



Consistency constraints and 3D building reconstruction

Sébastien Horna, Daniel Meneveaux, Guillaume Damiand, Yves Bertrand

► To cite this version:

Sébastien Horna, Daniel Meneveaux, Guillaume Damiand, Yves Bertrand. Consistency constraints and 3D building reconstruction. *Computer-Aided Design*, 2009, 41 (1), pp.13–27. 10.1016/j.cad.2008.11.006 . hal-00440844

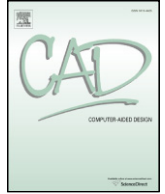
HAL Id: hal-00440844

<https://hal.science/hal-00440844v1>

Submitted on 12 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Consistency constraints and 3D building reconstruction

S. Horna^{a,*}, D. Meneveaux^a, G. Damiand^b, Y. Bertrand^a

^a Poitiers University, XLIM-SIC, UMR6172 CNRS, F-86962, France

^b Lyon 1 University, LIRIS, UMR5205 CNRS, F-69622, France

ARTICLE INFO

Article history:

Received 11 June 2008

Accepted 11 November 2008

Keywords:

3D modeling
Architecture
Indoor building
Reconstruction
Topological model
G-maps

ABSTRACT

Virtual architectural (indoor) scenes are often modeled in 3D for various types of simulation systems. For instance, some authors propose methods dedicated to lighting, heat transfer, acoustic or radio-wave propagation simulations. These methods rely in most cases on a volumetric representation of the environment, with adjacency and incidence relationships. Unfortunately, many buildings data are only given by 2D plans and the 3D needs varies from one application to another. To face these problems, we propose a formal representation of consistency constraints dedicated to building interiors and associated with a topological model. We show that such a representation can be used for: (i) reconstructing 3D models from 2D architectural plans (ii) detecting automatically geometrical, topological and semantical inconsistencies (iii) designing automatic and semi-automatic operations to correct and enrich a 2D plan. All our constraints are homogeneously defined in 2D and 3D, implemented with generalized maps and used in modeling operations. We explain how this model can be successfully used for lighting and radio-wave propagation simulations.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Before construction, buildings are often virtually represented in 3D so as to design a site and conduct studies related to lighting, heat transfer, radio-waves propagation, and so on. These environments are composed of a set of 3D geometric primitives with various types of additional information to expand their use: adjacency relationship between volumes, object types, or material properties used for building construction.

Many simulation programs cannot rely on commercial modeling software because their internal structure is not available. Consequently for each scene, the user has to manually pre-process the model in order to make it compatible with the required representation. This is why some integrated prototypes both build a 3D model and apply the simulation process [1]. However, the representation is not sufficiently generic for being used in different applications. To our knowledge, there is no common framework for both modeling indoor scenes and applying various types of simulations.

In this paper, our aim is to propose a representation of buildings homogeneous in 2D and 3D, taking geometry, topology and semantics consistency into account. The model should be generic enough to be used in as many applications and simulation systems as possible. To do this, we formally define a set of

constraints adapted to most common buildings. Generalized maps (G-maps [2]) are used as a topological basis and our definitions rely on their terminology. We also propose a semi-automatic reconstruction pipeline for generating 3D buildings from 2D architectural plans (see Fig. 1). Our main contributions are:

- a formal description of constraints providing a generic representation of geometry, topology and semantics of architectural indoor environments;
- the use of these constraints for automatically reconstructing 3D buildings from 2D vector plans, including topology and semantics;
- a set of semi-automatic operations used to detect and correct errors in the 2D plans before 3D construction.

The resulting scenes can be edited with a general modeler based on G-maps [3]. In addition, cell and portal structures often used in the context of large buildings [4–8] can be naturally produced from our structure. We have also produced files for radio-wave propagation simulations used in our laboratory [9,10] (Section 7).

This paper is organized as follows. Sections 2 and 3 present the existing methods in 3D architecture and describe G-maps. Section 4 presents our consistency constraints. Section 5 provides the 2D processing of an architectural vector plan. Section 6 explains how we detect and correct errors. Sections 7 and 8 present and discuss our results.

* Corresponding author. Tel.: +33 549496610; fax: +33 549706570.

E-mail address: horna@sic.sp2mi.univ-poitiers.fr (S. Horna).

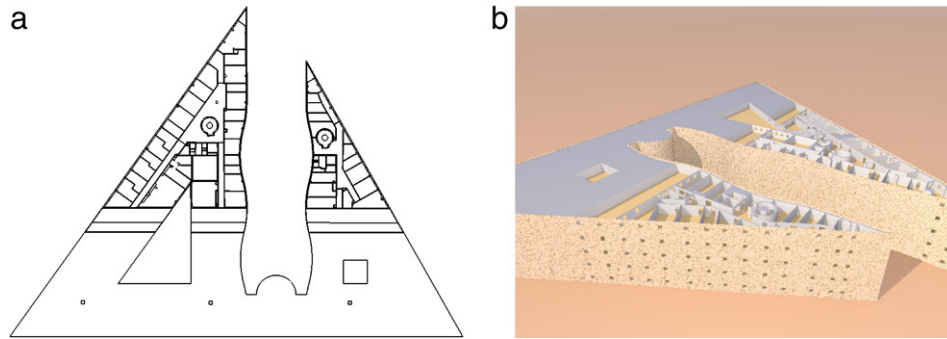


Fig. 1. (a) Building 2D plan; (b) 3D reconstruction with proper topology and semantics.

2. Related work

Topological models

We wish to represent large buildings, actually corresponding to 3D topological objects, made up of vertices, edges, faces and volumes which do not necessarily have regular shapes. Many topological models have been proposed in the literature for handling different classes of subdivisions (oriented surfaces, manifold, non-manifold, etc.) in any dimension. Examples of such structures are adjacency graphs [11], ordered models (as defined in [12]), 2D or 3D edge-based models [13–15] or higher-dimensional models [12,2].

Adjacency graphs (such as [11]) do not allow multi-incidence (see [2]). Moreover, the topological correctness of the model is difficult to express [16]. This is the reason why ordered models have been introduced. They are mainly defined with a single type of basic elements and links between these elements [12,2].

For the reasons explained above, we choose an ordered topological model. Complexity studies have shown that for 2D and 3D manifold (surface subdivisions or volume subdivisions), costs for representing an ordered model and an incidence graph are comparable [17]. This is even more true when taking geometry, photometry and other attributes into account.

Buildings are composed of volumes (floors, walls, rooms, etc.) sharing faces. Topologically speaking, it corresponds to 3D orientable manifolds. In [18], it has been shown that models defined to represent 3D manifolds are comparable either to 3D map (for orientable ones without boundaries) or to 3D Generalized maps (for orientable or not orientable ones, with or without boundaries). Generalized maps are defined homogeneously in any dimension, so that operations implemented in 2D can be naturally extended in 3D or in higher dimension.

Analysis of architectural bitmap images

Several methods have been proposed for reconstructing 3D buildings from existing images of 2D architectural plans (i.e. bitmap images) [19,20]. Bitmap images contain low-level geometrical and semantical information and before reconstruction, the first step is extracting richer information. However, the vectorization process usually employed to this task does not provide any topological information.

Even though the 3D reconstruction can be achieved with these methods, none of them does provide any topological information. However, these types of analyses are complementary to our work since much semantic information can be automatically deduced from images.

Google Earth [21], the French National Geographic Institute (IGN) [22,23] and other authors [24–26] also propose frameworks used to rebuild the geometry of real urban scenes based on various types of images (satellite, aerial, ground, etc.). Again, no topological information is produced with these approaches.

3D building analysis

Some authors have also proposed to extract topological information from a list of polygons, making it possible to reduce visibility calculations for lighting simulation and visualization. Airey et al. and Teller et al. [5,6] propose a binary space subdivision method (Binary Space Partitioning or BSP) while Meneveaux et al. [27] propose a method using rules for extracting rooms of buildings. These subdivision schemes produce a set of regions called *cells*, separated by openings (*portals*); the topological description corresponds to adjacency relations between 3D cells. However, this model does not provide any incidence/adjacency relations between lower-dimensional elements.

Building modelers

Fradin et al. [28] propose a building modeler for manually designing complex indoor scenes. The topological structure relies on G-maps, with hierarchy and multi-partitions, also adapted to various types of simulations [8,29]. The topological basis is also G-maps, but does not include any consistency constraints specifically related to architecture. Moreover, the operations defined in this modeler are designed so that the user has to construct the whole 3D building manually, without the benefit of using existing 2D data.

Several industrial software [30,31] propose to design 2D plans and/or 3D buildings. Many operations are provided such as: walls, stairs, or doors drawing; boolean operations; extrusion, revolutions and so on. However, even though those software allow us to construct 2D plan and 3D buildings simultaneously, they still do not provide any topological representation nor consistency control (superimposed walls, disjoint segments, and so on).

Many actors in architectural business need to share information for formalizing electronic documents. To this end, the IFC (Industry Foundation Classes) have been proposed recently. Since the first version appeared in 1997, more and more applications have employed this formalism. However, the associated file format essentially contains semantical information; it has not been defined for representing low-level topological relationships, which are necessary for many modeling operations such as those described in this paper.

Procedural modeling

Procedural systems which do not require any manual intervention have been used to produce urban scenes. Several authors propose applications for randomly constructing realistic (but not real) virtual cities and facades [32–35]. Unfortunately, these programs only produce cities and buildings geometry, without topology. In addition, reproducing a real environment is not possible due to the random processes applied.

Discussion

Many tools are dedicated to the (re)construction of 3D architectural environments. Some of them are dedicated to simulation, but

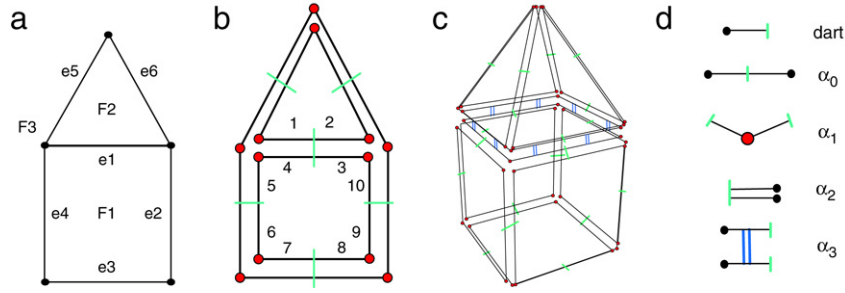


Fig. 2. G-Map representation: (a) a 2D object containing 3 faces, 6 edges and 5 vertices; (b) corresponding generalized map: the set of darts {1, 2, 3, 4} represents the edge e_1 , the set of darts {3, 4, 5, 6, 7, 8, 9, 10} represents the face F_1 ; (c) 3D object composed of 2 volumes represented with 3G-map, volumes are linked by α_3 ; (d) symbolic representation of involutions. In most cases, α_0 links are not represented for the sake of readability.

no generic model is currently provided for general purpose applications. Furthermore, without any topological structure nor consistency constraint, reconstructing a 3D building from 2D plans for producing topology requires many manual processing that cannot be done automatically. To our knowledge, the only application that provides topological information is the one proposed in [28]. However, no specific constraint is defined for the building structures so as to detect and correct errors in models.

