

## Research Article

# Automatic Generation of 3D Building Models Based on Line Segment Vectorization

Qiao Wen<sup>1</sup> and Rui-Guang Zhu<sup>2,3</sup> 

<sup>1</sup>Department of Civil Engineering, Tsinghua University, Beijing 100084, China

<sup>2</sup>Key Laboratory of Green Construction and Intelligent Maintenance for Civil Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China

<sup>3</sup>School of Civil Engineering and Mechanics, YanShan University, Qinhuangdao 066004, China

Correspondence should be addressed to Rui-Guang Zhu; [zrg179@163.com](mailto:zrg179@163.com)

Received 30 July 2020; Revised 15 September 2020; Accepted 18 September 2020; Published 10 October 2020

Academic Editor: Łukasz Jankowski

Copyright © 2020 Qiao Wen and Rui-Guang Zhu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Operations and maintenance (O&M) management for existing buildings is of high importance since it consumes the most cost during buildings' lifecycle. Its effectiveness could be significantly improved through the systematic use of building information modeling (BIM). However, BIM relies on full-fledged digital models, which, for most buildings, are not available. This paper introduces a recognition algorithm aiming at the automatic generation of 3D building models from 2D drawings. The algorithm is able to generate separated wall segment 3D models with their topology relations. The algorithm is implemented and tested by several real projects. The results are very promising and show that the proposed algorithm could be a key component of future digital toolkits for O&M management.

## 1. Introduction

Operation and maintenance management for existing buildings relies on complex building information to perform different tasks. Although building information models (BIMs) provide an integrated information repository, currently, most existing buildings were still built with 2D drawings [1]. The majority of these buildings that were relatively newly constructed within the last several decades usually have 2D drawings in electronic formats such as CAD files [2]. In general, the CAD drawing is a reliable choice for analyzing geometry and semantic information of existing buildings [3]. However, for most buildings, existing 3D models are not available [4]. For example, in Europe, more than 80% of the residential buildings were built before 1990 [5] and do not have BIM which means that the building owners or facility managers cannot benefit from using BIM during the O&M phase [1]. Manually reconstructing BIM based on these 2D drawings has been proven to be not only very labor-intensive [1], but also time-consuming [6, 7].

Because of poor information availability and accessibility, O&M of these existing buildings is not efficient and has significant room for improvement [8].

While BIM tools are undoubtedly beneficial to building design practices and are now commonplace in new building design processes, they are still underexploited in existing buildings [4]. Such advanced design approaches require full-fledged 3D digital models in order to realize their full potential [4]. Therefore, one critical short-term research challenge is to devise effective and reliable methods and tools to reconstruct 3D digital models of existing buildings [4].

Tombre [9, 10] firstly discussed the challenges of automatic recognition of architectural drawings. Ah-Soon and Tombre [11] analyzed the specificities of architectural drawings and recognized objects from the scanned drawings. They proposed a network-based recognition method [12, 13] to identify symbols, following an idea proposed by Messmer and Bunke [14]. Park and Kwon [15] recognized the main walls of apartments in floor plan architectural drawings using dimension extension lines. Some systems were

developed to recognize the different elements of the drawing [16, 17]: an architectural structure drawing interpretation system which used geometric features of the architectural drawing objects as the recognition criteria [16] and a novel complete system for automated floor plan analysis, which introduced novel preprocessing methods, e.g., the differentiation between thick, medium, and thin lines and the removal of components outside the convex hull of the outer walls [17]. Some other research proposed different models [18], e.g., a hierarchical SINEHIR (self-incremental axis-net-based hierarchical recognition) model to recognize structural objects (e.g., columns, beams, slabs, walls, and holes) from construction structural drawings [18]. The most common research focused on recognition methods or algorithms [19–22]: statistical grouping method for partitioning line drawings into shapes, those shapes which represent meaningful parts of the symbols that constitute the line drawings [19]; a new rotation invariant corner detection method for architectural line drawing images [20]; a generic method for floor plan analysis and interpretation [21], in which basic building blocks, i.e., walls, doors, and windows are first detected using a statistical patch-based segmentation approach, and structural pattern recognition techniques are second applied to further locate the main entities, i.e., rooms of the building; and a novel approach to automatically recognize architectural and electrical symbols [22].

Besides the above studies which focused on recognition, some further research investigated extracting information from 2D CAD drawings to generate 3D models for the purpose of O&M [1, 3, 23–25]. According to the review [26], which gave an overview on techniques used to generate 3D building models, there is currently no general answer about the best approach to generate 3D models of existing buildings [26]. The review also highlighted the breadth of the area and the numerous techniques that aim at the creation of 3D models of existing buildings [5]: photogrammetry [27], laser scanning [28], tagging, and use of preexisting information like sketches and tape measurers. Despite this significant offer, no approach seems to allow for the generation of exploitable 3D building models that would truthfully represent the geometry, topology, and semantics of the buildings [4]. For example, point clouds could often have noisy data and miss data such as texture in addition to their time-consuming and tedious process [29].

The aim of this paper is to develop, implement, and test a line segment vectorization-based algorithm that can generate 3D building models with both geometry and topology. The algorithm relies on easily accessible data: CAD drawings are usually available for existing buildings [2, 3]. At the same time, the algorithm can be easily implemented at reasonable costs.

The remainder of this paper is organized as follows: Section 2 gives the idea of how this algorithm comes. Section 3 explains all the principles used in the algorithm. Section 4 gives an application of the algorithm. Section 5 gives the conclusion and future work.

## 2. Algorithm Overview

The drawings of regular layout buildings are investigated by the algorithm. These types of drawings are deconstructed to obtain the minimum unit, as shown in Figure 1.

Step in Figure 1 is to deconstruct the walls in the drawings into 13 parts, as discussed in Section 2.1. Step is to further deconstruct the 13 parts into 4 types of right angles, which will be detailed in Section 2.2. Step is to deconstruct the right angles, which is discussed in Section 2.3. The final result is to obtain the horizontal and vertical line segments, which are the minimum processing units of the algorithm.

### 2.1. Deconstruction Drawings of Regular Layout Buildings.

A shear-wall structure can effectively resist the horizontal load, which is a type of load that should be frequently considered in structural design. Generally, a shear-wall structure building has a regular shape, and the entire building is flat and vertical, which is a typical regular layout building. This paper considers a shear-wall structure as an example to illustrate the algorithm.

Figure 2 displays part of a standard layer of an actual shear-wall building located in Harbin. The building is an 11-story shear-wall structure. For simplicity, only part of the walls and the necessary dimensions are retained: the outer part is the dimensions, and the inner part is the walls. The rightmost curve consists of a broken line, which indicates that Figure 2 comprises only part of a drawing.

By analysis of a large number of shear-wall building drawings, the walls in the shear-wall drawings can be deconstructed into 4 parts: wall-corner, T-wall, cross-wall, and wall-end. Three of the units can be further deconstructed depending on the orientation. The wall-corner is further deconstructed into 4 parts: lower left wall-corner, lower right wall-corner, upper left wall-corner, and upper right wall-corner. The T-wall is further deconstructed into 4 parts: left T-wall, right T-wall, up T-wall, and down T-wall. The wall-end is further deconstructed into 4 parts: left wall-end, right wall-end, up wall-end, and down wall-end. The cross-wall cannot be further deconstructed due to symmetry. The deconstruction processes are shown in Figure 3.

After the two-step deconstruction shown in Figure 3, 13 types of parts of the shear-wall drawings are obtained. Their graphic representations, which correspond to those in Figure 3, are shown in Figure 4. The 13 types of deconstruction parts shown in Figure 4 are named by their orientations.

According to the deconstruction methods, the walls of any shear-wall drawings comprise a number of the 13 parts. These 13 parts will be referred to as “deconstruction parts” in the following sections.

