Topological models for boundary representation: a comparison with *n*-dimensional generalized maps

Pascal Lienhardt

In boundary representation, a geometric object is represented by the union of a 'topological' model, which describes the topology of the modelled object, and an 'embedding' model, which describes the embedding of the object, for instance in three-dimensional Euclidean space. In recent years, numerous topological models have been developed for boundary representation, and there have been important developments with respect to dimension and orientability. In the main, two types of topological models can be distinguished. 'Incidence graphs' are graphs or hypergraphs, where the nodes generally represent the cells of the modelled subdivision (vertex, edge, face, etc.), and the edges represent the adjacency and incidence relations between these cells. 'Ordered' models use a single type of basic element (more or less explicitly defined), on which 'element functions' act; the cells of the modelled subdivision are implicitly defined in this type of model. In this paper some of the most representative ordered topological models are compared using the concepts of the n*-dimensional generalized map and the* n*-dimensional map. The main result is that ordered topological models are (roughly speaking) equivalent with respect to the class of objects which can be modelled (i.e. with respect to dimension and orientability).*

computational geometry, computational topology, geometric modelling, boundary representation

Classically, in boundary representation, a 'solid' is defined by a *subdivision of a surface* (informally, a *partition* of this surface into vertices, edges and faces, that is into cells of dimension 0, 1 and 2), embedded in 3D Euclidean space[1]. This surface divides the space into two distinct areas: 'inside' and 'outside' the modelled solid. More precisely, Baer *et al*[2] define the boundary of a solid as a subdivision of an orientable surface without boundaries, bounded, connected and with no self-intersection. Boundary representation thus implies the representation of topologic information – for instance, adjacency and incidence relations between the different cells of the modelled subdivision. Moreover, this topological information is explicitly defined in a boundary representation model (cf. the distinction made by Weiler[3] and Hillyard[4] between 'evaluated' and 'unevaluated' representations of geometric objects).

More precisely, a geometric object is defined in boundary representation by:

- a 'topological' model, which describes the topology of the modelled object. Examples of topological models are the winged-edge data structure[5], vertex-edge and face-edge data structures[6], the quad-edge data structure[7], pavings[8] and pavements[9]; the facet-edge data structure[10]; *n*-dimensional maps[11], the cell-tuple structure[12]; and *n*-dimensional generalized maps[11].
- an 'embedding' model, which defines the embedding of the object. For instance, for modelling solids, a subdivision of a 2D space is embedded in 3D space. Generally, a 2D manifold homeomorphic to a disc is associated with each topological face; two topological faces which are incident to the same edge are embedded in such a way that they intersect along a curve homeomorphic to a segment. Each topological face, for example, is embedded as a planar face[5], a cylindrical face, or a face defined by a parametric surface[13,14].

Among the operators used for handling these models, we distinguish:

- topological operators, which exclusively act on the topological model – for instance, Euler operators[15–17];

Département d'Informatique, Université Louis Pasteur, 7 rue René Descartes, 67084 Strasbourg Cedex, France
Paper received: 4 May 1990. Revised: 22 October 1990

 0010–4485/91/010059–24

- 'embedding' operators, which exclusively act on the embedding model – for instance, some cases of tweaking, bending, and twisting[4,14,18];
- 'mixed' operators, which act on the topological model and on the embedding model – for instance, chamfering, Boolean operations[4,13,14,18–20].

The boundary representation approach has been used for modelling subdivisions of the Euclidean plane; subdivisions of orientable or non-orientable surfaces, with or without boundaries; subdivisions of 3D space; and, more generally, subdivisions of orientable or non-orientable *n*-dimensional topological spaces, with or without boundaries, thus extending the domain of boundary representation. As a consequence of the fact that boundary representation is not only used for modelling solids, topological models and embedding models will be considered to be as independent as possible, i.e. with no embedding constraints.

For instance, for modelling subdivisions of surfaces, two faces which are not adjacent may be embedded in such a way that they intersect (for instance, a Klein bottle embedded in Euclidean three-space). Similarly, two adjacent faces – for instance, incident to the same edge – may be embedded discontinuously, i.e. in such a way that they do not intersect (see, for example, the two faces of Figure 3, embedded in the plane in a discontinuous manner, each edge incident to *two* distinct faces is embedded as *two* segments).

Thus, one of the main principles on which boundary representation is based is the distinction between topology and embedding. This distinction allows the differentiation of problems which appear in geometric modelling, and more generally in computational geometry, and sometimes enables a hierarchization of these problems. For instance, topology may be invariant under geometric transformations[3], in particular in animation of articulated objects[21]. Computing Voronoi diagrams is simplified when topological and embedding aspects are clearly distinguished[7,10].

It will be observed that embedding does not mean geometry in this instance. If topology can be deduced from geometry[1,22], topology and embedding are more independent, i.e., as observed above, topological models and embedding models may be as independent as possible. For instance, in dimension 2, assume that an embedding model consists in associating a polygon with each face of the topological model. If a discontinuous embedding is allowed, it is impossible to deduce the topological information from the embedding model (for example, it is clear that the relations of adjacency between faces cannot be restored). In addition, it may be quite difficult to deduce topological information from geometric information[3]. For instance, how can topological invariants (number of boundaries, Euler characteristic, orientability factor and genus) be computed, given a parametric surface, or a surface defined by implicit equations?

Topological information such as adjacency and incidence relations between cells, and topological invariants (for instance, the genus of a surface), are important for a complete description of a geometric object. Such information provides considerable control over the modelled object. Thus, the definition, manipulation and control of the topology of geometric objects have been extensively studied in boundary representation. Numerous topological models and many topological operators (in particular Euler operators) have been defined in recent years.

Conversely, the control of embedding has not yet been solved satisfactorily. For instance, the author considers that no embedding constraints exist, but it is clear that, in solid modelling for instance, embedding constraints must be satisfied (for example, self-intersections are not allowed). If these constraints are not satisfied, the embedding of the modelled object is *incoherent* with regard to its topological characteristics. Algorithms have been studied, for detecting self-intersections, or for computing the results of Boolean operations applied to polyhedra, defined by planar faces, or free-form surfaces[23–26]. Some of them are used to detect or modify incoherent embeddings. New approaches concerning the control of embedding have been researched[27].

Finally, mention must be made of two aspects of the distinction between topology and embedding which are important for geometric modelling. First, this distinction raises certain problems when creating and manipulating geometric objects. For instance, not only the topology of an object must be defined but also its embedding[1,22]. Second, this distinction makes it possible to associate different embedding models with some topological model, in the same geometric modeller[3,18]. For a more general study, see Miller[28].

Many topological models have recently been defined, extending the field of boundary representation, sometimes considered to be too restricted[22], for instance:

- concerning subdivisions of surfaces (2D spaces):
  - Baumgart[5] and Weiler[6] have studied the winged-edge data structure, and extensions of this data structure, while Ansaldi *et al*[17] define the face-adjacency graph representation, so as to model the topology of subdivision of closed oriented surfaces (and thus for modelling solids[1]);
  - to model the topology of subdivisions of orientable surfaces, Guibas and Stolfi[7] define an edge algebra, and deduce from it the quad-edge data structure, while Tutte[29] and Bryant and Singerman[30] study the concept of the combinatorial map (see also Chiyokura[31] and the data structure presented in Mäntylä[32]);
- concerning subdivisions of 3D spaces:
  - to model the topology of subdivisions of closed oriented 3D spaces, Weiler[33] defines the radial-edge data structure, while Spehner[8], Arquès and Koch[9] and Lienhardt[34] define V-maps and pavings, approaches which are extensions of the concept of the combinatorial map;
  - to model the topology of subdivisions of orientable or non-orientable 3D spaces, Dobkin and Laszlo[10] define the facet-edge data structure (which is an extension of the edge algebra of Guibas and Stolfi[7]).

In particular, for CAD, these representations make it possible to model inhomogeneous volumes:

- concerning subdivisions of nD spaces:
  - to model the topology of subdivisions of oriented nD spaces without boundaries, Lienhardt[11] defines nD maps, or n-maps (an extension of the concept of the combinatorial map);
  - to model the topology of subdivisions of nD topological spaces, orientable or non-orientable, with or without boundaries Brisson[12] defines the cell-tuple structure, while Lienhardt[11] defines nD generalized maps, or n-G-maps (again, an extension of the concept of the map); Brisson[35], Edelsbrunner[36] and Rossignac and O'Connor[37] define, respectively, incidence posets, incidence graphs and selective geometric complexes, while Sobhanpanah[38] defines a polytopal mesh representation in order to represent nD polytopes.

Much attention has been given to problems concerning nD objects (see for instance, Bieri and Nef[25] and Putman and Subrahmanyan[26]). Some problems involve dealing with subdivisions of nD topological spaces – for instance, computing Voronoi diagrams in nD Euclidean space.

This paper is mainly concerned with the evolution of topological models used in boundary representation. Two main types of model are distinguished:

- 'Incidence graphs', are graphs or hypergraphs, where generally the nodes correspond to the cells of the modelled subdivision, and the edges correspond to the adjacency and incidence relations between these cells.
- 'Ordered' topological models (the concept of ordering is presented and discussed by Weiler[6] and Brisson[12]). These models use a single type of basic elements (more or less explicitly defined), on which 'element functions' act. The different cells, boundaries, and connected components of the modelled subdivision are implicitly defined in this type of model.

Some of the most representative topological models used in boundary representation are studied in a later section, and ordered topological models are (albeit incompletely) compared. For simplicity, a single formalism is used as a reference, i.e. the notions of the nD generalized map and the nD map[11,39]. In the next section, these ideas are recalled in detail. The main result of this comparison is that ordered topological models are based on the same ideas, and that they are (roughly speaking) equivalent to each other, with respect to the class of objects that can be modelled. Notice that data structures deduced from ordered topological models are not compared. Different possibilities are merely related for the implementation of such models. It is hoped that this comparison will simplify the comprehension of topological models. Another aim of this paper is to point out some of the basic problems of ordered topological models (and, more generally, some basic problems which appear in geometric modelling).

In the fourth section, some significant developments are described concerning embedding models and operators used in boundary representation (topological, embedding and mixed operators). For instance, for modelling subdivisions of surfaces, embedding models use planar faces[5], cylindrical faces and parametric surfaces[13,14], quadrics[40], and free-form surfaces[18]. Concerning topological operators, Euler operators are introduced by Baumgart[15], and studied by others[14,16,17,19]. New operators have been defined for handling the most recent ordered topological models, among them the splice[7], splice-edge and splice-facet[10], merging[8] and sewing[11]. These new operators can be distinguished from earlier ones, in particular by the fact that a small number of operators (two or three, inverse operators included) is enough to handle the related topological model (this answers criticisms of some workers concerning the variety of basic operators which are necessary in boundary representation[1,22,41]). Moreover, these new operators can often be easily extended to higher dimensions. Concerning operators, Dufourd[42] examines the formal specification of geometric objects, and programming geometric constructions, using n-G-maps and n-maps.

A final section presents some conclusions.

n-DIMENSIONAL MAPS

In this section, the *combinatorial* definitions of the concepts of n-*dimensional generalized map* (or n-G-map) and n-*dimensional* (or n-map) are recalled. The formalism of n-G-maps and n-maps is used in the next section to compare some of the most representative ordered topological models. Some of the important properties of these concepts (in particular, the relationship between n-G-maps and n-maps) which are useful for this comparison, are also recalled (Lienhardt[11,39] presents a more complete study of n-G-maps and n-maps).

In the next section, the concept of the n-G-map is used to study models which represent the topology of subdivisions of *orientable* or *non-orientable* topological spaces (quad-edge[7], facet-edge[10] and cell-tuple[12] structures, for instance). The n-map concept is used to study models which represent the topology of subdivisions of *oriented* topological spaces (winged-edge[5], vertex-edge and face-edge[6] structures, pavings[8] and pavements[9], for instance). The relationship between n-G-map and n-map concepts illuminates the relations between models used for the representation of the topology of subdivisions of orientable or non-orientable topological spaces and models used for the representation of the topology of subdivisions of oriented topological spaces.

The n-G-map and n-map concepts are extensions of the map concept, defined in 1960 by Edmonds[43], and studied by many authors (cf. Tutte[29], Bryant and Singerman[30], Jacques[44], Cori[45] and Vince[46,47]; some uses of the map concept in geometric modelling are described by Lienhardt[21], Michelucci and Gangnet[48], Braquelaire and Guitton[49], Baudelaire and Gangnet[50] and Michelucci and Peroche[51]).

n-G-MAPS

n-G-maps are defined by using a single type of elements called *darts*. Intuitively, a dart may be defined as a zero-dimensional cell or *vertex*. This definition of a dart is discussed in the next section (in particular, a dart has also been defined as an oriented edge[45], as understood in graph theory – see Berge[52]). A dart will be graphically represented either by an oriented edge or by a half-edge (see Figure 1(a)). This dart concept can be found in most of the models used in boundary representation; but, in most cases, it is not explicitly defined[5–12,29,30,32,34,53] (cf. the next section).

