3D Reconstruction of Detailed Buildings from Architectural Drawings

Tong Lu[1], Chiew-Lan Tai[2], Li Bao[1], Feng Su[1] and Shijie Cai[1]

[1]Nanjing University, lu_tong@263.net
[2]The Hong Kong University of Science and Technology, taicl@cs.ust.hk

ABSTRACT

Due to the difficulties in designing and modifying a complex 3D architectural building in 3D space, 2D drawings are often a more effective means for design. However, for various applications such as quantity surveying and visualization, it is necessary to convert 2D architectural drawings to 3D models. In this paper, we propose a new method for accurate 3D reconstruction from real-life architectural drawings. The method integrates and normalizes architectural information dispersed in multiple drawings and tables under the guidance of semantics and prior domain knowledge. The reconstructed detailed 3D building can be used for quantity surveying, construction, and 4D modeling.

Keywords: 3D reconstruction; architectural drawings; detailed 3D building.

1. INTRODUCTION

In the last two decades, we have seen a great development in 3D modeling using solid modeling techniques and parametric representations for mechanical engineering and computer-aided production. In contrast, there has been little progress in the development of 3D design in the field of architecture [1]. The main reasons are that architecture design involves complex products, consisting of a great many distinct parts, and creating and editing these kinds of 3D models in 3D space are difficult. Hence, architecture design continues to be done on 2D drawings. However, for various construction applications, such as quantity surveying, inventory, construction, and visualization, it is necessary to convert the widely used 2D architectural drawings to accurate 3D models.

In real life, an architect designs a building by drawing a set of 2D drawings, depicting the layout, dimensions, and internal structures of various structural objects as well as the mechanical design (e.g., steel bars or concrete) of these structural objects (see Fig. 1 and details in Section 2). These drawings are called Construction Structural Drawings (CSDs). They are the most important kind of architectural drawings for quantity surveying and construction. To reconstruct a detailed 3D building, it is necessary to use the CSDs.

In practice, the conversion from the 2D architectural construction drawings to 3D models is always carried out manually or semi-automatically. Skilled persons are needed to interpret the paper drawings or electronic drawings based on their acquired domain knowledge. They must understand the designer's intentions, reconstruct the 3D building in their mind, and finally, calculate the useful engineering data manually for various applications. This is an imprecise and time-consuming approach. The low efficiency of human interpretation is a huge obstacle in communication between designers and engineers, which not only hinders the information flow among different users, but also becomes the bottleneck of the automation effort in construction. Therefore, the automatic reconstruction of detailed 3D buildings for real-life applications is an important and urgent research topic.

There have been abundant research results in mechanical reconstruction from drawings. However, these algorithms have some common characteristics that limit their direct applicability to the field of architecture. Nevertheless, they form a good basis for the problem of reconstruction from architectural drawings, especially the purely geometric methods [2-6]. One family of algorithms, known as fleshing-out projections, is based on the reconstruction of a wire-frame model by matching vertices and edges between drawings depicting orthographic views of the object, and then finding the faces of the object by propagating the constraints on the wire-frame model. Another family of algorithms is volume-oriented, which first finds 3D subparts and then combines them to obtain the whole object. These algorithms share the basic idea of looking for certain sets of consistent matching hypotheses between the features extracted from the projection views of an object. They share two characteristics that limit their domain of applicability [7]:

- The data on which they work have to be perfect. This is necessary for the reconstruction from paper drawings or from electronic drawings, because the deficiencies will finally be reflected in the form of geometry primitives in a reconstruction system. In real life, perfect paper or electronic architectural drawings are non-existence, because vectorization of paper drawings is easily affected by noises and electronic drawings are drawn as a multitude of lines that often overlaps or missing.
- These methods only apply to the geometric part of the drawing. But, a large part of the architectural construction drawing is symbolic, rather than geometric. As a technical document, it does not only contain a set of geometric patterns, but also possesses a “linguistic” dimension. Not much work has been done on mixing the geometry with semantics in the interpretation process. A generic solution may even be impossible to reach since the semantic part of the information is strongly context dependent.

Therefore, before the reconstruction methods for mechanical engineering drawings can be applied to architectural drawings, we face the challenge of associating a geometric reconstruction process with the functional and symbolic information retrieved by semantic analysis of each architectural drawing. As noted by Ablameyko [8], the second limitation is perhaps more difficult to deal with, and only partial solutions is possible. It is especially challenging for architectural drawings due to certain characteristics of these drawings (see details in Section 2) [9-10]. To our knowledge, very few researchers have investigated the problem of reconstruction of 3D architectural objects from drawings [11].