### 2.2. Further Deconstruction of the 13 Deconstruction Parts.

In Figure 4, the 13 deconstruction parts can be considered to be composed of right angles of 4 different orientations. Thus, the 13 deconstruction parts can be further deconstructed

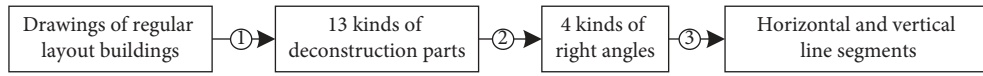


FIGURE 1: Deconstruction of drawings of regular layout buildings.

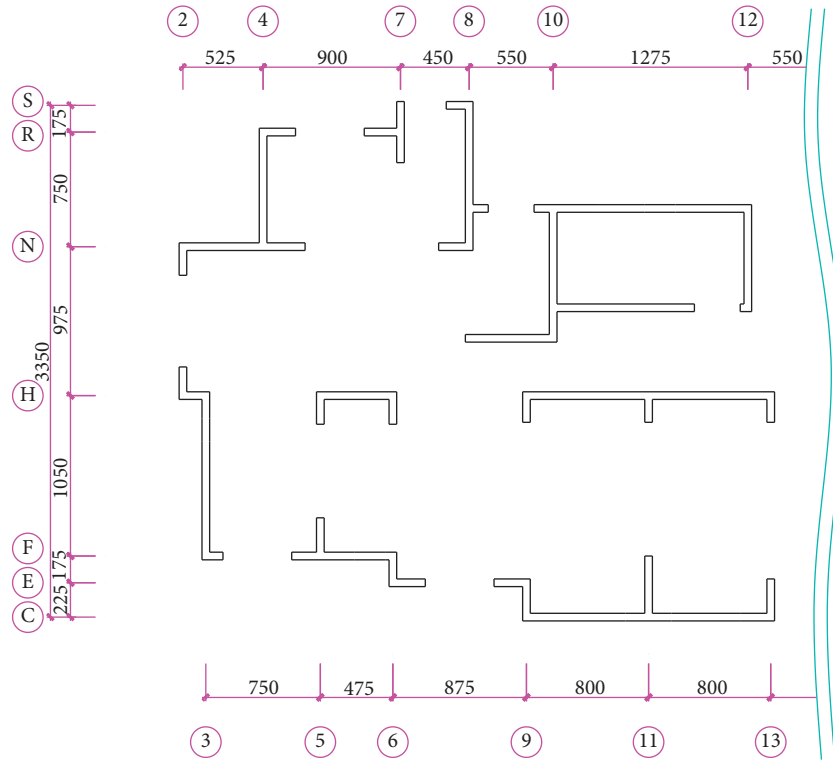


FIGURE 2: Real shear-wall building drawing example.

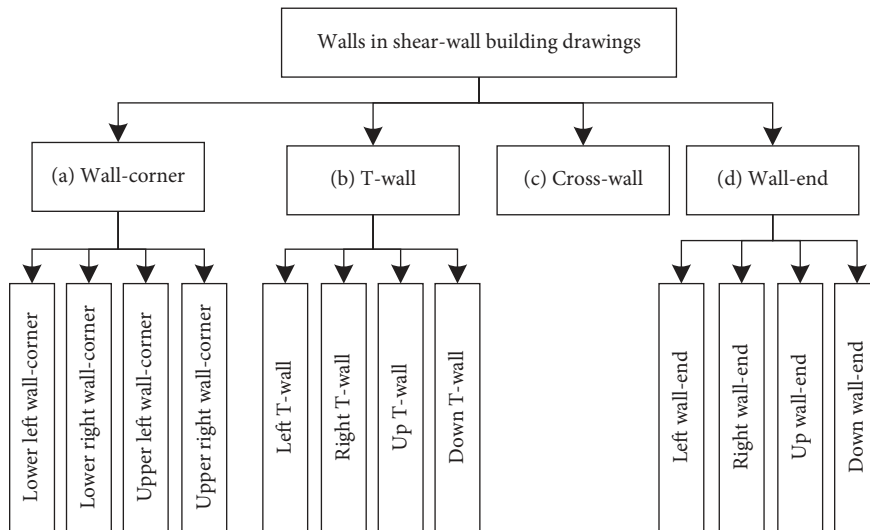


FIGURE 3: Deconstruction processes of walls in shear-wall building drawings.

into 4 types of right angles. The deconstruction process is shown in Figure 5.

After the further deconstruction shown in Figure 5, a graphic representation of the 4 types of right angles, which correspond to those in Figure 5, is shown in Figure 6. In

Figure 6, the 4 right angles are named according to their position in a rectangle.

In Figure 6, only 4 right angles with 4 different orientations are considered. In most cases, the walls in a shear-wall drawing only contain these 4 types of right angles. These

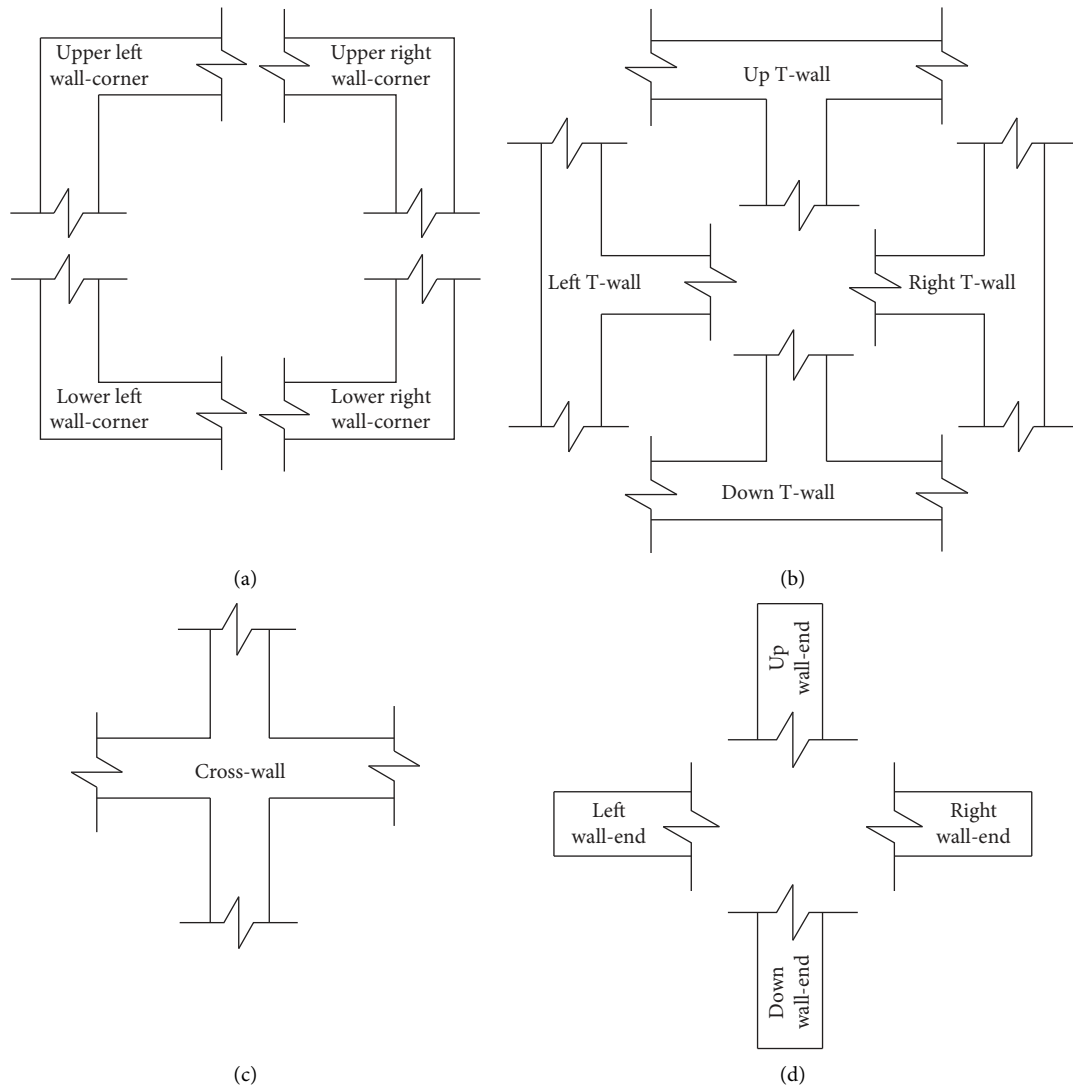


FIGURE 4: Graphic representations of 13 deconstruction parts: (a) wall-corner, (b) T-wall, (c) cross-wall, and (d) wall-end.