Definition 1.
Let $n \geqslant 0$; an *n*-G-map is defined by an $(n+2)$-tuple $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$, such that (see Figures 1–4):

- B is a finite, non-empty set of darts;
- $\alpha_0, \alpha_1, \ldots, \alpha_n$ are involutions on B (i.e. $\forall\, i,\ 0 \leqslant i \leqslant n$, $\forall\, b \in B,\ \alpha_i^2(b) = b$), such that:
  - $\forall\, i \in \{0, \ldots, n-1\}$, α_i is an involution without fixed points (i.e. $\forall\, i,\ 0 \leqslant i \leqslant n-1,\ \forall\, b \in B,\ \alpha_i(b) \neq b$);
  - $\forall\, i \in \{0, \ldots, n-2\}$, $\forall\, j \in \{i+2, \ldots, n\}$, $\alpha_i\alpha_j$ is an involution.

If α_n is an involution without fixed points, G is without boundaries, or closed, else G is with boundaries, or open (cf. below).

Notation
For each set Φ of permutations of B, let $\langle \Phi \rangle$ be the *group of permutations* of B generated by Φ (notice that an involution is a permutation); for each dart b of B, $\langle \Phi \rangle(b) = \{\phi(b),\ \phi \in \langle \Phi \rangle\}$ is the *orbit* of b relative to group $\langle \Phi \rangle$, and $z(\Phi)$ is the number of orbits of $\langle \Phi \rangle$ in B. Let b be a dart of B, and let τ, τ' be two permutations on B; $b\tau = \tau(b)$, and $b\tau\tau' = (\tau' \circ \tau)(b)$. A *connected component* of an *n*-G-map $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$, incident to a dart b of B, is defined by $\langle \alpha_0, \alpha_1, \ldots, \alpha_n \rangle(b)$ (i.e. all darts which can be 'reached', starting from b, by successive applications of $\alpha_0, \alpha_1, \ldots, \alpha_n$). A connected *n*-G-map has only one connected component (i.e. $z(\langle \alpha_0, \alpha_1, \ldots, \alpha_n \rangle) = 1$). In order to simplify, we assume that *n*-G-maps are connected.

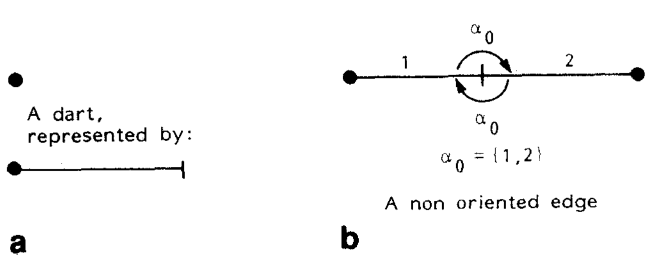


Figure 1. Dart and non-oriented edge

To explain the definition of *n*-G-maps, *n*-G maps are presented constructively by recursion on dimension n (this is similar to Lienhardt's[illegible] presentation of *n*-G-maps).

- $n = 0$. Let $G = (B, \alpha_0)$ be a 0-G-map. Intuitively, α_0 is a 'tie', which puts together at most two distinct darts (Figure 1(b)). Let G be connected. If α_0 is without fixed points, G models the boundary of a 1D cell (or a non-oriented edge: see Figure 1(b)). This corresponds intuitively to the fact that the boundary of an edge is defined by two vertices, i.e. by two darts.
- $n = 1$. Let $G = (B, \alpha_0, \alpha_1)$ be a 1-G-map, and let b, be a dart of B. The 0-G-map (B, α_0) defines the boundaries of edges (α_0 is without fixed points; cf. definition above), and G may be constructed by 'tying' together these edges around the vertices. Formally, the orbit $\langle \alpha_0 \rangle(b)$ is an edge of G. The orbit $\langle \alpha_1 \rangle(b)$ is a vertex of G. If α_1 is with fixed points, the 1-G map is graphically represented by a simple elementary path of edges (Figure 2(a)), else it is graphically represented by a simple elementary cycle of edges (Figure 2(b)). If b is invariant by α_1, b is incident to the boundary of G (in Figure 2(a), dart 1 is incident to a vertex extremity of this path). If G is connected and α_1 is without fixed points, G defines the boundary of a 2D cell (i.e. a face).
- $n = 2$. Let $G = (B, \alpha_0, \alpha_1, \alpha_2)$ be a 2-G-map. The 1-G-map (B, α_0, α_1) defines the boundaries of faces (α_1 is without fixed points; cf above), and G may be constructed by tying together these faces along their edges. Intuitively, a 'tie' α_2 'sews' the edges, by tying at most two distinct darts together (cf. Figure 3), i.e. α_2 is an involution on B. Moreover, as edges are tied by α_2, $\alpha_0\alpha_2$ is an involution (cf. Figure 3).
- $n = 3$. Similarly, a 3-G-map $G = (B, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$ may be constructed by tying together three-dimensional cells (the 2-G-map $(B, \alpha_0, \alpha_1, \alpha_2)$ defines the boundaries of these cells) along their faces, by an involution α_3, which ties the darts incident to the faces (cf. Figure 4). As faces are 'sewn' together (the 1-G-map (B, α_0, α_1) defines the boundaries of these faces), α_0, α_1 and α_3 are not independent i.e. $\alpha_0\alpha_3$ and $\alpha_1\alpha_3$ are involutions.
- In the general case of dimension n, an *n*-G-map $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ may be constructed by putting

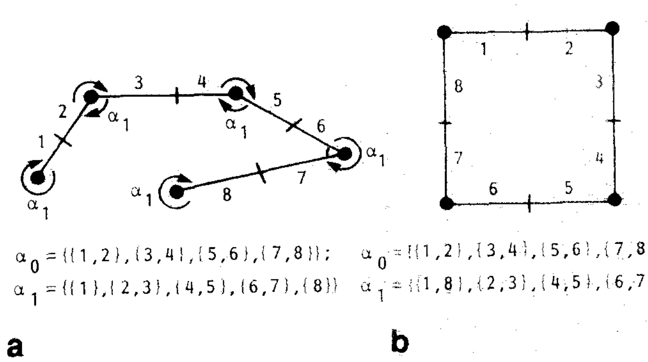
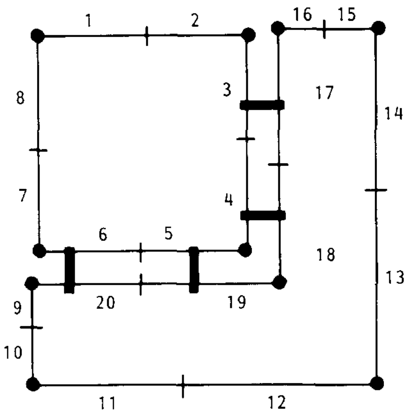


Figure 2. (a) Simple elementary path; (b) simple elementary cycle



$\alpha_2 = \{\{1\},\{2\},\{3,17\},\{4,18\},\{5,19\},\{6,20\},\{7\},\{8\},\{9\},\{10\},\{11\},\{12\},\{13\},\{14\},\{15\},\{16\}\}$

Figure 3. 2-G-map embedded discontinuously (α_2 is symbolised by a thick line)

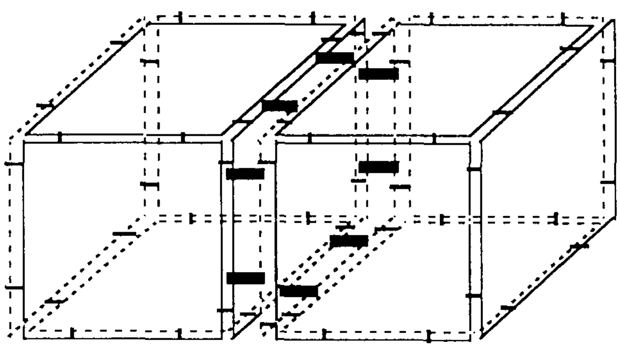


Figure 4. 3-G-map

together nD cells the $(n-1)$-G-map $(B, \alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ defines the boundaries of these nD cells) along their $(n-1)$-dimensional cells (the $(n-2)$-G-map $(B, \alpha_0, \alpha_1, \ldots, \alpha_{n-2})$ defines the boundaries of these $(n-1)$-dimensional cells), by an involution α_n. α_n ties the darts incident to these $(n-1)$-dimensional cells[46,54]. Thus α_n and all α_i $(\forall i \in \{0, 1, \ldots, n-2\}$ are not independent (i.e. $\alpha_i\alpha_n$ is an involution $\forall i \in \{0, 1, \ldots, n-2\}$).

Theorem 1 (due to Bryant and Singerman[30]). It is possible to associate the topology of a subdivision of a surface (orientable or non-orientable, with or without boundaries) with any connected 2-G-map; conversely, it is possible to associate a connected 2-G-map with the topology of any subdivision of any surface (cf. Tutte[29], · Bryant and Singerman[30] and Lienhardt[53]; Griffiths[55] presents a constructive approach to topological surfaces). Moreover, the representation of the topology of a subdivision of a surface by a 2-G-map is unique, up to isomorphism.

An outline of a constructive proof is the following. Griffiths[55] shows that any subdivision of any topological surface can be constructed by gathering topological faces (homeomorphic to a disc) along their edges, by the topological operation of identification of edges (and conversely). The proof of this theorem will consist in proving that it is possible to associate a topological face with any two-dimensional cell of a 2-G-map, and vice versa; and the topological operation of identification of edges with the combinatorial operation of 'sewing' edges. Such a theorem does not exist for higher dimensions. It is clear that it is possible to associate the topology of subdivisions of nD spaces with n-G-maps, and vice versa[9,10,12]. But the set of all subdivisions of all nD spaces, such that their topologies can be defined by n-G-maps and n-maps, is still a subject of active research (similarly for other ordered models – see below). It has been shown[56] that it is possible to associate any n-G-map with its barycentric triangulation, i.e. an n-G-map where the different n-dimensional cells are simplex n-G-maps (a simplex n-G-map defines the topology of a topological nD simplex). The topological operation which corresponds to the combinatorial operation of 'sewing' is still being studied (cf. below). This study could provide a constructive definition of subdivisions of spaces associated with n-G-maps (the idea is similar to the idea of the constructive proof presented above for $n = 2$).

We conclude this sub-section with a definition of cells associated with an n-G-map.

Definition 2. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ be an n-Gap-map (with $n \geqslant 1$); then $n+1$ $(n-1)$-G-maps of the elements $(G_i)_{i=0,\ldots,n}$ are defined, such that:

- $G_0 = (B, \alpha_1, \ldots, \alpha_n)$;
- $G_n = (B, \alpha_0, \alpha_1, \ldots, \alpha_{n-1})$;
- $\forall i \in \{1, \ldots, n-1\}, G_i = (B, \alpha_0, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n)$.

G_n is closed; the other $(n-1)$-G-maps $(G_i)_{i=0,\ldots,n-1}$ are open or closed depending on whether G is open or closed. For each i, each connected component of G_i is an i-dimensional cell of G, i.e. each connected component of G_0 is a vertex of G, each connected component of G_1 is an edge of G, each connected component of G_2 is a face of G ... (cf. Figure 5), and G may be constructed by tying together the connected components of G_n (nD cells) by the involution α_n.

Remark. Let G be a 2-G-map, and S be a subdivision of surface, such that G models the topology of S. To each iD cell of G $(i = 0, 1, 2)$ there corresponds an iD cell of S, and vice versa[29,30,53].

Definition 3. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ be an n-G-map, and $b \in B$. If $b\alpha_n = b$ (i.e. b is a fixed point of α_n), b is *free*, else b is *tied*.

Theorem 2. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ be an n-G-map $(n \geqslant 2)$, and $b \in B$; if b is free (tied), then all darts incident to the connected component $\langle \alpha_0, \alpha_1, \ldots, \alpha_{n-2} \rangle(b)$ are free (tied).

Let $G_{n,n-1}$ be the $(n-2)$-G-map $(B, \alpha_0, \alpha_1, \ldots, \alpha_{n-2})$. Any connected component of $G_{n,n-1}$ is free (tied) if and only if it is incident to a free (tied) dart.

