Building 3D indoor scenes topology from 2D architectural plans
Sébastien Horna, Guillaume Damiand, Daniel Meneveaux, Yves Bertrand

To cite this version:
BUILDING 3D INDOOR SCENES TOPOLOGY FROM 2D ARCHITECTURAL PLANS

Sebastien Horna, Guillaume Damiand, Daniel Meneveaux, Yves Bertrand
SIC laboratory, University of Poitiers, France
{horna,damiand,meneveaux}@sic.univ-poitiers.fr, bertrand@univ-poitiers.fr

Keywords: 3D modelling, architectural scenes, reconstruction, topological model.

Abstract: This paper presents a new method for reconstructing geometry and topology of 3D buildings from 2D architectural plans. A complete topological model expresses incidence and adjacency relations between all the elements. It is necessary for both recovering accurately 2D information and constructing a coherent 3D building. Based on an existing topological kernel, several high-level operations have been developed in 2D for creating walls, portals, stairs, etc. Semantic information is associated with all volumes for specifying openings, walls, rooms, stairs, facade, etc. The resulting 2D model is extruded for generating a 3D environment, taking the semantic information into account since doors are not processed as walls for instance. Floors are superimposed using volumes corresponding to upper and lower ceilings linked according to stairways. The resulting models are suitable for various applications such as walkthrough, lighting/wave propagation/thermal simulation.

1 INTRODUCTION

Accurate three dimensional descriptions of architectural environments is an important need for many building trades such as lighting engineering, thermal simulations, etc. Generally, the models produced by the architects are handled in two dimensions without any topological information. However, in addition to a three dimensional description of the model, many simulation algorithms require adjacency and incidence relationships between volumes.

Unfortunately, manually reconstructing complex architectural scenes using a 3D modeller is a long and tedious process. This is why we propose a new method for automatically reconstructing 3D buildings from 2D architectural plans (figure 1). Our method inherently integrates a complete topological description of the environment. The resulting scenes can thus be edited in a topological modeller for adding furniture, moving walls, etc. In addition, semantic information is used for defining object attributes such as rooms, floor, corridors and so on.

![Figure 1: 2D plan and 3D reconstruction.](image)

Our aim is to define a building model corresponding to a 3D partition of space. Each room should correspond to a closed and orientable volume, incident to closed and orientable faces. Amongst the existing topological models, we have chosen generalized maps which allow to represent space subdivisions and incidence and adjacency relations.

The reconstruction method we propose is based on four main phases: (i) 2D edges processing for removing geometrical inconsistencies, (ii) topological reconstruction with semantic information, (iii) 3D building extrusion (iv) superimposing of floors.
During the first phase, most 2D imprecisions (90%) are automatically corrected. However, remaining awkward elements can be semi-automatically processed. Semantic information can be deduced from the plans when they exist, or defined manually by the user. The extrusion operation is guided by the semantics. For instance, walls, doors or stairs are extruded using specific rules. Superimposing of floors is applied using semantics and topology.

The resulting topological indoor models have been designed so that volumes adjacency can be efficiently used for various types of simulation. For instance such models prove efficient in the context of radiosity or photon mapping approaches [11, 6].

This paper is organized as follows. Section 2 describes the existing methods for 3D architectural scenes construction. Section 3 justifies our choice of generalized maps. Section 4 presents our method of geometrical and topological rebuilding in two dimensions. Section 5 details the extrusion and superimposing of several floors. Section 6 discusses the results obtained with several examples of real buildings.

2 RELATED WORK

Many methods in the literature propose to rebuild urban environments. For instance, the MATIS team in the research section of the French Institut Geographique National (IGN) proposes an elevation method starting from satellite photographs [7]. Ah-Soon et al. processes digitized 2D drawings (i.e. images of plans) for 3D reconstruction [1]. Recognition is based on the detection of vertical and horizontal symbols. The aim is to analyze the interior geometry of a building as well as the openings location (doors, windows, etc). This work primarily concerns methods of image analysis. The geometry reconstruction produces 3D scenes without topological information.