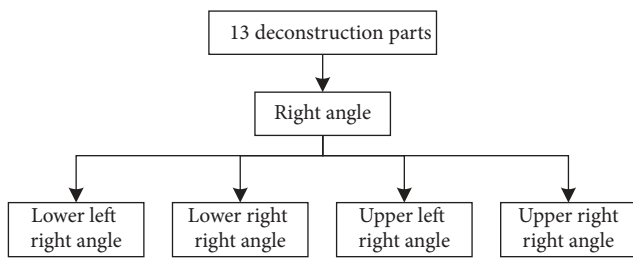


FIGURE 5: Deconstruction processes of the 13 deconstruction parts.

4 right angles are distinguished from each other and can be used to reversely recognize the 13 deconstruction parts shown in Figure 4.

2.3. *Right Angle Deconstruction.* The 4 right angles in Figure 6 can be further deconstructed. Each of these 4 right angles is composed of 2 line segments: a horizontal line segment and a vertical line segment. This deconstruction step is shown in Figure 7.

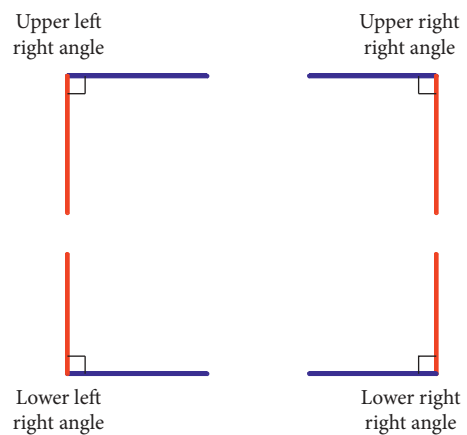


FIGURE 6: Right angle classifications.

Generally, the walls in shear-wall drawings, which are similar to those shown in Figure 2, contain only horizontal and vertical line segments. At this point, the walls in the

shear-wall drawings, which are represented in Figure 2, are deconstructed into the minimum units: line segments.

### 3. Algorithm Principles

According to the description in Section 2, the algorithm starts with the minimum unit and performs the line segment vectorization as step . In step , the combination of vectorized line segments and their endpoints are utilized to distinguish the 4 types of right angles. Each combination corresponds to a type of right angle. The objective of step is to sort the 4 types of right angles according to different rules. The objective of step is to recognize the 13 deconstruction parts by the orientation features of the right angles. The steps of the algorithm are shown in Figure 8.

The steps shown in Figure 8 have the opposite direction of those shown in Figure 1. By reverse operation, the recognition of drawings of regular layout buildings can be achieved. The principles applied in each step are described in Sections 3.1, 3.2, 3.3, and 3.4.

**3.1. Line Segment Vectorization.** As shown in Figure 6, the 4 types of right angles only consist of horizontal and vertical line segments. Each line segment contains 2 endpoints, and fixing the relative positions of these 2 endpoints is the base of the algorithm. The method is to vectorize the horizontal and vertical line segments according to certain rules to ensure that the relative positions of the 2 endpoints of each line segment are fixed: the first endpoint is the starting point, and the second endpoint is the ending point. A set of feasible rules is shown in Table 1, and other rules similar to these rules may also be feasible.

According to probability theory, some line segments do not comply with the rules in Table 1. Half of the horizontal line segments have starting points on the right and ending points on the left; half of the vertical line segments have starting points at the top and ending points at the bottom. For the horizontal and vertical line segments that do not comply with the rules shown in Table 1, exchange the relative positions of the starting points and ending points, as shown in Figures 9 and 10, respectively. These line segments can comply with the rules shown in Table 1; they remain unchanged.

In Figures 9 and 10, circles are used to represent the endpoints of line segments. Circles with a cross inside serve as starting points and circles with blank space inside serve as ending points. This method is used to distinguish the starting points and ending points in the following sections. The original line segments (red) in both Figures 9 and 10 do not comply with the vectorization rules in Table 1 and need to exchange the relative positions of the starting points and ending points.

After this step, any line segment in the drawings has 2 endpoints with fixed relative positions. This feature is the base of the algorithm, which is simple and universal: any line segment has 2 endpoints, which can be vectorized without any exceptions.

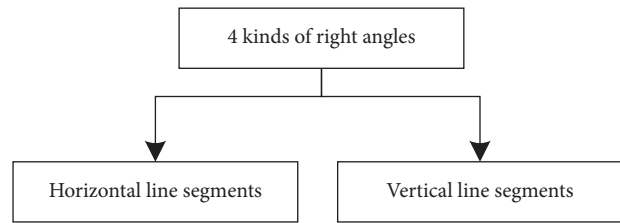


FIGURE 7: Deconstruction processes of 4 types of right angles.

**3.2. Right Angle Recognition.** After the vectorization of the line segments in Section 3.1, the 4 types of right angles in Figure 6 can be distinguished from each other. The principles and corresponding combinations are shown in Table 2, in which HLS = horizontal line segment, VLS = vertical line segment, SP = starting point, and EP = ending point.

The principles shown in Table 2 can be visually represented by graphs, as shown in Figure 11. Similar to Figures 9 and 10, circles are used to represent the endpoints of line segments. Circles with a cross inside serve as starting points, and circles with blank space inside serve as ending points. In Figure 11, consider the lower left right angle as an example, the 2 starting points of a horizontal line segment and a vertical line segment coincide to form a lower left right angle. The remaining 3 combinations in Table 2 form 3 right angles in a similar manner.

Four principles and 4 corresponding combinations are shown in Table 2 and Figure 11. Since the horizontal and vertical line segments are mutually distinguishable, and the starting and ending points of the 2 types of line segments are also mutually distinguishable, 4 different combinations exist, as shown in Table 2. These 4 combinations correspond to the 4 right angles shown in Figure 11. Thus, the 4 types of right angles can be distinguished from each other.

**3.3. Right Angle Sorting.** Right angle sorting is an indispensable step of the algorithm to recognize wall-corners. Each right angle has only one vertex; thus, the vertex can represent a right angle when sorting. Each type of right angle must be separately sorted, and a set of feasible sorting rules is shown in Table 3.

Consider the real project shown in Figure 2 as an example. After sorting, the sequence number of the lower left right angles is shown in Figure 12. The footer is the ordinal number after sorting according to the lower left right angle rule shown in Table 3. The red right angles belong to wall-corners, the blue right angles belong to wall-ends, and the pink right angles belong to T-walls.

Each type of right angle will form a sequence for looping before deconstruction parts recognition. After sorting, the following 2 goals can be achieved, as shown in Table 4.

Consider Figure 12, for example, right angle  $P_1$ , which is located on the most lower left outer contour of the walls of the entire shear-wall drawing, will be the initial element of the looping of lower left right angle sequence. During the looping,  $P_1, P_4, P_{11}$ , and  $P_{20}$  will be the looping elements, whereas  $P_3, P_5, P_{13}$ , and  $P_{22}$ , respectively, are the corresponding target elements. In Figure 12,  $P_1, P_4, P_{11}$ , and  $P_{20}$

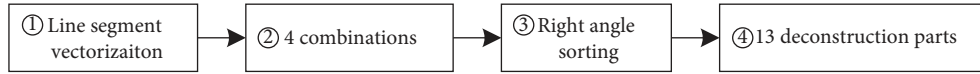


FIGURE 8: Algorithm steps.

TABLE 1: Set of vectorization rules for line segments.