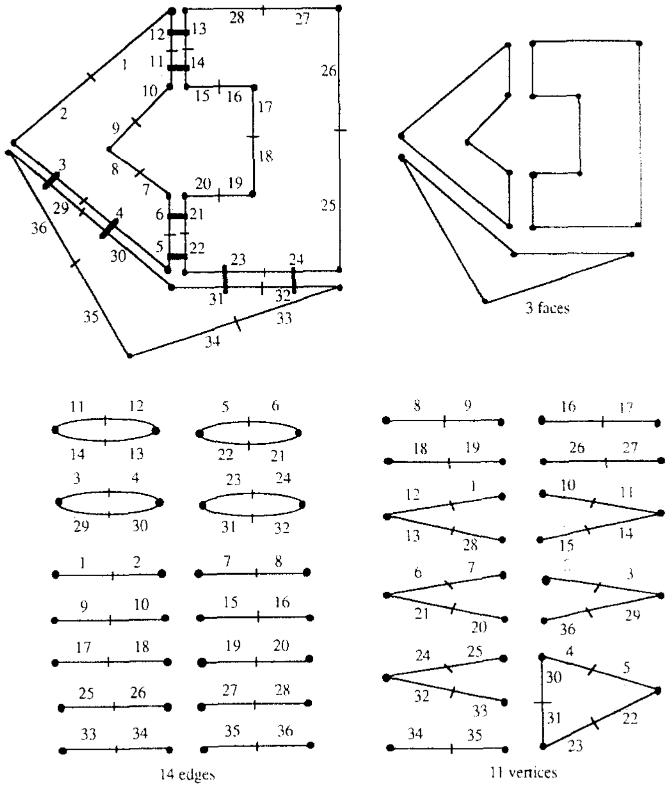


Figure 5. 2-G-map and its 1-G-maps of vertices, edges and faces

Definition 4. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ be an n-G-map ($n \geqslant 1$). The $(n-1)$-G-map of the boundaries of G is defined as the $(n-1)$-G-map $\partial(G) = (B', \alpha'_0, \alpha'_1, \ldots, \alpha'_{n-1})$, defined (see Figure 6) by:

- $B' = \{b \in B \mid b\alpha_n = b\}$ = set of the free darts of B
- $\forall i \in \{0, \ldots, n-2\}$, α'_i is the restriction of α_i to B' ($\forall b \in B'$, $b\alpha'_i = b\alpha_i$)
- α'_{n-1} is defined as follows. Let $b \in B'$; there exists exactly one free dart b', such that b and b' are distinct and are incident to the same connected component of the 1-G-map $(B, \alpha_{n-1}, \alpha_n)$ (this connected component is open, and then defines a simple elementary path, where b is incident to one extremity, and b' is incident to the other extremity). Then $b\alpha'_{n-1} = b'$, $b'\alpha'_{n-1} = b$, and α'_{n-1} is formally defined by:

$$\alpha'_{n-1} = \{\{b, b'\} \in B'^2 \mid b \neq b' \text{ and } b' \in \langle \alpha_{n-1}, \alpha_n \rangle(b)\}.$$

Theorem 2. $\partial(G)$ is a closed $(n-1)$-G-map.

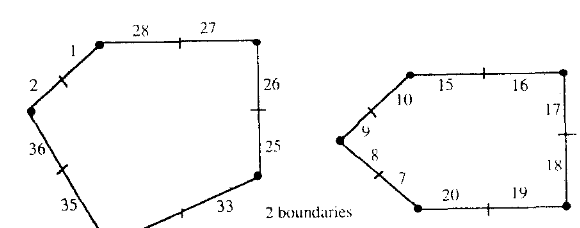


Figure 6. Boundaries of 2-G map of Figure 5

Any connected component of $\partial(G)$ is a boundary of G (the boundaries are made up by tying together the free connected components of G_{n-1} by the involution α'_{n-1}).

Remark. It is easy to compute the relationships of adjacency and incidence between the different cells of an n-G-map. For instance, let $G = (B, \alpha_0, \ldots, \alpha_{n-1}, \alpha_n)$ be an n-G-map, and let b be a dart of B. Let $V = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle(b)$ be the vertex incident to b, and suppose we want to compute the set SV of the vertices which are adjacent to V. SV is defined in the following way. For each dart b' of $\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle(b)$, let V' be the vertex which is incident to $b'\alpha_0$ (i.e. $\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle(b'\alpha_0)$). SV is the union of all these vertices. All the different relationships of adjacency and incidence are defined in this way.

n-Maps

Only the combinatorial definition of the concept of the n-map is given here (see Figure 7, for a more complete presentation of this notion, see Lienhardt[11,39]).

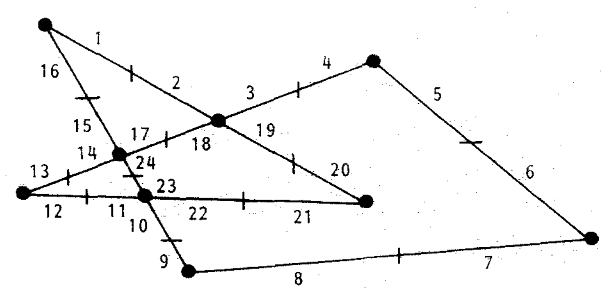
Definition 5. An n-map ($n \geqslant 1$) is defined by an $(n+1)$-tuple $C = (B, \alpha_0, \ldots, \alpha_{n-1})$, such that:

- B is a finite, non empty set of darts;
- $\alpha_0, \alpha_1, \ldots, \alpha_{n-2}$ are involutions on B, α_{n-1} is a permutation on B, such that, $\forall i, j$, $0 \leqslant i < i+2 \leqslant j \leqslant n-1$, $\alpha_i\alpha_j$ is an involution.

By extension, a 0-map is defined as a 1-tuple (B), where B is a finite, non empty set of darts.

Definition 6. Let $C = (B, \alpha_0, \ldots, \alpha_{n-2}, \alpha_{n-1})$ be an n-map. The n-map $C^{-1} = (B, \alpha_0, \ldots, \alpha_{n-2}, \alpha_{n-1}^{-1})$ is the inverse n-map of C.

Definition 7. Let $C = (B, \alpha_0, \ldots, \alpha_{n-1})$ be an n-map. If $n \geqslant 2$, we deduce from C $n+1$ $(n-1)$-maps of the



$\alpha_0 = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\},\{9,10\},\{11,12\},\{13,14\},\{15,16\},$
$\{17,18\ ,\ 19,20\},\{21,22\},\{23,24\}\}$

$\alpha_1 = \{\{1,16\},\{2,3,19,18\},\{4,5\},\{6,7\},\{8,9\},\{20,21\},\{10,11,23,$
$22\},\{12,13\},\{14,15,17,24\}\}$

$\alpha_1^{-1}\alpha_0 = \{\{1,15,13,11,9,7,5,3\},\{2,17,16\},\{4,6,8,10,21,19\},$
$12,14,23\},\{18,20,22,24\}\}$

Figure 7. N-*map and associated vertices, edges and faces*

elements C_i ($i = 0, \ldots, n$), defined by (see Figure 7):

$$C_0 = (B, \alpha_1, \ldots, \alpha_{n-1}); \forall i \in \{1, \ldots, n-2\},$$
$$C_i = (B, \alpha_0, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_{n-1});$$
$$C_{n-1} = (B, \alpha_0, \alpha_1, \ldots, \alpha_{n-2});$$
$$C_n = (B, \alpha_{n-1}^{-1}\alpha_0, \alpha_{n-1}^{-1}\alpha_1, \ldots, \alpha_{n-1}^{-1}\alpha_{n-2}).$$

Each connected component of each $(n-1)$-map C_i ($i = 0, \ldots, n$) deduced from C is an oriented iD cell of C (each connected component of C_0 is an oriented vertex, each connected component of C_1 is an oriented edge, etc.).

Relationship between *n*-G-maps and *n*-maps

In this subsection the notion of orientability of n-G-maps is considered. In particular, a relation is established between the notions of n-G-map and n-map, which makes it possible to compute the orientability of an n-G-map.

Definition 8. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ be an n-G-map ($n \geqslant 1$); the n-map $HG = (B, \alpha_n\alpha_0, \alpha_n\alpha_1, \ldots, \alpha_n\alpha_{n-1})$ is the n-map of the hypervolumes of G (by extension, if $n = 0$, $HG = (B)$). Let $b \in B$; the connected component $\langle \alpha_n\alpha_0, \alpha_n\alpha_1, \ldots, \alpha_n\alpha_{n-1} \rangle (b)$ is a hypervolume of G.

Definition 9. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ ($n \geqslant 1$) be an n-G-map with boundaries, and let BG' be an $(n-1)$-G-map such that G and BG' are disjoint, and BG' is isomorphic to the $(n-1)$-G-map of the boundaries deduced from G, by an isomorphism ϕ. The *closure* of G is defined as an n-G-map G'' without boundaries, deduced from G by (informally) gathering each boundary of G with the corresponding connected component of BG' using the isomorphism ϕ (the closure is formally defined by Lienhardt[11,39]). By extension, if G is an n-G-map without boundaries, the closure of G is equal to G.

A fundamental theorem is the following, which defines in particular the relationship between the concept of the n-G-map and that of the n-map (cf. Figure 8).

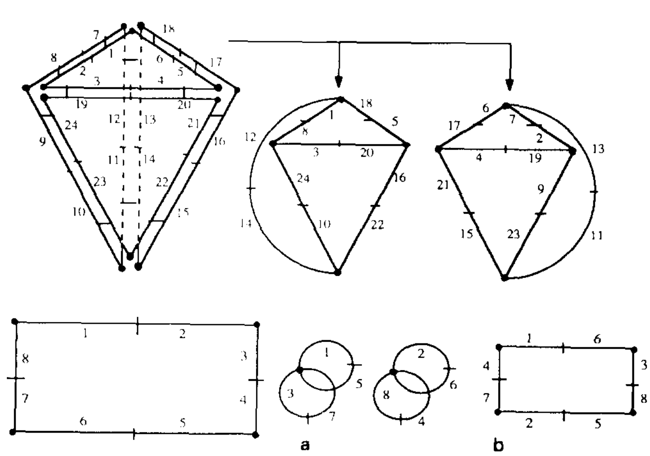


Figure 8. (top) tetrahedron and two connected components of 2-map of hypervolumes; (a) orientable; (b) non-orientable

Theorem 4. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ be a connected n-G-map ($n \geqslant 1$), and let HG be the n-map of the hypervolumes deduced from G. Then:

- HG has at most two connected components.
- If G is an n-G-map with boundaries, HG has exactly one connected component.
- If G is an n-G-map without boundaries and if HG has exactly two connected components vg_1 and vg_2, vg_1 is isomorphic to the inverse n-map of vg_2. Conversely, given two n-maps C_1 and C_2, such that C_1 is isomorphic to the inverse n-map of C_2, we can construct an n-G-map G without boundaries, such that the n-map of the hypervolumes of G has exactly two distinct connected components, which are isomorphic to C_1 and C_2.

Definition 10. An n-G-map G without boundaries is orientable (non-orientable) if and only if the n-map of the hypervolumes deduced from G has exactly two distinct connected components (one connected component). An n-G-map with boundaries is orientable (non-orientable) if and only if its closure is orientable (non-orientable).

Remark. Let G be a 2-G-map, and S be a subdivision of a surface, such that G models the topology of S. G is orientable if and only if S is a subdivision of an orientable surface[29]. If G is orientable, each connected component of the 2-map of the hypervolumes deduced from G corresponds to a possible orientation of the surface (cf. Tutte[29] and Bryant and Singerman[30]; on the orientability of subdivisions of surfaces and nD spaces, see, for instance, Griffiths[55] and Agoston[57]). If G is non-orientable, the connected component of the 2-map of the hypervolumes models the topology of the corresponding subdivision of the usual two-sheeted covering surface[30] – for instance, a sphere for a projective plane, a torus for a Klein bottle.

Remarks

Some properties of *n*-G-maps and *n*-maps

Definition 11. Let $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ be an n-G-map without boundaries. The n-G-map G^*, the *dual* of G, is defined by $G^* = (B, \alpha_n, \alpha_{n-1}, \ldots, \alpha_0)$.

Definition 12. Let $C = (B, \alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ be an n-map. The n-map C^* the dual of C, is defined by $C^* = (B, \alpha_0^{-1}, \alpha_0^{-1}\alpha_{n-1}, \ldots, \alpha_0^{-1}\alpha_1)$.

These definitions are needed for the following theorems:

Theorem 5. Let G be an n-G-map without boundaries. The dual of G^* is G. If $n \geqslant 1$, let $(G_i)_{i=0,\ldots,n}$ be the $(n-1)$-G-maps of the elements of G, and $(G_i^*)_{i=0,\ldots,n}$ be the $(n-1)$-G-maps of the elements of G^*. For each i, such that $0 \leqslant i \leqslant n$, G_i^* is the dual of G_{n-i}, G_i^* is orientable if and only if G is orientable.