3. Generalized maps

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

The rest of this section gives the main definitions and properties of n-G-maps. They can be found in [2,18] and are included here for the convenience of readers unfamiliar with G-maps.

Definition 1 (Generalized Map [2]). A generalized map in dimension $n \geq 0$ (or n-G-map) is an algebra $G = (D, \alpha_0, \dots, \alpha_n)$, where:

- D is a finite set of darts;
- $\alpha_0, \dots, \alpha_n$ are involutions;
- $\alpha_i \alpha_j$ is an involution $\forall i, j$ such that $0 \leq i < i + 2 \leq j \leq n$.

From a single type of basic elements (called *darts*) and involutions¹(called α) defined on these darts, generalized maps represent object cells and adjacency/incidence relationships. Each involution α_i , with $0 \leq i \leq n$ (n being the dimension of the considered space), represents the adjacency relationships between i -dimensional cells. α_0 represents link between two vertices, α_1 links two edges, α_2 links two faces and α_3 links two volumes (Fig. 2).

Two darts d and d' are i -sewed if $\alpha_i(d) = d'$ with $d \neq d'$, and d is i -free if $\alpha_i(d) = d$. An *orbit* is described by a dart and a set of involutions. It provides the set of all darts that can be reached by any composition of the given involutions: given dart d , the orbit $\langle f_1, \dots, f_k \rangle(d)$ corresponds to all the darts obtained starting from d , and using any composition of all the involutions f_1, \dots, f_k .

Definition 2 (i -cell). Let $d \in D$, $N = \{0, 1, \dots, n\}$ and let $i \in N$. The i -cell incident to d is the orbit $\langle \alpha_{N-[i]}(d) = \langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle(d)$.

An i -cell is a particular orbit composed with all involutions except α_i . Two cells are disjoint when their intersection is empty, i.e. when no dart is shared by the cells. In Fig. 3, i -cells which compose a 3G-map are represented with their corresponding orbit.

The *degree* of an i -cell C is the number of distinct $(i + 1)$ -cells incident to C (for instance, the vertex degree is the number of edges incident to it).

Definition 3 (Dangling Elements). $\forall i \in [1, 2]$ the i -cell C_i is a *dangling* cell iff the degree of C_i is 1 and if one of its incident $(i - 1)$ -cell C_{i-1} is also of degree 1.

In Fig. 4a, E_1 is a dangling edge since it is incident to a one-degree vertex and its degree is also equal to one; E_2 is not dangling, because even if its degree is one, it is not incident to a one-degree vertex (see Fig. 4b).

On the basis of G-maps representation, we have used a 3D topological modeler, comprising many operations such as sewing two cells along a face or more complex operations like sweeping or corefining [3]. We associate G-maps with linear embeddings because the reconstruction process starts with a set of segments corresponding to the 2D plan geometry.

4. Consistency constraints for 3D buildings

For each type of simulation using buildings, specific attributes are required by the program (adjacency relationship between volumes, object types, or materials properties used for building construction). However, the general building structure remains the same in most buildings, so that generic rules can be defined for various applications. Such a model should:

- correspond to a structured representation;
- ensure consistency of topology, geometry and semantics;
- reduce as manual processing of buildings as possible when 2D data already exists.

G-maps alone are not sufficient for representing an architectural model with semantics, topology and geometry such as neighbourhood relationships or volume thickness. This is why we propose additional consistency constraints, that can be used for various applications: reconstruction from 2D plans, error detection and correction, model modifications, and so on.

Our set of constraints is dedicated to building models. Each constraint is defined as a geometrical, topological and/or semantical property. In this section, we firstly focus on the space subdivision representation of buildings; secondly, we express constraints in terms of elements thickness; thirdly, we enrich the model with semantical rules.

4.1. Space subdivision

We need to represent objects of the real world, corresponding to disjoint volumes: rooms are defined as volumes, as well as walls, doors, windows. This is why we use a 3D space subdivision. The model must be oriented and composed of closed volumes, faces and edges. Furthermore, elements cannot be superimposed. This representation corresponds to a closed and oriented 3D *partition*.

¹ An involution f is a one-to-one mapping such that: $f^2 = Id$.

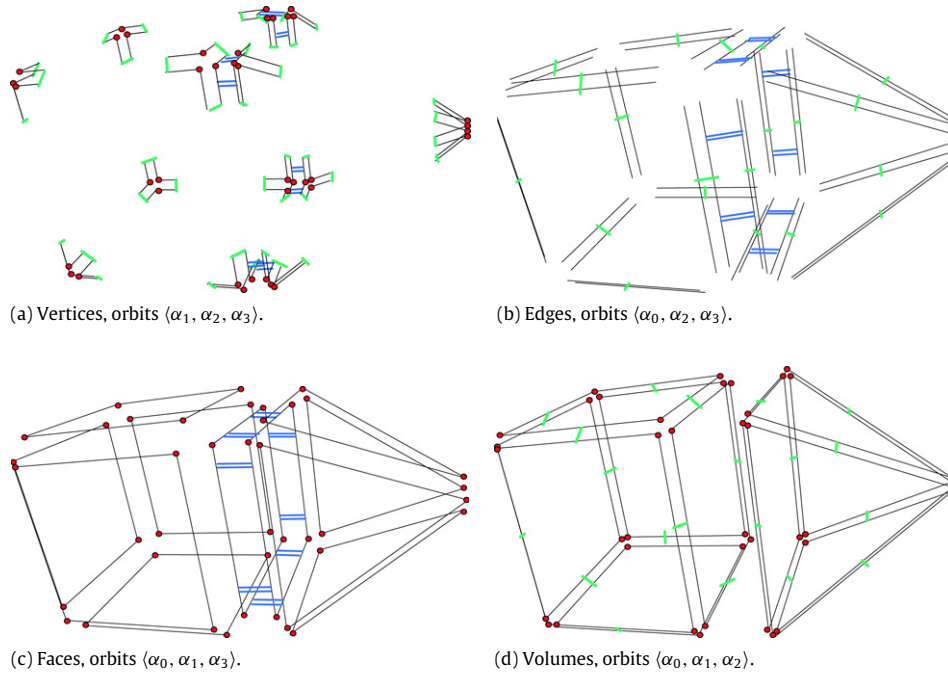


Fig. 3. Each cell can be obtained from a dart and an orbit: $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$ for vertices, $\langle \alpha_0, \alpha_1, \alpha_3 \rangle$ for edges, $\langle \alpha_0, \alpha_1, \alpha_3 \rangle$ for faces and $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$ for volumes.

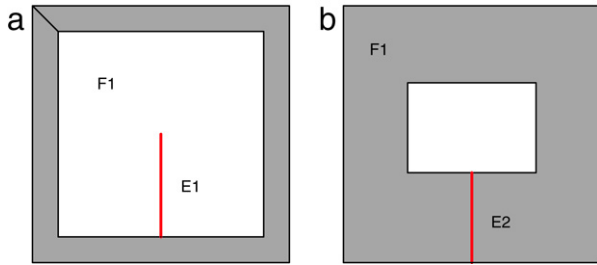


Fig. 4. (a) Dangling edge: E_1 is inside face F_1 . Its degree is 1 and it is incident to a vertex of degree 1; (b) E_2 is a degree 1 edge which is not dangling.

Property 1 (Space Partition of \mathbb{R}^3). Let C_i^1 and C_i^2 be two i -cells of a partition P with $i \in \{0, 1, 2, 3\}$; $C_i^1 \cap C_i^2 = \emptyset$ or $C_i^1 \cap C_i^2 = \{C_j | j \in \{0, 1, 2\} \text{ and } j < i; C_j \in P; C_j \text{ be incident to } C_i^1 \text{ and to } C_i^2\}$.

Fig. 5 gives example of models that do not comply with this property. It shows two distinct faces with various embeddings.

Fig. 6 shows configurations with several distinct faces in \mathbb{R}^2 . In Fig. 6a, F_1 and F_2 are two open faces that do not cover \mathbb{R}^2 thus not corresponding to a space partition. In Fig. 6b, F_1' and

F_2' are the complement of F_1 and F_2 and their intersection is not empty (it is a face): this configuration also does not correspond to a space partition. Fig. 6c illustrates three closed faces where F_3 is the complement of F_1 and F_2 . The union of F_1 , F_2 and F_3 covers \mathbb{R}^2 and the intersections are composed of edges and vertices included in the partition. This configuration is thus a space partition, complying with Property 1.

Note that with Property 1, a model composed of several connected components is not a space partition (see Fig. 6a and b). With 3G-maps, a model is composed of a single connected component iff: $\forall d \in G, d \in \langle \alpha_0, \alpha_1, \alpha_2, \alpha_3 \rangle(d')$ until $d' \in G$.

Property 2 (Closeness). The model is closed in all dimensions.

For a G-map, this property is equivalent to: $\forall d \in G$ and $\forall i, 0 \leq i \leq 3, \alpha_i(d) \neq d$ (a closed G-map has no free dart).

Property 3 (Orientation). The model is oriented.

G-maps can be used to represent objects orientable or not. For buildings, we specify that the model is oriented. When a 3D connected oriented model is represented by a G-map, the orbit $\langle \alpha_0 \alpha_1, \alpha_0 \alpha_2, \alpha_0 \alpha_3 \rangle(d)$ contains one out of two darts of the G-map.

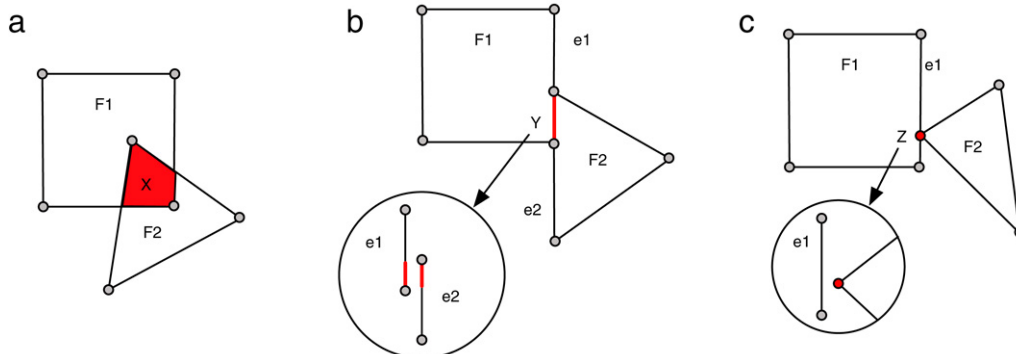


Fig. 5. Series of configurations with distinct faces F_1 and F_2 composed of the set of cells $S = \{f_1, f_2, e_1, \dots, e_7, v_1, \dots, v_7\}$: (a) the geometrical intersection of F_1 and F_2 is a face X that does not belong to S ; (b) identically, $F_1 \cap F_2$ is an edge Y that does not belong to S ; (c) $F_1 \cap F_2$ is a vertex Z that is not incident to F_1 .

