# Combinatorial models for topology-based geometric modeling

Pascal Lienhardt, Laurent Fuchs, Yves Bertrand

# Combinatorial Models for Topology-Based Geometric Modeling

*Pascal Lienhardt, Laurent Fuchs, and Yves Bertrand*

Contents

## 1. Introduction.

### 1.1. Representation of subdivisions.

Topological models[1] studied in this paper are defined in order to represent *subdivided* objects, i.e. partitioned into *cells* of different dimensions: vertices, edges, faces, volumes, etc. Such models are often used for the following reasons:

1. « The object » one intend to handle has a *structure*, which is important for the application [12, 10, 51], for instance the representation of geological layers for geological applications (cf. figure 1): such layers are often broken by faults, and thus they are composed by many blocks. Similarly for architectural applications, buildings are composed by different rooms, walls, doors, etc. For these applications, assemblies of volumes have to be represented: such volumes share faces, faces share edges and edges share vertices.

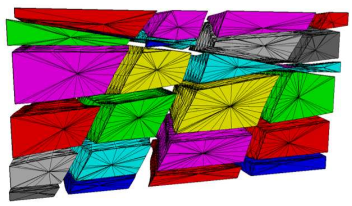


*Figure 1* – Geological layers.

The representation of *neighborhood relations* between cells is also important, for instance in order to know the doors and windows which close a room. So we want to represent the cells *and their incidence and adjacency relations* [80]. A similar need arises in image analysis [20], since it is important for many applications to represent the structure of an image segmented into regions. For these examples, cells can be *any* cells, and thus *cellular topological models* are often used (cf. section 3).

[1] All notions here are based upon Combinatorial Topology [24].

2. It is often useful or efficient to handle *discretized* objects, for instance [36, 46, 89, 31, 88, 28]:

   – due to the modeling method itself: an object can be modeled by a set of patches (e.g. Bézier patches); reconstructing objects from images, using the marching-cubes method, produces triangulations;

   – due to the operations: for instance for realistic rendering, objects are often triangulated, in order to optimize the computation of ray-object intersections; many simulation applications handle meshes.

   As before, it is necessary to represent cells and their incidence or adjacency relations. For such applications, cells are often *regular* ones (e.g. tetrahedra, cubes, etc.), and *simplicial models* are mainly used (cf. section 2).

Note also that topology-based geometric modeling methods[2] make it possible to represent local informations and to apply local operations, since the represented objects are subdivided.

### 1.2. Distinction between topology and embedding.

Topological models represent the *structure* of subdivided objects: usually, such models represent cells as *abstract* objects, and incidence or adjacency as *relations* between cells. For instance for incidence graphs (cf. section 3.1), the nodes of a graph correspond to the cells, and the edges of the graph correspond

[2]This terminology has been proposed by Jean Françon at the end of the 80's, in order to distinguish more clearly between the modeling of subdivisions and Boundary Representation methods (B-rep). Schematically, B-rep methods intend to model a « solid » (i.e. a volume) by the surface which bounds it; and usually, a subdivided surface (orientable without boundary) is represented. So, many work dealt with the definition of models and operations for handling surface subdivisions. When new works dealt with subdivisions of the 3D space [103] (i.e. objects composed by several volumes), some confusion arises, i.e. what has to be modeled: sets of faces which bound volumes, or sets of volumes? This question has important consequences, since the topological dimension of faces (resp. volumes) is equal to 2 (resp. 3). So, it became important to distinguish more clearly between *boundary representation* and *subdivision representation*. Moreover, the boundary of a solid is not necessarily subdivided. Note also that an other meaning for Boundary Representation is mentioned in section 3.1

to incidence relations between cells. In order to get a whole geometric model, it is necessary to link the topological model with an *embedding* model, which describes the *shapes* of cells and thus the shape of the object [9]. For instance, an object modeled by a set of triangular Bézier patches can be represented by a semi-simplicial set (cf. section 2.2), which represents the structure of the set of patches, and each « abstract » simplex is associated with a patch [68]. For representing polyhedra, each abstract vertex (resp. edge, face, etc.) is associated with (i.e. *embedded* into) a point (resp. a line segment, a part of a plane, etc.). More generally, a topology-based geometric model is a semi-explicit representation: the structure (topology) is explicitly described, the shape is more or less explicitly represented. For instance for a subdivision of the plane, the shapes of the faces can be deduced from the shapes of vertices and edges which bound them (and when edges are embedded as line segments, their shapes can be deduced from the points which are associated with their extremity vertices).

Several important interests come from this distinction between topology and embedding:

– for *computing informations* [55, 76, 5, 37, 57, 91]. Several informations can be deduced from the topological model, for instance the topological characteristics of a surface [58]. These characteristics (number of boundaries, orientability factor, genus) make it possible to distinguish between different types of surfaces (cf. figure 2), and they can be useful in order to check the object validity during the construction, or to control the construction process itself. Similarly, the explicit representation of the object structure by a topological model can be very useful, for instance for comparing or matching objects (e.g. automatic matching between image regions, automatic re-application of a construction process when several parameters have been modified, etc.).

– for *constructing* objects [70, 40, 19, 9, 53]. For instance for animating articulated objects, the shape is modified during time, but not the structure: so, no topological operation is applied. Other operations can be decomposed into more local and basic operations. For instance, cell rounding can be defined as the composition of a topological chamfering operation and a geometric rounding operation (cf. figure 3): the topological model provides informations which are used in order to control the shape modifications. The converse exists for other operations [18]: for instance, boolean operations can be defined as compositions of local operations (e.g. cell split and merge), and embedding information control the topological modifications.

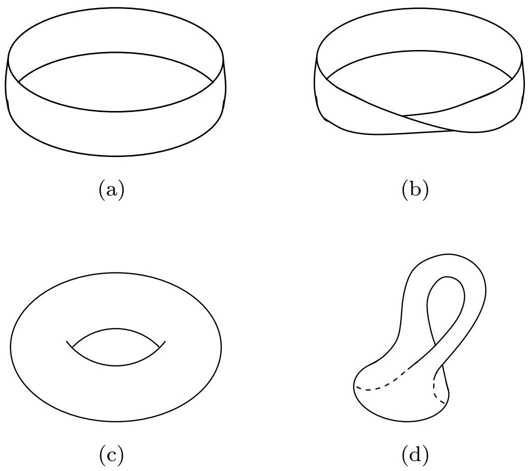



*Figure 2* – Several surfaces. The ring (a) (resp. the Möbius strip (b)) is an orientable (resp. non orientable) surface with one boundary. The torus (c) (resp. the Klein bottle (d)) is an orientable (resp. non orientable) surface without boundary.

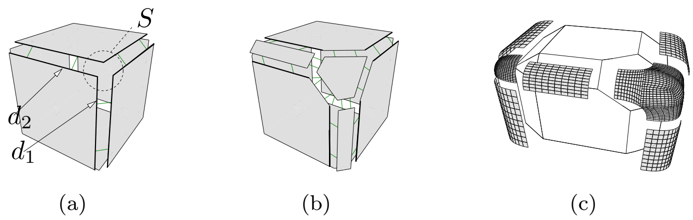



*Figure 3* – Chamfering and rounding vertices and edges. Edges $d_1$, $d_2$ and vertex $S$ (a) are chamfered (b). Rounding, by associating surface patches (c).

### 1.3. Different classes of models, and general frame.

Many models (and operations) have been proposed within different fields, in order to represent and handle subdivided objects, e.g. geometric modeling, computational geometry, image processing and analysis, topology, mechanics of solids, etc [6, 3, 77, 101, 59, 48, 103, 38, 42, 29, 13, 72, 71, 39, 4, 94, 60, 78, 87, 98, 49, 23, 14, 61, 89, 66, 74, 45, 90, 67, 50, 17, 8, 69, 11, 15, 32, 28, 30, 26, 83, 92]. We have to notice (and this is reassuring) that the proposed solutions are often similar, whatever their original field is. Schematically, models can be classified according to (cf. figure 4):

1. the *type* of cells: we can distinguish between models which represent assemblies of:

   – *regular* cells: simplices (triangles, tetrahedra, etc.), cubical simplices (squares, cubes, etc), simploids (which are cartesian products of simplices)[3], etc;

   – *any* cells: in fact, cells are never « any » cells; cellular models make it possible to handle more general cells, but such cells generally satisfy topological properties (cf. section 3);

2. the *type of assembly*: we can distinguish between topological models which make it possible or not to represent subdivisions in which cells are incident several times between them (i.e. *multi-incidence*). For instance, a loop is an edge the two extremities of which are identified into one vertex: so the edge is incident twice to the vertex. For both cases, we can distinguish other sub-classes, mainly:

   – *quasi-manifolds*, which are characterized by important topological properties, for instance surfaces[4]. Models exist for representing

[3] The set of simploids contains thus simplices themselves, cubes, since they are products of 1-dimensional simplices (i.e. edges), $3D$ prisms which are products of a triangle and an edge, etc.

[4] A subdivided surface without multi-incidence can be constructed by « gluing » faces along their boundary edges, in such a way that any edge is incident to at most two faces; more generally, an $n$-dimensional quasi-manifold can be constructed by gluing $n$-cells by

any quasi-manifolds (orientable or not orientable, with or without boundary), or for representing oriented quasi-manifolds without boundary;

– *complexes* (such objects are usually called *non-manifolds* in the geometric modeling literature): they are « any » assemblies of cells[5], the word « any » having to be more precisely defined (cf. section 3).

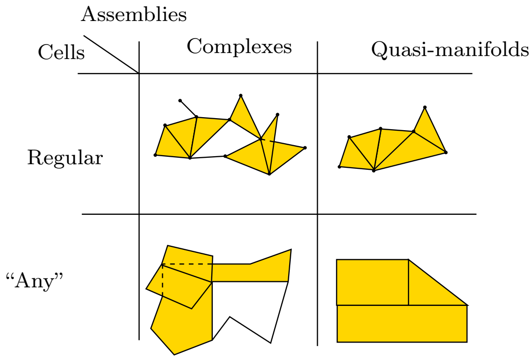



*Figure 4* – Different classes of subdivisions.

Instead of enumerating topological models, the goal of this paper is to present a general frame for defining simplicial, simploidal and cellular models (cf. figure 5 and figure 6). For instance, it is possible to deduce models for representing sub-classes of simplicial objects from « general » simplicial models, by mechanisms based upon the topological properties satisfied by these sub-classes. In fact, it is a classical way for optimizing data structures (for instance, optimized data structures can be deduced from a general graph data structure in order to represent graphs satisfying some regularities). Cubical models and

identifying their boundary $(n-1)$-cells, in such a way that an $(n-1)$-cell is incident to at most two $n$-cells. Quasi-manifolds exist, the geometric representation of which are not manifolds, i.e. which contain points the neighborhoods of which are not homeomorphic to $n$-balls. In fact, it is well-known that manifolds can not be combinatorially characterized, but quasi-manifolds can be. Note also that any quasi-manifold is a pseudo-manifold, but the converse is not true.

[5]For instance in dimension 2, an edge can be incident to more than two faces, faces can be « glued » along a vertex, etc.

more generally simploidal models can be deduced from simplicial models, using the fact that cells are cartesian products of simplices. Similarly, cellular models can be deduced from simplicial models, using a simple numbering mechanism which induces a notion of cell. As for simplicial models, mechanisms can be conceived in order to represent sub-classes of simploidal and cellular objects (quasi-manifolds, etc.) [73, 14, 74, 45, 7, 2].