Category	Vectorization rules
Horizontal	Horizontal line segment has the starting point on the left and ending point on the right
Vertical	Vertical line segment has the starting point at the bottom and ending point at the top

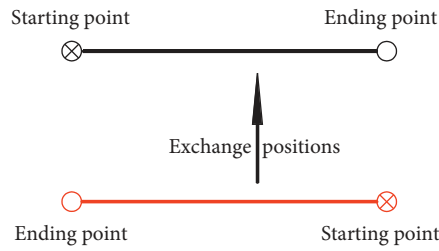


FIGURE 9: Vectorization of horizontal line segments.

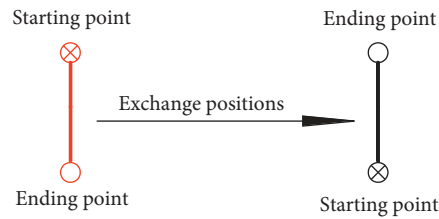


FIGURE 10: Vectorization of vertical line segments.

TABLE 2: Right angle recognition principles.

Category	Principle description	Combination
Lower left right angle	An HLS's SP coincides with a VLS's SP	(HLS' SP, VLS' SP)
Upper left right angle	An HLS's SP coincides with a VLS's EP	(HLS' SP, VLS' EP)
Lower right right angle	An HLS's EP coincides with a VLS's SP	(HLS' EP, VLS' SP)
Upper right right angle	An HLS's EP coincides with a VLS's EP	(HLS' EP, VLS' EP)

TABLE 3: Set of right at angle sorting rules.

Category	Sorting rule
Lower left	By X coordinates ascending; when X is the same, by Y coordinates ascending.
Lower right	By X coordinates descending; when X is the same, by Y coordinates ascending
Upper left	By Y coordinates descending; when Y is the same, by X coordinates ascending
Upper right	By Y coordinates descending; when Y is the same, by X coordinates descending

are on the lower left side of  $P_3, P_5, P_{13}$ , and  $P_{22}$ , respectively. Thus, during the looping of lower left right angle sequence, the same judgment condition can be applied to recognize the 4 wall-corners, which are formed by the red right angles shown in Figure 12.

**3.4. Deconstruction Parts Recognition.** After recognition and sorting of the 4 types of right angles, the 13 different deconstruction parts shown in Figure 4 can be recognized using the combination and relative position relation between right angles.

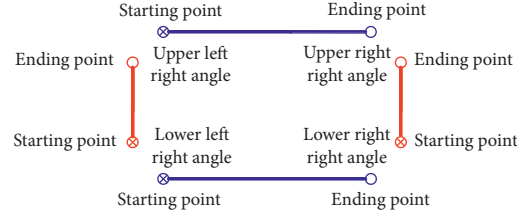


FIGURE 11: Distinguishing 4 kinds of right angles.

TABLE 4: Purposes of right angle sorting.

No.	Purpose
1	The right angles on the outer contours of the walls in the drawings are placed at the first position of the sequence of its types
2	For any wall-corner in the drawing, ensure that the external right angle is in front of the internal right angle

**3.4.1. Wall-Corner Recognition.** Two right angles of the same type form a wall-corner; each wall-corner has a similar recognition algorithm. Consider the lower left wall-corner as an example. Two lower left right angles form a lower left wall-corner, as shown in Figure 13. The relative positions of these 2 lower left right angles are fixed. The relative position rules that they must satisfy are shown in Table 5, in which both rules must be simultaneously satisfied.

Figure 13 shows the relative positions of the 2 right angles of a lower left wall-corner. The  $P_N$  is the external right angle, and  $N$  is its ordinal number after sorting.  $P_{N+i}$  is the internal right angle, and  $N+i$  is its ordinal number after sorting. Both  $N$  and  $i$  are positive integers, and  $N < N+i$ , which indicates that  $P_N$  is in front of  $P_{N+i}$  after sorting. Consider Figure 12 as an example, the corresponding  $(N, N+i)$  of the 4 lower left wall-corners (formed by red right angles) are (1, 3), (4, 5), (11, 13), and (20, 22).

After the lower left right angle sorting of Section 3.3, in Figure 13, the right angle  $P_N$  is in front of  $P_{N+i}$  in the sequence. When these 2 right angles form a lower left wall-corner, the mathematical expression of the relative position rules shown in Table 5 is shown in equations (1) to (4). WallThickness represents the wall thickness in millimeters, and the wall thickness in the shear-wall drawings are generally 100, 150, 200, 250, and 300. A thickness of more than 300 is rare, and all walls have a modulus of 50.  $D$  is the deviation range, which is a small positive number that must satisfy  $D \ll \text{WallThickness}$ ; for example,  $D=5$ . Both  $D$  and WallThickness have the same meanings in the following sections.

$$P_{N+i}.X - P_N.X > D > 0, \quad (1)$$

$$P_{N+i}.Y - P_N.Y > D > 0, \quad (2)$$

$$|P_{N+i}.X - P_N.X - \text{WallThickness}| \leq D, \quad (3)$$

$$|P_{N+i}.Y - P_N.Y - \text{WallThickness}| \leq D. \quad (4)$$

As expressed in equation (1), in the  $X$  direction,  $P_{N+i}$  locates in the positive direction of  $P_N$ . As expressed in

equation (2), in the  $Y$  direction,  $P_{N+i}$  locates in the positive direction of  $P_N$ . These 2 equations guarantee that  $P_{N+i}$  locates on the upper right side of  $P_N$ . According to equation (3), in the  $X$  direction,  $P_{N+i}$  locates in the positive direction of  $P_N$ , and the distance between them is exactly a wall thickness. As expressed in equation (4), in the  $Y$  direction,  $P_{N+i}$  locates in the positive direction of  $P_N$ , and the distance between them is exactly a wall thickness. These 4 equations ensure accurate recognition of the lower left wall-corner after sorting the lower left right angles in Section 3.3.

The recognitions of the other 3 wall-corners have similar recognition methods. The recognition algorithms of these 3 wall-corners are shown in Tables 6–8, respectively.

The interpretations of the mathematical expressions contained in Tables 6–8 are similar to those of equations (1) to (4) of the lower left wall-corner but are not separately explained.

**3.4.2. Wall-End Recognition.** Each wall-end contains 2 right angles of different types, and each wall-end has a similar recognition algorithm. Consider the left wall-end as an example, a lower left right angle and an upper left right angle form a left wall-end. The relative positions of these 2 right angles are fixed, and the relative position rules that they must satisfy are shown in Table 9, among which all 3 rules must be simultaneously satisfied.

Figure 14 shows the relative position of 2 right angles that form the left wall-end. The  $P_{LL}$  is the lower left right angle;  $P_{UL}$  represents the upper left right angle. The 2 right angles are not of the same type.

In Figure 14, when these 2 right angles form the left wall-end, the mathematical expression of the relative position rules shown in Table 9 is shown in the following equations:

$$|P_{UL}.X - P_{LL}.X| \leq D, \quad (5)$$

$$P_{UL}.Y - P_{LL}.Y > D > 0, \quad (6)$$

$$|P_{UL}.Y - P_{LL}.Y - \text{WallThickness}| \leq D. \quad (7)$$

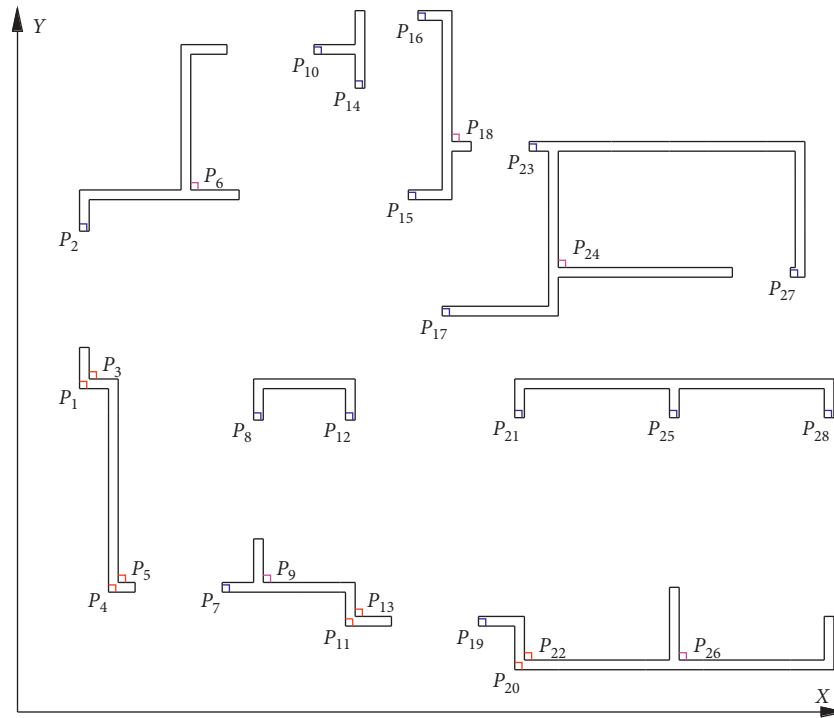


FIGURE 12: An example of lower left right angle sorting.