Several methods aim at extracting topological information from a list of polygons, making it possible to reduce calculations of visibility for lighting simulation and visualization. Airey et al. propose a method of binary space subdivision (Binary Space Partitioning or BSP) for axis-aligned environments [2]. Teller et al. present an extension of this method for all types of walls [15]. Meneveaux et al. propose a method containing rules to find the parts of the buildings [12]. All these subdivision schemes produce a set of regions called cells, separated by openings. The topological description corresponds to adjacency relations between 3D cells, there no incidence/adjacency relations between lower-dimensions elements.

Complex urban scenes can also be produced using procedural modeling, such as cityEngine [14, 13]. Several parameters can be taken into account: population density or height maps. The road network is generated using a L-System mechanism. A construction grammar is used to create the building facade. Although these methods generate realistic (but not real) geometrical environments, topological information is not managed.

3 GENERALIZED MAPS

We wish to represent buildings made up of volumes (floors, walls, rooms, etc), each of them being a orientable 3D object. We need a subdivision of space into faces, edges, vertices, defined by their boundaries (boundary representation) in any dimension. Maps and generalized maps offer an implicit representation of cells with efficient operations since a local modification in the map is automatically propagated to the incident edges.

Several topological models allowing space subdivisions have been proposed in the literature: structures containing adjacency graphs [4], 2D/3D models based on edges [3, 8, 18] or models capable of handling higher dimensions [4, 10]. As explained in the following section, many reasons have motivated our choice for generalized maps [10].

It has been shown in [9] that existing topological models representing 3D subdivisions are comparable with 3D maps (for orientable models without bound-
ary) or with 3D generalized maps (for orientable or not models, with or without boundary). Even though 3D generalized maps are more expensive than 2D maps in the memory, we have chosen this model since it provides a homogeneous representation in all dimensions. This simplifies many operation definitions.

From a single type of basic elements (called darts) and one to one mappings \( \alpha \) defined on these darts, generalized maps represent object cells and adjacency/incidence relationships. Each mapping \( \alpha_i \), with \( 0 \leq i \leq n \) (\( n \) being the highest dimension used), represents the adjacency relations between \( i \)-dimensional cells;

Definition 1 (Generalized map [10])

A generalized map in dimension \( n \geq 0 \) (or \( n \)-G-map) is an algebra \( G = (D, \alpha_0, \ldots, \alpha_n) \), where:

- \( D \) is a finite set of darts;
- \( \alpha_0, \ldots, \alpha_n \) are involutions \(^1\);
- \( \alpha_i \alpha_j \) is an involution for all \( i, j \) such that \( 0 \leq i < i + 2 \leq j \leq n \).

Two darts \( d \) and \( d' \) are \( \alpha_i \)-sewed if \( d \alpha_i = d' \) with \( d \neq d' \), and \( d \) is \( \alpha_i \)-free if \( d \alpha_i = d \). The \( i \)-cell associated with a given dart \( d \) is composed of all the darts obtained by a coverage starting from \( d \) and using all the involutions except \( \alpha_i \) (see figure 2). The number of distinct edges incident to a vertex defines its degree. When the vertex is incident to only one edge, the edge is called a dangling edge.

On the basis of this representation, we have used the 3D topological modeller MOKA [17], comprising many operations such as sewing two cells along a face or more complex operations like sweeping or corefining.

4 2D RECONSTRUCTION

For extruding a 3D building, a valid topology has to be reconstructed from the 2D plan. Therefore, the dataset has to comply with three fundamental properties: (i) edges should not be merged, (ii) edges should not intersect (iii) edge vertices should all be incident to another edge. (i) and (ii) ensure that the plan is a partition of a 2D space in faces, edges and vertices. In a 2D architectural plan every object is usually defined with a given thickness. Consequently edges should not be isolated, which corresponds to (iii). When these 3 properties are verified, the set of edges is said valid.