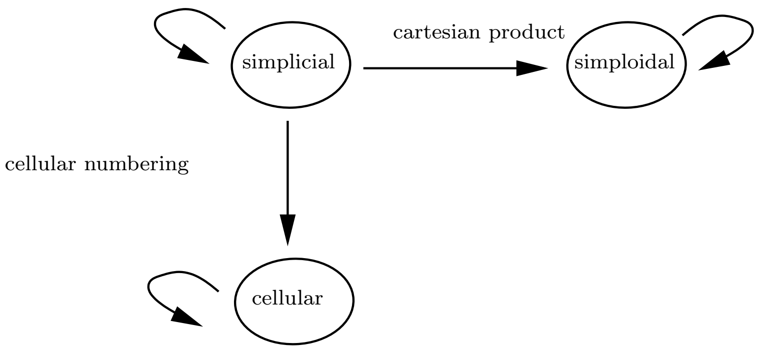



*Figure 5* – General frame for the definition of topological models. Loops correspond to optimization mechanisms according to the topological properties of sub-classes.

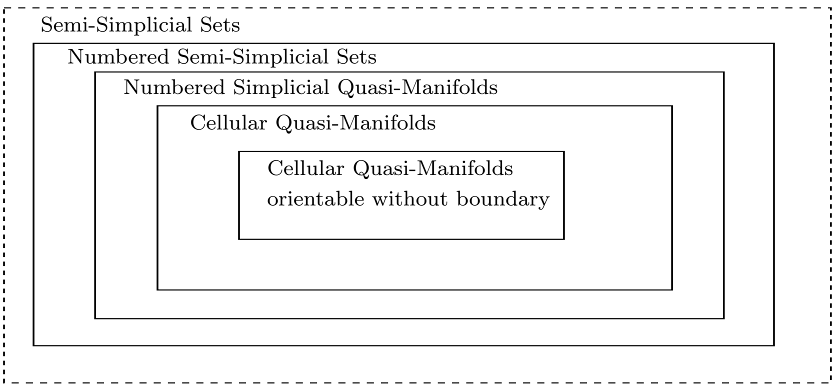



*Figure 6* – Examples of sub-classes: we could also distinguish between models according to the fact that multi-incidence is allowed or not.

Such a general frame makes it possible to:

– extend to cellular models important properties defined in combinatorial topology for simplicial models;

– convert cellular models into simplicial ones (and conversely under certain restrictions); so, cells of cellular models are not so general, and it is possible to study their properties.

It is thus also possible to choose a topological model for a particular application, according to several criteria.

### 1.4. Model choice.

The choice of a topological model obviously depends upon the type of objects one intend to handle, and also upon the operations which have to be applied, the space/time complexity and the complexity of software development [9, 41]. It is thus a classical problem: choosing a data structure, taking into account:

1. the « type » of objects. It is possible to define a general model in order to represent « any » type of cellular complex, and the other models are in fact optimizations adapted for particular sub-classes. Such optimizations are deduced from the topological properties which characterize the sub-classes, by making implicit some information. It is obvious that using a model defined for a sub-class in order to handle objects of a larger class can lead to important errors. For instance, incidence graphs represent cellular complexes without multi-incidence. Even if it is possible to intuitively deduce an incidence graph from a subdivision in which some cells are multi-incident, the formal interpretation of the resulting graph does not correspond in any way to the initial subdivision (cf. section 3.1);

2. the operations applied to the objects. For instance, boolean operations (union, intersection, difference) applied to simplicial objects do not directly produce simplicial objects (cf. figure 7). A first way consists in using a cellular model for representing simplicial objects, applying the operation and triangulating the result. A second way consists in splitting

simplices when intersections are processed: the problem here lies in the fact that new simplices are added which perhaps produce new intersections, and examples exist showing that the whole process may be non convergent. As far as we know, this convergence problem has never been carefully studied.

3. the space/time costs of models and operations. Given a class $C$ of objects, a model $M_L$ corresponding to a larger class uses generally more memory space than an optimized model $M_C$ corresponding to $C$: some information is explicit within $M_L$, and implicit within $M_C$. It is clear that the time complexity of operations has also to be taken into account: when some operations often need to make explicit some implicit information, a less efficient model (according to space complexity) could be a better choice. For instance, representing a polyhedron by a list of faces can seem efficient; in fact, since adjacency and incidence relations are implicit, this representation is often not efficient when constructing objects, since adjacency and incidence information are used by many construction operations. Conversely, many algorithms do not need all information contained into the whole geometric model, and specialized models can be a better choice [93] (e.g. a list of faces for rendering algorithms).

4. the cost of operation conception. For instance, the definitions of several models do not take into account the constraints of consistency which have to be satisfied by the modeled objects (e.g. an edge incident to three vertices can be represented by an incidence graph: cf. section 3.1). The construction process has thus to control the modeled object validity. For instance, Euler operators have been defined in order to construct any subdivision of any orientable surface without boundary [81, 79]: each operator simultaneously creates or removes several cells; so, the implementation of Euler operators for handling incidence graphs, even if not complicated, is not so easy than for other basic operations defined for handling models in which the constraints of consistency are explicitly defined.

So, it is clear for us that « one best model » does not exist; some models are more adapted for a particular use, and thus conversion algorithms which can be deduced from the general frame presented in this paper are very important for practical purposes.

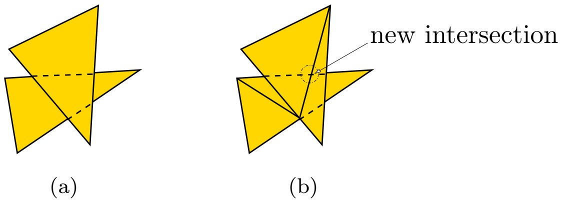



*Figure 7* – Boolean operations are not internal to the set of finite simplicial objects. (a) the intersection of two simplicial objects is not a simplicial one, (b) simplex splitting generates new intersections.

Basic simplicial and cellular models are presented in section 2 and section 3 respectively, together with basic related notions and operations. Obviously, it is necessary for particular applications to adapt such models and operations, and / or to define more elaborated models and operations from the basic ones.

## 2. Simplicial models.

Abstract simplicial complexes (ASCs) are one of the most known topological models [1, 89, 21, 32]. Semi-simplicial sets (SSSs) generalize them [82, 67, 68], i.e. a SSS can be associated with any ASC, but the converse is not true. In particular, SSSs can represent « curved » objects, maybe multi-incident, but it is not possible for ASCs. The *definition type* also differs: ASCs are sets of sets, SSSs are algebra.

Their definitions are here recalled, and also basic notions and operations (operations for computing topological properties are not discussed here, though many works deal with them [35, 37, 91]). Conversion operations are mentioned, and also classical model optimizations as for simplicial quasi-manifolds.

### 2.1. Abstract simplicial complexes

**Definition.** An abstract simplicial complex $K$ is defined upon a set $V$ of (abstract) vertices in the following way (cf. figure 8):

– a $p$-dimensional simplex is a set of $p+1$ vertices;

– $K$ is a set of simplices, such that any non empty subset of any simplex of $K$ is a simplex of $K$.

The dimension of $K$ is the highest dimension of the simplices of $K$.

Let $\sigma = \{v_0, ..., v_p\}$ be an abstract simplex. A *proper face* of $\sigma$ is a non empty subset of vertices of $\sigma$, different from $\sigma$. The *principal face* of $\sigma$ is $\sigma$ itself. The *boundary* of $\sigma$ is the ASC made of the proper faces of $\sigma$. The *star* of $\sigma$ is the set of simplices for which $\sigma$ is a (proper or principal) face. $\sigma$ is a *principal simplex* if it is not the proper face of any simplex.

Traversal algorithms can be defined using these notions, i.e. by traversing simplices of the boundary and / or the star of a simplex. These « neighborhood traversal » algorithms are very important for many operations. They correspond for topological models to « connectivity traversal » algorithms for graphs; they are either fundamental ones, and they can be defined in similar ways[6].

**Geometric realization.** The geometric realization of an ASC is a *simplicial complex* (cf. figure 8). An euclidean $p$-dimensional simplex is the convex hull of $p+1$ linearly independent points of an euclidean space. A face of an euclidean simplex is a simplex defined by a non empty subset of the points which define the simplex. A simplicial complex $L$ is a set of euclidean simplices which satisfy the two following properties:

– any face of any simplex of $L$ is a simplex of $L$;

– the (geometric) intersection of two simplices of $L$ is empty, or it is a face common to the two simplices.

[6]Note that simple graphs correspond to 1-dimensional ASCs.

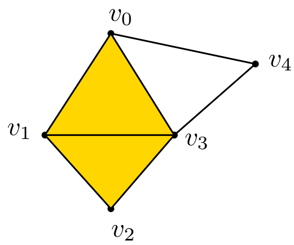



*Figure 8* – The geometric realization of ASC $K$, defined by:

$$\begin{array}{rcl} V & = & \{v_0, v_1, v_2, v_3, v_4\}, \\ K & = & \{\{v_0\}, \{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \\ & & \{v_0, v_1\}, \{v_0, v_3\}, \{v_0, v_4\}, \{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_3, v_4\}, \\ & & \{v_0, v_1, v_3\}, \{v_1, v_2, v_3\}\}. \end{array}$$

$\{v_0\}$ and $\{v_1, v_3\}$ are proper faces of $\{v_0, v_1, v_3\}$, which is a principal simplex. $\{\{v_0\}, \{v_1\}\}$ is the boundary of $\{v_0, v_1\}$. The star of $\{v_0\}$ is composed by $\{v_0, v_1\}$, $\{v_0, v_3\}$, $\{v_0, v_4\}$, $\{v_0, v_1, v_3\}$. The principal simplices of $K$ are $\{v_0, v_1, v_3\}$, $\{v_1, v_2, v_3\}$, $\{v_0, v_4\}$, $\{v_3, v_4\}$.

It is clear that several simplicial complexes can be associated with one ASC: the *geometric realization* of an ASC $K$ is the set of all (isomorphic) simplicial complexes which can be associated with $K$ (or sometimes it is one element of this set)[7].

**Representations.** Due to the definition of ASCs, they can be represented using well-known methods conceived for handling sets; from a practical point of view, the notions of face, boundary and star are handled using set inclusion operations, which have to be managed very efficiently. Note also that we can choose to explicitly represent:

- *all* simplices: the consistency constraint of ASCs has thus to be satisfied (i.e. any face of any simplex is a simplex of the ASC);
- the *principal* simplices: implicitly, all proper faces belong to the ASC. For the example of figure 8, we have to represent $K' = \{\{v_0, v_1, v_3\}, \{v_1, v_2, v_3\}, \{v_0, v_4\}, \{v_3, v_4\}\}$.

Embedding an ASC into an euclidean space consists in associating a simplicial complex with it, i.e. in associating a point with any vertex in such a

[7]It is possible to associate a simplicial complex in $\mathbb{R}^{2n+1}$ with any $n$-dimensional ASC [86].

way that the geometric constraints over simplicial complexes are satisfied (i.e. the intersection of two simplices is empty or it is a common face).

### 2.2. Semi-simplicial sets

**Definition.** An $n$-dimensional semi-simplicial set $S = (K, (d_j))$ is defined by (cf. figure 9):

- $K = \bigcup_{i=0}^{n} K^i$, where $K^i$ is a set of abstract objects called $i$-dimensional simplices, for any $i$ between 0 and $n$;
- maps $(d_j)$ are boundary operators: no operator is defined on $K^0$; for any $i$ between 1 and $n$, $i+1$ operators are defined on $K^i$, which associate each $i$-simplex with the $i+1$ $(i-1)$-simplices of its boundary[8]. These operators satisfy the following condition: for any simplex of dimension at least 2, the successive applications of two operators $d_i$ then $d_j$, with $j < i$ has the same result than the application of $d_j$ followed by $d_{i-1}$ [9].

The consistency of the simplicial structure is given by this constraint: without it, for instance a 2-simplex could be incident to six vertices[10].

Notions of proper face, principal face, principal simplex, boundary and star are here defined using boundary operators: for instance, $j$-simplex $\mu$ is a proper face of $i$-simplex $\sigma$ if a non empty sequence of boundary operators $d_{p_{i-1}}$, ..., $d_{p_j}$ exists, such that $\sigma d_{p_{i-1}} ... d_{p_j} = \mu$.