Theorem 6. Let C be an n-map. The dual of C^* is C. If $n \geq 2$, let $(C_i)_{i=0,\ldots,n}$ be the $(n-1)$-maps of the elements of C, and $(C_i^*)_{i=0,\ldots,n}$ be the $(n-1)$-maps of the elements of C^*. For each i (with $0 \leq i \leq n$), C_i^* is the dual of C_{n-i}.

Theorem 7. Let G be an n-G-map without boundaries, and G^* be the dual of G ($n \geq 1$). The n-map of the hypervolumes deduced from G^* is the dual of the n-map of the hypervolumes deduced from G.

Classification for $n = 2$

It is well-known that four integer-valued characteristics can be associated with any subdivision of any surface[55]. They are the *number of boundaries*, the *Euler characteristic*, the *orientability factor* and the *genus*. The set of all subdivisions which have the same number of boundaries, orientability factor and genus defines a *topological surface*. For instance, a ring is defined by its two boundaries, an orientability factor of 0 (meaning it is orientable), and genus 0; a Möbius band has one boundary, an orientability factor of 1, and genus 0.

We define here for 2-G-maps (and for 2-maps) four equivalent characteristics (also integer-valued), which make it possible to classify 2-G-maps (and 2-maps), according to the classification of the subdivisions whose topologies are modelled by these 2-G-maps (and 2-maps). These characteristics can be easily computed on 2-G-maps and 2-maps. Such a classification is not known to exist for higher dimensions.

Let $G = (B, \alpha_0, \alpha_1, \alpha_2)$ be a 2-G-map. Associated with G are the following characteristics[39,53]: $b(G)$ is the number of boundaries of G, which equals the number of connected components of the 1-G-map of the boundaries deduced from G; $c(G)$ is the Euler characteristic of G, and is equal to $(z(\alpha_0\alpha_1) + z(\alpha_1\alpha_2) + z(\alpha_2\alpha_0) - |B|)/2$, where $|B|$ is the number of darts, and $z(\alpha_0\alpha_1)$, $z(\alpha_1\alpha_2)$ and $z(\alpha_2\alpha_0)$ are, respectively, the numbers of the orbits of the permutations $\alpha_0\alpha_1$, $\alpha_1\alpha_2$ and $\alpha_2\alpha_0$; $q(G)$ is the orientability factor, defined so that, if G is orientable, $q(G) = 0$; else, if $b(G) + c(G)$ is odd, $q(G) = 1$ (for instance, for a Möbius band), else $q(G) = 2$ (for instance, for a Klein bottle); and $g(G)$ is the genus of G, and equals $1 - (b(G) + c(G) + q(G))/2$ (for instance, 0 for a sphere, 1 for a torus with one 'hole').

Let $C(B, \alpha_0, \alpha_1)$ be a 2-map. By definition, C is oriented without boundaries, so that $b(C) = 0$ and $q(C) = 0$; thus, it is necessary to define only the following[39,53]: $c(C)$ is again the Euler characteristic of C, and equal to $z(\alpha_0) + z(\alpha_1) + z(\alpha_0\alpha_1) - |B|$, where $|B|$ and z are defined as for 2-G-maps; and $g(C)$ is the genus of C, defined as $1 - c(G)/2$.

Operations

Two operations are defined for constructing any n-G-map, and consequently any n-map[11,39]. Given an $(n-1)$-G-map G without boundaries, the first operation creates an n-G-map $G' = (B', \alpha'_0, \alpha'_1, \ldots, \alpha'_n)$, such that α'_n is the identity on B', and the $(n-1)$-G-map of the boundaries deduced from G' is equal to G. The second operation is (informally) the following (see Figure 9): any n-G-map can be defined by *putting together* $(n-1)$-G-maps without boundaries (i.e. the connected components of G_n), by 'sewing' together free connected components of $G_{n,n-1}$ ($(n-2)$-dimensional) (see the constructive presentation of n-G-maps at the beginning of this section; 'sewing' is achieved by the second operation). The most representative cases of 'sewing' operation in two dimensions are studied by Lienhardt[53], according to the variations of the characteristics associated with a 2-G-map.

The operations presented above make a set of basic operations, from which more elaborated operations can be defined (cf., for instance, operations defined by Tutte[29], for subdivisions of surfaces). These operations

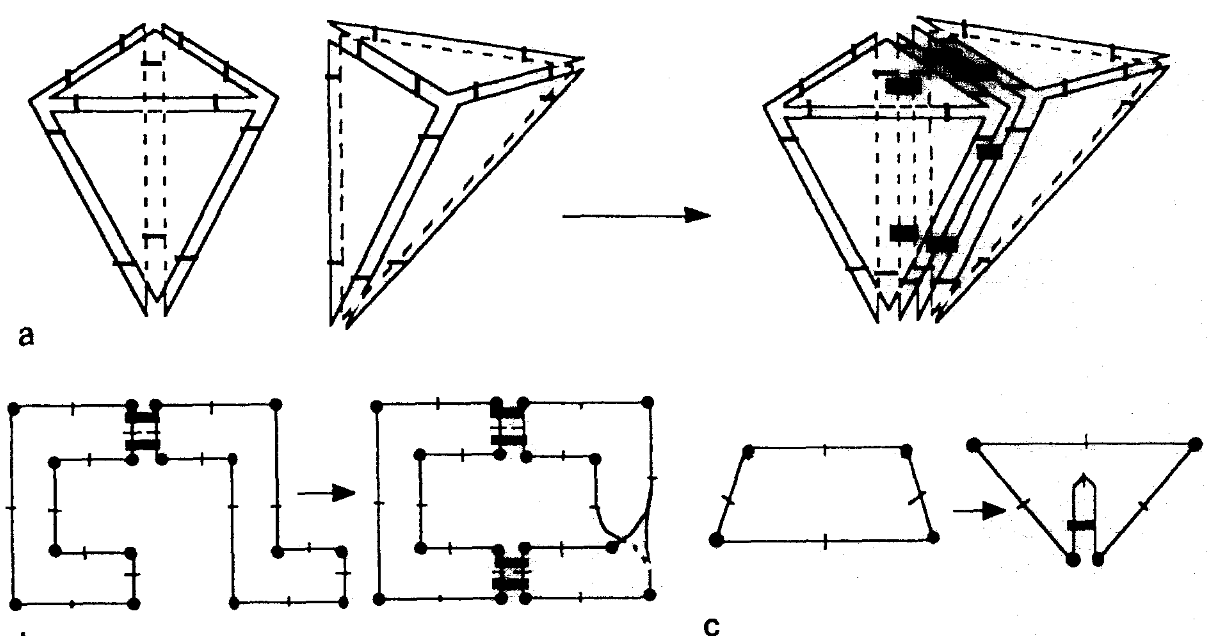


Figure 9. Different cases of 'sewing': (a) sewing two distinct connected components (dimension 3); (b) treating Möbius band (dimension 2); (c) bending edge back on itself (dimension 2)

have been implemented and verified in two modellers: a modeller of subdivisions of surfaces, based on 2-G-maps and 2-maps, and a modeller of subdivisions of 3D spaces, based on 3-G-maps and 3-maps. Higher-level operations have also been defined, based on these operations. Other basic operations are defined by Dufourd[42], for a study of the algebraic specification of n-G-maps and n-maps. They also allow any n-G-map and any n-map to be constructed.

Data structures

Data structures can be easily deduced from the definition of n-G-maps and n-maps (for instance, using pointers, or relational data structures; see, for instance, Lienhardt[53] and Dufourd *et al*[58], who present implementations of 2-G-maps and 2-maps). A possible implementation of n-G-maps is as follows. Three types of object exist: n-G-maps, connected components and darts. An n-G-map is defined by a set of connected components, a connected component is defined by a *reference dart*, and a dart contains $n+1$ pointers symbolizing the involutions α_i. The consistency constraints, necessary for such a data structure to define an n-G-map (or an n-map), can be directly deduced from the definition of n-G-maps and n-maps. For instance, the relations in the data structure between two darts (symbolizing the involutions α_i) must satisfy the constraints given in the definition of n-G-maps (all α_i are involutions, and the composition of α_i and α_j, for $i+2 \leqslant j$, is also an involution).

It is obvious that it is possible to deduce other implementations. For instance, for geometric modelling, other types of object are needed, such as vertices, edges and faces, mainly for the information associated with them (embedding, etc.). Such implementations have been studied at CNRS, in particular for the two modellers of subdivisions we are developing. An implementation in the general case of n dimensions has been realized (in this implementation, n is a parameter, and the data structure can handle simultaneously generalized maps of different dimensions). Darts, cells, boundaries, connected components and n-G-maps are explicitly represented in this data structure, and all constraints of consistency have been directly deduced from the definition of n-G-maps (just apply definitions of cells, boundaries, connected components, etc.: cf. above).

TOPOLOGICAL MODELS USED IN BOUNDARY REPRESENTATION

In this section some of the most representative topological models used in boundary representation are studied (the choice of the models studied here does not reflect any value judgement). Two main types of model are distinguished:

- 'incidence graphs', where the different cells of the modelled subdivision, and some adjacency and incidence relations between these cells, are explicitly represented;
- 'ordered' topological models, which use, more or less explicitly, a single type of basic elements (for instance, darts in n-G-maps and n-maps), and where the different cells are implicitly defined[5-12]; this implicit definition may be explicit in a data structure deduced from the model (cf., for instance, the above discussion on data structures). 'Element functions' act on these basic elements (using the terminology of Guibas and Stolfi[7] and Dobkin and Laszlo[10], which define 'edge functions' and 'traversal functions').

The aims of this section are twofold: to show the important evolution of topological models, as a consequence of which, the field of boundary representation (i.e. the set of objects which can be modelled) has been notably extended; and to compare ordered topological models, using the n-G-map and n-map concepts, to bring out the ideas on which these models are based. An (incomplete) study of the topological operators which act on these models is made in the next section.

Incidence graphs

Models of this type represent explicitly the different cells of the modelled subdivision, and some adjacency and incidence relations between these cells. Three incidence graphs, quite equivalent, are described in recent work: the 'polytopal mesh representation'[38]; the 'incidence poset'[35] and the 'incidence graph' itself[36], for the representation of the topology of (respectively) nD polytopes, finite regular CW-complexes (see below), and arrangements of hyperplanes in nD Euclidean space. Informally, these incidence graphs represent a finite collection of open disjoint iD cells ($0 \leqslant i \leqslant n$) whose union is an nD manifold. The boundary of each iD cell is equal to the union of cells which have a dimension lower than i. Each iD cell belongs to the boundary of an $(i+1)$-dimensional cell.

The topology of the modelled subdivision can be represented schematically by a graph, where the nodes correspond to the cells of the subdivision, and the edges represent the incidence relations between i-dimensional cells and $(i+1)$-dimensional cells (for i between 0 and $n-1$). For notational convenience, Brisson[35] and Edelsbrunner[36] assume that two cells always exist: an $(n-1)$-dimensional cell, incident to all vertices, and an $(n+1)$-dimensional cell, incident to all n-dimensional cells. Brisson[35] has shown that this representation is sufficient for the representation of all topological information (up to isomorphism).

Edelsbrunner[36] describes an implementation of such an incidence graph, where the incidence relations are represented in the following way. Two lists of cells are associated to each iD cell γ: a list of all $(i-1)$-dimensional cells incident to γ and a list of all $(i+1)$-dimensional cells incident to γ. Sobhanpanah[38] defines only the first list (for instance, a polyhedron is defined by a list of faces, a face is defined by a list of edges, an edge is defined by a list of two vertices).

Selective geometric complexes

Selective geometric complexes (SGCs) are employed in the representation of nD pointsets[37] (more precisely, subsets of real algebraic varieties in Euclidean n-space).

Informally, a geometric complex is a finite collection of open disjoint iD cells. The boundary of each iD cell is equal to the union of cells which have a dimension lower than i. It should be observed that an iD cell may not belong to the boundary of an $(i+1)$-dimensional cell. For instance, the boundary of a face may be a vertex; a 3D cell may be combined with dangling faces and edges in Euclidean three-space (cf. presentation of 'non-manifold' objects below).

Two sets of cells are associated with each cell c: the set of all cells which belong to the boundary of c (c.boundary), and the set of all cells such that c belongs to their boundaries (c.star). Let c and c' be two cells of the geometric complex. If c belongs to c'.boundary, c' belongs to c.star, and vice versa. A geometric complex may be represented by an incidence graph, where the nodes correspond to the cells of the geometric complex, and the edges represent the relation 'belongs to the boundary of' and its inverse. Notice that edges may exist between cells of dimensions i and $i+k$, with $k > 1$.