Unfortunately, modeller software used by architects is not devised for 3D topological extrusion. Consequently, in most cases, none of the above properties is maintained. The reconstruction robustness of our method highly depends on the detection and correction of all geometrical inconsistencies contained in the 2D plans. Our application corrects geometry and builds up the topological model. It is composed of two parts: the first one consists of geometry error detection and correction while the second one constructs topological information. The final goal is to link edges so as to produce 2D faces.

In practice, for the plans we have used, 90% of imprecisions are automatically corrected. However, some remaining awkward elements have to be processed. Therefore, we propose semi-automatic operations for correcting the plans (see section 4.4).

Semantic information can be deduced from the plans when it exists, or defined manually by the user. Finally, each type of object contained in plans is associated with semantics: walls, rooms, openings, stairs, etc.

The general algorithm of 2D reconstruction is broken up into the following steps: (1) edge extraction from source file (2) geometry correction (3) topological construction (4) semi-automatic finalization (5) semantics association.

4.1 Geometrical correction

Once the edges have been identified in the source file, the plan analysis is performed. Therefore, a threshold \( \varepsilon \) is defined for testing whether two edges are superimposed and finding all the edges incident to a given vertex. In practice, we have fixed \( \varepsilon = 1 \) mm.

Two edges are superimposed if they have the same slope, the same origin ordinate and at least one extremity included in the other edge. In this case, both edges are merged into a single one.

All the edge intersections are processed two by two. If an intersection is found, a vertex is added at the intersection point on the concerned edges.

\(^1\)A bijection \( f \) is an involution iff \( f^2 = Id \)

\(^2\)If \( \beta \) and \( \gamma \) are applications of \( E \rightarrow E \), \( \beta \gamma \) corresponds to the composition \( \gamma \circ \beta \), and \( \beta \gamma \) is the application of this composition to element \( b \) of \( E \)
4.2 Topological reconstruction

The above processing produces a set of valid segments used to construct topology. All the adjacency and incidence relationships between vertices, edges and faces have to be defined.

4.2.1 Edges creation

Each edge is associated with four darts corresponding in 2D to an edge shared by two faces. Links \( \alpha_0 \) and \( \alpha_2 \) are immediately set on the corresponding darts. Only \( \alpha_1 \) remain to be processed for creating faces.

Since buildings are orientable objects, composed of orientable elements, we also need to set an orientation to the whole generalized map. This is why darts are associated with a boolean mark indicating the edge orientation. For a dart \( d \) marked, \( d\alpha_0 \), \( d\alpha_1 \) and \( d\alpha_2 \) are not marked.

4.2.2 Angle arrangement

For 1 and 2-degrees vertices, the corresponding darts are directly connected by \( \alpha_1 \). For each vertex of higher degree, the incident edges are stored and sorted according to their angle around the vertex. The algorithm is the following:

1. search for dart \( d \alpha_1 \)-free;
2. search for all darts \( \{d_i\} \alpha_1 \)-free, incident to \( d \);
3. sort \( \{d_i\} \) according to the angle with \( d \) (corresponding to the angle formed by the associated edges);
4. \( \alpha_1 \)-sew the darts two by two according to this order, with respect to the orientation constraints.

4.2.3 Face inclusion

In most plans, some objects are included in others. For instance, stairs are included in rooms. Unfortunately, with boundary representations, these objects are not connected. Consequently, there is no relative position between elements. This is why we have used fictive edges for linking the existing connected components. On the floor, a fictive edge is thus used to link an external face to the included ones. These edges are called fictive edges since they do not represent the boundary of a face.

4.3 Accelerating structure

With the process described above, many operations require testing couples of darts according to their location in the plans. The use of an accelerating structure makes it possible to reduce the processing time. Since the plans scale is defined in meters, we choose a uniform grid made up of \( 1\text{ meter} \times 1\text{ meter} \) tiles. Each tile is associated with the list of segments which cross it. Thus, for each segment, tests are performed only in a local neighborhood. Note that segments corresponding to walls only belong to a few tiles.