**Geometric realization.** The notion of simplicial complex can not be used in order to define the geometric realization of an SSS: this can be seen on the two following examples:

- $K^0 = \{\mu\}, K^1 = \{\sigma\}$, and $\sigma d_0 = \sigma d_1 = \mu$. This SSS describes a loop, i.e. an edge the extremities of which are identified. A simplicial complex

[8] $\forall i, 1 \leq i \leq n, \forall j, 0 \leq j \leq i : d_j : K^i \to K^{i-1}$.

[9] $\forall p, 2 \leq p \leq n, \forall \sigma \in K^p, \forall i, 0 \leq i \leq p, \forall j, 0 \leq j < i, \sigma d_i d_j = \sigma d_j d_{i-1}$ (where $d(\sigma)$ is denoted by $\sigma d$).

[10] A 2-simplex $\sigma$ can have three distinct 1-simplices in its boundary: $\sigma d_0$, $\sigma d_1$, $\sigma d_2$, and any 1-simplex $\mu$ can have two distinct 0-simplices in its boundary: $\mu d_0$ and $\mu d_1$. The constraint corresponds to the fact that vertices are equal two by two: $\sigma d_2 d_1 = \sigma d_1 d_1$, $\sigma d_2 d_0 = \sigma d_0 d_1$, $\sigma d_1 d_0 = \sigma d_0 d_0$.

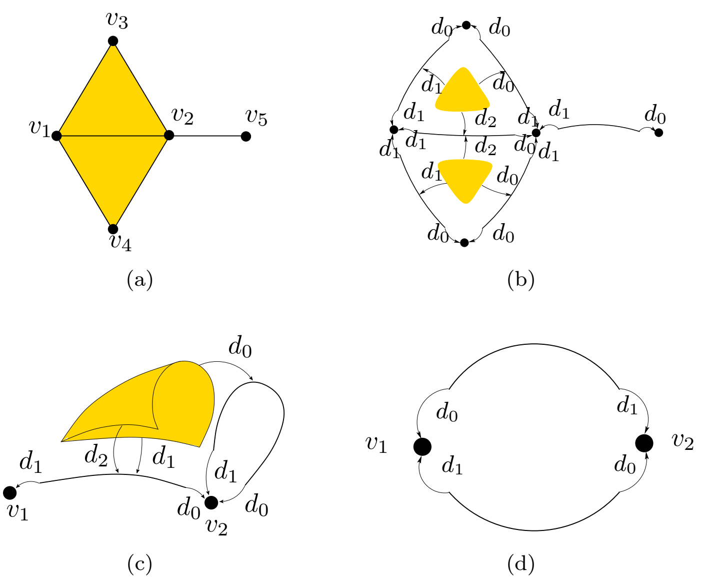



*Figure 9* – Examples of semi-simplicial sets. (a) $C$ is a simplicial complex. (b) An SSS describing the structure of $C$. (c) and (d) These two SSSs can not be associated with any ASC.

associated with this SSS is composed by a point, associated with vertex $\mu$, and edge $\sigma$ is *degenerated* into this point.

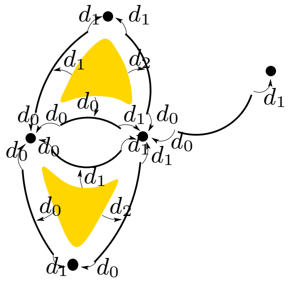
– $K^0 = \{\mu_0, \mu_1\}, K^1 = \{\sigma_0, \sigma_1\}$, and $\sigma_0 d_0 = \sigma_1 d_0 = \mu_0, \sigma_0 d_1 = \sigma_1 d_1 = \mu_1$. This SSS describes two edges sharing two vertices. A simplicial complex associated with this SSS is composed by two points, associated with the two vertices, and the two edges are embedded into one line segment.

The geometric realization of an SSS is a CW-complex (it is useless to recall here this notion) [82, 52]. The important fact for geometric modeling is the following: SSSs can represent simplicial objects, maybe with multi-incidence and maybe « curved » objects; ASCs are naturally associated with « linear » objects.

**Representations.** An oriented graph can easily be associated with any SSS: a node of the graph corresponds to a simplex (the dimension of it can also be associated with the node); an edge of the graph links two nodes corresponding to two simplices which are linked by a boundary operator (the index of the boundary operator is associated with the edge). This graph satisfies some properties: for instance, all nodes sharing a dimension have the same number of edges issued from them. So, it is possible to represent SSSs using methods conceived for representing oriented graphs. In practice, it could be useful to represent the inverse edges of graphs, for star traversal for instance.

SSSs can be embedded using parametric models (cf. figure 10), for instance Bézier simplicial spaces: for data structures, this consists in associating control points with simplices (according to the association method, $C0$-continuity can be implicitly controlled).

(a)




(b)

*Figure 10* – A semi-simplicial set, embedded using Bézier simplicial patches.

## 2.3. Basic operations

Two basic operations can be used in order to construct any simplicial object: *cone* and *identification*. We here recall their basic topological definition: note that many more elaborated operations have been proposed in order to control the structures and shapes of the resulting objects:

– schematically, a cone operation consists in adding a new vertex, and to create, for any initial $i$-simplex, a new $(i + 1)$-simplex linking it with the new vertex (cf. figure 11). As a particular case, any simplex (an its boundary) can be created by successive applications of the cone operation. This operation is easily defined on both ASCs and SSSs.

– the identification operation consists in identifying two simplices having same dimension within a simplicial object (cf. figure 11). For ASCs, this operation always consists in identifying vertices, and simplices can be degenerated by this operation (identifying two vertices incident to an $i$-simplex degenerates it into an $(i-1)$-simplex). If degeneracy has to be avoided, it is necessary to control the operation by adding a constraint (the cost for checking this constraint has to be taken into account). Degeneracy is not a problem for SSSs, since simplices are basic objects (identifying simplices of the boundary of a simplex has no effect on its existence); but identifying simplices could have surprising effects, due to the consistency constraint of SSSs.

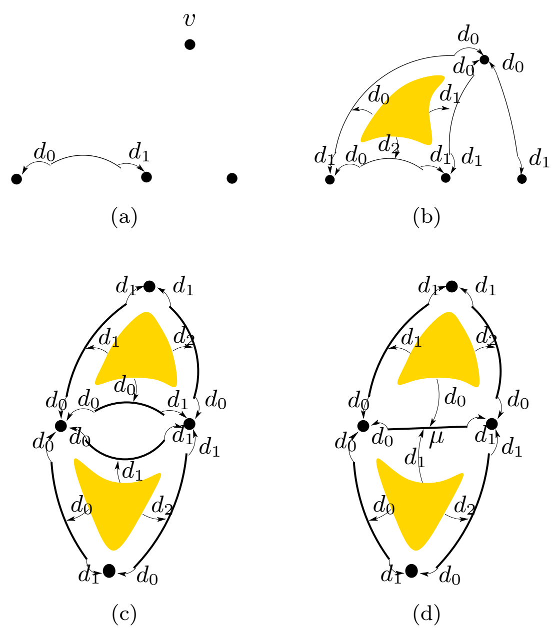



*Figure 11* – Cone operation $(a)$ and $(b)$. Identification operation $(c)$ and $(d)$ (two edges are identified into a single edge).

Many other basic operations have been defined [49, 89, 75] (cf. figure 12):

– *edge flip* is often used for handling triangulations of surfaces in many applications (cf. figure 12(a) and 12(b));

– *splitting* (cf. figure 12(c) and 12(d)) a simplex by a vertex results in splitting the simplex *and* its star[11]. This « propagation » explains why the definition of boolean operations by successive applications of split operations possibly do not converge (cf. section 1.4).

– *sweeping* (cf. figure 12(e) and 12(f)), and more generally cartesian product operations are defined on ASCs and SSSs.

Note also that many works deal with the computation of topological properties of ASCs and SSSs (e.g. homology groups), which provide information about the modeled objects which can be useful during a construction process (number of boundaries, of « holes », etc.).

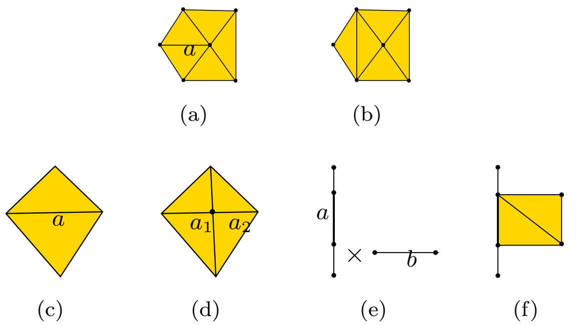



*Figure 12* – Some operations for handling simplicial objects. ($a$) and ($b$) flip of edge $a$. ($c$) and ($d$) split edge $a$ into $a_1$ and $a_2$, by adding a new vertex: note that the simplices of the star of $a$ are also split. ($e$) and ($f$) sweeping edge $a$ along edge $b$.

## 2.4. Classes of simplicial objects

**Conversion between ASCs and SSSs.** Simplicial objects associated with ASCs make a sub-class of objects associated with SSSs: so, it is always

[11]This « propagation » is general for all topological models with regular cells: cubic, simploidal, etc.

possible to associate an SSS with an ASC[12]. Conversely, it is possible to associate an ASC with any SSS without multi-incidence, i.e.:

– any $i$-simplex is incident to $i+1$ distinct vertices;

– two distinct $i$-simplices are incident to two distinct sets of vertices.

**Quasi-manifolds.** For many applications, it is important to handle subclasses of simplicial objects, for instance triangulations of surfaces or their combinatorial extensions in higher dimensions: the simplicial *quasi-manifolds*. Many works have dealt with the definition of local characteristic properties, or with *constructive* characterization. For instance, an $n$-dimensional simplicial quasi-manifold can be defined as a simplicial object which can be constructed by:

– adding $n$-simplices (and their boundaries), which are incident to $(n+1)$ distinct vertices;

– identifying $(n-1)$-simplices, in such a way that any $(n-1)$-simplex is incident to at most two $n$-simplices.

This constructive definition fits well with SSSs, and optimized models can be deduced for representing quasi-manifolds: schematically, we have to represent $n$-simplices and to replace boundary operators by « adjacency » operators between $n$-simplices (cf. figure 13). It is more difficult to apply this definition to ASCs, since degeneracy can not be dissociated from identification [31, 33]. For ASCs, several authors have proposed to explicitly represent principal simplices, and to add « pointers » corresponding to adjacency relations between $n$-simplices: this information is redundant, but it is useful in order to optimize traversal algorithms [89].

**Extensions and other models with regular cells.** The notion of simplicial set [82] generalizes SSSs, by adding a second class of operators: degeneracy operators. It is then possible to handle simplicial objects in which some simplices are degenerated ones: for instance, an edge of the boundary of a triangle

[12] Given an ASC, it is generally possible to associate several SSSs with it: intuitively, this is similar to the fact that several sequences can be associated with a set.

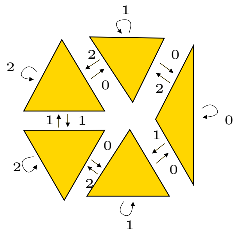



*Figure 13* – Optimizing SSSs for representing quasi-manifolds.

can be degenerated into a vertex, but the triangle itself is not degenerated. *A priori*, this has few interests for geometric modeling; a theoretical interest lies in the fact that cartesian product can easily be defined for simplicial sets, and it is thus possible to optimize it for semi-simplicial sets. Other extensions have been proposed in order to handle for instance « incomplete » simplicial objects, i.e. such that all simplices of the boundary of a simplex do not necessarily belong to the simplicial object. This kind of extension has also been proposed for cellular objects.

*Cubical sets* and more generally *simploidal sets* [95, 54, 92] have been defined in order to handle objects the cells of which are cartesian products of simplices (1-simplices for the cubical case). Simploidal sets can be used for instance for handling assemblies of Bézier patches (i.e. products of Bézier simplicial patches) [25, 85] (cf. figure 14).