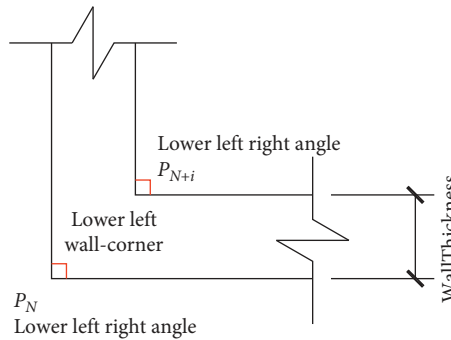


FIGURE 13: Schematic of 2 lower left right angles that form a lower left wall-corner.

TABLE 5: Rules of 2 lower left right angles that form a lower left wall-corner.

No.	Rule
1	An external right angle on the lower left side and an internal right angle on the upper right side
2	The vertices of 2 right angles differ by one wall thickness in the X and Y directions

As shown in equation (5), in the X direction,  $P_{LL}$  and  $P_{UL}$  are at the same locations. According to equation (6), in the Y direction,  $P_{UL}$  locates in the positive direction of  $P_{LL}$ . These 2 equations guarantee that  $P_{UL}$  locates directly above  $P_{LL}$ . As expressed in equation (7), in the Y direction,  $P_{UL}$  locates in the positive direction of  $P_{LL}$ , and the distance between them is exactly a wall thickness. These 3 equations ensure accurate recognition of the left wall-end.

Similar to the recognition of the left wall-end, recognition of the other 3 wall-ends follows similar recognition methods. The recognition algorithms of these 3 wall-ends are shown in Tables 10–12, respectively.

The interpretations of the mathematical expressions contained in Tables 10–12 are completely similar to those of equations (5) to (7) of the left wall-end and are not separately explained.

3.4.3. *T-Wall Recognition.* Each T-wall contains 2 right angles of different types, and each T-wall has a similar identification algorithm. Consider the left T-wall as an example, a lower left right angle and an upper left right angle form a left T-wall. The relative positions of these 2 right angles are fixed, and the relative position rules that they must



TABLE 6: Lower right wall-corner recognition algorithm.

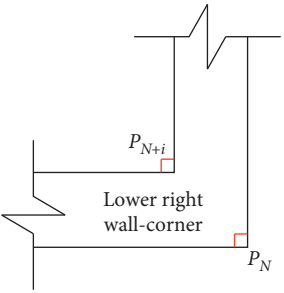
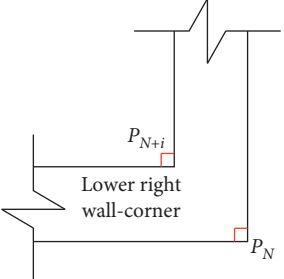
Schematic	Recognition rules descriptions and mathematical expressions
	<p>External <math>P_N</math> on the lower right side; internal <math>P_{N+i}</math> on the upper left side</p> <p>The 2 vertices differ by one wall thickness in the X and Y directions</p>
	$P_N \cdot X - P_{N+i} \cdot X > D > 0$ $P_{N+i} \cdot Y - P_N \cdot Y > D > 0$ $ P_N \cdot X - P_{N+i} \cdot X - \text{WallThickness}  \leq D$ $ P_{N+i} \cdot Y - P_N \cdot Y - \text{WallThickness}  \leq D$

TABLE 7: Upper left wall-corner recognition algorithm.

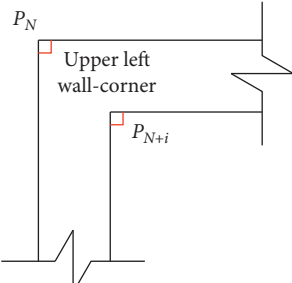
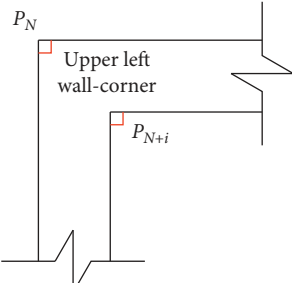
Schematic	Recognition rules descriptions and mathematical expressions
	<p>External <math>P_N</math> on the upper left side; internal <math>P_{N+i}</math> on the lower right side</p> <p>The 2 vertices differ by one wall thickness in the X and Y directions</p>
	$P_{N+i} \cdot X - P_N \cdot X > D > 0$ $P_N \cdot Y - P_{N+i} \cdot Y > D > 0$ $ P_{N+i} \cdot X - P_N \cdot X - \text{WallThickness}  \leq D$ $ P_N \cdot Y - P_{N+i} \cdot Y - \text{WallThickness}  \leq D$

TABLE 8: Upper right wall-corner recognition algorithm.

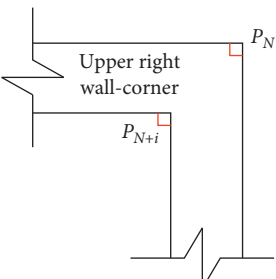
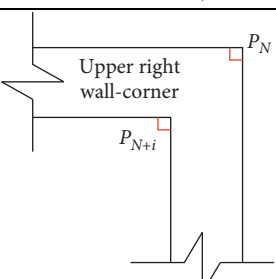
Schematic	Recognition rules descriptions and mathematical expressions
	<p>External <math>P_N</math> on upper right side; internal <math>P_{N+i}</math> on lower left side</p> <p>The 2 vertices differ by one wall thickness in the X and Y directions</p>
	$P_N \cdot X - P_{N+i} \cdot X > D > 0$ $P_N \cdot Y - P_{N+i} \cdot Y > D > 0$ $ P_N \cdot X - P_{N+i} \cdot X - \text{WallThickness}  \leq D$ $ P_N \cdot Y - P_{N+i} \cdot Y - \text{WallThickness}  \leq D$

TABLE 9: Rules for a lower left right angle and an upper left right angle that form a left wall-end.

No.	Rule
1	A lower left right angle on the downside and an upper left right angle on the upside
2	The 2 vertices of the 2 right angles are on the same vertical line
3	The 2 vertices of the 2 right angles differ by one wall thickness in the Y direction

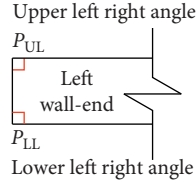


FIGURE 14: Schematic of 2 different right angles that form a left wall-end.

TABLE 10: Right wall-end recognition algorithm.

Schematic	Recognition rules descriptions and mathematical expressions
	<p>Lower right <math>P_{LR}</math> on the downside; upper right <math>P_{UR}</math> on the upside  The 2 vertices are located on the same vertical line  The 2 vertices differ by one wall thickness in the Y direction</p> $ P_{UR}.X - P_{LR}.X  \leq D$ $P_{UR}.Y - P_{LR}.Y > D > 0$ $ P_{UR}.Y - P_{LR}.Y - \text{WallThickness}  \leq D$

TABLE 11: Up wall-end recognition algorithm.