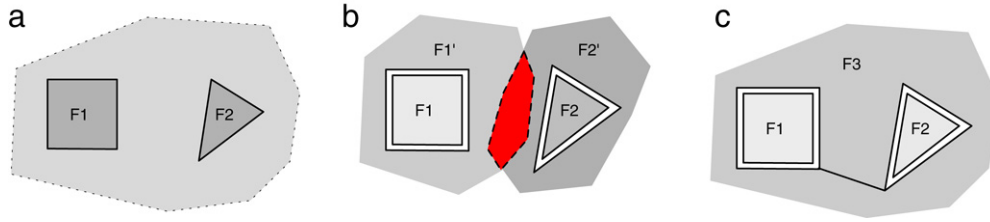


Fig. 6. Space partition property: (a) F1 and F2 do not cover \mathbb{R}^2 ; (b) the intersection of F1' and F2' is not empty (it is a face); (c) a model made up of a single connected component which complies with [Property 1](#).

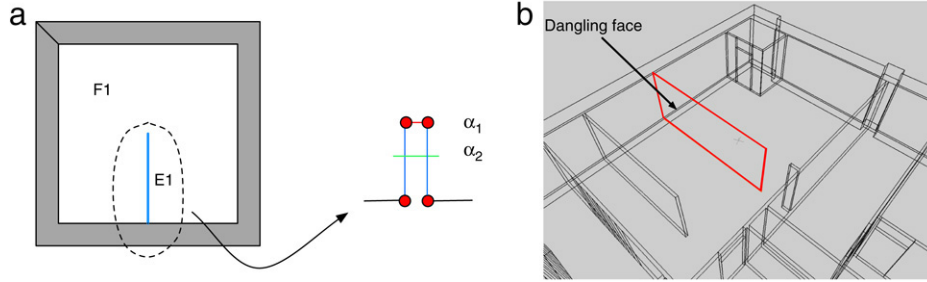


Fig. 7. Dangling cells: (a) in 2D the dangling edge E_1 does not represent a thick object; (b) 3D example with a wall having no thickness, represented by a dangling face.

Then, this orbit cardinal is equal to half the number of darts in the G-map.

Let $d \in D$, $|\langle \alpha_0 \alpha_1, \alpha_0 \alpha_2, \alpha_0 \alpha_3 \rangle(d)| = |D|/2$.

With a G-map, this property can be ensured using a boolean mark for each dart indicating the orientation and verifying: $\forall d \in D, \forall i, 0 \leq i \leq 3, \text{orientation}(d) = 1 \Leftrightarrow \text{orientation}(\alpha_i(d)) = 0$.

4.2. Volumes and thickness

We are mainly interested in a structured representation of the main building work in 3D: doors, windows, walls, stairs, rooms, and so on. In practice, each element must be defined with a significant thickness: flat volumes/edges or dangling faces/edges (see [Fig. 7](#)) cannot appear in a real world model.

Property 4 (Dangling Elements). A partition of \mathbb{R}^3 representing a building does not contain any dangling edge or face.

With G-maps, the degree of dangling faces and edges is 1. Furthermore, a dangling edge contains at least one dart d such that $\alpha_1(d) = \alpha_2(d)$, and a dangling face contains at least one dart d such that $\alpha_2(d) = \alpha_3(d)$.

4.3. Semantic constraints

In applications concerning building simulations or modeling, the type of each volume should be precisely defined. This is why we have identified several semantic key markers matching objects common to most buildings (walls, rooms, stairs, etc.). Their number can easily be increased depending on application needs. Our basis set of marks contains the following elements:

| | | | |
|-------|---------|----------|---------|
| -WALL | -STAIR | -ROOM | -FACADE |
| -DOOR | -WINDOW | -CEILING | -GROUND |

Property 5 (Semantical Markers). Each volume should be associated with one (and only one) basis key marker.

Note that the FACADE describes the infinite volume outside the building, and each of its faces is incident to the building WALL, WINDOW, DOOR, CEILING or GROUND volumes.

With G-maps, this property implies that all the darts belonging to the same 3-cell have the same semantic:

$\forall d_1 \in D, \forall d_2 \in \langle \alpha_0, \alpha_1, \alpha_2 \rangle(d_1), \text{semantic}(d_1) = \text{semantic}(d_2)$.

Property 6 (Neighbourhood Constraints). Volume neighbourhood must comply with the constraints given in [Table 1](#).

Using G-maps allows us to detect semantical inconsistencies. Each face is incident to 2 volumes and volumes are connected by an α_3 involution. Then, with G-maps, semantic consistency is controlled with the following property: $\forall d \in G, \text{Table 1}[\text{semantic}(d), \text{semantic}(\alpha_3(d))] \neq \text{error}$.

Two neighbour volumes can have the same semantic markers in several cases: architects sometimes separate walls or ground volumes deliberately. For instance, when two walls are not built with the same material, a face must exist between the two corresponding volumes.

Property 7 (Building Facade). The outside volume is unique, infinite and complementary to the set of building volumes; the building facade corresponds to the (interface) faces between outer wall volumes and the outside volume.

With G-maps, this property implies that all darts with semantic facade are contained in the same volume:

$\forall d_1, d_2 \in D, \text{with } \text{semantic}(d_1) = \text{FACADE} \text{ and } \text{semantic}(d_2) = \text{FACADE}, d_1 \in \langle \alpha_0, \alpha_1, \alpha_2 \rangle(d_2)$.

Note that when two houses are adjacent, there is only one shared wall. The facade is defined all around the whole building (corresponding to the two houses). If each house is modeled independently and linked later, the common walls and facades should be merged to compose a single volume. In addition, for several floors the corresponding facade faces are linked, producing one single facade volume.

5. Building reconstruction and constraints compliance

This section describes our 3D reconstruction system, with the assumption that 2D digital plans do not contain any error: no superimposed elements, no dangling edge, no semantic inconsistency. Without any error, the process is automatic. In

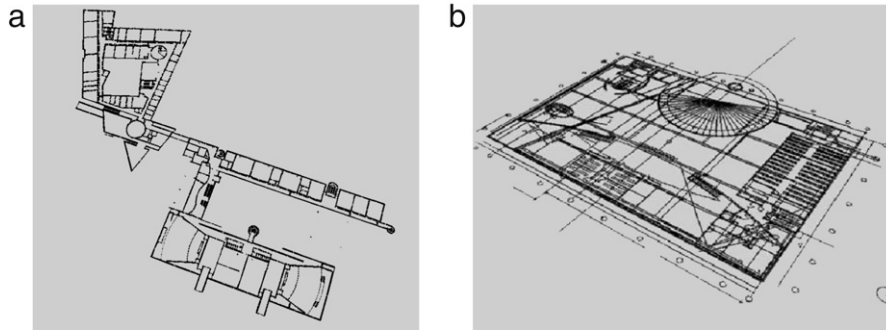


Fig. 8. 2D geometry.

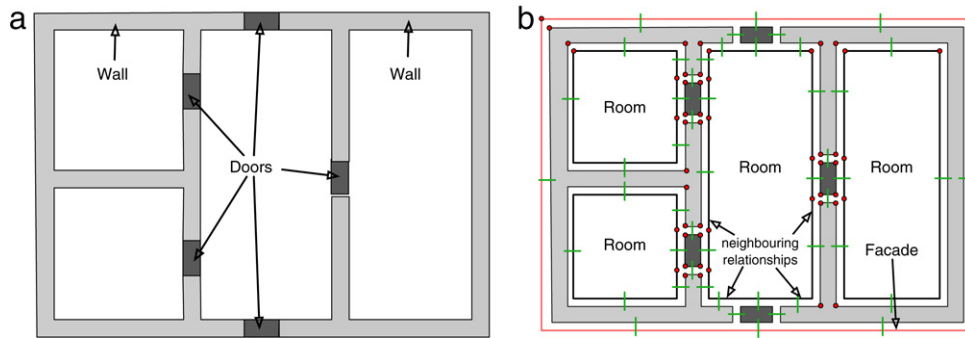


Fig. 9. (a) 2D architectural plan representation (b) 2D topological and semantic information. Each face corresponds to one element (room, wall, door, window, etc.) while topology represents neighbouring relationships, and semantics specify types.

Table 1

Neighbourhood constraints; cells containing 'error' indicate that the corresponding objects cannot be neighbours. Objects WALL, GROUND and CEILING can be neighbour to all other objects, their corresponding columns are not represented.

| | FACADE | STAIR | ROOM | DOOR | WINDOW |
|---------|--------|-------|-------|-------|--------|
| WALL | – | – | – | – | – |
| FACADE | error | error | error | – | – |
| STAIR | error | – | – | – | – |
| ROOM | error | – | – | – | – |
| GROUND | – | – | – | – | – |
| CEILING | – | – | – | – | – |
| DOOR | – | error | – | error | error |
| WINDOW | – | error | – | error | error |

addition, our method automatically detects the errors, using our model properties. Section 6 details error detection and operations provided to the user for correcting plans.

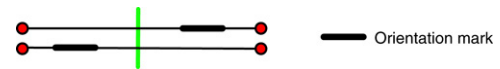
The reconstruction process starts from the geometry of an extracted file produced with a traditional CAD software (see Fig. 8). It is composed of five steps [36]:

- 2D partition construction;
- semantic instantiation for all 2D elements;
- 3D extrusion taking semantic information into account;
- 3D floor, ceiling, stair extrusion;
- floors superimposition.

We show how each step produces a model compliant with the properties defined in the previous section, firstly during the 2D processing from the CAD plan, and secondly with our 3D extrusion system.

5.1. 2D Reconstruction

Based on G-maps, our model is homogeneous in 2D and 3D: 3D properties can be transposed directly in 2D since each face corresponds to one 3D volume. This is why our first goal is to

Fig. 10. Topological oriented edge, links α_1 are free (not connected) and an orientation mark is defined.

build a 2D oriented and closed partition, composed of a single connected component from a set of geometrical 2D segments without topological nor semantical information.