In this paper, we propose a new knowledge-based method for the automatic 3D reconstruction from 2D architectural drawings. The method is based on the idea of Integration and Normalization of Dispersive Architectural Information (INDAI), thus we call it the INDAI method. It is built upon our previous work in automatic recognition of dimensions, coordinate systems, and structural objects (e.g., columns, beams, slabs, walls, and holes) from electronic CSDs [12, 18] and the vectorization of paper CSDs [16-17].

The remainder of this paper is organized as follows. Section 2 introduces the characteristics and difficulties of 3D reconstruction from architectural drawings. Section 3 gives an overview of the INDAI reconstruction method. Section 4 presents the details of the INDAI method, which gives good results even for complex architectural drawings. Finally, Section 5 offers conclusions of our work.

2. CHARACTERISTICS OF ARCHITECTURAL DRAWINGS FOR 3D RECONSTRUCTION

Current 3D reconstruction methods [2-6, 8] are mostly proposed for mechanical engineering drawings. For mechanical objects, each object is usually defined by six, or at least three, orthographic views (front, side and top). Geometric constraints correspond strictly to each other in the different views. Idesawa [13] proposed a method to generate a wireframe model from its orthographic views. The method projects the orthographic views into a 3D environment and recognizes a face by finding a closed loop of edges. But the search process becomes increasingly complex because of the complexity of architectural objects. Wesley and Markowsky [14] proposed a method that recognizes blocks to obtain the solution, where each block is composed of a closed region of faces. The method is more effective than Idesawa’s method because the number of blocks is much fewer than the number of faces; but since the number of combinations of blocks to consider is 2^n, where n is the number of blocks, it is still difficult to apply this method to complex architectural drawings. Tanaka [6, 15] proposed a method using dimensions to automatically convert 2D assembly drawings to 3D part drawings and generate a unique solution.

These methods cannot be directly applied for 3D reconstruction from architectural drawings due to the great differences between mechanical drawings and architectural drawings. To reconstruct a solid model from 2D architectural drawings, the following difficulties must first be considered:

- Structural objects in a construction project are usually drawn in multiple drawings (e.g., column drawings, beam drawings, and slab drawings). Each type of structural objects on various floors may be grouped and drawn in different drawings (for example, we may have a column drawing for floor 2-3 and a beam drawing for floor 3-5). So an architectural project usually has many related drawings; typically, a real-life building is described by hundreds of architectural drawings.
- The 3D architectural information is distributed in plane drawings, section drawings, elevation drawings, or tables. Furthermore, certain information is conveyed by multitude descriptive annotations, prior knowledge, and domain experiences.
- To reconstruct a solid model corresponding to a structural object, a series of drawings must be cross-referenced because the information of an object is usually dispersed or implied in more than one drawing. For example, the geometric contour of a column may be defined by an outline in a column section drawing,

but its size is described by annotations in another column section table, and its height can only be calculated from a level table and its adjacent structural objects (i.e., beams and slabs).

- Sketched drawings are used to define the contours of objects, and the geometric primitives in the sketched drawings only reflect the architect's intentions. The "real" contours is known only after the semantics are analyzed,.
- There is a great deal of information on the detailed internal structures of objects (e.g., the steel bars in a column, the distributed steel bars on a slab) in an architectural drawing. This complicates the automatic reconstruction process.

As an example, Fig.1 shows a plane beam drawing of floor 2-4, where the details of the beams are annotated. The details of the other structural objects, such as columns and slabs, appear in other drawings which are not shown in this paper.