Schematic	Recognition rules descriptions and mathematical expressions
	<p>Upper right <math>P_{UR}</math> on the right side; upper left <math>P_{UL}</math> on the left side  The 2 vertices are located on the same horizontal line  The 2 vertices differ by one wall thickness in the X direction</p> $ P_{UR}.X - P_{UL}.X  \leq D$ $P_{UR}.X - P_{UL}.X > D > 0$ $ P_{UR}.X - P_{UL}.X - \text{WallThickness}  \leq D$

satisfy are shown in Table 13, among which the 3 rules must be simultaneously satisfied.

Figure 15 shows the relative position of 2 right angles that form the left wall-end. Among them,  $P_{LL}$  is the lower left right angle and  $P_{UL}$  represents the upper left right angle. The 2 right angles are not of the same type.

In Figure 15, when these 2 right angles form the left T-wall, the mathematical expression of the relative position rules shown in Table 13 is shown in the following equations:

$$|P_{LL}.X - P_{UL}.X| \leq D, \quad (8)$$

$$P_{LL}.Y - P_{UL}.Y > D > 0, \quad (9)$$

$$|P_{LL}.Y - P_{UL}.Y - \text{WallThickness}| \leq D. \quad (10)$$

As expressed in equation (8), in the X direction,  $P_{LL}$  and  $P_{UL}$  are at the same locations. According to equation (9), in the Y direction,  $P_{LL}$  locates in the positive direction of  $P_{UL}$ . These 2 equations guarantee that  $P_{LL}$  locates directly above  $P_{UL}$ . As demonstrated in equation (10), in the Y direction,  $P_{LL}$  locates in the positive direction of  $P_{UL}$ , and the distance between them is exactly a wall thickness. These 3 equations ensure accurate recognition of the left T-wall.

Similar to the recognition of the left T-wall, the recognition of the remaining 3 T-walls follows the similar recognition methods. The recognition algorithms of these 3 T-walls are shown in Tables 14–16, respectively.

The interpretations of the mathematical expressions contained in Tables 14–16 are similar to those of equations (8) to (10) of the left T-wall and are not separately explained.

TABLE 12: Down wall-end recognition algorithm.

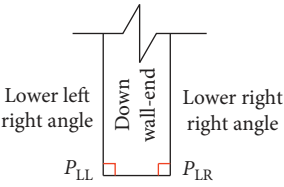
Schematic	Recognition rules descriptions and mathematical expressions
	<p>Lower right <math>P_{LR}</math> on the right side; lower left <math>P_{LL}</math> on the left side                      The 2 vertices are on the same horizontal line                      The 2 vertices differ by one wall thickness in the X direction</p> $ P_{LR}.Y - P_{LL}.Y  \leq D$ $P_{LR}.X - P_{LL}.X > D > 0$ $ P_{LR}.X - P_{LL}.X - \text{WallThickness}  \leq D$

TABLE 13: Rules for a lower left right angle and an upper left right angle that form a left T-wall.

No.	Rule
1	A lower left right angle on the upside and an upper left right angle on the downside
2	The 2 vertices of the 2 right angles are located on the same vertical line
3	The 2 vertices of the 2 right angles differ by one wall thickness in the Y direction

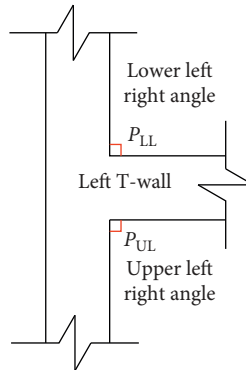
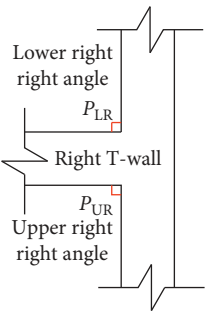


FIGURE 15: Schematic of 2 different right angles that form a left T-wall.

TABLE 14: Right T-wall recognition algorithm.

Schematic	Recognition rules descriptions and mathematical expressions
	<p>Lower right <math>P_{LR}</math> on the upside; upper right <math>P_{UR}</math> on the downside                      The 2 vertices are on the same vertical line                      The 2 vertices differ by one wall thickness in the Y direction</p> $ P_{LR}.X - P_{UR}.X  \leq D$ $P_{LR}.Y - P_{UR}.Y > D > 0$ $ P_{LR}.Y - P_{UR}.Y - \text{WallThickness}  \leq D$

3.4.4. *Cross-Wall Recognition.* Each cross-wall contains 4 right angles of different types. The structural features of the cross-wall reveal that the cross-wall contains the features of the T-walls. For example, remove the left limb of the cross-wall and the cross-wall becomes the left T-wall. Therefore, the rules used to recognize the T-walls are also valid for the cross-wall recognition.

However, none of the 4 types of T-walls simultaneously contains both a lower left right angle and an upper right

right angle; none of them simultaneously contains both a lower right right angle and an upper left right angle. Each pair of these 2 pairs of right angles can distinguish the cross-wall from the 4 types of T-walls. Choosing the pair of a lower left right angle and an upper right right angle, the relative position rules of these 2 right angles are shown in Table 17.

Figure 16 shows the relative position of 2 right angles that form the cross-wall.  $P_{LL}$  is the lower left right angle;  $P_{UR}$

TABLE 15: Up T-wall recognition algorithm.

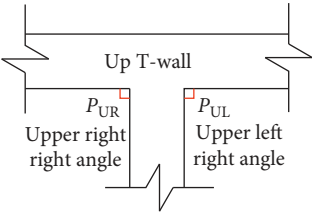
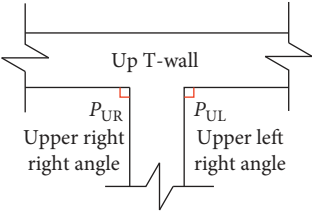
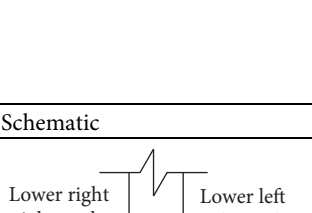
Schematic	Recognition rules descriptions and mathematical expressions
	Upper right $P_{UR}$ on the left side; upper left $P_{UL}$ on the right side The 2 vertices are on the same horizontal line
	The 2 vertices differ by one wall thickness in the $X$ direction
	$ P_{UL}.Y - P_{UR}.Y  \leq D$ $P_{UL}.X - P_{UR}.X > D > 0$  $ P_{UL}.X - P_{UR}.X - \text{WallThickness}  \leq D$

TABLE 16: Down T-wall recognition algorithm.

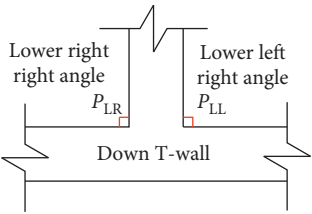
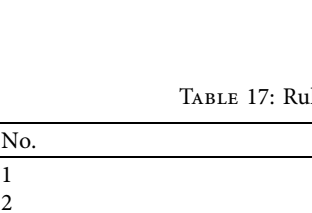
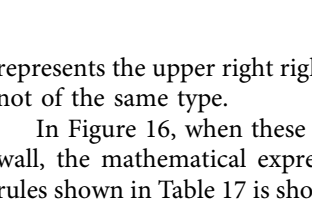
Schematic	Recognition rules descriptions and mathematical expressions
	Lower right $P_{LR}$ on the left side; lower left $P_{LL}$ on the right side The 2 vertices are on the same horizontal line
	The 2 vertices differ by one wall thickness in the $X$ direction
	$ P_{LL}.Y - P_{LR}.Y  \leq D$ $P_{LL}.X - P_{LR}.X > D > 0$  $ P_{LL}.X - P_{LR}.X - \text{WallThickness}  \leq D$

TABLE 17: Rules for a lower left right angle and an upper right right angle that form a cross-wall.

No.	Rule
1	Lower left right angle on the upper right and upper right right angle on the lower left
2	The vertices of 2 right angles differ by one wall thickness in the $X$ and $Y$ directions

represents the upper right right angle. The 2 right angles are not of the same type.