2D digital plans are made up of various types of geometric elements such as polygons, circles, ellipses, and so on. In order to deal with homogeneous basic elements, we have chosen to convert the initial set of objects into a set of line segments. From this set of independent segments, topology and semantics are deduced as explained in the following (result example shown in Fig. 9).

The first step of our process consists in associating each segment with a topological edge composed of 4 darts (with G-maps), with α_0 and α_2 defined (Fig. 10). This step already produces 0-closed and 2-closed topological edges; an orientation mark is fixed on the edge darts for satisfying the orientation property ($\forall d$ and $\forall i$, if d is marked, $\alpha_i(d)$ is not marked).

In the second step, α_1 is set for all darts, producing the plan faces and the 2D space partition. Therefore, an angle arrangement is performed around all the model vertices: incident edges are stored and sorted according to their angle with an initial edge. Then, darts are successively 1-sewed around the vertex, according to this angle and orientation marks (Fig. 11).

This construction process complies with the defined consistency constraints, in terms of space partition, closeness, orientability, as explained in the next paragraphs: in 2D, the reconstruction process automatically produces a closed and oriented space partition composed of faces corresponding to walls, doors, windows, rooms and so on.

Space partition property

According to the 2D reconstruction process, each face is delimited by a set of edges. We consider in this section that plans do not

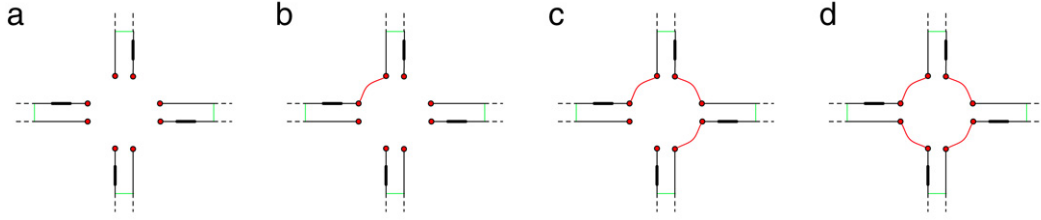


Fig. 11. Angle arrangement around an m -degree vertex: (a) all darts are 1-free, edges are stored according to their angles around vertices; (b) according to their angles, an α_1 link is defined with proper orientation; (c) $m - 1$ links are sewed around the vertex; (d) after the last sewing, orbit (α_1, α_2) around the vertex is closed and oriented.

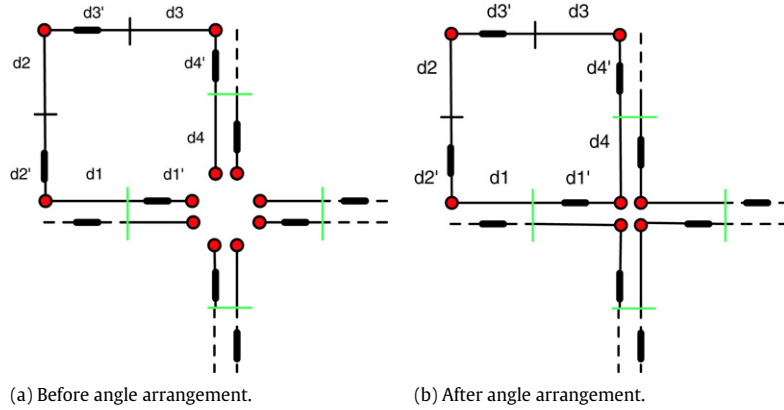


Fig. 12. Closure of a 4 edges face: $d4 = \alpha_0(\alpha_1\alpha_0)^3(d1')$ is unmarked while $d1'$ is marked.

contain any superimposed elements. Thus, faces do not overlap, as well as edges. Moreover, edges intersection always correspond to one extremity of each edge. Consequently, given 2 faces C_1 and C_2 , $C_1^1 \cap C_2^2 = \emptyset$ or $C_1^1 \cap C_2^2 = \{C_j\}$, with C_j an edge or a vertex already existing in the plan, complying with [Property 1](#). \square

Closeness property

During edges creation, α_0 and α_2 links are immediately defined, so that the model is directly 0-closed and 2-closed. In addition, each vertex is incident to at least 2 darts (see [Fig. 10](#)). Our angle arrangement algorithm is then used to sew all α_1 links locally for each vertex: all the darts incident to one vertex are 1-closed and all vertices are processed (see [Fig. 11](#)). Thus all the darts are 1-sewed and the model is 1-closed by construction, and the 2-G-map is closed in all dimensions, complying with [Property 2](#). \square

Orientation property

A 2G-map is oriented if $\forall d \in D, \forall i, 0 \leq i \leq 2, \text{mark}(d) = \text{true} \Leftrightarrow \text{mark}(\alpha_i(d)) = \text{false}$.

During edge creation, 4 darts are created and linked with α_0 and α_2 as described in [Fig. 10](#). Thus, α_0 and α_2 orientation is correctly set for all the darts in the plan. In addition, if m edges are incident to a given vertex v , $2m$ darts are incident to v : m darts marked and m darts unmarked. Angle arrangement sorts edges around the vertex and each marked dart is linked by α_1 with an unmarked dart (see [Fig. 11](#)). Consequently around each vertex, the orientation property is correctly defined. \square

Note: this angle arrangement is applied to every vertex of the plan, thus constructing faces. A face incident to k edges is composed of $2k$ darts such that $(\alpha_0\alpha_1)^k(d) = d$. Linking darts around vertices corresponds to closing faces. Consequently, when $k - 1$ vertices are linked for a face, the last one links d with $\alpha_0(\alpha_1\alpha_0)^{k-1}(d)$, i.e. a marked dart with an unmarked one (see [Fig. 12](#)).

Dangling elements property

Since we assume that the initial plan is correctly defined, it does not contain any inconsistency, and each vertex is incident to at least two edges. The angle arrangement algorithm rebuilds topology around all model vertices without adding any 1-degree element.

Dangling elements detection and correction are explained in next sections.

Semantic properties

Semantics can sometimes be extracted from the plan layers. Many plans are defined with one layer corresponding to walls, another layer defining doors and so on. Rooms correspond to undefined faces in this context. When this information is not complete, our user interface provides tools for providing semantic associations.

5.2. 3D Reconstruction

The 3D reconstruction process ([Fig. 13](#)) corresponds to 4 main steps:

- (1) extrusion of walls and openings (doors and windows);
- (2) construction of grounds and ceilings;
- (3) floors superimposing;
- (4) stairs construction.

Extrusion process

With G-maps, the usual extrusion operation along a path is defined locally for each dart. Given a dart d_0 on the original plan, for each component, six darts are added and sewed along the extrusion path ([Fig. 14a](#)). The darts corresponding to the path are linked to the remaining of the plan according to α_i links of d_0 (see [Fig. 14c](#)). Thus, no dangling element is produced and all the G-maps properties are maintained.

This local process automatically produces for each i -cell of the 2D plan an $(i + 1)$ -cell connected to the rest of the building. For instance, two faces F_1 and F_2 connected by α_2 in the plan are extruded as 2 volumes V_1 and V_2 connected by α_3 ([Fig. 14c](#)).

According to each face semantic, various extrusion paths are used, with specific parameters. For instance, wall and room volumes are defined with a height of 2.5 m ([Fig. 14](#)); doors are built with 2 volumes: opening and upper wall part ([Fig. 15a](#)); windows are defined by 3 volumes ([Fig. 15b](#)) with a 1 m opening placed at the middle of the wall. Each parameter can be fixed by the user.

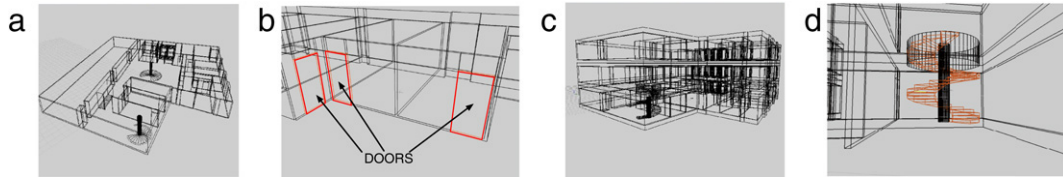


Fig. 13. 3D reconstruction: (a) sample floor with walls extrusion; (b) result of doors extrusion; (c) building made up of two superimposed floors; (d) result of a spiral staircase in a building with ceiling opening.

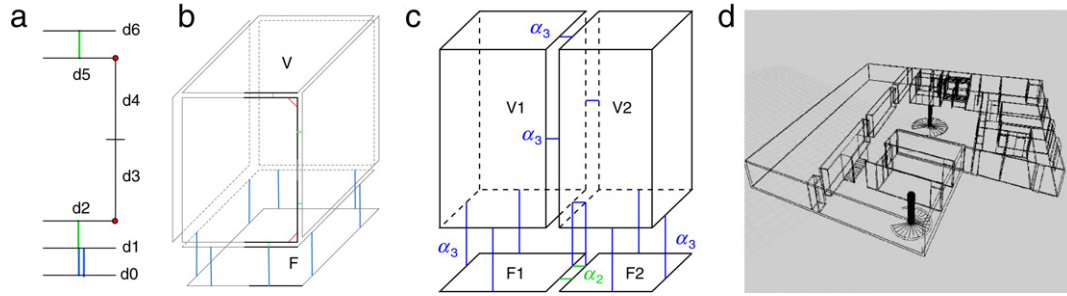


Fig. 14. Extrusion process: (a) extrusion path; (b) face extrusion along the path defined in (a); (c) if d_0 is linked by α_2 , α_3 links are propagated; (d) wall extrusion result.

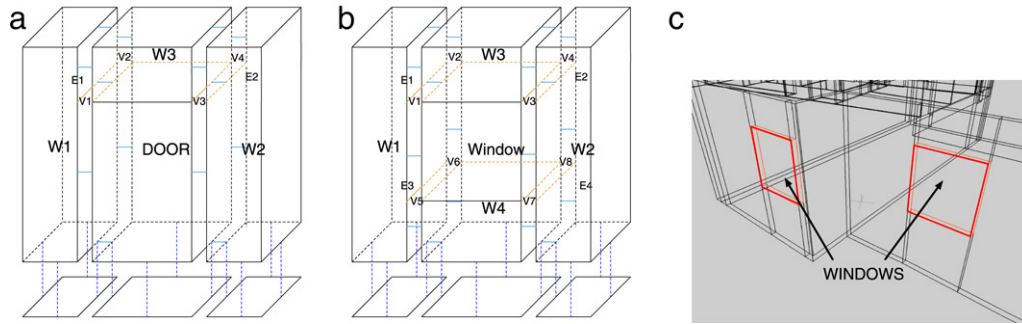


Fig. 15. Doors and windows extrusion: (a) extrusion and connexion to the walls W_1 and W_2 ; the section of wall named W_3 is built above the door; four vertices (V_1, V_2, V_3, V_4) and two edges, E_1 and E_2 , are inserted on W_1 and W_2 ; (b) extrusion and connexion to the walls W_1 and W_2 , the section of wall named W_3 is built above the window and W_4 is built under the window. Eight vertices (V_1, \dots, V_8) and four edges, E_1, E_2, E_3 and E_4 , are inserted on W_1 and W_2 ; (c) result of windows extrusion.