A *neighbourhood* parameter is associated with each pair of cells (c, c'), such that c belongs to the boundary of c'. In a simplified way, the neighbourhood is defined if c is iD and c' is $(i+1)$-dimensional (for a more precise study, see[37]). The neighbourhood of c with respect to c' can take one of the following three values: 'full', 'left' and 'right'. If its value is 'full', c is not in the boundary of the topological closure of c' (in other words, c is an *interior* boundary of c'). When its value is not 'full', it must be 'left' or 'right', and denotes the 'side' on which c' is located with respect to c (c is an *exterior* boundary of c'). The neighbourhood parameter can be used, for instance, to define an orientation of the cells.

Selective geometric complexes are defined as geometric complexes where *active* and *non-active* cells are distinguished. Pointsets associated with active cells should be included in the pointset defined by the SGC.

Consequently, the pointset associated with an SGC is the union of all pointsets associated with active cells (this enables an SGC to represent a non-closed pointset). SGCs are represented by incidence graphs (as geometric complexes can be represented by incidence graphs). Attributes are associated with the nodes of a graph, in particular an active attribute (meaning that the corresponding cells are active or not). Other (mainly embedding) information is also associated (for more details, see Rossignac and O'Connor[37]).

Face-adjacency graphs

Another type of incidence graph, the face-adjacency graph[17,59], is used to model the topology of orientable surfaces without boundaries, i.e. for solid modelling. This model is an extension of the 'edge-face relational scheme'[60]. It is in fact a hypergraph, defined by a set of nodes, a set of arcs and a set of hyperarcs. Each node corresponds to a face of the subdivision (and vice versa); each arc corresponds to an edge of the subdivision (and vice versa), and joins the two nodes representing the two faces which are incident to the edge; each hyperarc corresponds to a vertex of the subdivision (and vice versa), and joins all nodes representing the faces which are incident to the vertex. Ansaldi *et al*[17] have shown that face-adjacency graphs are sufficient for representing all topological information for subdivisions of orientable surfaces without boundaries, up to isomorphism.

Remarks

The foregoing raises some apparently important problems. First, what are the constraints of consistency with respect to these models, in order that they represent 'valid' objects? For instance in two dimensions, incidence graphs exist which do not model the topology of a subdivision of a surface. Control over consistency is exerted in Ansaldi *et al*[17] through topological operators defined for handling face-adjacency graphs (Euler operators). Similarly, Sobhanpanah[38] presents a method for constructing an incidence graph, given a convex polytope defined by a set of hyperplanes; but no topological operator is defined (cf. the next section).

Second, the topological information contained in this type of model is sufficient, but perhaps its representation does not enable easy computation of certain important topological properties (orientability, for instance). In this regard, notice that an incidence graph can easily be deduced from an ordered topological model. For instance, the definition of the cells can be deduced directly from an n-G-map, as can the relationships of adjacency and incidence between these cells (cf. the previous section). No algorithm is known to exist for converting an incidence graph into an ordered topological model (a hint of such an algorithm is contained in the definition of the cell-tuple structure[12]: cf. below). It may be useful to study such algorithms, for instance in order to deduce algorithms for computing on incidence graphs such topological properties as orientability.

'Ordered' topological models

The topological models presented below are based on ideas which are very close to each other. They use (more or less explicitly) a single type of basic elements, on which a set of 'element functions' act. The cells of the modelled subdivision are defined in an implicit manner, using the basic elements and the element functions[6,12]. A characteristic of these models is the notion of 'ordering'[6,12,37] (see, for instance, Brisson[12] on the ordering of i-dimensional and $(i+1)$-dimensional cells 'about' an $(i-1)$-dimensional cell, and 'within' an $(i+2)$-dimensional cell).

Different models are presented according to the class of objects which can be modelled, i.e. taking into account dimension and orientability. In particular, models defined for the representation of the topology of subdivisions of *oriented* topological spaces are compared with n-maps, models defined for the representation of the topology of subdivisions of *orientable* or *non-orientable* topological spaces are compared with n-G-maps. The relations between the two types of model can be deduced from the relations between n-G-maps and n-maps (see the previous section). In this sub-section topological models are compared, but not the data structures deduced from these models, except to point out different approaches. Notice that the topological model subjacent to a data structure is sometimes not explicitly defined (in the cases of the winged-edge, vertex-edge and radial-edge data structures, for instance).

Models for representing the topology of subdivisions of orientable surfaces

Winged-edge data structure. The 'winged-edge' data structure is presented by Baumgart[5] for modelling the topology of subdivisions of orientable surfaces without boundaries. The main element in this data structure is the *edge*. With each edge e are associated pointers which maintain the topological information. They are: two pointers to the faces incident to e; two pointers to the vertices incident to e; and four pointers to the edges which are 'directly' adjacent to e, i.e. the edges which can be reached, starting from e, turning clockwise and counter-clockwise around each extremity vertex (see Figure 10). These pointers define doubly-linked cycles of edges around the vertices. The dart (intuitively oriented edge) concept is not expressed explicitly, but can be found here by the fact that all pointers which represent the adjacency and incidence relations can be, for any edge, partitioned into two similar disjoint sets. The element functions which act on these implicit half-edges are symbolized by the pointers which define cycles of edges around the vertices. The ordering of cells about or within cells with lower or higher dimension is achieved through these pointers.

Initially, the winged-edge data structure was used for the representation of subdivisions of orientable surfaces without boundaries, where each edge is incident to exactly two distinct vertices (no loops) and to exactly two distinct faces (no isthmuses), and where two distinct edges are not incident to the same two vertices (no multiple edges). Weiler[6] has shown that this data structure is sufficient to model unambiguously subdivisions including loops, isthmuses and multiple edges. But case studies are then necessary to restore all topological information (on the modelling of such 'degeneracies', see Guibas and Stolfi[7]).

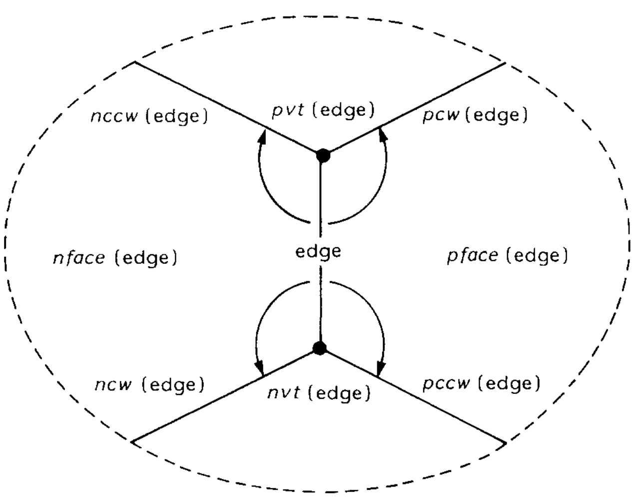


Figure 10. pvt *and* nvt *define vertices incident to edge,* pface *and* nface *define faces incident to edge,* pcw, pccw, ncw, nccw *define edges 'directly' adjacent to edge, turning clockwise or counter-clockwise around vertices incident to edge*

We can formally prove that any 2-map (B, α_0, α_1) (where α_0 is without fixed points) can be represented by a winged-edge data structure (pointers defining cycles of edges around the vertices correspond to permutation α_1, and to its inverse), and vice versa. This can be proved, using the results of Weiler[6] (see below).

Assume that a winged-edge structure W and a 2-map $C = (B, \alpha_0, \alpha_1)$ represent the topology of the same subdivision of the same surface (in the general case where no loop or isthmus is represented, in order to avoid studies of cases). We have seen that implicit 'element functions' act on implicit half-edges in W. Thus, we define exactly two half-edges for each edge. A dart in C corresponds to each half-edge, and vice versa. α_0 'ties' two darts such that these darts are associated with the two half-edges corresponding to the same edge (consequently, α_0 is an involution without fixed points). We have seen that pointers define doubly-linked cycles of edges around the vertices. It is easy to deduce cycles of half-edges from these cycles of edges. These cycles are represented in the 2-map C by involution α_1 (α_1 is a permutation, and defines cycles of darts around the vertices: see Figure 7). So, α_1 and its inverse α_1^{-1} define doubly-linked cycles of darts, which correspond to the double-linked cycles of half-edges (a similar comparison is made below between 2-maps and vertex-edge data structures).

Let $G = (B', \alpha'_0, \alpha'_1, \alpha'_2)$ be an orientable 2-G-map without boundaries, such that the 2-map of the hypervolumes deduced from G has exactly two connected components, one of them isomorphic to C. If α_0 is without fixed points in the 2-map C then $\alpha'_0\alpha'_2$ is without fixed points in G. If we consider that G is constructed by putting together faces along their edges, using 'sewing' operations then the fact that $\alpha'_0\alpha'_2$ is without fixed points means that no edge has been bent back on itself (cf. Figure 9).

Modified winged-edge, vertex–edge and face–edge data structures. Weiler[6] defines three structures for modelling the topology of subdivisions of orientable surfaces without boundaries. The first is an extension of the winged-edge data structure, modified by adding indicators. These allow, for each edge, a distinction to be made between its two sides, i.e. between half-edges. However, the basic element of this structure is still the edge. Two other data structures are defined: the vertex–edge data structure, and the face–edge data structure. For the two structures, the basic element is the half-edge. Element functions which act on the half-edges are symbolized by pointers: in the vertex–edge data structure, the half-edges are organized into cycles around the vertices; in the face–edge data structure, they are organized into cycles around the faces. Moreover, each half-edge is associated with its opposite half-edge. Weiler[6] has proved that vertex–edge and face–edge data structures are equivalent (the face–edge data structure is a 'dual version' of the vertex-edge data structure). Also, any winged-edge data structure can be converted into a vertex–edge data structure, and vice versa.

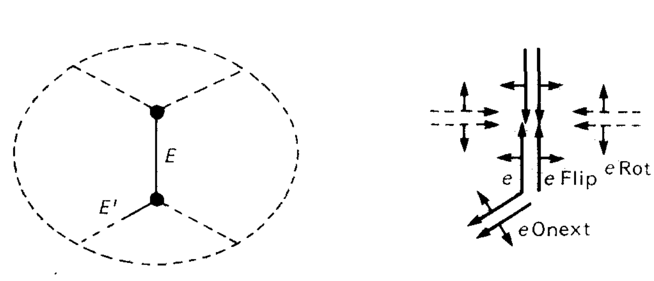
It is easy to prove that the vertex–edge data structure is equivalent to the concept of a 2-map, where α_0 is without fixed points. A dart corresponds to each half-edge, and vice versa; α_0 corresponds to the pointer which joins two opposite half-edges (in particular, α_0 is without fixed points); α_1 corresponds to the pointers which join half-edges into cycles around the vertices.

The consistency constraints which must be satisfied by these data structures, and the consistency constraints which must be satisfied by data structures deduced from the notion of G-maps, are obviously similar. This means that the pointers must satisfy the definition of α_0 and α_1 (α_0 is an involution without fixed points, α_1 is a permutation). Moreover, the explicit definition of the cells in the data structure must be consistent with respect to their implicit definition (for instance, in a 2-map, all darts which belong to the same orbit of α_1 are incident to a same vertex; cf. the discussion of data structures in the previous section).

Models for representing the topology of subdivisions of orientable or non-orientable surfaces

Edge algebra and quad-edge data structure. The notion of 'edge algebra'[7] is used for modelling the topology of subdivisions of surfaces, orientable or non-orientable, without boundaries. This topological model allows the simultaneous representation of the topologies of the subdivision, the dual subdivision and the mirror-image subdivision. Four 'directed and oriented' edges are associated with each edge of the subdivision; similarly, four directed and oriented edges are associated with each edge of the dual subdivision (cf. Figure 11).