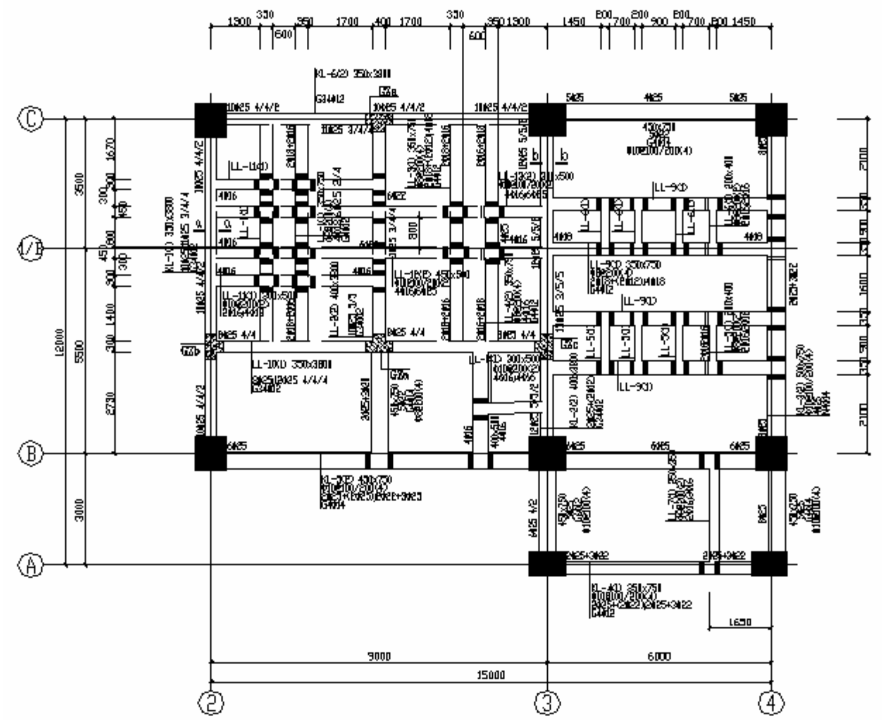


Fig. 1. An example of plane beam drawing of floor 2-4

3. OVERVIEW OF THE INDAI METHOD

We refer to the above-mentioned characteristics and difficulties of the reconstruction problem from architectural drawings as dispersion and diversity. That is, the 3D information may be distributed in multiple related drawings in different ways – in the form of outlines, annotation texts, different kinds of tables, etc. To reconstruct the solid models, we must collate the dispersed and diversified information. We propose the INDAI method to solve this problem. The main idea is to transform the explicit representations, such as graphic primitives in the plane drawings or section drawings, the annotations and tables, into the normalized global graphic primitives of three orthogonal views – the top, side and front views. The implied information is extracted from texts, tables, or domain knowledge and is projected onto the normalized views. Thus, from the normalized drawings, we can build a solid model step by step to obtain a single final solution regardless of the complexity and difficulties of the original architectural drawings. Since dimensions and names of structural objects are widely used in CSDs, the transformation is feasible under the guidance of the semantics of structural objects and prior knowledge. Finally, the internal structures of each solid are added according to the relationships among the solids to obtain the detailed target building.

The inputs to the INDAI method are electronic CSDs drawings in DWG or DXF CAD format. These formats enable lines, arcs, and texts to be distinguished as different entities. The main steps of INDAI are as follows:

1. Recognize the high-level structural objects automatically from the input drawings. Build the relationships of the

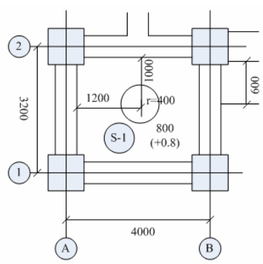
structural objects within each drawing and across different drawings. This process simplifies the subsequent analysis of 3D reconstruction of structural objects.
2. For each recognized structural object, integrate the geometric primitives and other representations from multiple drawings to obtain three normalized drawings, which are orthogonal views of a valid 3D object.
3. Reconstruct a solid model for each structural object from the normalized drawings.
4. Output the detailed 3D building.

In [12], we presented a recognition model, called SINEHIR (Self-Incremental Axis-Net-based Hierarchical Recognition) method, for recognizing high-level structural objects from electronic drawings. In this paper, we use the SINEHIR method in Step 1 of INDAI. The details of step 2 and 3 are described in the following section.

4. CONVERTING 2D STRUCTURAL OBJECTS TO SOLID MODELS

4.1 Normalized 2D Drawings

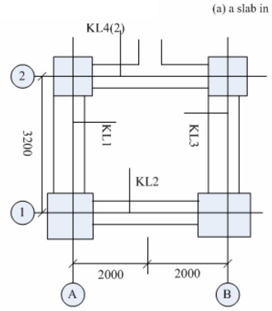
After all the architectural drawings are analyzed using the SINEHIR method [12], we obtain the functional and structural descriptions of the dimensions, structural objects, tables and their relationships. To briefly illustrate the subsequent reconstruction process that we will describe, we will use the recognition results shown in Fig. 2 as an example of output of SINEHIR. Fig. 2 shows parts of a slab drawing, a beam drawing, and a column drawing, depicting a slab surrounded by four columns and four beams. The details of the beams are recognized from the beam plane drawing (Fig. 2(c)), where there is a name associated to each beam (e.g., KL1, KL2) annotated by a leader line. The size and the internal steels are listed in the beam table; using the name in the beam plane drawing, we can retrieve the attributes in the table. Observe that only two of the recognized columns in the column drawing (Fig. 2 (d)) have the detailed outlines and dimensions. The information of the internal steels is in the column table. The internal structures of each column are implied in the column sections: the tiny circles represent the internal long steel bars, which spans through the whole column and bends into its adjacent structural objects at the top and bottom. The internal structures will be analyzed and added after the solids of all the structural objects are reconstructed. Fig. 2(b) is a part of the level table, which describes the elevation level of each floor in the project.