The *type* of a simploid is defined by a $k-$tuple $(a_1, \ldots, a_k)$ of strictly positive integers; $k$ is the *length* of the simploid, $\sum_{l=1}^{k} a_l$ is its *dimension* (intuitively, a simploid is the product of simplices of respective dimensions $a_1, \cdots a_k$). It should be noted that a $p-$simplex is a simploid of type $(p)$ and that a $p-$cube is a simploid of type $(1, \ldots, 1)$ with length $p$.

A $n$-dimensional simploidal set $S = (K, (\epsilon^i_j))$ is the union $\bigcup_{p=0}^{n} K^p$ of sets of $p$-dimensional simploids, $0 \leq p \leq n$ equipped with *boundary operators* $\epsilon^i_j$ such that (figure 14):

$$\begin{array}{r} (\ldots, a_i, \ldots)\epsilon^i_j \\ \text{with } 0 \leq j \leq a_i \end{array} \quad :\longrightarrow \begin{cases} (\ldots, a_i - 1, \ldots) & \text{if } a_i > 1 \\ (\ldots, \widehat{a}_i, \ldots) & \text{otherwise} \end{cases} \tag{2.1}$$

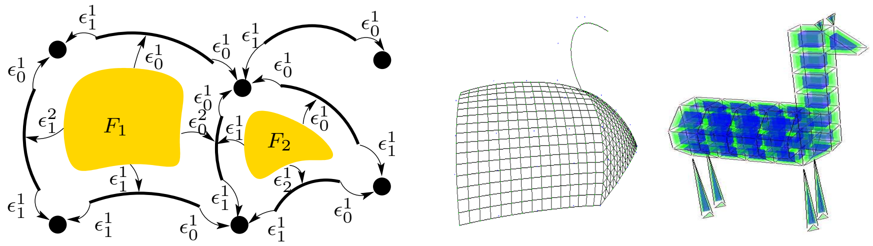
$$(2.2) \qquad (\ldots, a_i, \ldots)\epsilon^i_k \epsilon^i_l = (\ldots, a_i, \ldots)\epsilon^i_l \epsilon^i_{k-1} \qquad \begin{array}{l}\text{with } 0 \le l < k \le a_i \\ \text{and } a_i > 1\end{array}$$

$$(2.3) \qquad \begin{array}{r}(\ldots, a_i, \ldots, a_j, \ldots)\epsilon^j_k \epsilon^i_l \\ \text{with } i < j, 0 \le k \le a_j, 0 \le l \le a_i\end{array} = \begin{cases} (\ldots, a_i, \ldots, a_j, \ldots)\epsilon^i_l \epsilon^j_k \\ \qquad \text{if } a_i > 1 \\ (\ldots, a_i, \ldots, a_j, \ldots)\epsilon^i_l \epsilon^{j-1}_k \\ \qquad \text{otherwise.}\end{cases}$$

where $\widehat{a}_i$ means that $a_i$ is removed.

Equation (2.1) denotes the action of a boundary operator on the simploid type. The cartesian product of a simploid $s$ by a simploid of type (0) (i.e. a vertex) is equal to $s$. Hence, if a zero appears after the application of a boundary operator (i.e. if $a_i = 1$), it is removed from the type. Equation (2.2) corresponds to the commutation relation of boundary operators for semi-simplicial sets. Equation (2.3) corresponds to the commutation relation of boundary operators, when they are successively applied to two different simplices. The second part of this equation takes into account the shifts issued from suppressed zeros (for example, $(2,1,1)\epsilon^3_0\epsilon^2_1 = (2,1)\epsilon^2_1 = (2) = (2,1)\epsilon^2_1 = (2,1,1)\epsilon^2_1\epsilon^2_0$).

Note that this definition can be optimized for cubical sets for instance.




(a) (b) (c)

*Figure 14* – Simploidal sets. (*a*) boundary operators are explicitly represented (*b*) a simploidal set embedded by Bézier spaces (*c*) prisms link cubes and tetrahedra.

## 3. Cellular structures.

Defining a cellular topological model as a simple generalization of simplicial models is difficult, since one main interest of cellular models is the fact that cells can have « any » structure, contrary to simplices.

Many works dealt with the definition of cellular topological models. Although these models come from different fields (e.g. geometric modeling, image processing and analysis), they are quite similar and we can distinguish between two main classes: incidence graphs and ordered models, according to the terminology proposed by Brisson[13].

Incidence graphs can represent a large class of objects (without multi-incidence), but few works dealt with the characterization of these objects and the definition of consistency constraints in order to control the topological properties satisfied by the modeled objects. These problems are solved by ordered models. Some partial results exist [2], showing the equivalence between some sub-classes of incidence graphs and some ordered models. In these cases, conversion operations can be defined between both types of models.

### 3.1. Incidence Graphs.

**Introduction.** Several notions based upon *partially ordered sets* have been defined [7, 94, 13, 97], in order to represent different classes of objects (for instance having or not a complete boundary, etc.). Several models make it easier to express some topological properties or some construction operations. We here study incidence graphs (IG) in their basic definition.

An IG is an oriented graph the principle of which is the following (cf. figure 15): a node corresponds to a cell of the modeled object (the dimension of the cell is associated with the node); an oriented edge corresponds to a boundary relation between an $i$-dimensional cell and an $(i-1)$-cell of its boundary. An

[13]The « order » of cells is not explicitly represented by incidence graphs: for instance, two different subdivisions are presented in figure 17, corresponding to one incidence graph (we will see later that this graph does not correspond to any of these subdivisions). If we follow the boundary of face $F_1$ counterclockwise, we get two distinct edge sequences $(a, c, e, d, f, b)$ and $(a, c, f, d, e, b)$. Similarly, if we go round vertex 3 counterclockwise, we get the edge sequences $(b, f, e, c)$ and $(b, e, f, c)$. This order notion, which is not explicitly represented by incidence graphs, is explicit for ordered models: cf.figure 22.

object is *connected* when its associated graph is connected. The set of cells corresponding to the sons (resp. fathers) of a node $n$ is the *boundary* (resp. *star*) of the cell corresponding to $n$. As for simplicial models, these notions are important since they are the basis for defining the *local neighborhood* of a cell (i.e. adjacency and incidence relations, used in order to traverse an object or to control its construction; such traversals are fundamental operations for topological models).

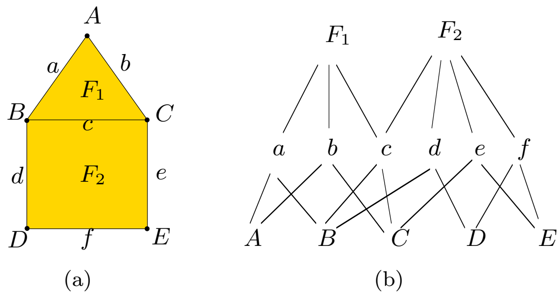



*Figure 15* – An incidence graph.

But as far as we know, no consistency constraints have been defined for IGs (equivalent to that of SSSs for instance). So, fundamental problems arise from the definition of IGs:

- what constraints have to be added to the definition of IGs in order to represent « valid » objects? For instance, an edge incident to three vertices can be represented by an IG (cf. figure 16). More precisely, what topological properties have to be satisfied by the cells? As said before, one main interest of cellular models is the fact that cells can be « any » cells, but this example shows that it could be necessary to be more precise;

- given a subdivided object, does the corresponding « intuitive » IG unambiguously represent this object? The answer is clearly no (cf. figure 17); more precisely, it is not possible to represent objects with multi-incidence using IGs, since it is necessary to know for these multi-incident cells the *order* of the cells of their boundaries and / or stars.

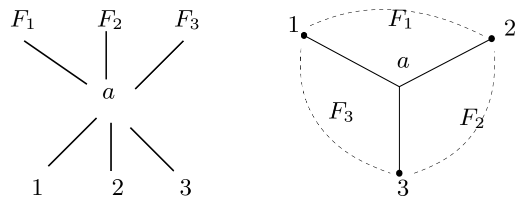



*Figure 16* – A part of a subdivision containing an edge incident to three vertices.

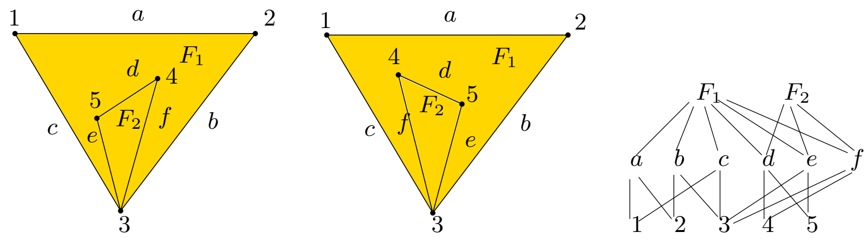



*Figure 17* – Two distinct objects corresponding to one incidence graph.

**Topological interpretation.** The first problem is studied in section 3.2 below. The second problem is solved by the fact that it is always possible to associate an ASC (and thus a SSS) *structured into cells* with any IG, in the following way (cf. figure 18):

– each simplex corresponds to a « partial path » within the IG: more precisely, each simplex corresponds to a sequence of incident cells of strictly increasing dimensions, i.e. to a particular sub-path of the IG;

– this ASC is structured. Each vertex of the ASC can be associated with the dimension of the corresponding cell, and the ASC can be partitioned in the following way: an $i$-dimensional subset corresponds to a vertex of the ASC numbered by $i$, and it is composed by this vertex and all simplices of its star which are incident to vertices numbered by a dimension lower than $i$. A 0-dimensional subset is thus defined by a vertex numbered by 0; a 1-dimensional subset is composed by a vertex numbered 1 and by the incident edges numbered by 0 and 1, etc. These subsets make a partition of the ASC and formally define the structure of cells and their incidence and adjacency relations as represented by the IG.

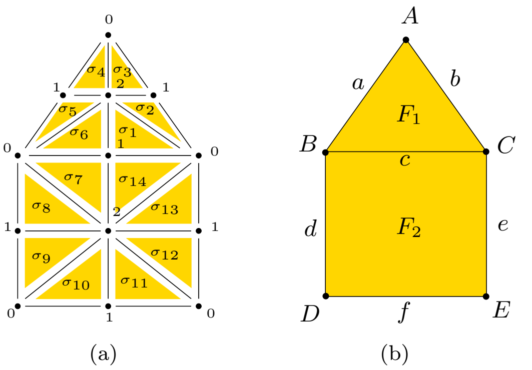



*Figure 18* – ($a$) an ASC structured into cells, corresponding to the IG of figure 15(b); ($b$) the corresponding object; for instance, 2-simplex $\sigma_3$ (resp. $\sigma_{14}$) corresponds to vertex $A$ (resp. $C$) seen from edge $b$ (resp. $c$), seen from face $F_1$ (resp. $F_2$), i.e to the path $(A, b, F_1)$ (resp. $(C, c, F_2)$). The 1-simplex incident to $\sigma_7$, the extremities of which are numbered 0 and 2, corresponds to path $(B, F_2)$. The ASC is structured into 5 vertices (i.e. the vertices numbered 0, corresponding to $(A)$, $(B)$, $(C)$, $(D)$ and $(E)$), 6 edges (i.e. the subsets composed by 0-simplices numbered 1 and the incident 1-simplices the extremities of which are numbered 0 and 1: $\{(a), (A, a), (B, a)\}$, $\{(b), (A, b), (C, b)\}$, $\{(c), (B, c), (C, c)\}$, $\{(d), (B, d), (D, d)\}$, $\{(e), (C, e), (E, e)\}$, $\{(f), (D, f), (E, f)\}$), and 2 faces (i.e. the subsets composed by 0-simplices numbered 2, the incident 1-simplices numbered 0 and 2, or 1 and 2, and the incident 2-simplices, i.e.

$$\{(F_1), (A, F_1), (B, F_1), (C, F_1), (a, F_1), (b, F_1), (c, F_1),$$
$$(A, a, F_1), (A, b, F_1), (B, a, F_1), (B, c, F_1), (C, b, F_1), (C, c, F_1)\}$$

and

$$\{(F_2), (B, F_2), (C, F_2), (D, F_2), (E, F_2), (c, F_2), (d, F_2), (e, F_2), (f, F_2),$$
$$(B, c, F_2), (B, d, F_2), (C, c, F_2), (C, e, F_2), (D, d, F_2), (D, f, F_2), (E, e, F_2), (E, f, F_2)\}.$$

We can now forget any intuition about cellular objects and *formally define a cellular object as a simplicial object structured into cells as described above* (as far as we know, all cellular models can be interpreted in a similar way): this solves the ambiguity problem of IGs (note that the object described by the IG of figure 17 is not one of the two subdivisions presented on this figure). So, IGs make an optimized model defined for representing a certain subclass of cellular objects, but the characterization of this class is still incomplete. Note also that simplicial objects exist, which can not be structured into cells (cf. 19).