These directed and oriented edges are the basic elements of the model. Three edge functions act on these basic elements: Flip, Onext and Rot. Let e be a directed and oriented edge. Flip reaches e', which has same direction but the inverse orientation to e, such that e and e' are associated with the same topological edge. Onext reaches e'', which is directly adjacent to e, turning counter-clockwise around the origin vertex of e, such that e and e'' have the same direction and same orientation. Rot reaches the directed and oriented edge, the dual of e' (the edge functions Flip, Onext and Rot satisfy ten properties presented by Guibas and Stolfi[7]). An edge algebra is formally defined as a 5-tuple (E, E^*, Onext, Rot, Flip), where E and E^* are, respectively, the sets of oriented and directed edges corresponding to the edges of the primal and the dual subdivisions.



a **b**

Figure 11. (a) edge E of subdivision of surface is represented, with part of edge E', directly adjacent to E, turning counter-clockwise around their common vertex. (b) four directed and oriented edges corresponding to E are represented, with two of the directed and oriented edges corresponding to E'. Direction is shown by large arrow, orientation by perpendicular little arrow. Dashed arrows correspond to directed and oriented edges, the duals of those corresponding to E

It is possible to define an equivalent structure using 2-G maps (with the results of the comparison made by Brisson[12], between the cell-tuple structure (see below) and the quad-edge data structure). Let $G = (B, \alpha_0, \alpha_1, \alpha_2)$ be a 2-G-map, such that α_2 and $\alpha_0\alpha_2$ are without fixed points. Let $G^d = (B^d, \alpha_0^d, \alpha_1^d, \alpha_2^d)$ be a 2-G-map which is isomorphic to the dual 2-G-map of G, by an isomorphism α^d (with B and B^d disjoint).

Let α_0^*, α_1^*, α_2^*, α^{*d} be involutions on $B \cup B^d$, such that for each i ($0 \leqslant i \leqslant 2$), the restriction of α_i^* to B is equal to α_i, the restriction of α_i^* to B^d is equal to α_i^d; the restriction of α^{*d} to B is equal to α^d, and the restriction of α^{*d} to B^d is equal to $(\alpha^d)^{-1}$. Notice that $\alpha_0^* = \alpha^{*d}\alpha_2^*\alpha^{*d}$ (see the previous section).

Let Γ be the 5-tuple (B, B^d, $\alpha_1^*\alpha_2^*$, $\alpha_2^*\alpha^{*d}$, α_2^*). Γ simultaneously defines the primal and the dual of G, and we can prove that such a 5-tuple corresponds to any edge algebra, and vice versa. To each directed and oriented edge there corresponds a dart of $B \cup B^d$ (more precisely, each element of E corresponds to a dart of B, and vice versa; each element of E^* corresponds to a dart of B^d, and vice versa). Flip corresponds to α_2^*. Onext corresponds to $\alpha_1^*\alpha_2^*$ and Rot corresponds to $\alpha_2^*\alpha^{*d}$.

Notice that α_2 and $\alpha_0\alpha_2$ are without fixed points in G, i.e. G is without boundaries and no edge of G is bent on itself. In this case, each edge of G (i.e. each orbit of $\langle\alpha_0, \alpha_2\rangle$) is composed of exactly *four* distinct darts, i.e. four directed and oriented edges. This corresponds in particular to the fact that edge algebras cannot directly represent the topology of subdivisions of surfaces *with boundaries*. In a similar way, it should be observed that the dual n-G-map G^* of an n-G-map G is only defined if G is without boundaries. This is a consequence of the definition of an n-G-map, where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ must be without fixed points (see the previous section). An extension of this definition of n-G-maps is presented below, in the 2D case, where all involutions may have fixed points.

It should be observed that simplifications of edge algebras are proposed by Guibas and Stolfi[7] for the representation of orientable surfaces. In particular, it is shown that the directed and oriented edges can be partitioned into two sets, each closed under Rot and Onext. Moreover, for the representation of a simple subdivision (i.e. without its dual), only primitive operators Onext and Rot^2 are needed. This corresponds to the relation between 2-G-maps and 2-maps (see the previous section); the 2-map of the hypervolumes deduced from an orientable 2-G-map without boundaries has exactly two connected components (closed under primitive operators $\alpha_2\alpha_0$ and $\alpha_2\alpha_1$, corresponding respectively to Rot^2 and $Onext^{-1}$).

The quad-edge data structure is deduced from the definition of edge algebras. It is important to notice that directed and oriented edges are partitioned into groups of eight, i.e. the data structure contains *edge records*. Each edge record corresponds to an edge of the modelled subdivision, and represents the four directed and oriented edges associated with the edge, and the four directed and oriented edges associated with the dual edge. This implementation is quite different from the implementation of n-G-maps presented in the discussion of data structures in the previous section (where a record in a data structure corresponds to each basic element, and vice versa; see also the vertex–edge and face–edge data structures in the previous sub-section). It could be interesting to study the generalization to higher dimensions of the idea presented in the quad-edge data structure (i.e. partitioning basic elements).

Notions of map. Other topological models (extending the notion of the combinatorial map) have been defined for modelling the topology of subdivisions of surfaces, orientable or non-orientable, with or without boundaries. In particular, Tutte[29] defines a map as a 4-tuple (S, θ, ϕ, P), where S is a finite set of 'crosses'; θ and ϕ are involutions without fixed points on S, such that $\theta\phi$ is an involution without fixed points; and P is a permutation on S, such that θP is an involution and such that for each cross b of S, $\langle P\rangle(b)$ and $\langle P\rangle(b\theta)$ are disjoint. It is easy to show that this definition is equivalent to the definition of 2-G-maps $(B, \alpha_0, \alpha_1, \alpha_2)$, such that α_2 and $\alpha_0\alpha_2$ are without fixed points. Crosses correspond to darts, θ corresponds to α_2, ϕ corresponds to α_0 and P corresponds to $\alpha_2\alpha_1$; this verifies in particular the definition of the vertices, edges and faces, given by Tutte[29]. It will be evident that, as for the edge algebras presented above, this definition of a map does not allow the direct representation of subdivisions of surfaces with boundaries.

Another extension of the notion of the combinatorial map is given by Bryant and Singerman[30]. This uses three involutions, τ, λ and ρ (corresponding, respectively, to involutions α_2, α_0 and α_1 of the 2-G-maps). But here, each of these involutions may have fixed points. This map is thus larger in scope than the 2-G-map. It is possible to represent subdivisions of surfaces with boundaries (α_2 has fixed points), or subdivisions where a face intersects a boundary of the surface at a vertex but not along an edge (α_1 has fixed points), or an edge intersects a boundary but not at a vertex (α_0 has fixed points, cf. Fig. 12; see Bryant and Singerman's whole topological interpretation of these fixed points[30]). Extensions of the notion of the n-G-map where such fixed points are allowed are being studied at Strasbourg. The definition of these extended n-G-maps would be more homogeneous (not only α_n may have fixed points; and the dual of an n-G-map with boundaries can be defined).

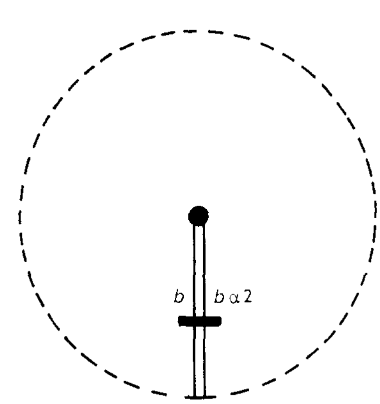


Figure 12. Dart b *is such that* $b\alpha_0 = b$; *topological edge, corresponding to edge* $\{b, b\alpha_2\}$ *intersects boundary of surface, but not at vertex*

Models for representing the topology of subdivisions of orientable three-dimensional topological spaces

Pavings. Spehner[8] presents a definition of 'paving', extending the notion of the two-dimensional combinatorial map. A paving is defined as a 4-tuple $(B, \alpha, \sigma, \phi)$, where:

- (B, α, σ) is a 2-map such that α is without fixed points. Each connected component of this 2-map is an oriented three-dimensional cell (cf. the definition of an n-map in the previous section);
- ϕ is a permutation which satisfies $\chi\alpha\chi = \alpha$, $\sigma\phi^{-1}\sigma = \phi$, and such that for each dart b of B, $b\phi \neq b\alpha$, $b\phi \neq b\sigma$ (i.e. ϕ gathers oriented three-dimensional cells along their faces).

Intuitively (and from a constructive point of view), the principle consists in putting together oriented three-dimensional cells (represented by connected components of the 2-map (B, α, σ)) along their oriented faces by permutation ϕ (conditions on ϕ involve faces being tied together by ϕ).

Let $G = (B, \alpha_0, \alpha_1, \alpha_2)$ be a 3-map such that α_0, α_1 and $\alpha_0\alpha_2$ are involutions without fixed points. It can be shown that any paving is equivalent to a 3-map of this type, and vice versa. α corresponds to $\alpha_0\alpha_2$, σ corresponds to $\alpha_1\alpha_2$ and ϕ corresponds to α_2 (this verifies, in particular, the definitions of vertices, edges, faces and volumes given by Spehner[8]). The fact that α_0, α_1 and $\alpha_0\alpha_2$ are without fixed points means that no face is bent back on itself in a paving and no edge is bent back on itself in a three-dimensional cell (in the sense of the previous subsection).

Pavements. The definition of 'pavement' is presented by Arquès and Koch[9,61]. This is another extension of the concept of the combinatorial map (see also Arquès and Jacques[62]). The basic element used in pavements is the 'oriented angular sector', which can be explained in the following way. Let S be a subdivision of Euclidean three-space, and let v be a vertex of S. Let s be a little sphere, centered in v, with a radius such that s does not contain any other vertex of S. The intersection of s with S defines a subdivisions of s. More precisely, the intersection with s of the edges of S incident to v defines vertices on s. The intersection with s of the faces of S which are incident to v defines non-oriented edges on s, i.e. 'angular sectors' (an angular sector is then defined by two 'oriented angular sectors', inverse to each other). Finally, the intersection with s of the three-dimensional cells incident to v defines faces on s. Formally, the topology of the subdivision defined on sphere s may be modelled by a 2-map (B, α, σ), where B is a set of oriented angular sectors, α is an involution without fixed points and σ is a permutation.

It is important to compare oriented angular sectors and darts (as intuitively presented in the previous section). For this purpose, the 1-G-map of the vertices deduced from the 2-G-map of Figure 5 is used (with a result similar to when using the 2-map of the vertices deduced from a 3-map). Each edge of this 1-G-map corresponds to an angular sector, and each of these edges is defined by two oriented angular sectors, which here are darts. This means (using the terminology of n-G-maps) that a dart, related to α_0, corresponds to an oriented edge; when related to α_1, it corresponds to an oriented angular sector; when related to α_2, it corresponds to an oriented angle between two faces which are incident to the same edge, and so on.

Other topological interpretations of the dart concept have been offered, involving the half-edge[6], directed and oriented edge[7], facet-edge element[10] (cf. below), cell-tuple[12] (cf. below). The most general interpretation is the concept of the cell-tuple (defined in the general case of n dimensions), but it will be seen that it is still not general enough (cf. below).

Pavements (*sensu* Arquès and Koch[9]) are defined by adding an involution, ϕ, without fixed points, such that $\delta = \phi\sigma$ is an involution without fixed points. Informally, ϕ gathers two oriented angular sectors (i.e. two darts), which are incident to the same face. Finally, a pavement is formally defined as a 4-tuple $(B, \phi, \alpha, \delta)$. We can show that such pavements, and 3-maps $(B, \alpha_0, \alpha_1, \alpha_2)$, such that α_0, α_1 and $\alpha_0\alpha_2$ are without fixed points, are equivalent. Oriented angular sectors correspond to darts, ϕ corresponds to α_0, α corresponds to α_1, and δ corresponds to $\alpha_0\alpha_2$.

Arquès and Koch[9] restrict the field of application of pavements to subdivisions of Euclidean three-space. In particular, the intersection of faces incident to a vertex with a little sphere centered at this vertex defines a subdivision of a sphere. This subdivision is represented by a 2-map, so this 2-map is *planar* (i.e. its genus is 0: see the definition of the characteristics associated with a 2-map in the previous section). But this condition is not explicitly defined in the topological model. It will be seen that similar problems arise in other *descriptive* approaches, i.e. approaches where a topological model is deduced, given an original set of subdivisions which has to be modelled (cf. below). In all cases, the set of subdivisions of topological spaces, such that their topologies can be represented by the topological model, is larger than the original set. Similarly, this problem is still not solved for *constructive* approaches, i.e. approaches which study the construction of (the topology of) subdivisions of topological spaces (see the constructive approach of n-G-maps in the previous section). In fact, the entire set of subdivisions of all topological spaces associated with a topological model is incompletely defined, for dimensions greater than 3. This is one of the main unsolved problems concerning ordered topological models.

Models for representing the topology of subdivisions of orientable or non-orientable three-dimensional topological spaces

Facet-edge data structure. Dobkin and Laszlo[10] define the notion of facet-edge data structure, for modelling subdivisions of Euclidean three-space (orientable, with or without boundaries), such that the 3D cells of these subdivisions are homeomorphic to a sphere. In fact, if the subdivision is with boundaries, an 'unbounded polyhedron' is considered, whose boundary coincides with the boundary of the subdivision. This is a way of defining the 'closure' of the subdivision, which is without boundaries (facet-edge structures are equivalent to a subclass of 3-G-maps without boundaries: cf. below). Consequently each face of the subdivision is incident to *two* polyhedra, possibly not distinct.