(a) a slab in slab CSD of floor 2

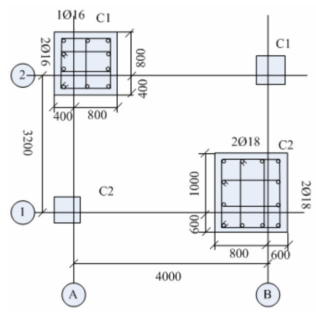
3	8.670
2	4.470
1	-0.030
-1	-4.630
FLOOR	LEVEL(m)

(b) the level table of this project



NAME	SIZE	STEEL	STEEL	EAR	FALL
KL1	800X1200	2Ø12	Ø8-100		+0.40
KL2	1200X1000 Y400X100	2Ø12	Ø8-100/200	E200X300/350	-0.40
KL3	800X1200	2Ø12	Ø8-100		
KL4	(1) 600X1200 Y400X100 (2) 800X1200 Y400X100	2Ø12	Ø8-100	E200X200/350 E200X400/750	+0.20

(c) four beams in beam CSD of floor 2



NAME	SIZE	STEEL	STEEL
C1	1200X1200	4Ø18	Ø8-100
C2	1600X1400	4Ø16	Ø8-100/200

(d) four columns in column CSD of floor 1

Fig. 2. Examples of architectural drawings and tables

Unfortunately, it is impossible to reconstruct the solid models of columns, beams and slabs only from the recognized structural objects in individual architectural drawings. Cross referencing among the different drawings is needed. We consider Fig. 2 as an example. From the slab drawing describes the details of slabs, as shown in Fig. 2 (a), we can neither find the alignments of beams nor the real width of each beam because the architect wants to emphasize only the details of slabs in slab drawings. Hence, the geometric contours of the beams in Fig. 2(a) only indicate the existence of beams, and it is impossible to obtain the accurate outline of the slab formed by the surrounding four columns and four beams in this drawing. We need to turn to the beam drawing (as shown in Fig. 2(c)), where we may find the details of the beams, such as the width, in the beam table. But one problem remains: we cannot calculate the alignments of the beams KL1 and KL3 because they are aligned via one of the sides of the connected columns, not by dimensions. Hence, we need to locate the outlines of the columns first. But we cannot find the accurate outline of any column in the beam drawing because the architect needn't have drawn them accurately there. Finally, we have to refer to the column drawing (Fig. 2(d)) to find the accurate outline of each column. In summary, before reconstructing the solid models of slabs from slab drawings, the column drawings and beam drawings have to be analyzed.

We will refer to the recognized objects that have accurate contours as *normalized 2D objects*. Normalized 2D objects are used to eliminate the implicit meanings of humans, which are important for manual reconstruction from architectural drawings. Normalized 2D objects can only be attained by interpreting the semantics and relationships of the recognized structural objects. A drawing that is composed of normalized 2D objects is called a *normalized 2D drawing*. It is a projection view of the real target 3D building.

The normalization of 2D objects involves the normalization of the following entities: location, geometric contour, level, height, length, and width. We will describe each of them in detail.

4.1.1 Normalization of location

We begin by normalizing the locations of the structural objects in the column drawing. In the column drawing in Fig. 2(d), the dimensions around the detailed columns indicate the distances from the grid lines. For columns without the detailed geometric primitives (e.g., the left C2), the distances will be copied from the corresponding detailed column of the same name (e.g., the right C2). For instance, the distance of the right detailed column C2 to the grid line B is 800mm, hence the distance of the left C2 to the grid line A is also 800mm. Then we calculate the alignments of the beams from the beam drawing in Fig. 2(c) as follows: the beams KL1 and KL3 are located by the columns C1 because the left side of KL1 overlaps with the left side of the left column C1, and the left side of KL3 overlaps with the left side of the right column C1. So columns C1 and C2 can be located from the column drawing in Fig. 2(d), and the beams KL1 and KL3 can be located from the beam drawing in Fig. 2(c). The other two beams KL2 and KL4 can be located from the grid lines because the two sides of each beam are symmetric along a grid line. If a beam meets both the above two conditions (located by a column and by a grid line), its location will only be calculated from the grid line, which is more probable according to prior knowledge about the habit of architects. Since the shape of a slab may be regular or irregular, the location of a slab will be determined from its surrounding columns and beams.

4.1.2 Normalization of width

The dimensions of the structural objects in their top view will first be extracted from the recognized tables in each drawing. For example, the text indicating sizes in the column table in Fig. 2(d), "1600X1400", means that the x-width and y-width of the column section are 1600mm and 1400mm, respectively. The size texts of in the beam table (Fig. 2(c)) are more complex. For example, the beam KL4 has two segments, and the text “(1)600X1200” means that the width and the height (the size of a beam section from the left view) of the first segment are 600mm and 1200mm, respectively; while “(2)800X1200” indicates the width and height of the second segment. The other texts such as “Y400X100” and “E200X200/350” are related to the side view and front view, which will be interpreted in Section 4.1.4 and Section 4.2.