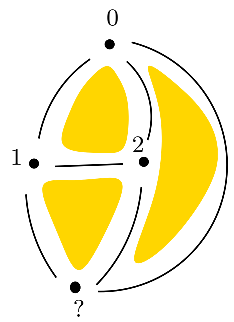



*Figure 19* – A simplicial object, which can not be structured into cells.

**Representation and operations.** Representations of IGs are obviously similar to representations of oriented graphs. In practice, it is often useful for some operations to explicitly represent the inverse edges of the graph, for instance for computing stars of cells.

Although formally defined as simplicial objects (structured into cells), IGs make a cellular model which can be associated with several embedding models. A cell is embedded by associating a geometric object of corresponding dimension with it (point, piece of curve, piece of surface, etc.). Cells of dimensions greater than 1 are often embedded into a support space, and their precise shape is deduced from this support space and the shapes of their boundaries[14]; this corresponds to the trimmed patch notion used in Computer-Aided Design (CAD). For instance, a face is embedded onto a surface, cut off by the boundary of the face; each edge of the face boundary is embedded onto a curve contained in the support surface of the face, and cut off by its extremity points.

Although it is easy to define basic operations for handling graphs (e.g. adding or removing nodes or edges), defining basic operations for handling IGs is more complicated, since they have to control the topological validity of the constructed object. For instance, several sets of *Euler operators* have

[14]This is the current meaning for Boundary Representation.

been defined for constructing any subdivision of any orientable surface without boundary [80][15]. Euler operators are such that the modifications of the topological characteristics of the surface subdivision are correct (according to the classification of surfaces). But the definition of similar operators for higher dimensions remains a problem, since no equivalent classification exists [102]. Several methods have been proposed in order to construct IGs from valid object representations (e.g. extracting objects from images by marching-cubes algorithms); the IG validity comes from the initial object validity. Mainly, the interest of IGs for this type of applications lies in the reduction of the represented information, while keeping the essential of topological information.

### 3.2. Ordered models and cellular quasi-manifolds.

We have seen above that the formal interpretation of IGs is based upon that of ASCs structured into cells. The definition of ordered models is directly deduced from this interpretation: numbered simplices (and not cells) are explicitly represented, providing more information (and avoiding thus problems related to multi-incidence). An other constraint which is taken into account is the fact that cells satisfy some topological properties: for instance for cellular models described in this section, cells are quasi-manifolds (taking into account such constraints was not directly possible for IGs: cf. section 3.1 and figure 16).

We here study three models derived from the combinatorial map notion [43, 62, 22, 100, 99, 16, 63] (see also the notions of Graph-Encoded Manifolds and the Crystallization Theory [56, 47, 77, 48, 76, 5], defined in Italy by Mario Pezzana during the 70's for studying piecewise-linear manifolds by combinatorial methods). These models are respectively generalized maps, maps[16], and chains of maps [74, 45] (cf. section 3.3), which can be used for representing cellular quasi-manifolds, orientable cellular quasi-manifolds without boundaries, and cellular complexes. We can show that each model corresponding to a sub-class of cellular objects can be derived from a model corresponding to

[15] For instance, insert a vertex and an edge incident to the vertex within a face, or split a face by inserting an edge between two vertices of the face boundary, etc.

[16] This notion extends for any dimension the notion of combinatorial map, initially defined for dimension 2.

a larger class: for instance, generalized maps (resp. maps) optimize chains of maps (resp. generalized maps) for the representation of cellular quasi-manifolds (resp. orientable quasi-manifolds without boundaries). At last, we study the links between these models and other ordered models.

**Generalized maps.** Given a cellular quasi-manifold, we can intuitively define the generalized map notion in the following way. The object represented on figure 20(a) is composed by two faces $F_1$ and $F_2$, « glued » along their common edge $c$. This edge can be split into two new edges (cf. figure 20(b)), in order to dissociate $F_1$ and $F_2$. These new edges correspond respectively to edge $c$ « seen » from face $F_1$ and to edge $c$ « seen » from face $F_2$. A « 2-dimensional relation » is added between these new edges, in order to remember that they initially correspond to a single one. This process is applied to all other edges, but no new edge is created since all edges belong to the boundary of the surface: in order to formalize this fact, each edge is linked with itself by the 2-dimensional relation (cf. figure 20(b)).

The boundary of each face $F_1$ or $F_2$ is a 1-dimensional quasi-manifold: the same process can be applied, and each vertex is split into two distinct vertices, linked by a 1-dimensional relation (cf. figure 20(c)). The 2-dimensional relation is now defined upon these new basic elements (cf. figure 20(c)). The boundary of each edge is now defined by two distinct vertices, linked by a 0-dimensional relation meaning that they correspond to a single edge (cf. figure 20(d)).

Split vertices obtained at the end of the process are the basic elements of the model[17], and they are usually called *darts* in the combinatorial map terminology. $i$-dimensional relations link pairs of darts, and this can be formalized in the following way:

A $n$-dimensional *pre-map* is an algebra $C = (B, \alpha_0, ..., \alpha_n)$, where:

- $B$ is a finite set of abstract objects called darts;
- $\forall i, 0 \leq i \leq n, \alpha_i$ is an involution[18] on $B$.

[17] At the end of the process, the resulting vertices correspond to the initial vertices, « seen » from the initial edges, « seen » from the initial faces, i.e. to triangles numbered $\{0, 1, 2\}$.

[18] An involution $\alpha : B \to B$ is a bijection on $B$ which is its own inverse; in other words, $\forall b \in B, b\alpha\alpha = b$, i.e. $b\alpha = b\alpha^{-1}$.

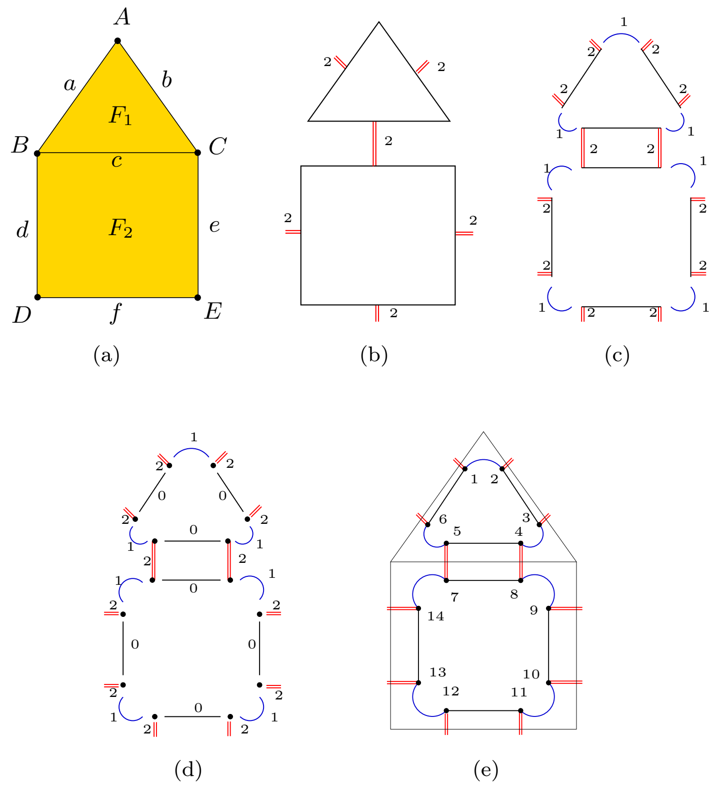



*Figure 20* – Constructing a generalized map by splitting a cellular quasi-manifold.

We can note an other constraint on involutions, which appears during the previous construction process: four darts, namely $4, 5, 7, 8$, have been created around edge $c$ common to faces $F_1$ and $F_2$. 4 and 5 (resp. 7 and 8) define the extremities of the edge « seen » from $F_1$ (resp. from $F_2$). Since faces share edge $c$, the darts have to be coherently linked by involution $\alpha_2$. More generally, we get the following definition of *generalized maps* (or G-maps):

A *generalized map* is defined as a pre-map $C = (B, \alpha_0, ..., \alpha_n)$ satisfying: $\forall i, j, 0 \leq i, j \leq n, i \notin \{j-1, j, j+1\}, \alpha_i\alpha_j$ is an involution[19].

[19]This constraint gets a real meaning when the dimension is greater than or equal to 2. For higher dimensions, if we think of a quasi-manifold as an object constructed by gluing

**Orbit notion.** The *cell* notion is a particular case of the *orbit* notion: let $B$ be a set and $\pi_0, ..., \pi_n$ be a set of permutations defined on $B$. The orbit of element $b$ of $B$ according to this permutation set, denoted $< \pi_0, ..., \pi_n > (b)$, is informally the set of all elements of $B$ which can be reached starting from $b$ by any composition of the permutations and their inverses. The set of orbits for $\{\pi_0, ..., \pi_n\}$ makes a partition of $B$ (cf. figure 21). Let $N = \{0, ..., n\}$, $I = \{i_1, ..., i_p\} \subseteq N$ and $b \in B$; $<>_I (b)$ is the orbit of $b$ relatively to the set of involutions the indices of which belong to $I$.

A pre-map (or a G-map) $C = (B, \alpha_0, ..., \alpha_n)$ is *connected* if and only if it contains a single orbit for the set of all involutions.

Different informations can now be extracted from maps. For instance, for any $i$ between 0 and $n$, the $i$-dimensional cells are formally defined as orbits for $<>_{N-\{i\}}$ (cf. figure 21). The G-maps of $i$-cells are defined by: $\forall i, 0 \leq i \leq n$, the $(n-1)$-G-map of $i$-cells is $C_i = (B, \alpha_0, ..., \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, ..., \alpha_n)$, where $\hat{\alpha}_i$ means that involution $\alpha_i$ is removed. Each connected component of this G-map describes the neighborhood of an $i$-dimensional cell (cf. figure 21(a, b, c)). We can also define the $(n-1)$-G-map of the boundaries, which describes the structure of an $(n-1)$-dimensional quasi-manifold which recovers the boundary of the $n$-dimensional quasi-manifold corresponding to the G-map (if non empty) (cf. figure 21(d)). The pre-map (or G-map) is without boundaries if and only if all involutions are *without fixed points*[20].

G-maps can represent multi-incident cells without ambiguities (G-maps associated with the objects of figure 17 are represented on figure 22).

Other consistency constraints have been proposed in order to represent subclasses of quasi-manifolds (for instance for avoiding multi-incidence [2]). It is thus possible to characterize the corresponding IGs and to define conversion operations between IGs and G-maps. For instance, if the G-map is without multi-incidence, an equivalent IG can be associated in the following way. A node of the graph (i.e. an $i$-cell) is associated with any orbit $<>_{N-\{i\}}$; an edge of the graph links two nodes associated with orbits $<>_{N-\{i\}}$ and $<>_{N-\{i-1\}}$ if and only if they share a common dart.

