doi:

10.1016/0146-664x(82)90104-6
Meijers M, Zlatanova S, Pfeifer N (2005) 3D geoinformation indoors: structuring for evacuation. In: Proceedings of Next generation 3D city models, June 21-22, 2005. Bonn, Germany, pp 21-22
OGC IndoorGML (2014) Indoor geography markup language (IndoorGML) encoding standard. http://indoorgml.net/ Accessed 3 December 2014
Oomes S, Snoeren P, Dijkstra T (1997) 3D shape representation: Transforming polygons into voxels. In: Romeny BT, Florack L, Koenderink J, et al (eds) Scale-space theory in computer vision. Springer, Heidelberg, Berlin, pp 349-352
Schaap J, Zlatanova S, van Oosterom PJM (2011) Towards a 3D geo-data model to support pedestrian routing in multimodal public transport travel advices. In: Rumor M (eds) Urban and regional data management. CRC Press, Boca Raton, pp 63-78
Stoffel E, Lorenz B, Ohlbach HJ (2007) Towards a semantic spatial model for pedestrian indoor navigation. In: Hainaut JL, Rundensteiner EA, Kirchberg M, Bertolotto M, Brochhausen M, Chen P, et al. (eds) Advances in conceptual modeling - foundations and applications. Springer, Heidelberg, Berlin, pp 328-37
Thill J-C, Dao THD, Zhou Y (2011) Traveling in the three-dimensional city: Applications in route planning, accessibility assessment, location analysis and beyond. J Transp Geogr 19:405-421. doi: 10.1016/j.jtrangeo.2010.11.007
Wallgrün JO (2005) Autonomous construction of hierarchical voronoi-based route graph representations. In: Freksa C, Knauff M, Krieg-Brückner B, Nebel B, Barkowsky T (eds) Spatial cognition IV, Reasoning, Action, Interaction. Springer, Berlin, pp 413-433
Xie X, Zhu Q, Du Z, Xu W, Zhang Y (2013) A semantics-constrained profiling approach to complex 3D city models. Comput Environ Urban Syst 41: 309-317. doi: 10.1016/j.compenvurbsys.2012.07.003
Xiong Q, Zhu Q, Zlatanova S, Huang L, Zhou Y, Du Z (2013) Multi-dimensional indoor location information model. In: Acquisition and Modelling of Indoor and Enclosed Environments, December 11-13. Cape Town, South Africa
Yuan W, Schneider M (2010) iNav: An indoor navigation model supporting length-dependent optimal routing. In: Painho M, Santos MY, Pundt H (eds) Geospatial Thinking. Springer, Heidelberg, Berlin, pp 299-313
Zlatanova S, Liu L, Sithole G (2013a) A conceptual framework of space subdivision for indoor navigation. In: The Fifth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, November 05-08, 2013. Orlando, FL, pp 37-41
Zlatanova S, Sithole G, Nakagawa M, Zhu Q (2013b) Problems in indoor mapping and modelling. In: Acquisition and Modelling of Indoor and Enclosed Environments 2013, December 11-13, 2013. Cape Town, South Africa