ISSN 2464-4617 (print)
ISSN 2464-4625 (CD-ROM)

WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016

Fast and Robust Construction of 3D Architectural Models from 2D Plans

Jalaj Pandey
IIIT-Delhi, India
jalaj13043@iiitd.ac.in

Ojaswa Sharma
IIIT-Delhi, India
ojaswa@iiitd.ac.in

ABSTRACT

In this work we present a simple and robust method to create 3D building models from a set of architectural plans. Such plans are created for human readability and thus pose some problem in automatic creation of a 3D model. We suggest a semi-automated approach for plan cleaning and provide an algorithm for alignment and stacking of the plans followed by generation of 3D building model. We show results of our method on floor plans that generate complex 3D models in near real-time.

Keywords

Architectural plans, 3D model creation.

1 INTRODUCTION

3D architectural models find applications in a number of fields including virtual reality, scientific simulations, military training simulations to name a few. Efficient and accurate creation of 3D building models is therefore very important for a large scale model creation process. In this paper we propose a fast and robust method to combine architectural floor plans and generate a complete 3D building model.

Starting with a set of 2D architectural floor plans, manual construction of a 3D model is tedious and involves a multitude of steps including creation of individual building levels, constructing various floors, interior and exterior walls, and pillars. Various building levels need to be aligned so that they can then be stacked up to create the basic building structure and lastly operations such as slicing and bridging need to be performed to place separately created windows and doors. In this paper we reduce the manual labor and time required to create 3D building models by automating the process to a major extent. Our system allows the user to semi-automatically generate 3D building models from 2D plans resulting in significant reduction of the time required to create models.

2 BACKGROUND

The fundamental basis for generation of 3D models of a building are the 2D architectural floor plans. These plans contain enough information about the architectural design and geometry of the building. Usually the architectural plans exist in two formats: vector and raster (scan of hand drawn floor prints). For computer assisted 3D model creation of buildings, most existing approaches use vector form of the floor plans as in Lewis and Séquin [5] and Zhu et al. [13], while some use the scanned images of the floor plans, by first converting them into vector form using image processing algorithms (for example, Xuetao et al. [11] includes both types of systems).

Many systems make use of scanned floor plans/raster images and convert them into CAD files or vector images using various pattern recognition and image processing techniques in order to retrieve architectural information. An extensive survey by Xuetao et al. [11] provides a detailed description of converting raster images as performed by various systems. Approaches (like ours) that use a vector format, allow the system to recognise basic information about the plan since most of the plan geometry is grouped and labeled uniquely. Easier storage, data processing and layered structure of vector formats (like AutoCAD DWG/DFX) have made their use more popular in construction of architectural floor plans.

So et al. [10] proposed one of the first automated system for generation of 3D buildings. The system automated wall extrusion along with ceiling and floor construction. However, the constructed models require manual intervention in order to get usable results. The automation process was only able to reduce the time required to construct the final 3D building to some degree.

Moloo et al. [8] developed a software that uses a 3-phase recognition approach to generate 3D building from 2D floor plans. In their approach the authors represented a wall by grouping lines into bounding boxes. They aimed at automating the process of 3D building generation by analysing the 2D floor plan.

Or et al. [9] proposed a highly automated approach to generate 3D model from 2D floor plans. The system takes as input a set of raster images of floor plans and converts these into a set of vector images. Symbol recognition is used to identity symbols for doors and windows. Once Symbol recognition is completed, the system generates 3D buildings and imports these in Genesis 3D game engine. The system generates 3D buildings by analysing relationship of connected segments, that are created by parsing floor plans into connected segments.

Dosch et al. [3] presented a complete system that aims to reconstruct 3D buildings by analyzing scanned 2D architectural drawings. The system is divided into two steps, 2D modeling step in which they describe a robust graphics recognition algorithm that is used by the system for image processing and feature extraction. They describe this by dividing the raster 2D image into tiles, processing them individually and finally merging them after vectorization. For feature extraction the system makes use of a skeleton based approach, where extracted lines are represented as segments using polygon approximation technique. Next they propose a 3D modeling process that is used to match the reconstructed floors. Also the system also provides a user interface that is flexible and capable for human interaction.

Horna et al. [4] described a system that presents a four phase construction method. Their algorithm starts with removing geometrical inconsistencies by processing 2D edges. This is followed by topology generation using semantic information, and extrusion for 3D building construction. Lastly the floor is superimposed corresponding to upper and lower ceilings that are linked by stairways to construct the final model. The complete topological model constructed by the system expresses incidence and adjacency relations between elements. Further, the system also associates semantic information with all volumes for specifying structures like rooms and walls. The system takes the semantic information into account during extrusion to 3D.