4.4 Additional operations

To eliminate inconsistencies that are not automatically corrected, we propose several semi-automatically operations. Based on the low-level operations \textit{sew} and \textit{removal} defined in [5], we propose higher-level operations for processing several edges at the same time: (i) for sewing two selected edges; (ii) for sewing several selected \textit{dangling edges} to the closest edge; (iii) for sewing all the \textit{dangling edges} to the closest edge; (iv) for topologically removing \( n \) selected edges.

It can be necessary to add doors on the plans. We also propose an operation for creating a door, starting from the selection of two walls. A door is inserted in the plan, and associated with its semantic.

4.5 Semantic definition

Semantic information allows the user to know the type of each element in the plan. The objects are classified into various categories: walls, doors, floor, ceiling, stairs, etc. Any type of new information can be conveniently added to the model. In practice, each dart holds a label corresponding to its semantic.

During the reconstruction process, it is possible to use the layers contained in the source file for indicating the semantic of objects. The user can also select part(s) of the building and manually modify semantics. This information is used for guiding the extrusion process described below.

5 3D EXTENSION

The starting point of the 3D extrusion is a 2D plan composed of faces, edges and vertices associated
with a consistent topology (i.e., a constant 2D
partition, closed and without dangling edges). Each
floor is handled using several types of operations.
Therefore, we have adapted the extrusion already
existing in the MOKA library. Each type of element
is specifically processed.

The topological 3D representation has to comply
with several important features.

1. The 3D model must be a closed space since each
room, wall and portal are defined as a closed vol-
umes. For instance, rooms are defined by volumes
with transparent faces corresponding to portals.
Consequently, each face should be incident to ex-
actly two volumes. Obviously, faces also have to
be closed as well as edges.

2. Each building must be composed of a single 3D
connected component. For instance, faces or vol-
umes defining holes have to be connected with
their respective faces or volumes.

3. The model must be oriented since each part of the
building should be clearly identified as the inside
or the outside.

These constraints are guaranteed by the properties
of the 2D plan and the extrusion operation.

Finally, the building extrusion is organized as fol-
lows: (i) extrusion of the floor (wall, doors), (ii)
creation of the ground and ceiling, (iii) superposition
of floors, (iv) stairs construction.

5.1 Extrusion of walls

For extruding walls, a vertical path is defined with a
height equal to 2.5 meters. From each face of the
2D plan, a volume is automatically created and con-
ected using $\alpha_3$ to the corresponding face (see
figure 3). Contrary to existing modeler, our topological extru-
sion of two faces connected by $\alpha_2$ produces two
volumes connected by $\alpha_3$. Note that fictive edges are
not extruded so that no useless fictive face be created.

The extrusion operation defined above allows the
construction of non-vertical walls (the extrusion path
has to be properly defined). However, in most cases
the slope is not defined on the plans. Moreover, the
modeler MOKA can be easily used for modifying the
upper wall edges.

5.2 Extrusion of openings

On the plans, door volumes are topologically con-
ected to the surrounding walls. The portion of

wall above to the door is created. Therefore, the 2D
polygon representing the door is extruded accord-
ing to a vertical path of two components (correspond-
ing to the opening and the portion of wall above the
door). The two resulting volumes are superimposed
and topologically connected. In a second step, they
are connected to the remainder of the building, along
the door stiles. Two vertices and an edge must
be inserted on the stiles to respect the topological
constraints (figure 4).

For windows, the same operation can be applied
with an additional component corresponding to the
wall part located under the window.

5.3 Creation of grounds and ceilings

With the extrusion system described above, the darts
of the 2D plan are $\alpha_3$-connected to volumes defining
walls or doors (figure 5.a).