---

$n$-cells by identifying $(n-1)$-cells, this constraint imposes that the identified $(n-1)$-cells have a same structure.

[20]For all $b \in B$, and all $i, 0 \leq i \leq n$, $b\alpha_i \neq b$.

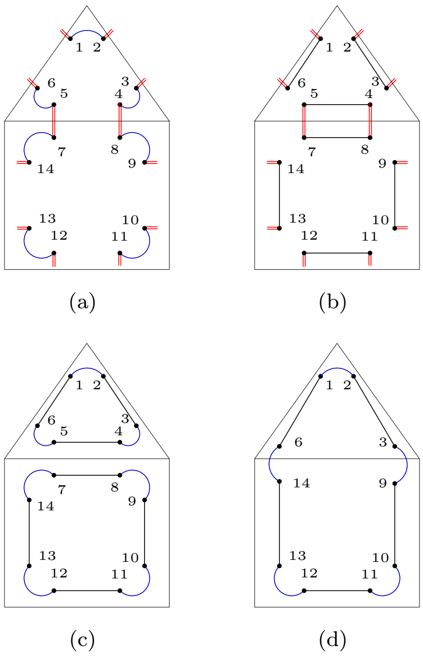



*Figure 21* – The G-maps of cells corresponding to the G-map of figure 20. 21(a): vertices, i.e. orbits $<>_{\{0,1,2\}-\{0\}} = <\alpha_1, \alpha_2>$; 21(b): edges, i.e. orbits $<>_{\{0,1,2\}-\{1\}} = <\alpha_0, \alpha_2>$; 21(c): faces, i.e. orbits $<>_{\{0,1,2\}-\{2\}} = <\alpha_0, \alpha_1>$. Figure 21(d) describes the map of boundaries.

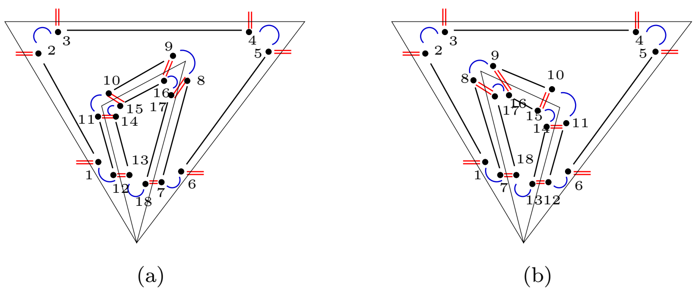



*Figure 22* – The G-maps associated with multi-incident objects of figure 17.

**Topological interpretation of generalized maps.** An intuitive presentation has been made above, but in fact, map notions can be deduced from simplicial objects structured into cells (as many other cellular models known in geometric modeling, IGs for instance). We have seen in section 2.4 that the representation of simplicial quasi-manifolds can be optimized, by representing principal simplices and their adjacency relations. In this case, pre-maps can be deduced from SSSs: darts correspond to principal simplices, and involutions are derived from boundary operators (indices of involutions are indices of boundary operators, constrained by the cell structure). For instance, the G-map of figure 20.e corresponds to the structured simplicial object of figure 18.a: darts correspond to 2-simplices, for instance 1, 5 and 11 correspond to $\sigma_4$, $\sigma_6$ and $\sigma_{11}$; for any $i$, involution $\alpha_i$ associates two darts corresponding to two 2-simplices which share a 1-simplex numbered $\{0, 1, 2\} - \{i\}$.

Conversely, we can associate a simplicial quasi-manifold structured into cells with any pre-map, using the orbit notion. The quasi-manifold associated with $C = (B, \alpha_0, \ldots, \alpha_n)$ is defined in the following way. Let $N = \{0, \ldots, n\}$ and $I = \{p_0, ..., p_i\} \subseteq N$ (for any $p_0, ..., p_i$):

– the set of orbits $<>_{N-I}$ defines the set of simplices numbered by the integers of $I$;

– let $b \in B, p_j \in I$, $\sigma$ and $\tau$ be the simplices associated with orbits $<>_{N-I}$ $(b)$ and $<>_{N-(I-\{p_j\})}(b)$. Then $\sigma d_j = \tau$.

So, we define a SSS which is a quasi-manifold structured into cells: all principal simplices have $n$ for dimension, and they correspond to the darts of $B$ (i.e. to orbits related to an empty set of involutions). An $(n-1)$-simplex is incident to at most two $n$-simplices, since all $\alpha_i$, $(0 \leq i \leq n)$ are involutions. The SSS is structured into cells, since any $n$-simplex $\sigma$ corresponding to dart $b$ is incident to $n+1$ distinct vertices associated to orbits $<>_{N-\{i\}}$ $(b)$, for any $i$ of $N$, and each vertex can be numbered by the corresponding index $i$.

Note that the SSS associated with a pre-map is a quasi-manifold, but its cells can be complexes (cf. figure 16.b). Due to their characteristic constraint, G-maps correspond to *quasi-manifolds such that all cells are quasi-manifolds*, i.e. to (what we called) *cellular* quasi-manifolds.

**Representation of G-maps.** It is easy to associate a non oriented graph with any pre-map or G-map: a node of the graph corresponds to a dart, and an edge numbered $i$ links two darts which are linked by involution $\alpha_i$. Each node of the graph is incident to $n+1$ edges, numbered from 0 to $n$. So, pre-maps and G-maps can be represented by using classical methods for representing non oriented graphs. Traversal algorithms (for computing connected components, cells, etc.) can be easily defined as graph traversals using all edges, or some edges characterized by their associated numbers.

The embedding of G-maps is similar to that of IGs, using support spaces. Note that since cells are defined as orbits, we can choose different methods for associating the embedding information with darts, according to the expected space / time complexity. For instance, the embedding information can be associated with:

– all darts of the orbit, and the information can be directly accessed from any dart of the orbit; this is efficient for extracting information, but not when the map is constructed. For instance, if the structure of the cell is modified (e.g. by splitting a cell), it can be necessary to traverse all darts of the orbit in order to change the associated information;

– one representative dart of the orbit: this is more efficient when constructing the map, but it could be less efficient for retrieving the information, since it can be necessary to traverse all darts of the orbit in order to find the dart which has the information.

**Operations.** Two basic operations can be defined in order to construct any $n$-dimensional pre-map $C = (B, \alpha_0, ..., \alpha_n)$. The first one consists in adding a new « isolated » dart, i.e. which is invariant for all involutions. The second one consists in « sewing », for a given dimension $i$ ($0 \leq i \leq n$), two darts $b$ and $b'$ which are invariant by $\alpha_i$ (i.e. involution $\alpha_i$ is modified in such a way that $b$ and $b'$ are linked by this involution).

These two operations can be used in order to construct any G-map, but it is then necessary to constrain the « sewing » operation. The precondition is that orbits $<>_{N-\{i-1,i,i+1\}}$ of $b$ and $b'$ have same structure, i.e. that the sub-maps:

$$(<>_{N-\{i-1,i,i+1\}} (b), \alpha_0, ..., \hat{\alpha}_{i-1}, \hat{\alpha}_i, \hat{\alpha}_{i+1}, ..., \alpha_n)$$

and

$$(<>_{N-\{i-1,i,i+1\}}(b'), \alpha_0, ..., \hat{\alpha}_{i-1}, \hat{\alpha}_i, \hat{\alpha}_{i+1}, ..., \alpha_n)$$

are isomorphic by $\alpha_i$[21]. The operation consists in « sewing » $b$ and $b'$ as before, and also all darts of the sub-maps defined above.

Many other operations have been defined for handling G-maps (and can be adapted for handling other classes of cellular objects), based upon these two basic operations: split, contraction, chamfering, sweeping and cartesian product, etc. Based upon these topological operations, many geometric operations have been defined, as rounding cells, refinement and boolean operations, etc [18, 27, 75].

### 3.3. Other classes of cellular objects.

**Maps.** Maps are deduced from G-maps in order to represent orientable quasi-manifolds without boundaries. An $n$-dimensional map is defined as an algebra $C = (B, \beta_1, ..., \beta_n)$, where:

- $\beta_1$ is a permutation on $B$;
- $\forall i, 2 \leq i \leq n, \beta_i$ is an involution on $B$;
- $\forall i, j, 1 \leq i, j \leq n, i \notin \{j-1, j, j+1\}, \beta_i\beta_j$ is an involution on $B$.

This definition is based upon the following property of G-maps (cf. figure 23). Let $C = (B, \alpha_0, ..., \alpha_n)$ be a connected G-map without boundaries[22]. $CO = (B, \alpha_0\alpha_1, \alpha_0\alpha_2, ..., \alpha_0\alpha_n)$ is the $n$-map of the orientations of $C$, and contains:

- one connected component: $C$ is not orientable;
- two connected components; $C$ is orientable, and each connected component of $CO$ corresponds to an orientation of $C$. Each orientation is

[21] This precondition for G-maps is due to their consistency constraint, i.e. $\alpha_i\alpha_j$ is an involution for all $j$ distinct from $i-1$ and $i+1$. It corresponds to the fact that the sewing operation in G-maps corresponds to the identification of some cells in the associated quasi-manifolds, and these cells must have the same structure.

[22] Remind that a G-map is without boundaries if all involutions are without fixed points.

the inverse of the other[23], so it is useless to explicitly represent both orientations. This completely defines the optimization and the possible conversion between G-maps and maps.

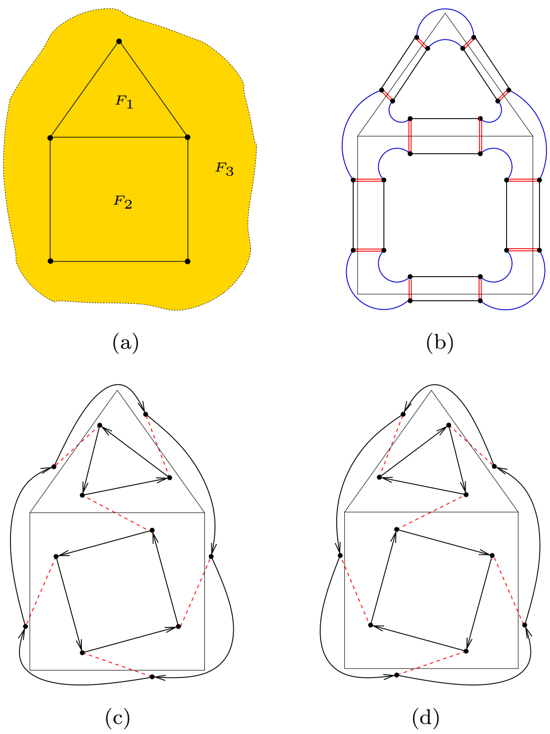



*Figure 23* – (a) A subdivision of the plane. (b) The corresponding G-map, which is orientable without boundaries (c) A connected component of the map of the orientations and (d) the other connected component. These two connected components describe the two possible orientations of the initial subdivision.

We can easily deduce from G-maps numerous notions and operations for handling n-maps (for instance the $(n-1)$-maps of $i$-cells). Note that the

[23]Let $CO_1 = (B_1, \beta_1, \beta_2, ..., \beta_n)$ and $CO_2 = (B_2, \gamma_1, \gamma_2, ..., \gamma_n)$ be the two connected components, then $CO_1$ is isomorphic to the *inverse map* of $CO_2$, i.e. $CO_2^{-1} = (B_2, \gamma_1^{-1}, \gamma_2, ..., \gamma_n)$.

definitions of some operations and constructions are more difficult to conceive for n-maps, since they correspond to quasi-manifolds *without* boundaries.