For doors, the 2D polygon is extruded with the same operation, according to a vertical path of two components (corresponding to the opening and the portion of wall above the door). The two resulting volumes have to be topologically connected and linked to walls and rooms. Therefore, two vertices and one edge are inserted on each adjacent volume (using topological operations) for complying with the topological constraints (Fig. 15a, edges E_1 and E_2). Windows are generated and connected to the walls using the same method (Fig. 15b).

Building construction

Several types of objects such as grounds, ceilings or stairs do not rely on extrusion. This is why we have developed specific construction operations complying (by construction) with our consistency constraints.

The 2D plan is used to create the ground volume (flagstone). The 2D plan faces correspond to the upper ground faces (Fig. 16b). The infinite outside face (facade) is modified so as to create the vertical ground volume properly linked to the remaining of the plan (Fig. 16b and c). Similarly, a ceiling is constructed at the top of each floor, so that the outside infinite volume be delimited by the outside building face. For superimposing two floors with same outer 2D shape, the ground of upper floor is connected to the lower floor ceiling. The face shared by these volumes can be used to separate the two objects.

When the shape of floors is different, a boolean operation has to be performed so as to connect the ceiling volume to the upper

ground volume. Split levels corresponds to floors with several facades and thus require fictive edges, which are correctly handled in our application.

We have defined a *generic* stair topological structure adapted to stair shapes (straight, spiral, elliptic, etc.): a topological volume composed of 6 faces describes each step while a second volume composed of 5 faces is used to link two steps (Fig. 17a). The geometry of each step is defined according to the data recovered from the plan (length, width, or diameter in the case of spiral staircases).

Steps are built between the ground level and the superior floor ground level. A hole is added in the upper ceiling and floor volumes so as to represent the opening (Fig. 17b). The hole is a volume linked by a fictive edge (see Section 6). The lower face of stairs volume is also connected to the floor upper face also using a fictive edge (Fig. 17c). Note that this defines a 3D hole in the corresponding room.

Space partition property

From the 2D space partition plan, extrusions produce a set of volumes, faces, edges and vertices. With this operation, two faces on the original plan sharing an edge are extruded into two volumes sharing a face, systematically (due to local processing). In addition, a set of k edges incident to one vertex produces k faces incident to one edge. In terms of G -maps, each α_i involution in the 2D plan generates an α_{i+1} involution in the 3D building. Moreover the remaining faces (ground and ceiling) are completed by the

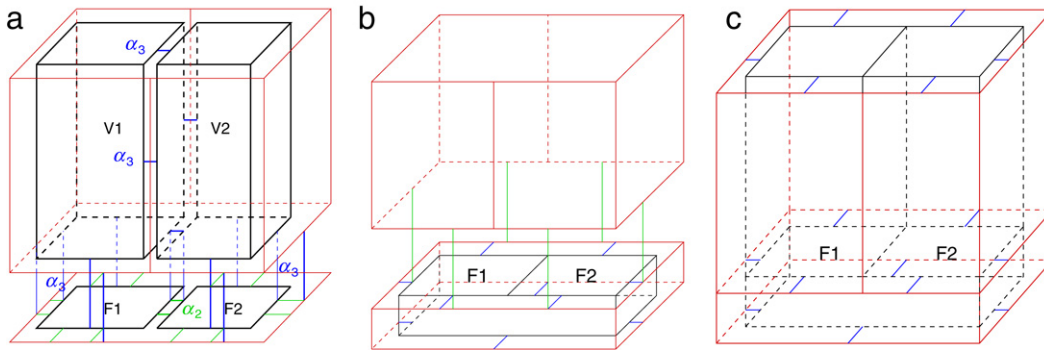


Fig. 16. Ground construction: (a) plan faces correspond to the upper ground faces; (b) ground volume construction; (c) resulting ground volume.

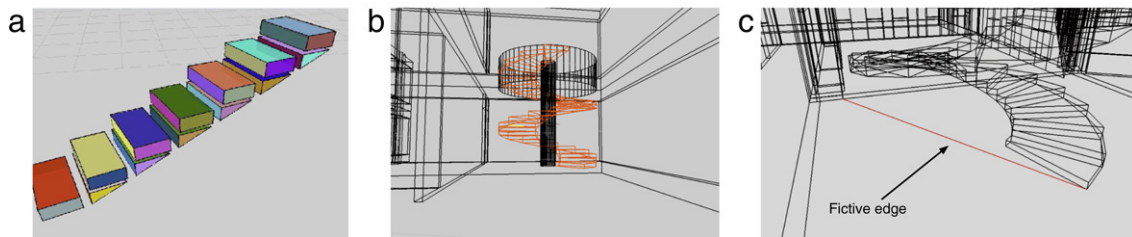


Fig. 17. Stairs construction: (a) each step of stairs is composed of 2 volumes; (b) the resulting of a spiral staircase extrusion (c) fictive edge between stairs and ground.

construction operations for producing the volumes corresponding to the building 3D partition.

Closeness property

In terms of G -maps, closeness is ensured when no dart is i -free. During the extrusion operation, for each original dart, a set of darts is built along the extrusion path (Fig. 18a). Along this path, d_1, d_2, d_5 et d_6 are 0-free, d_1 and d_6 are 1-free, d_3 and d_4 are 2-free and d_2, d_3, d_4 and d_6 are 3-free. Since d_0 is linked with $\alpha_0, \alpha_1, \alpha_2$ and α_3 , the extrusion progressively links all the new darts with all the involutions.

- According to α_0 :
 - . darts d_3 and d_4 are linked by construction along the extrusion path;
 - . dart d_0 is linked by α_0 , because 2D model is closed, by extrusion operation definition d_1, d_2, d_5 and d_6 are connected by α_0 (Fig. 18b).
- According to α_1 :
 - . darts d_2, d_3, d_4 and d_5 are linked by construction along the extrusion path;
 - . dart d_0 is linked by α_1 , thus, by construction d_1 and d_6 are connected by α_1 (Fig. 18c).
- According to α_2 :
 - . darts d_1, d_2, d_5 and d_6 are linked by construction along the extrusion path;
 - . dart d_0 is linked by α_1 , thus, by construction d_3 and d_4 are linked by α_2 (Fig. 18c).
- According to α_3 :
 - . darts d_0 and d_1 are linked by construction along the extrusion path;
 - . dart d_0 is linked by α_2 , thus, by construction d_2, d_3, d_4 and d_5 are linked by α_3 (Fig. 18d);
 - . ceiling construction allows us to create the upper floor volume, which links α_3 dart d_6 (Fig. 18e).

Orientation property

Starting with a 2D oriented plan, the extrusion operation automatically produces a 3D oriented model: for each extruded dart, the orientation mark is propagated along the extrusion path

as shown in Fig. 19a and b. The orientation is already correct after sewing (Fig. 19c and d). Therefore, for each oriented face of the 2D plan, the corresponding volume is correctly oriented (Fig. 19e) and the resulting 3D faces are automatically oriented (Fig. 19f). In addition, each 3D object built by our system is oriented according to the 3D building orientation configurations (ground, ceiling, stairs, etc.).

Dangling element property

The extrusion of an i -cell of degree k produces an $(i + 1)$ -cell of degree k . Since there is no dangling cell in the 2D plan, no dangling cell can be produced in the 3D building. Moreover, our construction operations all produce volumes delimited by faces, without any dangling edges nor faces.

For doors and windows, the same type of process is applied: from each 2D edge, a set of 3D faces is produced (thus without any dangling element); each face is α_2 -linked with its neighbour face coherently. When windows or doors are linked to walls, the new created edges are also controlled and thus cannot be dangling. In these conditions, the overall system can thus not produce any dangling element.

Semantic properties

The semantic properties defined for faces in the 2D plan are propagated to the corresponding building volumes (Fig. 20): walls, rooms, doors, windows, and so on. The objects that do not appear on the plan such as grounds, ceilings and stairs are created together with their semantic properties. Note that the original plan faces are finally associated with the *ground* semantic.

This reconstruction system of a 3D building from a 2D architectural plan is automatic when the 2D original set of linear segments complies with the following constraints: no superimposed element, no dangling edge, known semantic.

Unfortunately, during the 2D plan conception, the designer does not control geometry approximations. The set of segments may thus include many errors (disconnected or overlapping edges, etc.). This is why pre-processing is required for controlling 2D data and correcting errors.

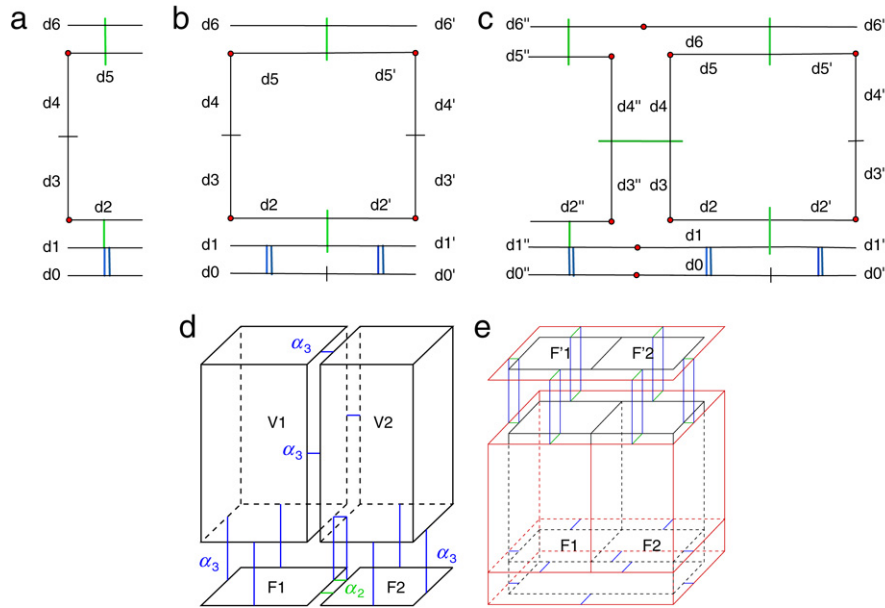


Fig. 18. Extrusion process: (a) extrusion of dart d_0 along a given path; (b) if d_0 is linked by α_0 , sewing is propagated; (c) if d_0 is linked by α_1 the 2 resulting faces are linked by α_2 ; (d) if two faces F_1 and F_2 are sewed by α_2 , resulting volumes V_1 and V_2 are linked by α_3 ; (e) during ceiling creation, top faces are linked by α_3 to the ceiling faces.