The 2D plan is used to create the ground volume
(flagstone). Therefore, the contour of the 2D plan
is $\alpha_1$ and $\alpha_3$-disconnected (figure 5.b) and the corre-
sponding edges are used to form the desired volume
(Figure 6). The external faces of this volume are $\alpha_2$-
sewed with the faces representing the floor contour
(figure 7).
The construction of the ceiling requires the copy of the 2D plan at the top of the floor (Figure 8.a). During duplication, each new dart is sewed by $\alpha_3$ to its corresponding dart. The external segment is used to construct the ceiling volume (Figure 8.b). The external faces are sewed by $\alpha_2$ with the darts of the 3D floor contour. This operation produces 4 types of volumes: ground, ceiling, indoor description, facade.

5.4 Superimposing of floors

For superimposing two floors with same outer 2D shape, the ground of the upper floor is connected to the ceiling of the lower floor (Figure 9). In practice, the two volumes are sewed by $\alpha_3$ and the shared face is removed. Thus, only one volume defines the space between the two floors.

5.5 Creation of stairs

Stairs can be defined with various shapes on the 2D plans: straight, snail, elliptic, etc. They are often disconnected from the rest of the plan or joined to the walls. We propose a generic method for creating the stairs topology. The steps geometry is computed according to the data recovered on the plan (length, width, or diameter in the case of spiral stairs). Presently, our method does not provide any automatic system for detecting the geometric type of stairs, the user manually selects the appropriate method.

Each step is composed of two volumes (figure 10.a). A surrounding volume ensures the model closeness (figure 10.b). Once created, the 3D stair is connected to the remainder of the plan by a fictive edge. The ceiling is perforated according to the stair shape using of a boolean operation. Therefore the stair contour is extruded according to the ceiling height and the resulting volume is subtracted from the ceiling so as to create the opening (Figure 11).
6 IMPLEMENTATION AND RESULTS

Our reconstruction system has been implemented in C++, using the MOKA library [17]. The source files have been produced by architects in dxf format. Computing times have been obtained with a centrino processor: 2Ghz and 1GB of RAM.

We have applied our reconstruction method to various 2D architectural plans. The processing times for the 2D analysis are presented Table 1. They depend on the distribution of the segments in the plan and on the uniform grid acceleration. The processing of a floor never exceeds one minute, even for complex plans.

With the proposed method, 90% of the plans incoherencies have been detected and corrected automatically. The time required for manually correcting the 2D models incoherencies is about a few hours, depending on the model complexity and the numbers of errors contained in the plan. From this point, the 3D reconstruction becomes completely automatic and takes only a few seconds. Moreover, all the topological and semantical information are automatically propagated.

7 CONCLUSION

This paper presents a new method for reconstructing a 3D description of buildings from 2D architectural plans. The resulting description includes geometry and topology so that the whole environment consistency be preserved according to constraints such as closeness, orientability, and connectivity.

The main steps of our method concern: (i) a geometrical correction of source data, (ii) a 2D topology construction, (iii) a 3D extrusion system, (iv) floors
superimposing. We also propose semi-automatic tools for correcting 2D plans. The results obtained show that 2D and 3D processing require only a few tens of seconds.

The main advantage of our method concerns the use of topology for validating the building structure coherence and editing the model using modelling operations. Furthermore, the resulting structure provides various types of information necessary for visualization or lighting/thermal/low-frequency wave propagation simulations [11, 6, 16].

The next step of this work consists in automatically defining additional semantics (such as rooms or furniture for instance). Thus operations dedicated to volume types can be explored for simplifying the 3D models. We also aim at automatically detecting stairs and their characteristics. Moreover, additional operations have to be defined, for instance related to windows, roofs or superimposed floor with different shapes.

In the future, we wish to apply our system to urban scenes as well, containing furnished buildings, etc. This implies the processing of larger data with missing information. We aim at coupling our system with procedural reconstruction methods.

REFERENCES