4.1.3 Normalization of geometric contours

After the normalization of the location and the widths of the recognized structural objects, we rebuild their contours. Firstly, we rebuild the column outlines from the column drawing. We calculate the contours of the detailed columns according to the location and the widths in the tables. Thus, the contour outlines of these columns are consistent with the global coordinate system. Secondly, we rebuild the columns that have no detailed graphic primitives according to their source detailed columns. Then we replace the original column outlines in all drawings with the reconstructed normalized column contours. Fig. 3(a) shows the result after replacing the columns in the slab drawing in Fig. 2 with the normalized column contours.

Similarly, we can rebuild the beam contours from the normalized location and widths. Fig. 3(b) shows the result after replacing the beams in Fig. 3(a) with the normalized beam contours.

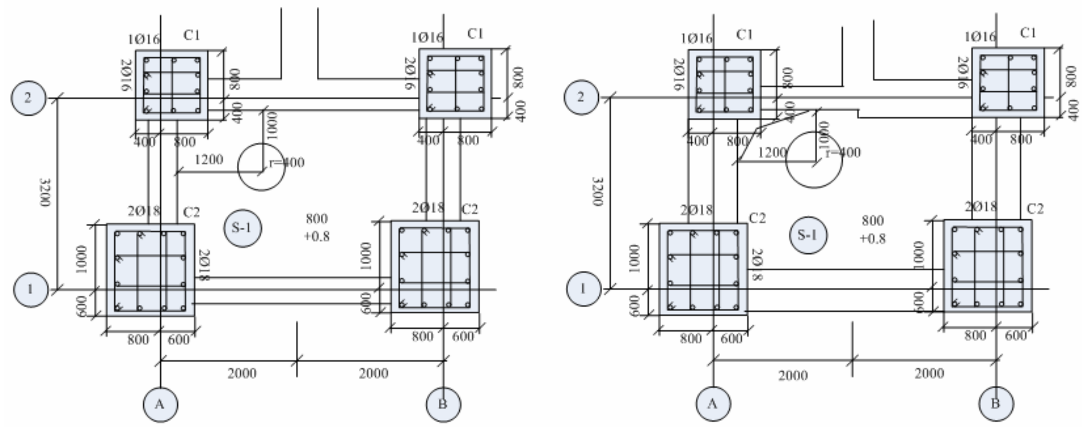


Fig. 3. Normalized geometric contours of columns and beams are added to Fig. 2(a)

4.1.4 Normalization of level, height and length

After the normalization of the geometric contours, the next step is to normalize the levels, heights and lengths of the slabs, columns and beams. The columns in a column drawing (Fig.2 (d)) are represented as sections. Usually architects do not label the height of a column in a section drawing or a table. In a real-life construction, columns are first built, then beams are added on the columns, and lastly, slabs are built over the beams and columns. Accordingly, we will first calculate the level of a slab, and then calculate the levels of the supporting beams based on the level of the slab. Lastly, the heights of the columns are calculated according to the levels of the beams.

We now illustrate the process by first normalizing the levels of the top and bottom faces of the slab in Fig. 2(a). In Fig.2 (a), there are three texts “S-1”, “800” and “(+0.8)” in the recognized slab area. The first text is the name of the slab, the second text is the thickness of the slab, and the last indicates the relative level of the top plane of the slab to the floor level. Since slab S-1 is on floor 2, we check the level of floor 2 from the level table in Fig.2 (b). Thus, the level of the top face of the slab should be 4.470+0.8=5.270m, and the level of the bottom face of the slab is 4.470+0.8-800/1000=4.470m. From the normalized geometric contour and the levels of the top and bottom faces of this slab, we can set up a front view of the slab. Fig. 2(a) also shows that the slab has a hole, dimensioned as 1200X1000. The normalized polygon of the hole will be subtracted from the polygon of the slab. Fig.4 shows the normalized views of the slab with the hole subtracted.

Next, we calculate the level of each of the four beams in Fig. 2(b). Usually, the level of the top face of a beam is equal to that of its connected slab. When a beam has a relative elevation text, e.g., “+0.20” in the beam table, it means that the level of the top face of that beam is raised by 0.20m. According to the existence of such elevation text, we calculate the top level of each beam. Similarly, the bottom level of each beam will be calculated using the formula "$level_{top}$ – height". Fig. 5 shows two views of the four recognized beams. The length of each beam is annotated in the front view, which is calculated from the normalized contours.