Let S be such a subdivision. This topological model is based on the notion of face-edge pair (f, e), such that face f is incident to edge e. For each face-edge pair of S four 'facet-edge' elements are defined, corresponding to the four possible ways of defining an orientation of the face-edge pair within the face, and around the edge. The 'facet–edge' elements are the basic elements of the model. Four 'traversal functions' are defined on the facet–edge elements (cf. Figure 13)

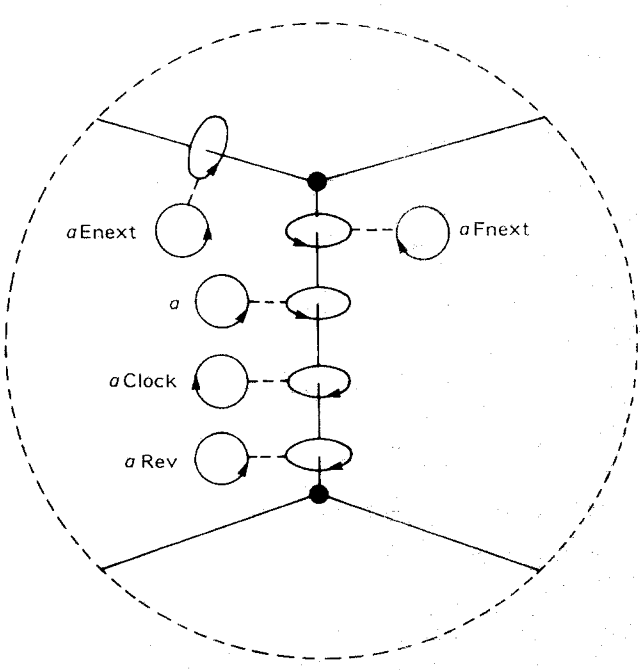


Figure 13. Facet–edge elements represented by two cycles joined by dashed line

– Clock, Enext, Fnext and Rev – which satisfy properties presented by Dobkin and Laszlo[10].

It is possible to show that this notion is equivalent to the notion of the 3-G-map $(B, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$, where the involutions α_3, $\alpha_0\alpha_2$, $\alpha_0\alpha_3$, $\alpha_1\alpha_3$ are without fixed points. Facet–edge elements correspond to darts, Clock corresponds to $\alpha_0\alpha_3$, Enext corresponds to $\alpha_0\alpha_1$, Fnext corresponds to $\alpha_3\alpha_2$ and Rev corresponds to α_3 (here the results of the comparison made by Brisson[12] between the cell-tuple structure and the facet–edge data structure, are used). More precisely, the four facet–edge elements of a face–edge pair correspond to the four darts of an orbit of $\langle\alpha_0, \alpha_3\rangle$ (four because $\alpha_0\alpha_3$ is without fixed points). It should also be observed that no edge is bent back on itself in the three-dimensional cells ($\alpha_0\alpha_2$ is without fixed points) and that no face is bent back on itself ($\alpha_0\alpha_3$ and $\alpha_1\alpha_3$ are without fixed points). Finally, 3-G-maps *without boundaries* correspond to facet–edge structures (α_3 is without fixed points).

The complete facet–edge data structure defines simultaneously the topologies of the primal and dual subdivisions. Each facet–edge element of the primal corresponds to a facet–edge element of the dual, by an element function Sdual, and vice versa. It is possible to define an extension of the notion of the 3-G-map, strictly equivalent to the concept of the facet–edge data structure, in the same way as used for the quad-edge data structure (i.e. a structure can be defined which represents the 'union' of a 3-G-map and its dual 3-G-map: see above).

Dobkin and Laszlo[10] have shown that the facet–edge data structure can represent the topology of orientable subdivisions of Euclidean three-space, made up by three-dimensional cells homeomorphic to spheres. As said above, the converse problem still remains: given any facet–edge structure which satisfies the combinatorial properties presented by Dobkin and Laszlo[10], does a subdivision of a 3D topological space exist, such that its topology is represented by the facet–edge structure? As for pavements, facet–edge data structures may possibly be applied to the representation of the topology of subdivisions of Euclidean three-space, but also to the representation of the topology of subdivisions of other 3D topological spaces.

Models for representing the topology of subdivisions of *n*-dimensional topological spaces

Only the notion of the n-map (cf. the previous section) appears to have been defined for modelling the topology of subdivisions of oriented nD spaces. So this section compares cell-tuple structures and n-G-maps, which have been defined for modelling the topology of subdivisions of orientable or non-orientable nD topological spaces.

In the previous section the notion of the n-G-map was presented constructively, i.e. n-G-maps were built by putting together nD cells, defined by $(n-1)$-G-maps without boundaries (see also Lienhardt[11,39]).

Brisson[12,35] presents the cell-tuple structure descriptively in the following way: let M be an nD manifold, and let C be a finite collection of open kD cells ($0 \leqslant k \leqslant n$), whose union is M (the cells are homeomorphic to open kD spheres). Informally, the pair (M, C) defines a subdivided nD manifold if the boundary of each kD cell of C does not self-intersect, and if it is equal to the union of cells of C, which have dimension lower than k (more formally, the class of objects considered is the finite, regular CW-complexes[63]).

Let c_1 and c_2 be two cells of $C \cdot c_1$ is 'a face' of c_2 if c_1 is contained in the boundary of c_2. Moreover, if c_1 is iD (note that $\dim(c_1) = i$), and if $\dim(c_2) = i+1$, then c_1 and c_2 are *incident* (for instance, a face contained in the boundary of a three-dimensional cell is incident to this 3D cell). A cell-tuple is defined as an $(n+1)$-tuple $(c_0, c_1, \ldots, c_n)$, such that for each i between 0 and $n-1$, $\dim(c_{i+1}) = \dim(c_i) + 1$, and c_i and c_{i+1} are incident (see Figure 14). Let T_M be the set of all cell-tuples defined by the cells of C.

The *switch operator* is defined in the following way. Let c_{k-1}, c_k and c_{k+1} be three cells such that $\dim(c_{k+1}) = \dim(c_k) + 1 = \dim(c_{k-1}) + 2$, and such that c_{k-1} and c_k, c_k and c_{k+1} are incident (for instance, in Figure 14, a cell-tuple of type (vertex, edge, face)). A single cell $c_{k'}$ exists such that $c_{k'} \neq c_k$, $\dim(c_{k'}) = \dim(c_k)$, and c_{k-1} and $c_{k'}$, $c_{k'}$ and c_{k+1} are incident. Then $\mathrm{switch}(c_{k-1}, c_k, c_{k+1}) = c_{k'}$ (for instance, in Figure 14, edge 1 and edge 2 are the two edges which are incident to both vertex a and face A, so switch$(a, 1, A) = (a, 2, A)$). Let $t = (c_0, \ldots, c_{k-1}, c_k, c_{k+1}, \ldots, c_n)$ be an element of T_M. The general *switch*$_k$ *operator* is then defined on T_M, for each k, $0 \leqslant k \leqslant n$, by $\mathrm{switch}_k(t) = (c_0, \ldots, c_{k-1}, c_{k'}, c_{k+1}, \ldots, c_n)$, with $c_{k'} = \mathrm{switch}(c_{k-1}, c_k, c_{k+1})$: see Figure 14. For notational convenience, Brisson[12] assumes the existence of two cells c_{-1} and c_{n+1}, with $\dim(c_{-1}) = -1$ and $\dim(c_{n+1}) = n+1$.

The cell-tuple structure is defined as the pair $(T_M, \{\mathrm{switch}_k\})$, with $0 \leqslant k \leqslant n$. Brisson[12,35] deduces some properties from the definition of switch$_k$. In particular,

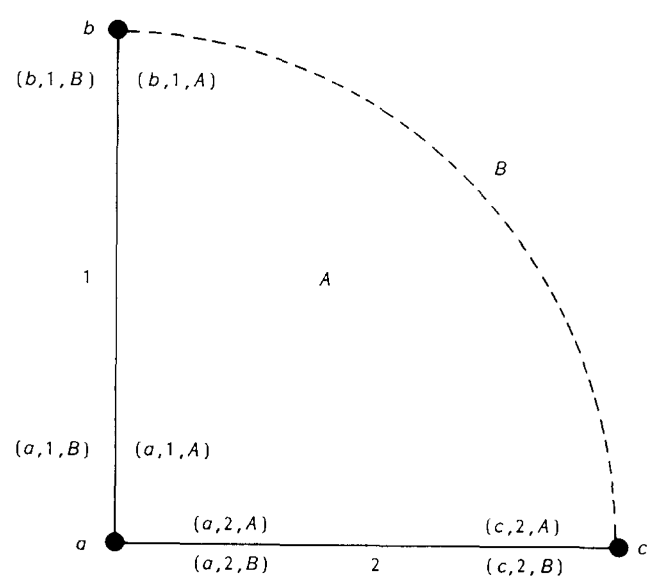


Figure 14. Only part of subdivision of surface is represented: three vertices a, b *and* c*; two edges 1 and 2; two faces A and B. Cell-tuples are 3-tuples of form (vertex, edge, face). Concerning the switch$_k$ operator, for instance, switch$_0$(*a, *1, A) = (*b, *1, A); switch$_1$(*a, *1, A) =* a, *2, A); switch$_2$(*a, *1, A) = (*a, *1,* B)

the switch_k operator is its own inverse (i.e. an involution), and does not have fixed points, for each k between 0 and n; the $\text{switch}_{k'}\text{switch}_k$ operator is also its own inverse, and does not have fixed points, for each k, k' between 0 and n, such that $k' < k - 1$. The implicit notions of vertex, edge, face, etc., are also defined by Brisson[12,35].

It should be evident that the definitions of cell-tuples and switch_k operator provide the basis for an algorithm for converting incidence graphs (as defined above) into an ordered topological model (i.e. a cell-tuple structure). But the definition of cell-tuples itself involves restrictions on the possible field of application of such an algorithm (see below).

The concept of 'ordering', formally defined by Brisson[12], should be mentioned briefly here. Given a $(k-2)$-dimensional cell contained in the boundary of a $(k+1)$-dimensional cell, all of the cells 'between' them may be put into a circular order 'around' the $(k-2)$-dimensional cell, such that this order alternates between $(k-1)$-dimensional cells and k-dimensional cells. For instance, in a polyhedron, circular sequences of edges and faces around vertices can be defined. Put simply, this notion of ordering corresponds to the fact that the composition $\text{switch}_{k-1}\text{switch}_k$ is a permutation, for each k $(1 \leqslant k \leqslant n)$, as it is a composition of two involutions.

It can be shown that the cell-tuple structure is equivalent to the notion of n-G-map without boundaries (α_n is without fixed points), such that, for each dart b, no dart b' exists, such that, for each i between 0 and n, b and b' are incident to the same iD cell. This condition involves, in particular, that $\alpha_i\alpha_j$ is without fixed points, for $0 \leqslant i < i+2 \leqslant j \leqslant n$ (cf. Lienhardt[64]). The condition means that the intersection of all iD cells c_i $(0 \leqslant i \leqslant n)$ incident to dart b is equal to $\{b\}$ (if the condition is not satisfied, at least two darts correspond to an identical cell-tuple). If the condition is satisfied, it is possible to define a unique cell-tuple for each dart of such an n-G-map, and vice versa. Thus, for each cell-tuple structure, it is possible to define an equivalent n-G-map. Conversely, it is possible to define an equivalent cell-tuple structure for each n-G-map without boundaries which satisfies the condition.

Concerning this condition, Brisson says that cell-tuple structures can not represent the topology of subdivisions such that the boundary of a cell self-intersects. In fact, this is not completely well defined. For instance, in two dimensions, a subdivision of a surface with a face which intersects itself only at a vertex may be represented by a cell-tuple structure.

More formally, it can be proved than an n-G-map $(B, \alpha_0, \alpha_1, \ldots, \alpha_n)$ does not satisfy the condition if and only if i exists $(0 \leqslant i \leqslant n-1)$ and a dart b of B exists, such that $\langle \alpha_0, \ldots, \alpha_i \rangle(b) \cap \langle \alpha_{i+1}, \ldots, \alpha_n \rangle(b) \neq \{b\}$[64]. For instance, in two dimensions $\langle \alpha_0, \alpha_1 \rangle(b) \cap \langle \alpha_2 \rangle(b) \neq \{b\}$ means that a face intersects itself along an edge.