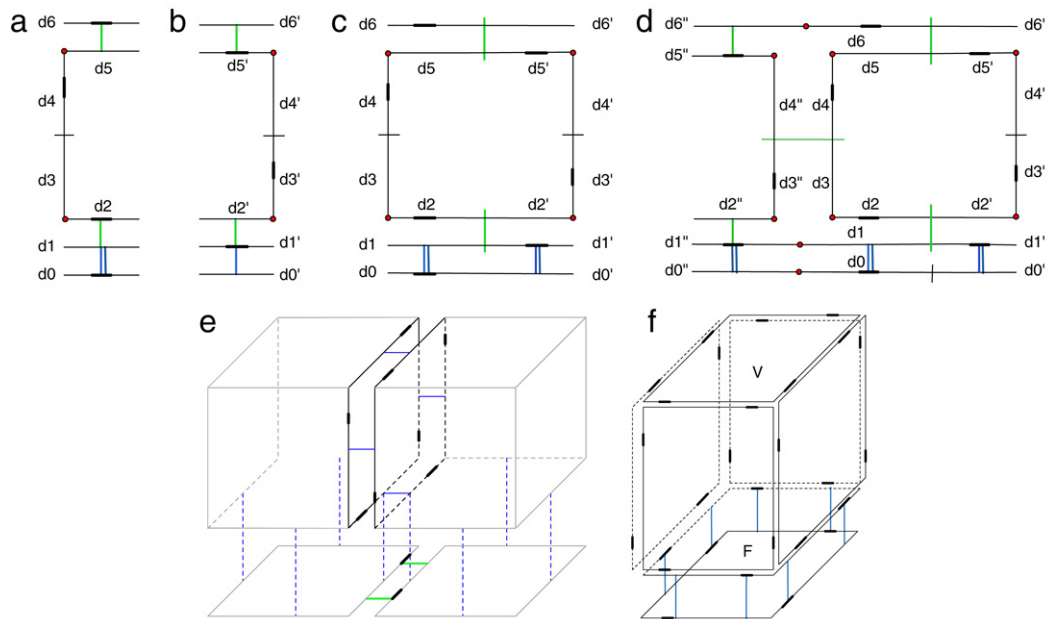


Fig. 19. Orientation properties of extrusion operation: (a) extrusion when d_0 is marked; (b) extrusion when d_0 is not marked; (c) α_0 links between 2 components; (d) links between two components by α_1 and α_2 ; (e) the extrusion of an oriented edge produces an oriented face; (f) the extrusion of an oriented face produces an oriented volume.

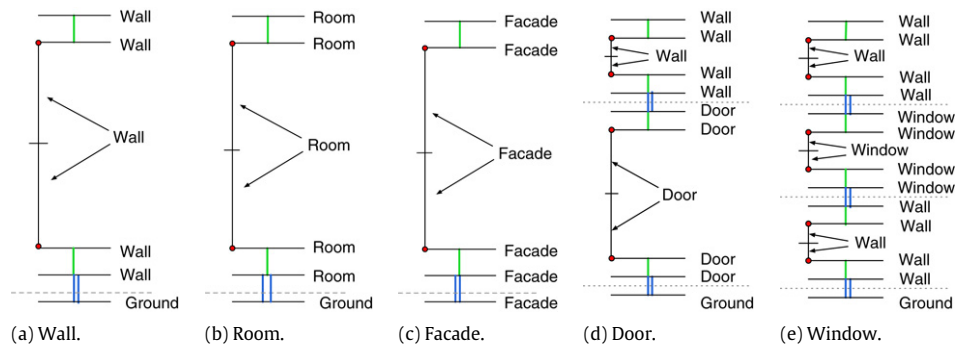


Fig. 20. Specific semantic definition along the extrusion path.

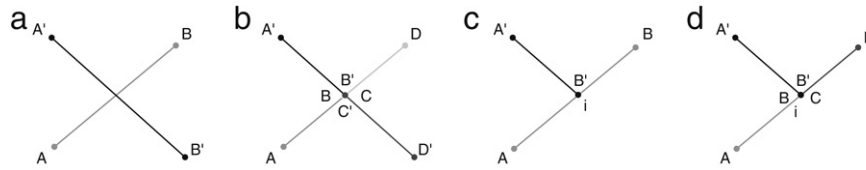


Fig. 21. Edges are processed two by two. If an intersection is found, a vertex is added at the intersection point on the concerned edges. Various cases have to be considered depending on the position of the intersection points (a) and (c) show the possible cases and (b) and (d) illustrate the result.

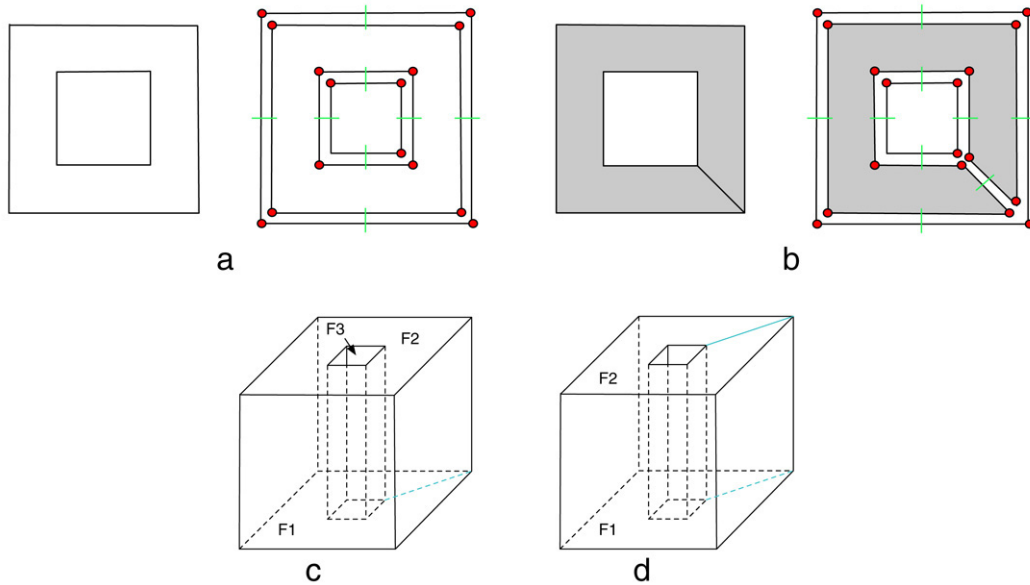


Fig. 22. Representation of faces with holes: (a) representation of two closed 2-cells superimposed without information; it does not correspond to a partition; (b) representation of a closed ring with a fictive edge; (c) Faces F_2 and F_3 do not represent a partition; (d) F_1 and F_2 correspond to the upper and lower faces of the volume boundary.

6. Inconsistency detection and correction

This section presents a set of methods for detecting and correcting geometrical, topological and semantical inconsistencies. Some of them are automatic (according to thresholds given by the user), while others are semi-automatic. With our application, given a valid 2D plan, our 3D reconstruction method produces a valid 3D building model. Consequently, corrections only need to be performed in 2D.

Given a 2D digital plan composed of line segments, we first aim at recovering a correct topology and geometry. Unfortunately, the original plan is (in most cases) not adapted to this type of reconstruction: many segments overlap, do not join or intersect. Moreover, walls are sometimes defined without thickness, some doors are missing, and so on.

In [36], the reconstruction method is detailed for correcting geometry and topology. This section illustrates how our system complies with the consistency properties proposed in this paper.

Space partition property

Since the 2D reconstruction process depends only on a list of line segments, the only inconsistencies come from:

- segments overlapping;
- segments intersection;
- several connected components.

We propose a pre-processing step that:

- merges overlapping segments;
- splits segments according to intersection points (see Fig. 21).

Once the set of segment is corrected, the processing described in Section 5 is applied to construct topological vertices, edges and faces. However, if the model is not composed of a single connected component, it does not correspond to a partition. To solve this problem, we have chosen to add fictive edges. This method is easy to manage in our application framework because no specific data structure is required (contrary to inclusion trees for instance). A face with holes is thus represented by only one face with one or several fictive edges (Fig. 22b). For volumes containing faces with holes, each holed face is processed in his 2D plan and a fictive edge is added if necessary (Fig. 22d).

Each 2D connected component (CC), is linked to another CC using a *fictive edge* (Fig. 23). Let us consider a CC c_1 ; the fictive edge is constructed using a half-line starting from a vertex v_1 of c_1 for finding a vertex v_2 on the closest CC c_2 (distance between v_1 and v_2 is minimized). For instance, Fig. 23 shows 3 connected components. In Fig. 23b, faces A and B are infinite and included in C, thus not correspond to a partition. In Fig. 23c a *fictive edge* links A and B to C. The 2d plan is then composed of only one CC and complies with the partition property.

During the extrusion process, fictive edges are not extruded. They are kept at the ground level and reproduced on the upper face (ceiling) of the model. Thus, the upper and lower volume faces have one boundary and comply with the partition property (Fig. 22d).

Dangling elements property

During the 2D reconstruction process, depending on the plan precision, some dangling edges can be detected. These edges correspond in most cases to architectural plan mistakes such as walls without thickness for instance. Dangling edges detection (and/or selection) can be processed automatically using the G-map

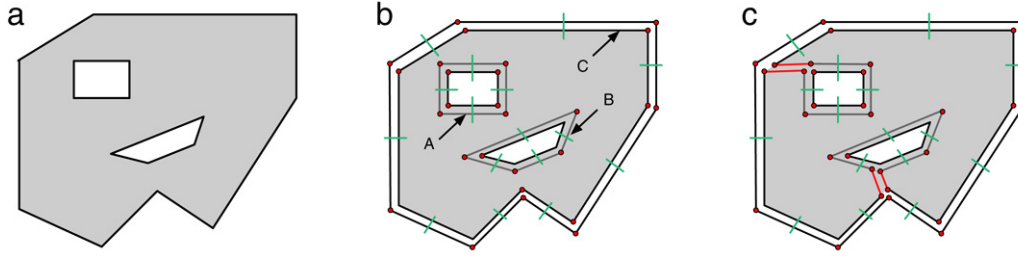


Fig. 23. 2D fictive edge management: (a) scheme composed of 3 components; (b) faces A, B and C define the same space; (c) 2 fictive edges are created in the model, which is connected.