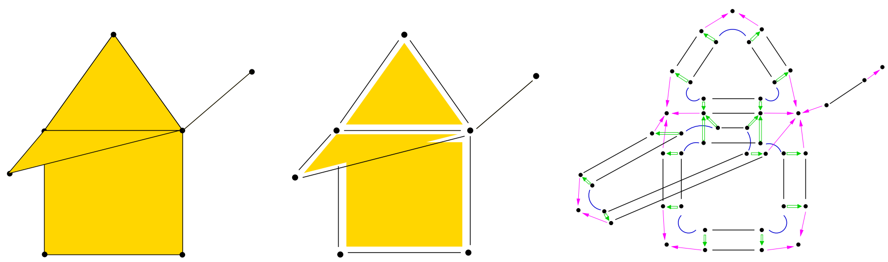
**Chains of maps.** Maps are an optimized model deduced from G-maps for the representation of a sub-class of cellular quasi-manifolds. Other topological models have been defined for representing other sub-classes, or *larger* classes. For instance, chains of maps [44, 45] represent cellular complexes, in which cells are quasi-manifolds.

A chain of maps is an algebra $((C^i)_{i=0,\dots,n}, (\sigma^i)_{i=1,\dots,n})$, such that (cf. figure 24):

– $\forall i, 0 \leq i \leq n, C^i = (B^i, \alpha_0^i, ..., \alpha_{i-1}^i, \alpha_i^i = \omega)$ is an $i$-dimensional G-map such that $\omega$ is undefined on $B^i$;

– $\forall i, 1 \leq i \leq n, \sigma^i : B^i \to B^{i-1}$ satisfies, for any dart $b$ of $B^i$:

  - $\forall k, 0 \leq k \leq i-2, b\alpha_k^i\sigma^i \in \{b\sigma^i, b\sigma^i\alpha_k^{i-1}\}$;
  - $b\alpha_{i-1}^i\sigma^i\sigma^{i-1} = b\sigma^i\sigma^{i-1}$.



(a) (b) (c)

*Figure 24* – A chain of maps. ($a$) a cellular complex; ($b$) the cells of the complex; ($c$) the chain of maps: the interior of each cell is described by a G-map.

For any $i$, each connected component of each G-map $C^i$ describes the *interior* of an $i$-dimensional cell: this is formally expressed by the fact that the involution of index $i$ is undefined. Each $i$-dimensional cell is linked to the cells of its boundary by $\sigma^i$; the consistency constraints corresponds to the facts that all cells are quasi-manifolds, and that the structure of each cell corresponds to the structure of its boundary (they are deduced from the consistency constraints of SSSs for the case of cellular complexes).

**Other ordered models.** Many other ordered models have been defined for different uses in different fields: computational geometry, geometric modeling, image structuration: equivalences have been proved, for instance between [73, 14]:

– G-maps and structures *quad-edge*, *facet-edge*, *cell-tuple*;

– maps and structures *winged-edge*, *face-edge*, *vertex-edge*, *radial-edge*. For instance in the 2-dimensional case, a dart of a map can be interpreted as an oriented edge, which is the basic object of structures *winged-edge*, *face-edge* and *vertex-edge*.

Main differences between these structures are the more or less important redundancy of the explicit information: either the minimal information is represented (e.g. darts and involutions) or cells may be also represented, in order to associate non topological information with them (embedding information, or other information related to the application, etc.). In this last case, it is necessary to satisfy the consistency constraints, i.e. all darts of a cell are associated with one explicit cell.

**Hierarchical topological models.** Many works deal also with the definition of hierarchical models, in order to represent subdivided objects at different levels of detail [84, 64, 65, 34, 96]. Each level of detail is represented by a basic topological model, and relations between two levels are defined by maps between the basic models corresponding to the levels. Several types of representations have been proposed, according to the type of applications, for instance:

– « bottom-up » representations: for instance for image structuration, all information can be expressed on the more detailed level; the other levels can be deduced by successive applications of simplification operations (in this case, the type and parameters of operations are explicitly represented);

– « top-down » representations: for other applications (e.g. incremental construction of objects, for instance in architectural applications), one can choose to represent the simpler level, and to incrementally precise the different « details ».

## References

[1] Agoston, M.K.: *Algebraic Topology, a first course*, Pure and applied mathematics, Marcel Dekker Ed. (1976).

[2] Alayrangues, S., Daragon, X., Lachaud, J.-O., and Lienhardt, P.: Equivalence between regular $n$-$G$-maps and $n$-surfaces, *10th Int. Workshop on Combinatorial Image Analysis*, Auckland, Nouvelle-Zélande, Décembre (2004).

[3] Ansaldi, S., de Floriani, L., and Falcidieno, B.: Geometric Modeling of Solid Objects by Using a Face Adjacency Graph Representation, *Computer Graphics* (3) **19** (1985), 131–139.

[4] Arquès, D. and Koch, P.: Modélisation de solides par les pavages, *Proc. Pixim '89*, Paris, France (1989), 47–61.

[5] Bandieri, P., Casali, M.R., and Gagliardi, C.: Representing manifolds by crystallization theory: foundations, improvements and related results, *Atti. Sem. Mat. Fis. Univ. Modena* **49** (2001), 283–337.

[6] Baumgart, B.: A polyhedron representation for computer vision, *Proc. AFIPS Nat. Conf.* **44** (1975), 589–596.

[7] Bertrand, G.: New notions for discrete topology, in Bertrand, G., Couprie, M., and Perroton, L., editors, Proc. DGCI '99, *Lecture Notes in Computer Science* **1568**, Springer, Marne-la-Vallée, France (1999), 218–228.

[8] Bertrand, Y., Damiand, G., and Fiorio, C.: Topological encoding of $3d$ segmented images, Proceedings of 9th Discrete Geometry for Computer Imagery, *LNCS* **1953**, Uppsala, Sweden (2000), 311–324.

[9] Bertrand, Y. and Dufourd, J.-F.: Algebraic Specification of a 3$D$-Modeler Based on Hypermaps, *CVGIP: Graphical Models and Image Processing* (1) **56** (1994), 29–60.

[10] Brandel, S., Schneider, S., Perrin, M., Guiard, N., Rainaud, J.-F., Lienhardt, P., and Bertrand, Y.: Automatic building of structured geological models, *Journal of Computing and Information Science in Ingeneering* (2) **5** (2005).

[11] Braquelaire, A., Damiand, G., Domenger, J.-P., and Vidil, F.: Comparison and Convergence of Two Topological Models for 3$D$ Image Segmentation, Proceedings of 4th IAPR-TC15 Workshop on Graph-Based Representations in Pattern Recognition, *LNCS* **2726**, York, England (2003), 59–70.

[12] Braquelaire, J.-P. and Guitton, P.: A Model for Image Structuration, *Proc. Computer Graphics International '88*, Genève, Switzerland (1988).

[13] Brisson, E.: Representing geometric structures in $d$ dimensions: topology and order, *Proc. 5th ACM Symp. Computational Geometry*, Saarbrücken, Germany (1989), 218–227.

[14] Brisson, E.: Representing geometric structures in $d$ dimensions: topology and order, *Discrete & Computational Geometry* (1) **9** (1993), 387–426.

[15] Brun, L. and Kropatsch, W.: Combinatorial pyramids, *IEEE Int. Conf. on Image Processing* **2**, Barcelona, Spain (2003), 33–37.

[16] Bryant, R. and Singerman, D.: Foundations of the Theory of Maps on Surfaces with Boundaries, *Quart. Journal of Math. Oxford* (2) **36** (1985), 17–41.

[17] Cavalcanti, P., Carvalho, P., and Martha, L.: Non-manifold modeling: an approach based on spatial subdivisions, *Computer-Aided Design* (3) **29** (1997), 299–320.

[18] Cazier, D. and Dufourd, J.-F.: A rewrite system to build planar subdivisions, *Proceedings of the 7th Canadian Conference on Computational Geometry*, Québec, Canada (1995), 235–240.

[19] Chen, X. and Lienhardt, P.: Modeling and Programming Evolutions of Surfaces, *Computer Graphics Forum* (5) **2** (1992), 323–341.

[20] Christensen, H.I. and Nagel, H.-H., editors: Cognitive Vison Systems: Sampling the Spectrum of Approaches, *Lecture Notes in Computer Sciences* **3948**, Springer (2006).

[21] Cigogni, P., de Floriani, L., Montani, C., Puppo, E., and Scopigno, R.: Multiresolution modeling and visualization of volume data based on simplicial complexes, *Symposium on Volume Visualization* (1994), 19–26.

[22] Cori, R.: Un code pour les graphes planaires et ses applications, *Astérisque* **27** (1975).

[23] Crocker, G. and Reinke, W.: An Editable Non-Manifold Boundary Representation, *Computer Graphics and Applications* (2) **11** (1991).

[24] Croom, F.H.: *Basic Concepts of Algebraic Topology*, Springer-Verlag, Heidelberg (1978).

[25] Dahmen, W. and Micchelli, C.A.: On the linear independence of multivariate $B$-splines I. Triangulation of simploids, *SIAM J. Numer. Anal.* **19** (1982), 993–1012.

[26] Damiand, G., Bertrand, Y., and Fiorio, C.: Topological model for two-dimensional image representation: definition and optimal extraction algorithm, *Computer Vision and Image Understanding* (2) **93** (2004), 111–154.

[27] Damiand, G. and Lienhardt, P.: Removal and Contraction for $N$-Dimensional Generalized Maps, *11th Discrete Geometry for Computer Imagery* (2003), 408–419.

[28] Daragon, X., Couprie, M., and Bertrand, G.: Discrete frontiers, in Nyström, I., Sanniti di Baja, G., and Svensson, S., editors, Proc. DGCI'03, *Lecture Notes in Computer Science* **2886**, Springer, Naples, Italie (2003), 236–245.

[29] De Floriani, L. and Falcidieno, B.: A hierarchical boundary model for solid object representation, *A.CM. Transactions on Graphics* (1) **7** (1988), 42–60.

[30] De Floriani, L., Magillo, P., Puppo, E., and Sobrero, D.: A multiresolution topological representation for non-manifold meshes, *Computer-Aided Design* (2) **36** (2004), 141–159.

[31] De Floriani, L., Mesmoudi, M., Morando, F., and Puppo, E.: Decomposing non-manifold objects in arbitrary dimensions, *Graphical Models* (1/3) **65** (2003), 2–22.

[32] De Floriani, L., Morando, F., and Puppo, E.: A representation for abstract simplicial complexes, in Nyström, I., Sanniti di Baja, G., and Svensson, S., editors, Proc. of DGCI '03, *Lecture Notes in Computer Science* **2886**, Springer, Naples, Italy (2003), 454–464.

[33] De Floriani, L., Morando, F., and Puppo, E.: Representation of non-manifold objects through decomposition into nearly manifold parts, in Shapiro, V. and Elber, G., editors, *8th A.C.M. Symposium on Solid Modeling and Applications*, ACM Press, Seattle, U.S.A. (2003), 304–309.

[34] De Floriani, L., Puppo, E., and Magillo, P.: A formal approach to multiresolution hypersurface modelling, in Strasser, W., Klein, R., and Rau, R., editors, *Geometric Modeling: Theory and Practice* (1997).

[35] Delfinado, C.J.A. and Edelsbrunner, H.: An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere, *Computer-Aided Geometric Design* (7) **12**, Elsevier Science (1995), 771–784.

[36] DeRose, T.: Composing Bézier simplexes, *ACM Transactions on Graphics* (3) **7** (1988), 198–221.

[37] Dey, T., Edelsbrunner, H., and Guha, S.: Computational Topology, in Chazelle, B., Goodman, J., and Pollack, R., editors, Advances in Discrete and Computational Geometry, *Contemporary mathematics* **223**, American Mathematical Society (1999), 109–143.

[38] Dobkin, D. and Laszlo, M.: Primitives for the Manipulation of Three-Dimensional Subdivisions, *3rd Symposium on Computational Geometry*, Waterloo, Canada (1987), 86–99.

[39] Dobkin, D. and Laszlo, M.: Primitives for the Manipulation of Three-Dimensional Subdivisions, *Algorithmica* (4) **5** (1989), 3–32.