Ahmed et al. [1] proposed a complete system for automated floor plan analysis. The system helps to apply and improve the present processing methods, further it also introduces preprocessing methods that improve the performance of the system. Some of the techniques used is the differentiation between thick, medium and thin lines and removing components that lie outside the convex hull of the outer walls.

The Berkeley WALKTHRU system , developed by Lewis and Séquin [5], is a semi-automatic system capable of generating 3D polyhedral buildings from AutoCAD DXF format with minimal user interaction. This system uses the concept of portals and spaces for stacking of floors in a multi-storey building. The models generated are a solid representation of the real buildings that can be used to develop a virtual walkthroughs and computer rendering. According to Zhu et al. [13], even though this approach involves minimal human intervention, it takes days to create a complex 3D model.

Zhu et al. [13] proposed a system to construct 3D building models automatically from vector floor plans by analysing semantic information and geometry from the plans. Their system made use of defined axes in each floor plan for stacking. The basic idea was to align every floor to the first floor plan by matching/equating their respective axes. Two axes are equal when they are of the same type and have the same label. The complexity of this algorithm is $O(n^2)$. The authors describe several interesting 3D building creation methods. The one by Zhi et al. [12] automatically creates a fire evacuation building simulator model, while the one by Domínguez et al. [2] introduces an interesting semi-automatic approach that detects the topology of building floors. Lastly the ones by Lu et al. [6, 7] indicate component types by making use of architectural drawings without labels and computer drawn construction structural drawings.

In this paper we present a system that allows the user to create 3D architectural models. Our novel contribution to the state-of-the-art is a robust algorithm for alignment and stacking of floor plans in absence of a common 2D coordinate system across plans. We provide a semi-automated end-to-end approach for creating usable 3D building models.

3 APPROACH

Our 3D model construction approach takes 2D architectural plans in a CAD vector format. The algorithm performs plan cleaning based on object attributes. The cleaned floor plans are then aligned and stacked by the system at their respective heights in the building. The building level heights are input parameters to our system. This is followed by generation of 3D model of the building by extruding the floor horizontally and the walls and other entities in the building vertically. Finally the process ends by performing a slicing operation that is used for placement of doors and windows. Algorithm 1 summarises our four-step approach to 3D building creation.

Next we describe steps of our proposed approach. The system is supplied with raw architecture vector plans. The process for generation of the 3D model in our system is divided into four basic steps.

3.1 Plan cleaning

The first step of the approach is to clean the raw vector plans so that they can be supplied to the system for fur-

Input: 2D floor plans, building parameters
Output: 3D building model
PLAN CLEANING
    Removal of text, and annotation tags
    Detailing identification and removal
PLAN ALIGNMENT AND STACKING
    Creation of tree of architectural elements
    Alignment using elements of structural stability
FLOOR GENERATION AND EXTRUSION
    Floor generation by horiozontal filling
    Vertical extrusion of walls
SLICING AND WINDOW PLACEMENT
    Creation of space for windows and doors

Algorithm 1: 3D building generation approach.

ther processing. An architectural plan consists of multiple elements required in a building design. Apart from these, the plans also contain text and supportive layout elements (like icons, grid lines, and guides). Only a subset of these elements is required for 3D model creation. Such elements include exterior and interior walls, columns, and elevator spaces. Plan cleaning is a complicated task and an automated approach requires machine recognition of these elements (see [1, 2, 3]). As a result, a complete analysis and recognition is not possible in all scenarios, and some geometrical elements needs user intervention to be removed in the cleaning process. We adopt a semi-automated approach to cleaning plans by combining geometric analysis with minimal user interaction. Earlier works have also resorted to processing raster 2D plans, which in our opinion is not only difficult but also leads to inaccuracies in final 3D reconstruction. We operate on vector 2D plans and assume that various elements have been tagged in some way by the architect who authored the plans.

Our plan cleaning follows a two step procedure. In the first step, we identify relevant elements within a plan by analysing the tags attached to each one of these and grouping similar elements together. Similarity of elements in this context refers to elements/objects falling under the same category such as doors and window, lifts (defined as portals in [2, 5]), and walls. Such an analysis is carried out by means of string matching with regular expressions on element text attributes. The system tries to extract element type attributes automatically with this analysis and presents it to the user for verification. The user can quickly correct attribute names or reject erroneous results. The system then removes unnecessary attributes and renames required attributes consistently.