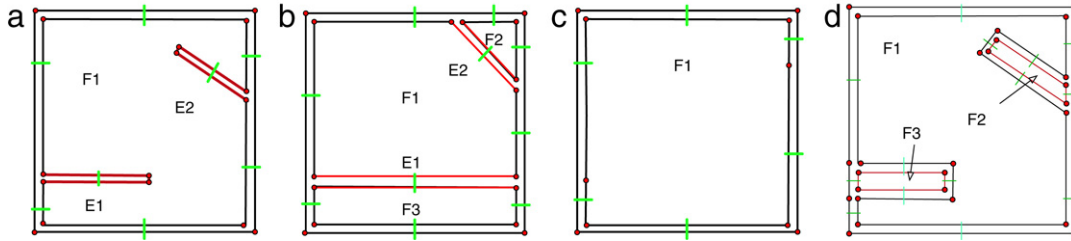


Fig. 24. Dangling edges processing: (a) dangling edges E_1 and E_2 in a face F_1 ; (b) E_1 and E_2 are extended; (c) E_1 and E_2 are removed and the remaining vertices are updated with an α_1 link; (d) faces F_2 and F_3 are constructed by thickening E_1 and E_2 .



Fig. 25. Geometric update for extending edges: (a) one edge is extended, one vertex must be inserted; (b) the extension corresponds to an existing vertex; (c) both edges are extended.

definition: a dangling edge contains a dart such that $\alpha_1(b) = \alpha_2(b)$. Unfortunately, correction depends on the configuration of building plans and can thus not be automatic. We propose three operations for correcting the corresponding inconsistencies, that can be used at any moment by the user. A dangling edge can be:

- extended up to the nearest intersection (Fig. 24b);
- removed (Fig. 24c);
- thickened to build a wall (Fig. 24d).

Extension operation: this operation is always processed for two edges of which at least one is dangling. Depending on their relative position, the intersection point is computed (see Fig. 25) and the corresponding darts are updated. The extension process splits a face F_1 (see Fig. 24a) into three faces F_1 , F_2 and F_3 (Fig. 24b). F_2 and F_3 are given semantic information of F_1 .

We also propose a selection system to reduce the processing time, so as to let the user define a set of dangling edges to be extended. This high-level operation allows us to correct quickly many dangling edges inconsistency which corresponds to a small geometric error.

Removal operation: removing an edge implies to remove four darts; the formerly adjacent darts are linked by α_1 [37], so as to ensure consistency. Note that any edge can be removed with this operation, including non-dangling edges. However, removed edges should not separate two faces with different semantic.

Thickening operation: a dangling edge in a face F_1 sometimes correspond to a wall without thickness (Fig. 24a). The thickening operation consists in choosing one side of the edge to construct a face (Figure: 24d); if the new edges intersect the boundary of F_1 , the other side is chosen; if the new edges still intersect F_1 boundary, the operation is cancelled.

Semantic property

Semantic information is deduced when possible from the architectural plans layers. In most plans, each layer corresponds to a given type of object (walls, doors, windows, etc.). However, these layers are constructed manually by architects and some errors recurrently occur (for instance doors are drawn in the walls layer). With topology, it is possible to check and detect semantic inconsistencies as explained in Section 4. Therefore, each face in the model is examined and the neighbour faces are checked. If the corresponding semantic does not comply with constraints, the corresponding faces are automatically selected for manual correction.

This detection process can also be used to detect walls without thickness (that do not correspond to dangling edges, such as in Fig. 26), using semantic properties. Corrections can then be applied by the user, according to removal or thickening operations (see Fig. 26c).

7. Results

7.1. From 2D to 3D

We have applied our reconstruction process to various buildings, with different sizes and shapes. Some of our test scenes are shown in Fig. 27 and computing times have been obtained with a CPU 2 GHz, Apple MacBook, Intel core duo, with 1 Gb of memory. The 2D reconstruction process is automatic:

- edges processing with intersection, overlap and topology;
- angle arrangement and face construction;
- detection of dangling edges.

Processing time for geometrical edges and topological face construction is given in Table 2. During this process, many

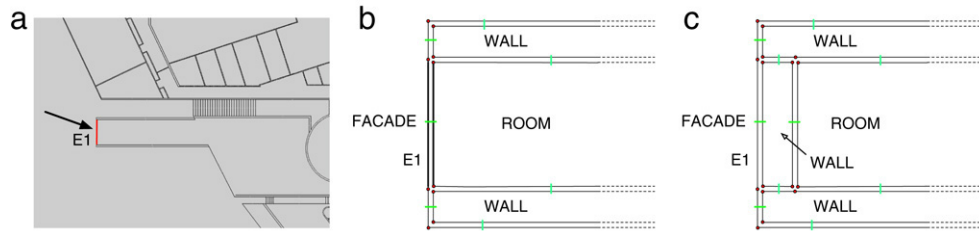


Fig. 26. Semantic inconsistency detection: (a) bold edge E_1 does not respect the semantic constraint; (b) zoom: E_1 is incident to the facade and a room, there is no wall which separates those objects; (c) a face wall is created to model separation between facade and room.

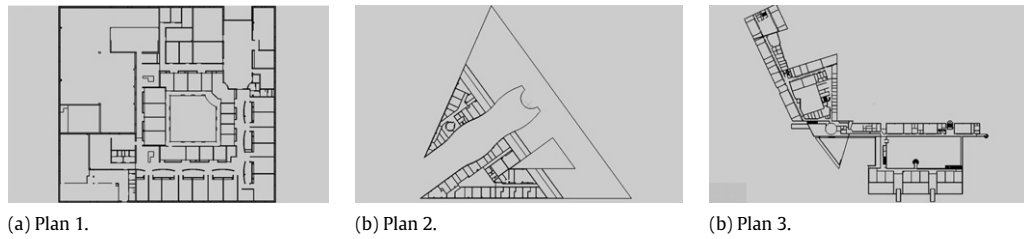


Fig. 27. Three of our test buildings: (a) 4352 edges; (b) 3188 edges; (c) 7006 edges.

Table 2

2D processing for 3 test buildings. The 4th column corresponds to the computing time required for edges correction and faces topological reconstruction. The second part of the table corresponds to dangling edge processing with the computing time for automatic correction. The last column indicates the manual processing time, including correction of remaining dangling edges and manual semantic instantiation.

| | # 2D edges | # edges overlap | # edges intersection | Reconstr. time (s) | # dangling edges | Time automatic (s) | Time manual (min) |
|--------|------------|-----------------|----------------------|--------------------|------------------|--------------------|-------------------|
| Plan 1 | 4352 | 39 | 4 | 11 | 22 | 1 | 15 |
| Plan 2 | 3188 | 0 | 31 | 10 | 59 | 2 | 5 |
| Plan 3 | 7006 | 26 | 902 | 30 | 986 | 40 | 40 |

Table 3

Buildings statistics: number of walls, rooms, doors and windows. The 4th column provides the computing time for the whole 3D reconstruction.

| | # walls | # rooms | # doors | # windows | 3D construction (s) |
|--------|---------|---------|---------|-----------|---------------------|
| Plan 1 | 807 | 63 | 73 | 39 | 3 |
| Plan 2 | 583 | 72 | 70 | 78 | 5 |
| Plan 3 | 750 | 128 | 142 | 138 | 6 |

operations require testing couples of darts according to their location in the plan. The automatic process relies on a uniform grid for reducing the processing time, with a tile size equal to one meter. Each tile is associated with the list of vertices it contains. Thus, for each segment, tests are performed only in a local neighbourhood. The algorithm complexity depends on the overall segments distribution in the scene.

In Table 2, the first 4 columns correspond to an automatic process that constructs a 2D partition and corrects 100% of edges overlaps and intersections. Column 5 corresponds to the *extension* operation (automatic), first applied to all dangling edges with an extension distance threshold given by the user (1 mm is used as a default value). This operation corrects about 80% of dangling edges, the corresponding time is given in column 6. Column 7 provides the time required by the user for correcting the remaining (20%) dangling edges, with the operations described in Section 6. No other processing is required for an entire 2D plan before (automatic) 3D extrusion. Without our application, the manual correction of such errors (in 2D only) requires to process independently each segment or geometric primitive. This can take many hours depending on the plans.

Note that due to our model properties, during the extension operation, the closest target edge is selected in the current face, so that complexity only depends on the number of edges in faces.

Table 3 gives results for the 3D reconstruction process, with the number of volumes in each building. The computing time corresponds to the whole 3D building automatic extrusion. Figs. 28 and 29 show the 3 test buildings reconstructed with our method.

Manually reconstructing such a topological representation in 3D is a tedious process for the user because volumes have to be manually modeled and each topological relationship must be explicitly defined. Once more, this manual process may require many hours. With our system, no manual operation is required during extrusion: the 3D building is automatically reconstructed from the 2D plan.

7.2. Topology for simulation

Lighting simulation in complex environments still remains a challenging process. However, several authors have proposed methods dedicated to large buildings, using a data structure called *cells and portals* [6–8]. Cells correspond to rooms and corridors while portals are faces separating cells. During simulation the aim is to store and process only a few cells at the same time, so as to reduce both computing time and memory loads. Our model contains all semantic information necessary for producing cells and portals. Each opening volume is associated with one of the two corresponding rooms, defining cell volumes; portals correspond to the face at the interface between two cells (see Fig. 30). Thus the input data needed by the lighting simulation system described in [8] or [6,27] can be easily generated by our modeler.

In addition, the XLim/SIC laboratory also proposes a radio-wave propagation simulation system adapted to indoor scenes [9,10]. The input data structure used is only composed of wall volumes. The input files describe the wall faces with transmission properties. Input files for this software have been produced for experiments.

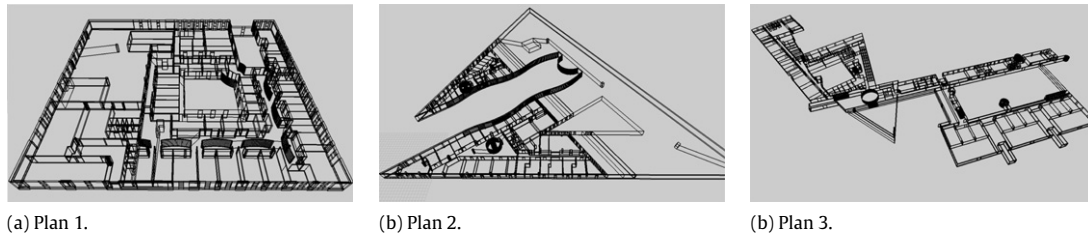


Fig. 28. 3D reconstructed buildings.