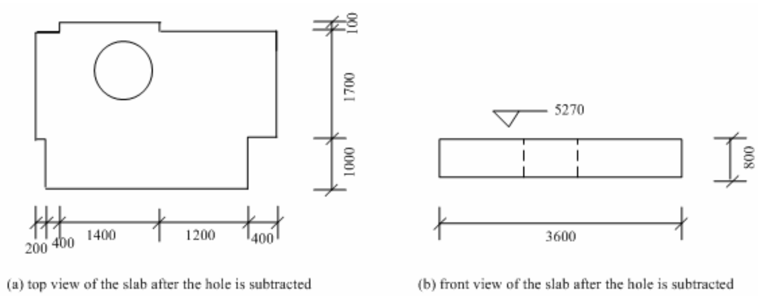


Fig. 4. Top view (a) and front view (b) of the slab in Fig. 2(a) after the hole is subtracted

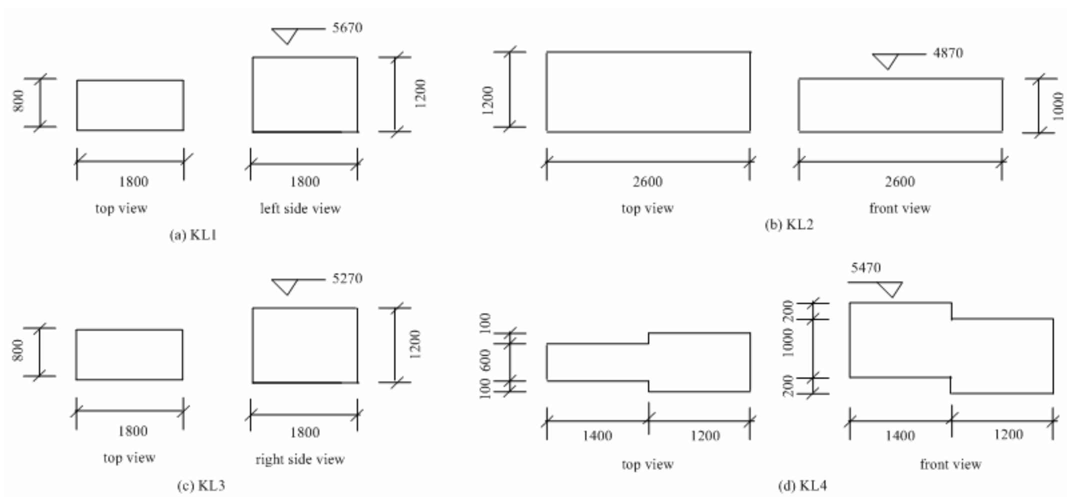


Fig. 5. Top view and front view of four beams, KL1 to KL4.

The shapes of the beams are a little complex because beams may have reinforcement components indicated by reinforcement texts. To reconstruct the solid models of beams, the texts have to be analyzed according to architectural semantics. As mentioned above, in the beam table of Fig. 2(c), the text "Y400X100" indicates a kind of beams that requires reinforcing the mechanics at the two ends. Fig. 6 shows the resulting views after analyzing this kind of text.

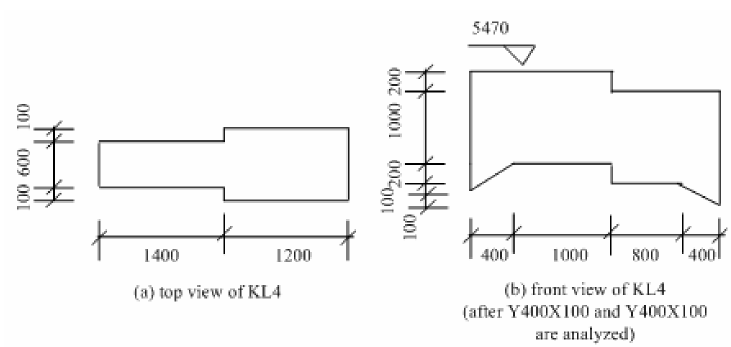


Fig. 6. Two views of the beam KL4 after the reinforcement texts are analyzed

4.2 Determining 3D faces and blocks of each object

From the three 2D normalized views of each structural object, we can determine a solid model. The lines in a view are called *2D edges*, and the end points of a 2D edge are called *2D vertices*. We first use a fleshing-out projection algorithm [6, 13-14] to create 3D vertices based on the combination of 2D vertices of the front and top views. A 3D edge that matches 2D vertices and 2D edges are then drawn between two 3D vertices. If two or more edges cross each other at a point other than a 3D vertex, then these edges are erased. Next, we recognize a face as a closed loop of edges, and recognize a block as a closed region of faces. Fig. 7(a) shows the 2D vertices and 2D edges of the beam KL4 in the top and side views. Fig. 7(b) shows the 3D vertices and 3D edges constituting the wire-frame model, and Fig. 7(c) shows the reconstructed solid model.