In Figure 16, when these 2 right angles form the cross-wall, the mathematical expression of the relative position rules shown in Table 17 is shown in the following equations:

$$P_{LL}.X - P_{UR}.X > D > 0, \quad (11)$$

$$P_{LL}.Y - P_{UR}.Y > D > 0, \quad (12)$$

$$|P_{LL}.X - P_{UR}.X - \text{WallThickness}| \leq D, \quad (13)$$

$$|P_{LL}.Y - P_{UR}.Y - \text{WallThickness}| \leq D. \quad (14)$$

As shown in equation (11), in the  $X$  direction,  $P_{LL}$  locates in the positive direction of  $P_{UR}$ . According to equation (2), in the  $Y$  direction,  $P_{LL}$  locates in the positive direction of  $P_{UR}$ . These 2 equations guarantee that  $P_{LL}$  locates on the upper right side of  $P_{UR}$ . As expressed in equation (13), in the  $X$  direction,  $P_{LL}$  locates in the positive direction of  $P_{UR}$ , and the distance between them is exactly a wall thickness.

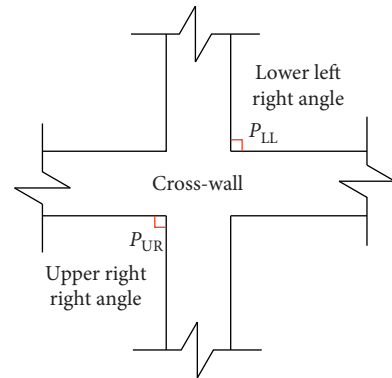


FIGURE 16: Schematic of 2 different right angles that form a cross-wall.

According to equation (14), in the  $Y$  direction,  $P_{LL}$  locates in the positive direction of  $P_{UR}$ , and the distance between them is exactly a wall thickness. These 4 equations ensure accurate recognition of the cross-wall.

TABLE 18: Cross-wall recognition algorithm using lower right right angle and upper left right angle.

Schematic	Recognition rules descriptions and mathematical expressions
	<p>Lower right <math>P_{LR}</math> on the upper left side; upper left <math>P_{UL}</math> on the lower right side                      The 2 vertices differ by one wall thickness in the X and Y directions</p> $P_{UL}.X - P_{LR}.X > D > 0$ $P_{LR}.Y - P_{UL}.Y > D > 0$ $ P_{UL}.X - P_{LR}.X - \text{WallThickness}  \leq D$ $ P_{LR}.Y - P_{UL}.Y - \text{WallThickness}  \leq D$

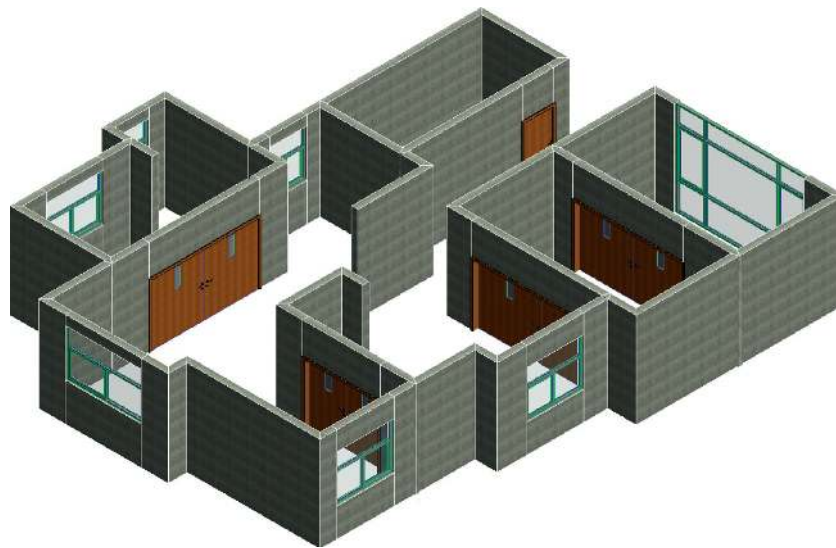


FIGURE 17: Automatic generation of 3D building models result.

If the pair of a lower right right angle and an upper left right angle is chosen to recognize the cross-wall, a similar recognition algorithm exists, as shown in Table 18.

The interpretations of the mathematical expressions contained in Table 18 are similar to those of equations (11) to (14). Both expressions can be used to independently recognize the cross-wall.

#### 4. Algorithm Tests

The algorithm directly focuses on the drawings' walls. The other parts, such as texts and dimensions, can be firstly detected and processed by the proposed methods [11, 14–16, 18]. The remaining parts of the drawing may look similar to Figure 2, which is then processed by the algorithm.

A construction company provides 10 real projects' drawings for algorithm tests. All 10 projects are residential buildings. Among them, 2 have small fillets on the outermost outline corners, while the rest of the drawings are all horizontal and vertical lines; the other 8 are all composed of horizontal and vertical lines. The algorithm implementation

was completed by Autodesk Revit customization. The test result is 8 successes and 2 failures.

The 2 failed tests are the 2 drawings with small fillets; the reason is straightforward: fillets contain arcs, so the fillets cannot be recognized as rectangles. The other 8 tests are all success. The actual project shown in Figure 2 is among the 8 success tests, and its generated 3D building models are shown in Figure 17.

In Figure 17, windows, doors, and the wall segments above or below them are generated first by adopting other study results [11, 14–16, 18]. Lots of white boundary lines exist in Figure 17. The biggest highlight is that the walls are not as a whole, but consist of separated wall segments. Thus, detailed information can be attached to these separated objects to support precise O&M tasks. Secondly, all windows, doors, and wall segments have recorded information of their adjacent objects. Thus, both geometry and topology information are generated for the 3D building models.

During the algorithm implementation, error thresholds are introduced to increase the algorithm's robustness in Section 3.1 and Section 3.2. Tables 19 and 20 show small

TABLE 19: Small rotation error thresholds.

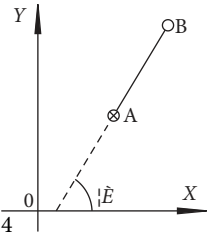
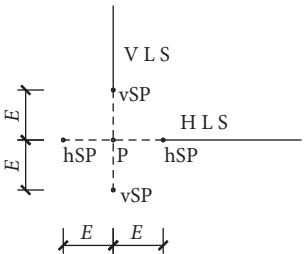
Schematic	Horizontal and vertical line segments with rotation error thresholds
	Horizontal: $ \theta  \leq E$ or $ \theta - 360^\circ  \leq E$ or $ \theta - 180^\circ  \leq E$  Vertical: $ \theta - 90^\circ  \leq E$ or $ \theta - 270^\circ  \leq E$

TABLE 20: False intersection error thresholds.

Schematic	Horizontal and vertical line segments with false intersection error thresholds
	HLS = horizontal line segment, VLS = vertical line segment, hSP = horizontal line segment starting point vSP = vertical line segment starting point $ \text{hSP}.X - \text{vSP}.X  \leq E$  $ \text{hSP}.Y - \text{vSP}.Y  \leq E$

rotation and false intersections can be well resisted, respectively.

In Table 19,  $\theta$  can be easily calculated by point A and B's coordinates. An error threshold  $E$  can be set to adapt small rotations. Horizontal and vertical line segments can be filtered out by the 2 equations in Table 19, even if they are a little rotated. Too much rotation can no longer be defined as errors, but inclined lines which is not included by this paper.

Table 20 takes lower left right angle for example, and error thresholds are both introduced in X and Y directions. By the 2 equations in Table 20, lower left right angle can still be recognized even if hSP may not strictly coincide with vSP.

In general, buildings' drawings are of good quality before they can be used to build buildings. With the help of error thresholds, the algorithm tests results are good.

## 5. Conclusions

This paper describes an algorithm to automatically generate 3D building models from 2D drawings. 3D building models are intended to be used in O&M management for example. The algorithm includes both geometrical and topological relations to provide 3D building model composed of separate wall segments. The algorithm has been implemented by Autodesk Revit customization and tested with several real projects.