[40] Dufourd, J.-F.: Formal specification of topological subdivisions using hypermaps, *Computer Aided Design* (2) **23** (1991), 99–116.

[41] Dufourd, J.-F.: Algebras and formal specifications in geometric modelling, *The Visual Computer* **13** (1997), 131–154.

[42] Edelsbrunner, H.: *Algorithms in Computational Geometry*, Springer, New-York (1987).

[43] Edmonds, J.: A Combinatorial Representation for Polyhedral Surfaces, *Notices Amer. Math. Soc.* **7** (1960).

[44] Elter, H. and Lienhardt, P.: Different combinatorial models based on the map concept for the representation of different types of cellular complexes, in Falcidieno, B. and Kunii, T.L., editos, *Proceedings of IFIP TC 5/WG II Working Conference on Geometric Modeling in Computer Graphics*, Springer, Genova, Italy (1993).

[45] Elter, H. and Lienhardt, P.: Cellular complexes as structured semi-simplicial sets, *International Journal of Shape Modeling* (2) **1** (1994), 191–217.

[46] Farin, G.: *Curves and Surfaces for Computer Aided Geometric Design: A Practical guide*, Academic Press, San Diego, CA 5th edition (2001).

[47] Ferri, M. and Gagliardi, C.: Crystallisation moves, *Pacific Journal of Mathematics* (1) **100** (1982), 85–103.

[48] Ferri, M., Gagliardi, C., and Grasselli, L.: A graph-theoretical representation of $PL$-manifolds: a survey on crystallizations, *Aequationes Mathematicae* **31** (1986), 121–141.

[49] Ferruci, V. and Paoluzzi, A.: Extrusion and Boundary Evaluation for Multidimensional Polyhedra, *Computer-Aided Design* (1) **23** (1991), 40–50.

[50] Fiorio, C.: A topologically consistent representation for image analysis: the frontier topological graph, *Proc. of DGCI '96*, Lyon, France (1996), 151–162.

[51] Fradin, D., Meneveaux, D., and Lienhardt, P.: A Hierarchical Topology-Based Model for Handling Complex Indoor Scenes, *Computer Graphics Forum* (2006), to appear.

[52] Fritsch, R. and Piccinini, R.: *Cellular Structures in Topology*, Cambridge University Press (1990).

[53] Fuchs, L., Bechmann, D., Bertrand, Y., and Dufourd, J.-F.: Formal specification for free-form curves and surfaces, *Proceedings of Spring Conference on Computer Graphics*, Bratislava, Slovakia (1996).

[54] Fuchs, L. and Lienhardt, P.: Topological structures for $d$-dimensional free-form objects, *Proceedings of Computer-Aided Geometric Design '97*, Lillehammer, Norway (1997).

[55] Gagliardi, C.: How to deduce the fundamental group of a closed $n$-manifold from a contracted triangulation, *Journal of Combinatorics, Information and System Sciences* (3) **4** (1979), 237–252.

[56] Gagliardi, C.: Regular imbeddings of edge-coloured graphs, *Geometriae Dedicata* **11** (1981), 397–414.

[57] González-Díaz, R. and Real, P.: Toward digital cohomology, in Sanniti di Baja, G., Svensson, S., and Nyström, I., editors, *Proc. DGCI'2003, LNCS* **2886**, Springer (2003), 92–101.

[58] Griffiths, H.-B.: *Surfaces*, Cambridge University Press, Cambridge, 2nd edition (1981).

[59] Guibas, L. and Stolfi, G.: Primitives for the Manipulation of General Subdivisions and the Computation of Voronoï Diagrams, *Transactions on Graphics* (2) **4** (1985), 74–123.

[60] Gursoz, E.L.C.Y. and Prinz, F.: Vertex-based Representation of Non-Manifolds Boundaries, in Wozny, M.T.J. and Preiss, K., editors, *Geometric Modeling for Product Engineering*, North-Holland (1990), 107–130.

[61] Hansen, O. and Christiansen, N.: A model for $n$-dimensional boundary topology, in Rossignac, J. and Turner, J., editors, 2*nd A.C.M. Symposium on Solid Modeling Foundations and CAD/CAM Applications*, Montréal, Canada (1993), 65–73.

[62] JACQUE, A.: Constellations et Graphes Topologiques, *Colloque Math. Soc. Janos Bolyai* (1970), 657–672.

[63] JAMES, L.: *Maps and Hypermaps: Operations and Symmetry*, PhD thesis, University of Southampton, Southampton, U.K. (1985).

[64] JOLION, J. and MONTANVERT, A.: The adaptive pyramid: a framework for $2d$ image analysis, *Computer Vision, Graphics and Image Processing* (3) **55** (1992), 339–348.

[65] KROPATSCH, W.: Building irregular pyramids by dual-graph contraction, *Vision, image and signal processing* (6) **142** (1995), 366–374.

[66] KUNII, T. and TAKAHASHI, S.: Area guide map modeling by manifolds and $CW$-complexes, in Falcidieno, B. and Kunii, T., editors, *Proc. IFIP TC 5/WG 2 Working Conf. on Geometric Modeling in Computer Graphics*, Genova, Italy (1993), 5–20.

[67] LANG, V. and LIENHARDT, P.: Geometric modeling with simplicial sets, *Proceedings of Pacific Graphics '95*, Seoul, Korea (1995).

[68] LANG, V. and LIENHARDT, P.: Simplicial sets and triangular patches, *Proceedings of CGI '96*, Pohang, Korea (1996).

[69] LEE, S. and LEE, K.: Partial entity structure: a fast and compact non-manifold boundary representation based on partial topological entities, *6th A.C.M. Symposium on Solid Modeling and Applications*, Ann Arbor, U.S.A. (2001).

[70] LIENHARDT, P.: Free-Form Surfaces Modeling by Evolution Simulation, *Proc. Eurographics '88*, Nice, France (1988), 327–341.

[71] LIENHARDT, P.: Subdivisions of $n$-dimensional spaces and $n$-dimensional generalized maps, *ACM Symposium on Computational Geometry* (1989), 228–236.

[72] LIENHARDT, P.: Subdivisions of Surfaces and Generalized Maps, *Proc. Eurographics '89*, Hamburg, Germany (1989), 439–452.

[73] LIENHARDT, P.: Topological models for boundary representation: a comparison with $n$-dimensional generalized maps, *Computer Aided Design* (1) **23** (1991), 59–82.

[74] LIENHARDT, P.: $N$-Dimensional Generalized Combinatorial Maps and Cellular Quasi-Manifolds, *International Journal of Computational Geometry and Applications* (3) **4** (1994), 275–324.

[75] LIENHARDT, P., SKAPIN, X., and BERGEY, A.: Cartesian Product of Simplicial and Cellular Structures, *Int. Journal of Computational Geometry and Applications* (3) **14** (2004), 115–159.

[76] LINS, S.: Gems, computers and attractors for 3-manifolds, *Series on Knots and Everything* **5**, World Scientific (1995).

[77] LINS, S. and MANDEL, A.: Graph-encoded 3-manifolds, *Discrete Mathematics* **57** (1985), 261–284.

[78] LUO, Y. and LUKACS, G.: A Boundary Representation for Form-Features and Non-Manifold Solid Objects, 1*st ACM/Siggraph Symposium on Solid Modeling Foundations and CAD/CAM Applications*, Austin, Texas, U.S.A. (1990).

[79] MÄNTYLÄ, M.: Computational Topology: a Study of Topological Manipulations and Interrogations in Computer Graphics and Geometric Modeling, *Acta Polytechnica Scandinavia* **37** (1983).

[80] MÄNTYLÄ, M.: *An Introduction to Solid Modeling*, Computer Science Press (1988).

[81] MÄNTYLÄ, M. and SULONEN, R.: GWB: A Solid Modeler with Euler Operators, *Computer Graphics and Applications* (7) **2** (1982), 17–31.

[82] MAY, J.P.: *Simplicial objects in algebraic topology*, Van Nostrand, Princeton (1967).

[83] MEINE, H. and KÖTHE, U.: The geomap: a unified representation for topology and geometry, in Brun, L. and Vento, M., editors, Proc. IAPR Graph-based representations in pattern recognition, *Lecture Notes in Computer Science* **3434**, Poitiers, France (2005).

[84] MONTANVERT, A., MEER, P., and ROSENFELD, A.: Hierarchical Image Analysis Using Irregular Tessellations, *PAMI* (4) **13** (1991), 307–316.

[85] MOORE, D.: *Graphic Gems III*, chapter V.10 Understanding simploids, Academic Press (1992), 250–255.

[86] Munkres, J.R.: *Topology*, Prentice-Hall, 2nd edition (2003).

[87] Murabata, S. and Higashi, M.: Non-manifold geometric modeling for set operations and surface operations, *Proc. IFIP/RPI Geometric Modeling Conference*, Rensselaerville, N.Y. (1990).

[88] Niethammer, M., Stein, A., Kalies, W., Pilarczyk, P., Mischaikow, K., and Tannenbaum, A.: Analysis of blood vessel topology by cubical homology, *IEEE Proceedings of the International Conference on Image Processing* **2** (2002), 969–972.

[89] Paoluzzi, A., Bernardini, F., Cattani, C., and Ferrucci, V.: Dimension-independent modeling with simplicial complexes, *A.C.M. Transactions on Graphics* (1) **12** (1993), 56–102.

[90] Pascucci, V., Ferrucci, V., and Paoluzzi, A.: Dimension-independent convex-cell based hierarchical polyhedral complex: representation scheme and implementation issues, *3rd Symposium on Solid Modeling and Applications* (1995), 163–174.

[91] Peltier, S., Alayrangues, S., Fuchs, L., and Lachaud, J.-O.: Computation of homology groups and generators, *Computers and graphics* (2006), to appear.

[92] Peltier, S., Fuchs, L., and Lienhardt, P.: Homology of simploidal sets, Discrete Geometry and Computer Imagery (DGCI 2006 Szeged, Hungary,) *Lecture Notes in Computer Science* **4245**, Springer, Berlin (2006), 235–246.

[93] Prat, S., Gioia, P., Bertrand, Y., and Meneveaux, D.: Connectivity compression in arbitrary dimension, *The Visual Computer* (8) **21** (2005), 876–885.

[94] Rossignac, J. and O'Connor, M.: SGC: A Dimension-Independant Model for Pointsets with Internal Structures and Incomplete boundaries, *Geometric modeling for Product Engineering*, North-Holland (1989), 145–180.

[95] Serre, J.P.: Homologie singulière des espaces fibrés, *Ann. of Math.* **54** (1951), 425–505.

[96] Simon, C., Damiand, G., and Lienhardt, P.: $n$D generalized map pyramids: definition, representations and basic operations, *Pattern Recognition* (4) **39** (2006), 527–538.

[97] Sobhanpanah, C.: Extension of a Boundary Representation Technique for the Description of $N$-Dimensional Polytopes, *Computer and Graphics* (1) **13** (1989), 17–23.

[98] Spehner, J.-C.: Merging in Maps and Pavings, *Theoretical Computer Science* **86** (1991), 205–232.

[99] Tutte, W.: Graph Theory, *Encyclopaedia of Mathematics and its Applications*, Addison-Wesley, Menlo Park (1984).

[100] Vince, A.: Combinatorial Maps, *Journal of Combinatorial Theory Series B* **34** (1983), 1–21.

[101] Weiler, K.: Edge-based data structures for solid modelling in curved-surface environments, *Computer Graphics and Applications* (1) **5** (1985), 21–40.

[102] Weiler, K.: Boundary Graph Operators for Non-Manifold Geometric Modeling Topology Representations, *Proc. IFIP WG 5.2 Working Conference*, Rensselaerville, U.S.A. (1986).

[103] Weiler, K.: The Radial-Edge Data Structure: A Topological Representation for Non-Manifold Geometry Boundary Modeling, *Proc. IFIP WG 5.2 Working Conference*, Rensselaerville, U.S.A. (1986).