Once the attributes are processed and elements are grouped, the system performs a geometric analysis to clean individual elements. This step is similar to the processing proposed in [2, 5], and primarily looks at geometric integrity of various elements required for extrusion in a subsequent step. Figure 1 shows result of cleaning on one of the plans.

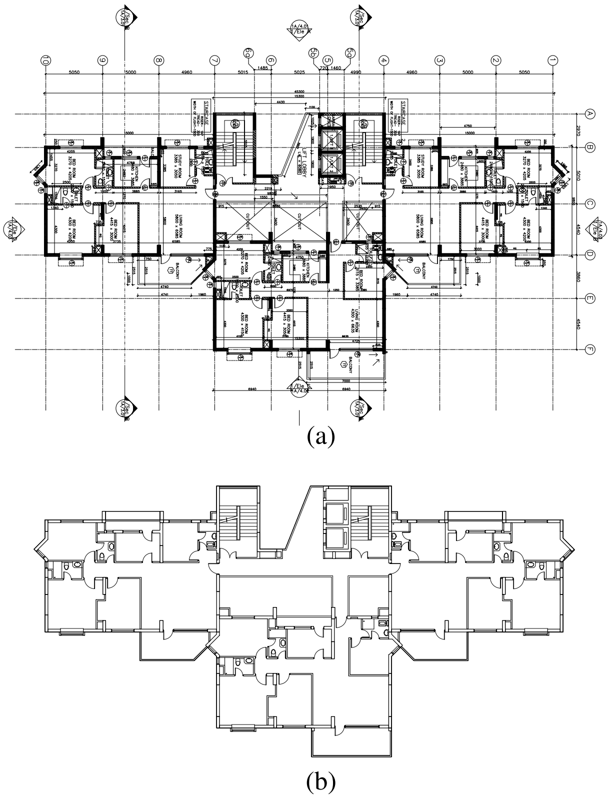


Figure 1: Result of cleaning on a highly detailed floor plan.

3.2 Plan alignment and stacking

Aligning plans of multi-floored models poses a challenge in generation of 3D building models [5, 13]. The process is complicated by the fact that floor plans are designed separately for multi floor buildings, thus each floor plan is designed with its own local 2D coordinate system. In our experience, these local coordinate systems do not always align with each other and thus it is not straightforward to stack up these plans for extrusion. Our system resolves the alignment problem by creating a hierarchical parent-child relationship between elements of the floor plan, which is the pivotal part of the algorithm.

The system prompts the user to identify architectural elements that identify structural stability and use those to align all floor plans. Such elements will always be aligned in any given building to satisfy structural stability (these include load bearing pillars, lift spaces and stairs). Such elements are grouped together as root element of the hierarchy in a plan and other elements are arranged below these. In our tree hierarchy, the stability element that is identified, acts as a pivot to which all other layers are attached. Consider this as a tree with one root and the remaining layers of the floor plan as its child nodes. The only property this parent-child relationship follows is that whenever the parent moves the child nodes move relative to it, thus not changing their respective positions among themselves, but if the user moves any child node, the parent or other nodes remain unaffected. The system iteratively aligns all floor plans by estimating a rigid transformation (including translation and rotation) between root elements from two plans at a time. All elements within a plan are transformed to a common coordinate system where all plans are aligned to each other. This gives us a robust approach to align plans under any circumstance. Many-a-times the floor plans are complex and asymmetric, but with our approach those are handled very well.