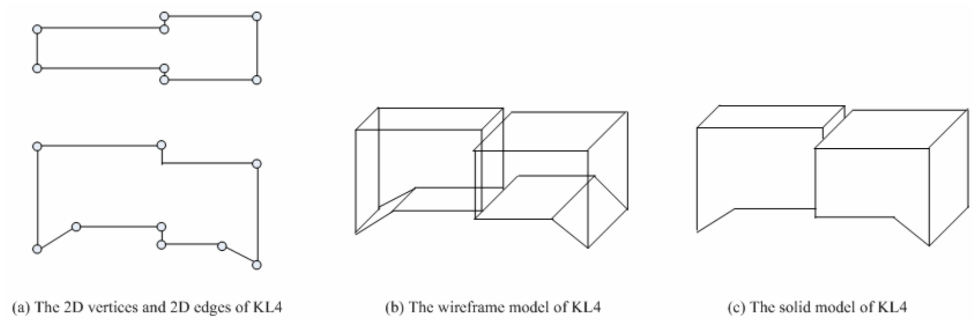


Fig. 7. (a) 2D vertices and 2D edges, (b) wireframe model and (c) solid model of beam KL4

4.3 Reconstruction of accessory parts

Structural objects may have accessory parts, which are usually implied by annotation texts and prior knowledge. For example, the text E200X200/350 in the beam table in Fig. 2 means that the first segment of the beam KL4 has a reinforcement part on one side. We first project the reinforcement part into a side view, then reconstruct the accessory part according to the front view and top view shown in Fig. 8(a). Finally, the solid model of KL4 is created using Boolean operations on the original solid model and the accessory parts. Fig. 8(c) shows the resulting wire-frame model after adding the accessory part, and Fig. 8(d) is the corresponding solid model.

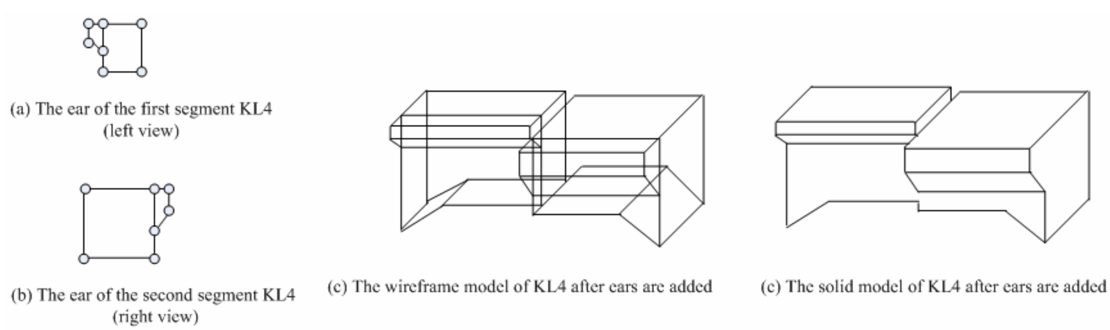


Fig. 8. Side view of the accessory parts, the final wireframe model and solid model of KL4

Using a similar method, holes can be reconstructed. Fig. 9 shows the solid model of slab S-1.

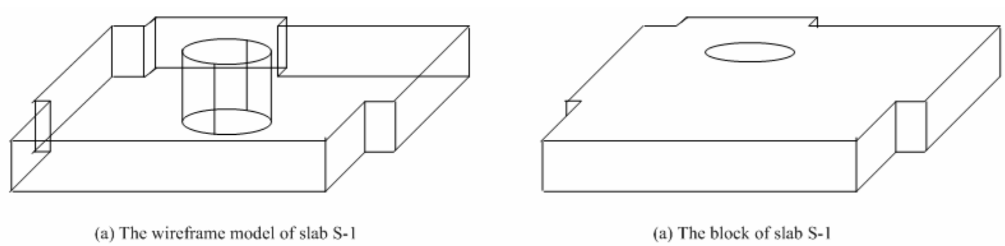


Fig. 9. The wire-frame model and solid model of slab S-1

4.4 Reconstruction of the detailed building

When all the structural objects are reconstructed, the whole 3D model is easily reconstructed by performing Boolean operations on the reconstructed solid models of the structural objects. Fig.10 shows the reconstructed building from the CSDs in Fig.2, consisting of a slab, four beams, and four columns.

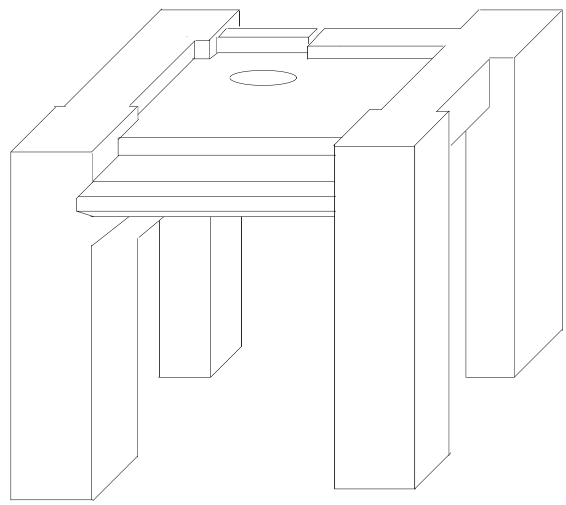


Fig. 10. The reconstructed example building from Fig.2

5. CONCLUSIONS

We have presented a new reconstruction method based on the integration of information distributed in multiple kinds of presentations. The proposed method mixes geometry with semantics in the reconstruction process, and is able to eliminate the complexity of architectural drawings and reconstruct a target detailed building efficiently.