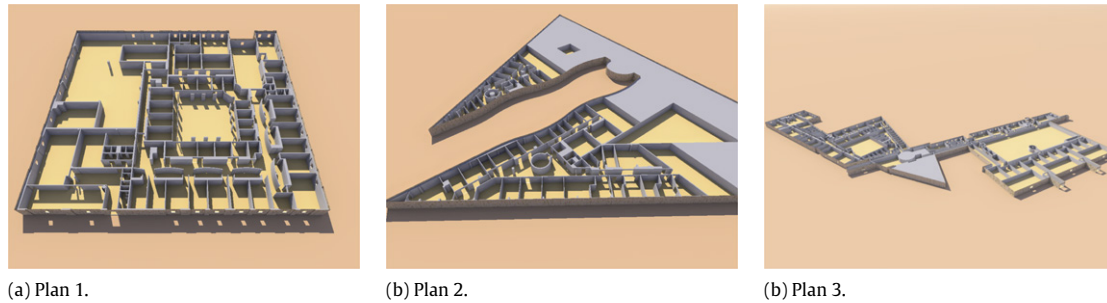


Fig. 29. Images generated with POV-Ray.

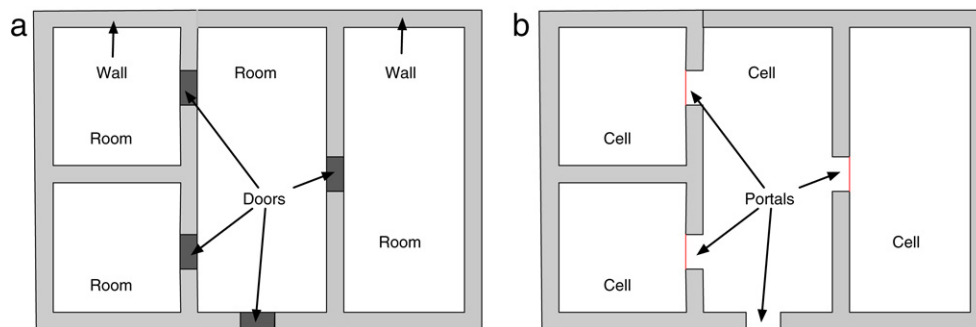


Fig. 30. Construction of cell and portal structures in 2D (a) 2D architectural plan representation; (b) each opening is associated with one of its 2 adjacency rooms, defining cell volumes; portals correspond to the face between the 2 rooms.

8. Limitations

The consistency constraints described in this paper correspond to most *usual* buildings, with floors clearly defined and vertical walls. In addition, 8 semantic key markers have been defined for most buildings; obviously, architects can specifically design buildings that would not fit with our system.

Our reconstruction system requires plans with 2D geometric data that can be clearly separated from the text. We use only linear embedding; curved objects such as circles have to be decomposed into many edges, and some geometric information is lost. In addition, buildings with non-vertical walls cannot be extruded without supplementary information. For instance blob buildings are not supported by our application. The processing of such environments would require specific extrusion operations so that roofs and walls be properly reconstructed and connected.

Finally, we make the assumption that floor plans are represented in the same orientation/coordinate system, so that they can be superimposed. In addition, when rooms extend over more than one floor, the corresponding ground/ceiling should be removed. This operation has not been defined in our application yet.

9. Conclusion

This paper presents a set of consistency constraints dedicated to architectural (indoor) environments as well as a reconstruction

system that produces 3D virtual buildings from 2D architectural plans. Our model is homogeneous in any dimension, and handles geometry, topology and semantics. It relies on the G-maps representation.

Our system is capable of detecting plans that do not contain errors, using topology and consistency constraints. Plans without error can be automatically reconstructed in 3D, while correction operations are proposed to the user for correcting errors in a few minutes.

More precisely, our reconstruction system is composed of the following steps. Firstly, 2D inconsistencies are automatically detected and corrected according to the defined consistency constraints. Secondly, several automatic and semi-automatic operations are provided to the user for editing the 2D model, while guaranteeing the overall building correctness (orientation, closeness, neighbourhood relationship, and so on). Finally, the 3D building is generated from the 2D representation using specific extrusion operations relying on 2D semantics. The resulting model also complies with our consistency constraints.

Our 3D model can be used on the one hand for lighting simulation based on the room and portal volumes [6–8] and on the other hand for radio-wave propagation based on wall volumes [9, 10]. We aim at performing other types of simulation such as acoustics, heat transfer and so on.

We are currently investigating 3D editing operations for coherently inserting, moving or removing elements such as walls,



Fig. 31. Images produced with Pov-Ray using radiosity. This image correspond to plan 1.

doors or windows. We are also interested in applications for automatically furnishing 3D buildings (see Fig. 31).

References

- [1] Punjabi S, Miranda V. Development of an integrated building design information interface. *Building Simulation* 2005;969–76.
- [2] Lienhardt P. *N*-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry & Applications* 1994;4(3):275–324.
- [3] Vidil F, Damiand G. Moka; www.sic.sp2mi.univ-poitiers.fr/moka/, 2003.
- [4] David Luebke CG. Portals and mirrors: Simple, fast evaluation of potentially visible sets. In: *Symposium on interactive 3D graphics*. 1995. p. 105–6.
- [5] John FPB, Airey M, Rohlf John H. Towards image realism with interactive update rates in complex virtual building environments. *Symposium on interactive 3D graphics*. 1990 p. 41–50.
- [6] Teller S, Fowler C, Funkhouser T, Hanrahan P. Partitioning and ordering large radiosity computations. *ACM SIGGRAPH Computer Graphics* 1994;443–50.
- [7] Meneveaux D, Bouatouch K, Maisel E. Memory management schemes for radiosity computation in complex environments. *Computer Graphics International* 1998;706–14.
- [8] Fradin D, Meneveaux D, Horna S. Out-of-core photon-mapping for large buildings. In: *Proceedings of eurographics symposium on rendering*. 2005. p. 65–72.
- [9] Vauzelle R, Pousset Y, Escarieu F. A sensitivity study for an indoor channel simulation. *Annals of Telecommunication*. 59:655–72.
- [10] Combeau P, Vauzelle R, Pousset Y, Aveneau L. An optimization in computation time for the prediction of radio coverage zones. *Radio Science* 42. RS1003. doi:10.1029/2006RS003492.
- [11] Rossignac J, O'Connor M.. A dimension-independent model for pointsets with internal structures and incomplete boundaries. *Geometric modeling for product engineering*. In: *Proceedings of the IFIP workshop on CAD/CAM*. 1989. p. 145–80.
- [12] Brisson E. Representing geometric structures in d dimensions: Topology and order. *Discrete & Computational Geometry* 1993;387–426.
- [13] Baumgart B. A polyhedron representation for computer vision. In: *AFIPS nat. conf. proc. 44*. Alirington (VA): AFIPS Press; 1975. p. 589–96.
- [14] Guibas L, Stolfi J. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. *Transactions on Graphics* 1985;131–9.
- [15] Weiler K. The radial-edge data structure: A topological representation for non-manifold geometry boundary modeling. In: *Proceedings of IFIP WG 5.2 working conference*. 1986. p.3–36.
- [16] Weiler K. Boundary graph operators for non-manifold geometric modeling representations. *Geometric Modeling for CAD Applications* 1988;37–66.
- [17] Françon J, Bertrand Y. Topological 3D-manifolds: A statistical study of the cells. *Theoretical Computer Science* 2000;234(1–2):233–54.
- [18] Lienhardt P. Topological models for boundary representation: A comparison with *n*-dimensional generalized maps. *Computer-Aided Design* 1991;23(1): 59–82.
- [19] Ah-Soon C, Tombre K. Architectural symbol recognition using a network of constraints. *Pattern Recognition Letters* 2001;22(2):231–48.
- [20] Lu T, Tai CL, Su F, Cai S. A new recognition model for electronic architectural drawings. *Computer Aided Design* 2005;37(10):1053–69.
- [21] Google. Google Earth. www.earth.google.com. 2004.
- [22] Lafarge F, Descombes X, Zerubia J, Pierrot-Deseilligny M. An automatic 3D city model: A bayesian approach using satellite images. In: *IEEE international conference on acoustics, speech and signal processing*. 2006.
- [23] Boudet L, Paparoditis N, Jung F, Martinoty G, Pierrot-Deseilligny M. A supervised classification approach towards quality self-diagnosis of 3D building models using digital aerial imagery. In: *ISPRS commission 3 symposium on photogrammetric computer vision*. 2006.
- [24] Hongjian Y, Shiqiang Z. 3D building reconstruction from aerial CCD image and sparse laser sample data. *Optics and Lasers in Engineering* 2006;44:555–66.
- [25] Suveg I, Vosselman G. Reconstruction of 3D building models from aerial images and maps. *International Journal of Photogrammetry and Remote Sensing* 2004;58:202–24.
- [26] Brenner C. Building reconstruction from images and laser scanning. *International Journal of Applied Earth Observation and Geoinformation* 2005;6: 187–98.
- [27] Meneveaux D, Maisel E, Bouatouch K, Delmont R. A new partitioning method for architectural environments. *The Journal of Visualization and Computer Animation* 1998;9(4):195–213.
- [28] Fradin D, Meneveaux D, Lienhardt P. A hierarchical topology-based model for handling complex indoor scenes. *Computer Graphics Forum* 2006;25(2): 149–62.
- [29] Fradin D. Modélisation et simulation d'éclairage à base topologique : Application aux environnements architecturaux complexes. Ph.D. thesis. France: University of Poitiers; 2004.
- [30] GraphiSoft. Archicad. www.archicad.com. 2008.
- [31] Autodesk. Autocad. www.autocad.com. 2008.
- [32] Parish Y, Mueller P. Procedural modeling of cities. *ACM SIGGRAPH Computer Graphics* 2001;301–8.
- [33] Mueller P, Wonka P, Haegler S, Ulmer A, Gool LV. Procedural modeling of buildings. *ACM SIGGRAPH Computer Graphics* 2006;614–23.
- [34] Wonka P, Wimmer M, Ribarsky FSansW. Instant architecture. *ACM Transactions on Graphics* 2003;22(4):669–77.
- [35] Chen Guoning, Esch Gregory, Wonka Peter, Müller Pascal, Zhang Eugene. Interactive Procedural Street Modeling. *ACM SIGGRAPH Trans Graph* 2008; 27(3):1–10.
- [36] Horna S, Damiand G, Meneveaux D, Bertrand Y. Building 3D indoor scenes topology from 2D architectural plans. *GRAPP* 2007;37–44.
- [37] Damiand G, Lienhardt P. Removal and contraction for *n*-dimensional generalized maps. *Discrete Geometry for Computer Imagery* 2003;2886: 408–19.