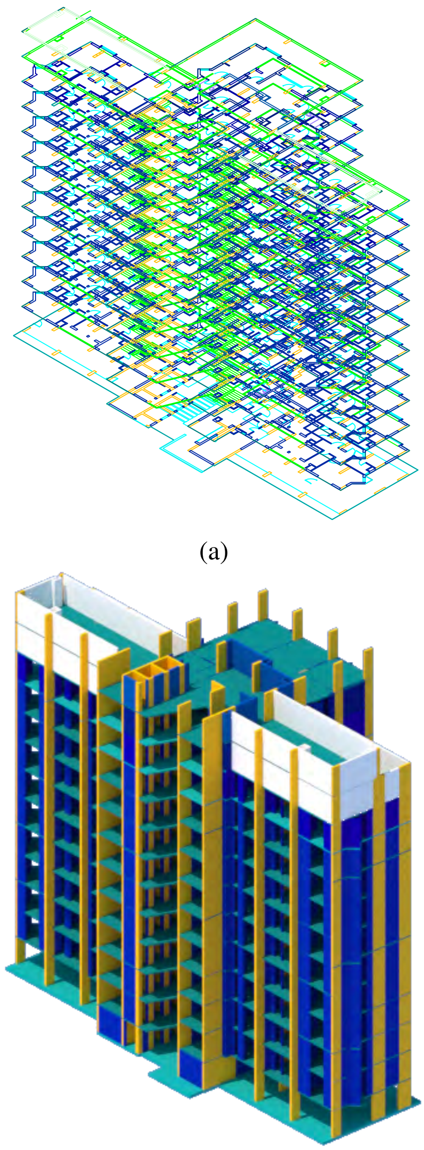
The floor plans are stacked once they are aligned. In our system we again use the tree root to stack the floor plans. The system is supplied the respective heights of all the floor plans from which each plan is transformed and placed at its respective height.

3.3 Floor generation and extrusion

We assume that geometric elements in a 3D building model are composed of extruded elements. Extrusion is a process of creating a cylindrical or planar element from its 2D footprint (a circle or a line segment) by extending it in the third dimension. For creation of a 3D model from stacked 2D floor plans, two types of geometries are required: horizontal floors and vertical walls. We utilise region fill and extrusion for creation of these.

Our system first generates floor polygons from wall boundaries by creating planar horizontal faces. Exterior and interior walls (and load bearing pillars) are then extruded vertically. This is performed completely automatically and creates a 3D model with solid faces. The system is capable of creating planar and cylindrical geometry by extrusion. Walls, floors and ceilings generated by our system have finite width (and are not merely thin planar geometry). In special cases, it can also handle surfaces of revolution by minimal modification (e.g. spherical geometry for tombs). Roof of a building is generated by replicating the plan of the top floor and placing it at the required height. A simple hipped roof, as seen in contemporary European-style buildings, may be generated by creating slanted planes guided by the top floor elevation. Figure 2 shows a complete extruded 3D model from stacked floor plans (the roof is removed to improve visualization).

Complicated elements like staircases require special treatment. These can be generated by combining information from both the plan and the elevation of a floor. A floor plan usually depicts steps in a staircase, while the elevation contains information about the slope and stride. Staircase generation is not currently implemented in our system.



(a)

(b)

Figure 2: Creation of extruded 3D building model. (a) aligned and Stacked 2D floor plans in 3D space, (b) extruded 3D building model.

3.4 Slicing and window placement

The 3D model constructed so far comprises of solid walls with no windows and doors. For a complete building creation, these structural elements are important. Our system handles window and door creation by slicing the 3D extruded model horizontally at various levels and creating placeholders for windows and doors. Input parameter to slicing is heights of door and window elements. The system extracts the width from the respective floor plan and creates a placeholder geometry for the window/door element. Detailed window and door 3D models can then be easily added to these models by the user.

4 RESULTS

We show results of our system with a set of complex floor plans of various campus buildings. We implemented our system completely in Autodesk 3ds Max using the MAXScript programming interface. All of our result are produced on an Intel Core i7 2.4 GHz processor with 6 GB memory (on a laptop computer).

In the results shown below, we purposefully omitted the building roofs for better visualisation of results and to highlight complexity of our models. Figure 3 shows two views of the 3D model of Academic block that consists of three distinct wings arranged at an angle to each other. Other reconstructions include a horseshoe shaped hostel building (see Figure 4) and the student center (see Figure 5). Figure 2(b) illustrates reconstruction of a 11-storey residence building.

Table 1 shows runtime information in seconds for generation of full 3D model using available floor plans. We note that these numbers depend on the complexity and number of the floor plans. These numbers show that our system is capable of generating highly detailed 3D models in near real-time on commodity mobile computer hardware.

Various tasks in 3D model creation can be automated to various degrees. We achieve a high automation in several intermediate steps. This is summarised in Table 2 along with comparison across existing methods.

5 CONCLUSION

In this paper, we presented a simple and robust system for 3D model generation from a set of unaligned 2D architectural floor plans. We illustrated our alignment algorithm that is a pivotal part of our system and allows us to align the floor plans and thus proceed with more general operations like extrusion to construct the 3D models. Currently our algorithm cannot identify the stability element on its own, which we would like to address in future. Also we plan to provide texture support in next version of our system. Generation of non-vertical complex structures is challenging. Such architectural geometries need to be handled on a case-by-case basis. In general, these may be split into multiple parts which may either be extruded along a non-standard axis or modelled by minimal surfaces. We would like to address these challenges in future.