Walls in drawings of regular layout buildings can be deconstructed into 4 categories: wall-corner, T-wall, cross-wall, and wall-end. According to the orientation, wall-corner, T-wall, and wall-end are further deconstructed into 4 different types. Because the cross-wall is symmetric in both the X direction and Y direction, it cannot be further deconstructed. Thus, 13 types of deconstruction parts are

obtained. The walls in a drawing are composed of different types of deconstruction parts.

All 13 types of deconstruction parts are composed of right angles. The orientations comprise 4 different types of right angles. 4 types of right angles consist of line segments. According to the orientation, line segments can be categorized into horizontal line segments and vertical line segments. A line segment has 2 endpoints: the starting point and the ending point. By vectorization, the starting point and ending point of a line segment are distinguished from each other. 2 different line segments with their 2 distinguished endpoints yield 4 different combinations, which correspond to the 4 types of right angles.

The algorithm is platform independent and can be easily implemented. Real project tests show that the algorithm can rapidly generate separated 3D building models with both geometry and topology, which can be used for precise O&M management for existing buildings.

The algorithm is based on vectorized line segments but only involves 2 special cases: horizontal line segments and vertical line segments. Therefore, this algorithm may be a special case. The general form of the algorithm is still to be found in the future.

## Data Availability

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to gratefully acknowledge the support of the Doctoral Scientific Fund Project of the Yanshan University (BL18053) and Postdoctoral Research Project of Hebei Provincial Department of Human Resources and Social Security (Personnel Department) (B2019003025).

## References

- [1] Y. C. Chi and X. Liu, "An automated reconstruction approach of mechanical systems in building information modeling (BIM) using 2D drawings," in *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, pp. 236–244, Seattle, WA, USA, June 2017.
- [2] Q. Lu and S. Lee, "Image-based technologies for constructing as-is building information models for existing buildings," *Journal of Computing in Civil Engineering*, vol. 31, no. 4, Article ID 04017005, 2017.
- [3] Q. Lu and S. Lee, "A semi-automatic approach to detect structural components from CAD drawings for constructing as-is BIM objects," in *Proceedings of the ASCE International Workshop on Computing in Civil Engineering (IWCCCE) 2017*, pp. 84–91, Seattle, WA, USA, June 2017.
- [4] L. Gimenez, S. Robert, F. Suard, and K. Zreik, "Automatic reconstruction of 3D building models from scanned 2D floor plans," *Automation in Construction*, vol. 63, pp. 48–56, 2016.
- [5] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (BIM) for existing buildings-literature review and future needs," *Automation in Construction*, vol. 38, pp. 109–127, 2014.
- [6] D. Santos, M. Dionísio, N. Rodrigues, A. Pereira, and I.-I. Inovação Leiria, "Efficient creation of 3D models from buildings' floor plans," *International Journal of Interactive Worlds*, vol. 2011, Article ID 897069, 30 pages, 2011.
- [7] X. Yin, P. Wonka, and A. Razdan, "Generating 3D building models from architectural drawings: a survey," *IEEE Computer Graphics and Applications*, vol. 29, no. 1, pp. 20–30, 2009.
- [8] S. Lee and Ö. Akin, "Shadowing tradespeople: inefficiency in maintenance fieldwork," *Automation in Construction*, vol. 18, no. 5, pp. 536–546, 2009.
- [9] K. Tombre, "Graphics recognition-general context and challenges," *Pattern Recognition Letters*, vol. 16, no. 9, pp. 883–891, 1995.
- [10] K. Tombre, "Ten years of research in the analysis of graphics documents: achievements and open problems," in *Proceedings of the 10th Portuguese Conference on Pattern Recognition*, pp. 11–17, Lisboa, Portugal, March 1998.
- [11] C. Ah-Soon and K. Tombre, "Variations on the analysis of architectural drawings," in *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 347–351, Ulm, Germany, August 1997.
- [12] C. Ah-Soon, "A constraint network for symbol detection in architectural drawings," in *Proceedings of the Selected Papers from the Second International Workshop on Graphics Recognition, Algorithms and Systems*, pp. 80–90, Nancy, France, August 1997.
- [13] C. Ah-Soon and K. Tombre, "Architectural symbol recognition using a network of constraints," *Pattern Recognition Letters*, vol. 22, no. 2, pp. 231–248, 2001.
- [14] B. T. Messmer and H. Bunke, "Automatic learning and recognition of graphical symbols in engineering drawings," "Automatic learning and recognition of graphical symbols in engineering drawings," in *Graphics Recognition Methods and Applications*, R. Kasturi and K. Tombre, Eds., pp. 123–134, Springer, Berlin, Germany, 1996.
- [15] J. Park and Y. B. Kwon, "Main wall recognition of architectural drawings using dimension extension line," *The Kips Transactions PartB*, vol. 3088, pp. 116–127, 2003.
- [16] R. Yang, Y. Cao, H. Li, S. Wang, and Q. Shen, "Automatic acquisition of rules for 3D reconstruction of construction components," *Automation in Construction*, vol. 13, no. 5, pp. 545–553, 2004.
- [17] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Improved automatic analysis of architectural floor plans," in *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, pp. 864–869, Beijing, China, September 2011.
- [18] T. Lu, C.-L. Tai, F. Su, and S. Cai, "A new recognition model for electronic architectural drawings," *Computer-Aided Design*, vol. 37, no. 10, pp. 1053–1069, 2005.
- [19] N. Nayef and T. M. Breuel, "Statistical grouping for segmenting symbols parts from line drawings, with application to symbol spotting," in *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, pp. 364–368, Beijing, China, September 2011.
- [20] M. Feltes, S. Ahmed, A. Dengel, and M. Liwicki, "Improved contour-based corner detection for architectural floor plans," in *Proceedings of the International Workshop on Graphics Recognition*, pp. 191–203, Bethlehem, PA, USA, August 2013.
- [21] L.-P. De Las Heras, S. Ahmed, M. Liwicki, E. Valveny, and G. Sánchez, "Statistical segmentation and structural recognition for floor plan interpretation," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 3, pp. 221–237, 2014.
- [22] J. Guo, C. Shi, G. Azzopardi, and N. Petkov, "Recognition of architectural and electrical symbols by COSFIRE filters with inhibition," "Recognition of architectural and electrical symbols by COSFIRE filters with inhibition," in *Computer Analysis of Images and Patterns*, G. Azzopardi and N. Petkov, Eds., pp. 348–358, Springer International Publishing, Cham, Switzerland, 2015.
- [23] H. C. Huang, S. M. Lo, G. S. Zhi, and R. K. K. Yuen, "Graph theory-based approach for automatic recognition of CAD data," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 7, pp. 1073–1079, 2008.
- [24] T. Li, B. Shu, X. Qiu, and Z. Wang, "Efficient reconstruction from architectural drawings," *International Journal of Computer Applications in Technology*, vol. 38, no. 1, pp. 177–184, 2010.
- [25] M. Liang and Z. Hu, "Reconstruction of 3D building models based on architectural construction drawings," *Applied Mechanics and Materials*, vol. 336–338, pp. 1349–1355, 2013.
- [26] L. Gimenez, J.-L. Hippolyte, S. Robert, F. Suard, and K. Zreik, "Review: reconstruction of 3D building information models from 2D scanned plans," *Journal of Building Engineering*, vol. 2, pp. 24–35, 2015.
- [27] J. Li, W. Huang, L. Shao, and N. Allinson, "Building recognition in urban environments: a survey of state-of-the-art and future challenges," *Information Sciences*, vol. 277, pp. 406–420, 2014.
- [28] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques," *Automation in Construction*, vol. 19, no. 7, pp. 829–843, 2010.

- [29] H. Fathi, F. Dai, and M. Lourakis, "Automated as-built 3D reconstruction of civil infrastructure using computer vision: achievements, opportunities, and challenges," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 149–161, 2015.