The 3D reconstruction process has proven to be efficient and accurate for complex real-life drawings. For a typical 7 floors architecture project made up of about 308,000 graphic primitives (e.g. lines, arcs, solids, polylines), the unique solution of the 3D building can be achieved in less than 10 minutes (not including the recognition cost). The reconstructed 3D detailed building has been used in real-life quantity surveying and construction.

Our method has some limitations. (1) The reconstruction process strongly depends on the automatic recognition of architectural drawings. Therefore, structural objects which are hard to be recognized, i.e., foundations and roofs, are also hard to be reconstructed. (2) The 3D information to be retrieved is strongly context dependent and domain knowledge dependent. A generic solution may be impossible to reach. Further research includes accurate recognition of more presentations in real-life drawings (such as complex tables), the analysis of other types of architectural drawings, such as decoration drawings to reconstruct the interior of a building (doors, windows, ceilings, floor boards, etc), and the reconstruction of various internal steel structures within different types of structural objects.

ACKNOWLEDGEMENT

This work was partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No. HKUST6210/02E).

6. REFERENCES

[1] J. Monedero, Parametric design: a review and some experience, *Automation in Construction*, Vol. 9, No.4, 2000, pp 369-377.

[2] D. Dori and K. Tombre, From engineering drawings to 3D CAD models: are we ready now? *Computer-Aided Design*, Vol.24, No. 4, 1995, pp 243-254.

[3] IV. Nagendra, UG G, 3D objects from 2D orthographic view – a survey. *Computer Graphics*, Vol. 12, No.1, 1988, pp 111-114.

[4] H. Masuda, M. Numao, A cell-based approach for generating solid objects from orthographic projections, *Computer-Aided Design*, Vol.29, No. 3, 1997, pp 177-187.

[5] Z. Chen, DB Peng, Automatic reconstruction of solid objects from 2D orthographic views, *Pattern Recognition*, Vol. 21, No. 5, 1988, pp 439-449.

[6] M. Tanaka, et al, A single solution method for converting 2D assembly drawings to 3D part drawings, *Computer-Aided Design,* Vol. 36, No.8, 2004, pp 723-734.

[7] K. Tombre, Graphics recognition – general context and challenges, *Pattern Recognition Letters*, Vol. 16, No. 9, 1995, pp 883-891.

[8] S. Ablameyko, V. Bereishik, A. Gorelik and S. Medvedev, 3D object reconstruction from engineering drawing projections, *Computing & Control Engineering Journal*, Vol. 10, No. 6, 1999, pp 277 –284.

[9] K. Tombre, Ten years of research in the analysis of graphics documents: achievements and open problems, *Proceedings of 10th Portuguese Conference on Pattern Recognition*, Lisbon, Portugal*,* 1998, pp 11-17.

[10] C. Ah-Soon and K. Tombre, Variations on the analysis of architectural drawings, *Proceedings of Fourth International Conference on Document Analysis and Recognition*, Ulm (Germany), 1997, pp 347-351.

[11] R. Lewis and C. Séquin, Generation of 3D building models from 2D architectural plans, *Computer-Aided Design*, Vol. 30, No. 10, 1998, pp 765-779.

[12] 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, 2005, 1053-1069.

[13] M. Idesawa, A system to generate a solid figure from three views, Bull, JSME 16 1973, 92, pp 216-225.

[14] MA. Wesley. and G. Markowsky, Fleshing out projects, *IMB J. Res. Devel.,* Vol. 25, 1981, pp 934-953.

[15] M. Tanaka, K. Iwama and T. Watanabe. Decomposition of a 2D assembly drawing into 3D part drawings, *Computer-Aided Design*, Vol. 30, No. 1, 1998, 37-46.

[16] J. Song, F. Su, C.-L. Tai and S. Cai, An object-oriented progressive-simplification-based vectorization system for engineering drawing: model, algorithm, and performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 8, 2002, pp1048-1060.

[17] J. Song, F. Su, J. Cheng and S. Cai, A knowledge-aided line network oriented vectorization method for engineering drawings, *Pattern Analysis and Application*, Vol.3, No. 2, 2000, pp 142-152.

[18] X. Xi, W. Dou, T. Lu and S. Cai, Research on automated recognizing and interpreting architectural drawings, *Proceedings of the First International Conference on Machine Learning and Cybernetics (IEEE)*, Beijing, China, 2002, pp 1000-1004.