6 ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

7 REFERENCES

[1] Sheraz Ahmed, Marcus Liwicki, Markus Weber, and Andreas Dengel. Improved automatic analysis of architectural floor plans. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 864–869. IEEE, 2011.

[2] B Domínguez, ÁL García, and Francisco R Feito. Semiautomatic detection of floor topology from CAD architectural drawings. *Computer-Aided Design*, 44(5):367–378, 2012.

[3] Philippe Dosch, Karl Tombre, Christian Ah-Soon, and Gérald Masini. A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition*, 3(2):102–116, 2000.

[4] Sebastien Horna, Guillaume Damiand, Daniel Meneveaux, and Yves Bertrand. Building 3D indoor scenes topology from 2D architectural plans. In *GRAPP (GM/R)*, pages 37–44. Citeseer, 2007.

[5] Rick Lewis and Carlo Séquin. Generation of 3D building models from 2D architectural plans. *Computer-Aided Design*, 30(10):765–779, 1998.

[6] Tong Lu, Chiew-Lan Tai, Feng Su, and Shijie Cai. A new recognition model for electronic architectural drawings. *Computer-Aided Design*, 37(10):1053–1069, 2005.

[7] Tong Lu, Huafei Yang, Ruoyu Yang, and Shijie Cai. Automatic analysis and integration of architectural drawings. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(1):31–47, 2007.

[8] Raj Kishen Moloo, Muhammad Ajmal Sheik Dawood, and Abu Salmaan Auleear. 3-phase recognition approach to pseudo 3D building generation from 2D floor plan. *arXiv preprint arXiv:1107.3680*, 2011.

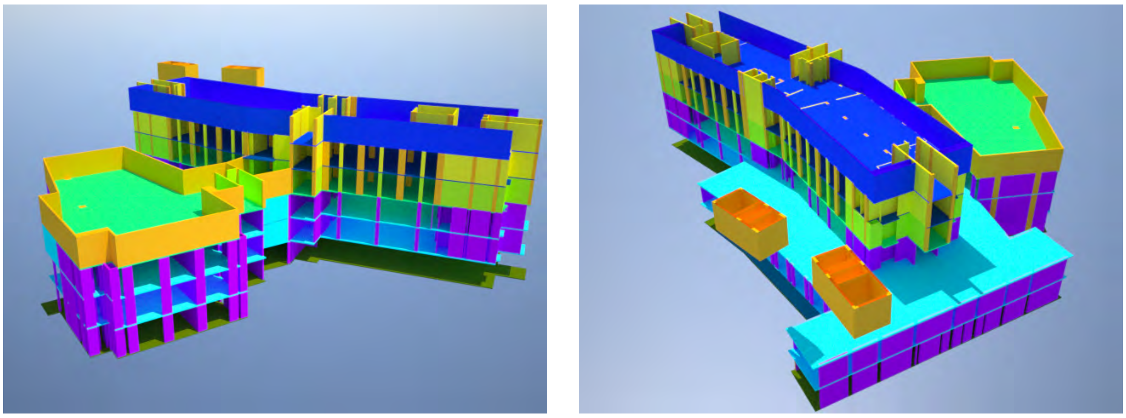
[9] Siu-Hang Or, Kin-Hong Wong, Ying-kin Yu, Michael Mingyuan Chang, and H Kong. Highly automatic approach to architectural floorplan image understanding & model generation. *Pattern Recognition*, pages 25–32, 2005.

[10] Clifford So, George Baciu, and Hanqiu Sun. Reconstruction of 3D virtual buildings from 2D architectural floor plans. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 17–23. ACM, 1998.

[11] Xuetao Yin, Peter Wonka, and Anshuman Razdan. Generating 3D building models from architectural drawings: A survey. *IEEE Computer Graphics and Applications*, (1):20–30, 2009.

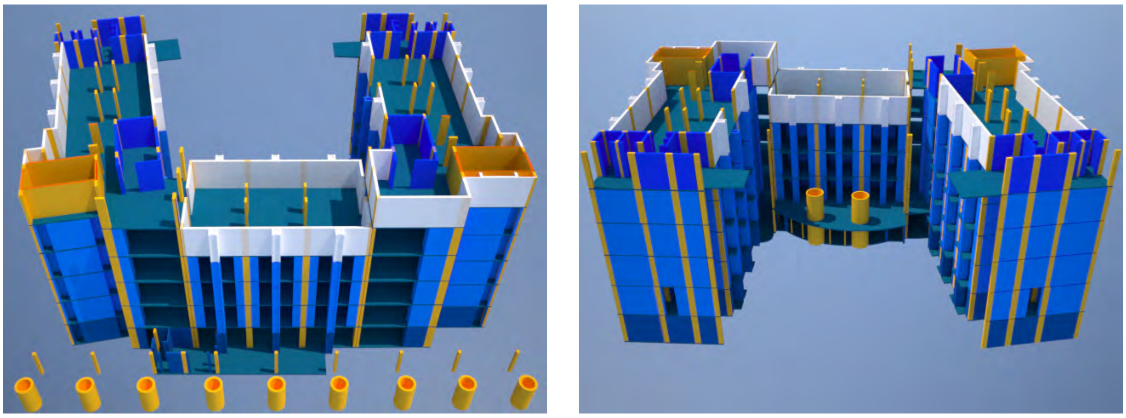
[12] GS Zhi, SM Lo, and Z Fang. A graph-based algorithm for extracting units and loops from architectural floor plans for a building evacuation model. *Computer-Aided Design*, 35(1):1–14, 2003.

[13] Junfang Zhu, Hui Zhang, and Yamei Wen. A new reconstruction method for 3D buildings from 2D



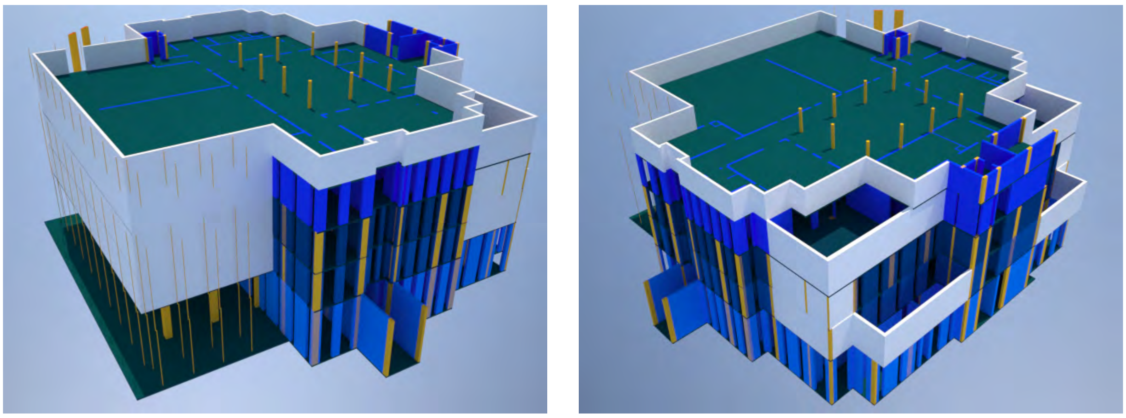
(a) (b)

Figure 3: Reconstructed Academic block model (a) front view, (b) back view.



(a) (b)

Figure 4: Reconstructed Hostel building (a) front view, (b) back view.



(a) (b)

Figure 5: Reconstructed Student center (a) front view, (b) back view.

Building	*# Input plans*	*# Output triangles*	*Without slicing (sec.)*	*With slicing (sec.)*
Faculty residence	13	130,498	15.280	29.160
Student center	5	41,166	7.083	10.183
Hostel	6	110,312	6.810	13.342
Academic block	6	86,984	7.302	14.784

Table 1: Construction times for entire model generation.

Process	*Lewis and Séquin* [5]	*So et al.* [10]	*Lu et al.* [7]	*Dosch et al.* [3]	*Or et al.* [9]	*Ours*
Vector plan input	Yes	Yes	Yes	No	No	Yes
Plan cleaning	Semi-Automatic	No	Manual	Manual	No	Semi-automatic
Plan alignment and stacking	Semi-automatic	No	No	Automatic	No	Semi-automatic
Floor generation and extrusion	Automatic	Semi-automatic	Automatic	Semi-automatic	Automatic	Automatic
Slicing and window placement	Automatic	Manual	Automatic	Automatic	Automatic	Automatic
Overall automation	High	Low	Medium	High	Medium	High

Table 2: Comparison of degree of automation with various approaches.

vector floor plan. *Computer-Aided Design and Applications*, 11(6):704–714, 2014.