This restriction of cell-tuple structures is mainly a consequence of the definition of the basic elements of the model, i.e. cell-tuples. But the notion of the cell-tuple is probably the best topological interpretation of the notion of the dart (as defined for any dimension). It could also be possible to extend this interpretation, in order to avoid restrictions such as the condition just mentioned (cf. below).

Finally Brisson[12] restricts the field of application of cell-tuple structures to subdivisions, such that the cells of these subdivisions are homeomorphic to spheres. But the whole set of subdivisions of topological spaces whose topology can be represented by cell-tuple structures would appear to be incompletely defined (cf. the above remark concerning the field of application of pavements and facet–edge data structures).

An extension of an ordered topological model: the radial–edge data structure

This subsection is confined to a discussion of the 'radial–edge' data structure as presented by Weiler[11]. This structure makes it possible to model the topology of (two-dimensional) 'manifold' objects, and of 'non-manifold' objects[33] (cf. also the presentation of selective geometric complexes[37] above). A non-manifold object is a three-dimensional geometric object, whose cells satisfy at least one of the following 'non-manifold conditions' (cf. Weiler[33]; see also his definition of separation surfaces):

- more than two faces are incident to the same edge;
- volumes (or faces) are adjacent through sharing a single vertex;
- dangling edges exist (i.e. edges which are not adjacent to a face).

The notion of volume does not explicitly appear in the radial–edge data structure, but this structure allows the topology of subdivisions of oriented three-dimensional topological spaces to be modelled (in particular, some of these subdivisions satisfy the first non-manifold condition above). In fact, a basic notion of the radial–edge data structure is the notion of 'edge use', which corresponds intuitively to the notion of the face–edge pair[10].

The radial–edge data structure is mainly an ordered topological model, defined for the representation of the topology of subdivisions of oriented three-dimensional spaces. It differs from the topological models presented above in particular by virtue of the fact that it is possible to model objects of 'mixed dimensionality' (for instance a polyhedron with dangling faces or dangling edges) and that the boundary of a cell may be unconnected (for instance, the boundary of a face may be defined by more than one cycle of edges). For these reasons, the radial–edge data structure is considered an extension of a three-dimensional ordered topological model.

The principle of the radial–edge data structure is as follows. Let (f, e) be a face–edge pair (i.e. face f is incident to edge e). Two edge-use elements are associated with (f, e), corresponding to the two possible ways of defining an orientation of the face–edge pair within the face (edge-use elements correspond to darts in 3-maps). With each edge-use element is associated the inversely oriented edge-use element, by a pointer (corresponding, for 3-maps, to the involution α_0, where α_0 is without fixed points). The edge-use elements are organized into cycles, defining the oriented faces (corresponding, for 3-maps, to the orbits of the permutation $\alpha_1^{-1}\alpha_0$). Finally, the cycles of faces around the edges are described by pointers, which gather the edge-use elements (corresponding, for 3-maps, to the involution $\alpha_0\alpha_2$, where $\alpha_0\alpha_2$ is without fixed points: cf. Figure 15). Thus the radial-edge data structure can be considered basically an ordered topological model (Weiler[33] makes no mention of constraints on the different pointers; so the correspondence between 3-maps and radial-edge data structures is, in fact, a personal interpretation).

This model is extended in order to take into account the following ideas:

- Inhomogeneity of the dimension: the ordered topological models, studied above, define a subdivision of an nD space by 'gathering' nD cells; each nD cell is defined by its boundary, i.e. by a subdivision of an $(n-1)$-dimensional space. However, the radial-edge data structure allows the topology of three-dimensional objects to be modelled, made up by 'gathering' cells, with dimensions less than or equal to 3 (for instance, gathering volumes and dangling edges – see the third non-manifold condition above). Moreover, the boundary of an iD cell may be a subdivision of a space with dimension less than $i-1$ (for instance, the boundary of a face may be reduced to a vertex).
- Disconnected cells: it was shown in the previous section that $n+1$ $(n-1)$-G-maps of the elements $(G_i)_{i=0,\dots,n}$ can be deduced from an n-G-map G; for each i between 0 and n, each connected component of G_i defines an iD cell of G (and similarly for n-maps). This formalism does not allow geometric objects, where an iD cell is defined by more than one connected component, to be modelled directly. For instance, the topology of two tetrahedra, adjacent through sharing a single vertex, cannot be directly modelled by a 3-G-map (this applies to all ordered topological models studied above: however, see Arquès and Koch[61]). The radial-edge data structure makes it possible, for instance, to model faces defined by a set of cycles of edges (multiply-connected faces), one of them defining the 'external' boundary of the face, the others defining 'internal' boundaries, or to model geometric objects (volumes, faces, edges) which are adjacent through a single vertex (cf. the second non-manifold condition above).

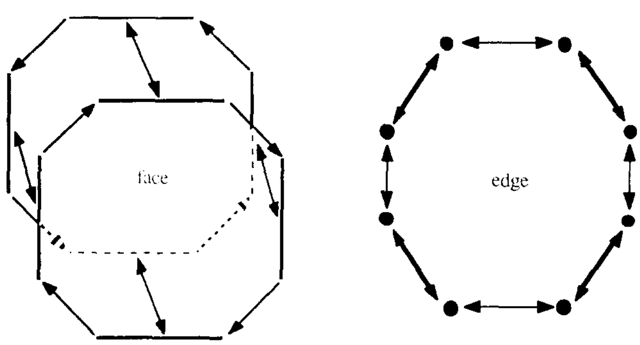


Figure 15. (a) face, defined by two cycles of edge-use elements (thick lines); arrows correspond to pointers that define such cycles (corresponding to 3-map to permutation $\alpha_1^{-1}\alpha_0$); double arrows correspond to pointers joining two corresponding inversely oriented edge-use elements (corresponding to 3-maps to involution α_0) (b) Cross-section of an edge: points correspond to edge-use elements; thin double arrows correspond to pointers gathering two corresponding inversely oriented edge-use elements; thick double arrows correspond to the edge-use radial pointers, which gather edge-use elements that are directly adjacent, turning around the edge (corresponding to 3-maps to involution $\alpha_0\alpha_2$)

This extension of an ordered topological model is achieved in the radial-edge data structure by using several basic elements ('vertex use', 'loop use', 'face use', etc.: cf. Weiler[33]), and by representing (in a more or less direct fashion) the different adjacency and incidence relations between cells. The radial-edge data structure is thus complex, and its extension to higher dimensions is not obvious. Moreover, it is quite redundant when two- and three-dimensional 'manifold' objects are represented. Nevertheless, this data structure is a very interesting approach for modelling the topology of three-dimensional CW-complexes (see Munkres'[63] definition).

Unsolved problems

Ordered topological models are based on similar ideas – definition using a single type of basic elements, on which element functions act. This simplifies the definition of data structures deduced from these models, the definition of consistency constraints that these data structures must satisfy, and also the definition, processing and control of operators which are applied to these models (cf. the following section).

Moreover, important topological properties (orientability, duality, number of boundaries, and topological characteristics for dimension 2 – Euler characteristic, orientability factor, genus) can be directly computed on ordered topological models. However, the definition of constraints of consistency for data structures deduced from incidence graphs seems more critical, and no study appears to have been made of this subject.

A problem, studied by many authors[2,6,65], is the following. Three types of cell exist in dimension 2 (vertex, edge, face), and so there are nine types of adjacency and incidence relation between these cells. What are the sets of relations which are complete, i.e. provide all topologic information? Among these sets, what are the minimal sets (i.e. those which are not redundant)?

Generalizing the problem to dimension n consists in determining such relations sets among $(n+1)^2$ types of relation (these questions are motivated by the following: proof of the sufficiency of data structures; definition of complete data structures; and limitation of redundancy). Brisson[15] has proved the sufficiency of the set of incidence relations between i-dimensional and $(i+1)$-dimensional cells, but the general problem remains. It should be observed that this problem does not directly exist for ordered topological models, the cells being implicitly represented. Nevertheless, no solution has been provided to the problem concerning the degree of necessary redundancy which must exist in a data structure in order to ensure good performances for basic algorithms (for instance, computing the number of connected components, orientability, etc.).

It has been shown that ordered topological models are equivalent (with respect to dimension and orientability), except for some differences concerning particularly fixed points of the element functions. It is not claimed here than a single model exists, that some model is better than any other, or that it is possible to define a universal data structure (obviously, many data structures can be deduced from a topological model, more or less adequate according to their uses). But it is important to highlight these equivalences, which may simplify the understanding of these models. In particular, some models have been deduced from descriptive approaches, others from constructive approaches, and it is remarkable that different approaches produce equivalent models.

Many problems concerning ordered topological models have yet to be solved. We have seen that the notion of dart has many topological interpretations: half-edge[6], oriented edge[45], directed and oriented edge[7], oriented angular sector[9], directed and oriented face–edge pair[10], cell-tuple[12]. Only the notion of cell-tuple is defined in the general case of n-dimensions, but this notion involves restrictions on the class of subdivisions of topological spaces whose topology may be represented by cell-tuple structures. Another possible topological interpretation of the notion of dart is suggested by Brisson[35] (see also Vince[46]): let M be a subdivision of an nD topological space, and let T be the *barycentric triangulation* of M. Informally, T is a particular decomposition of M into n-dimensional simplices, such that the topological spaces subjacent to T and M are homeomorphic (cf. Figure 16; for a more complete definition, see Agoston[57] and Munkres[63]). A dart corresponds to each nD simplex in T, and vice versa. This interpretation of the notion of dart is equivalent to the cell-tuple concept if self-intersections of the boundaries of cells are not allowed[35]. However, this interpretation does not appear to present the restrictions involved by the definition of cell-tuples, though no proof of this is known to exist as yet; nevertheless, see the examples in Figure 16.

Another problem concerns *fixed points* of element functions. For instance, fixed points are not allowed for any switch_k operator in a cell-tuple structure[12] $(0 \leqslant k \leqslant n)$, nor for any $\text{switch}_k\ \text{switch}_{k'}$ operator (with $k \neq k'$). Quad-edge structures[7] are equivalent to 2-G-maps $(B, \alpha_0, \alpha_1, \alpha_2)$, where α_2 and $\alpha_0\alpha_2$ are without fixed points (there are similar restrictions for facet–edge structures). In particular, these restrictions on fixed points involve problems for the representation of the topology of subdivisions of topological spaces *with* boundaries. Similarly, n-G-maps are defined as $(n+2)$-tuples $(B, \alpha_0, \alpha_1, \ldots, \alpha_n)$, where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ are without fixed points. This definition involves the fact that duality can only be defined for n-G-maps without boundaries (and the definition of n-G-maps is inhomogenous ...). On the other hand, the map concept by Bryant and Singerman[30] allows fixed points for any involution, and these fixed points have a coherent topological interpretation. It could be useful to study the generalization of this approach to the general case of n dimensions (i.e. remove the condition on fixed points for n-G-maps), and to study the topological interpretation of these fixed points.

It has been proved that the topology of subdivisions of topological spaces can be represented by such models (generally, subdivisions composed by cells which are homeomorphic to spheres), and that the representation is unique, up to isomorphism[9,10,12]. But the class of all subdivisions of all topological spaces, such that the topology of these subdivisions may be represented by such ordered models, is still incompletely defined.

The evolution of topological models has notably extended the class of objects which can be modelled in boundary representation (thus answering remarks made by Takala[22] concerning the restricted field of application of boundary representation). Nevertheless, some classes of object (for instance, subdivisions where cells are multiply-connected) cannot be directly represented by these topological models. It should be possible to extend to the general case of n dimensions, some of the well-known methods used in boundary representation (the distinction between 'active' and 'non-active' cells[37]; inclusion trees: cf. Weiler[6,33], Michelucci and Gangnet[48] and Dufourd *et al*[58]), but there is no known general study of this subject. Finally, it may be very useful to extend (if possible) ordered topological models in order to represent the topology of CW-complexes[63].

EVOLUTION OF OTHER ASPECTS OF BOUNDARY REPRESENTATION

This section takes a brief look at some of the recent developments in boundary representation concerning embedding models and operators.

Embedding models

Embedding models associated with topological models have evolved during the last years. For instance, to a topological edge may correspond a straight line[5], a polyline[58], a discrete curve consisting of a sequence of pixels[49], a free-form curve[50,66] ... To a topological face may correspond a planar face[5], cylindric faces, or parametric surfaces[4,13], faces defined by quadrics[40], free-form surfaces (which give important local control[18]: see, for instance, Bartels *et al*[67] and Mortenson[68]).

Nevertheless, fundamental problems remain. We have seen that the whole set of subdivisions of topological spaces whose topology may be associated with ordered topological models is still incompletely defined for dimensions higher than 3. This points to the problem of the relation between topological models and subdivisions of topological spaces, and thus the

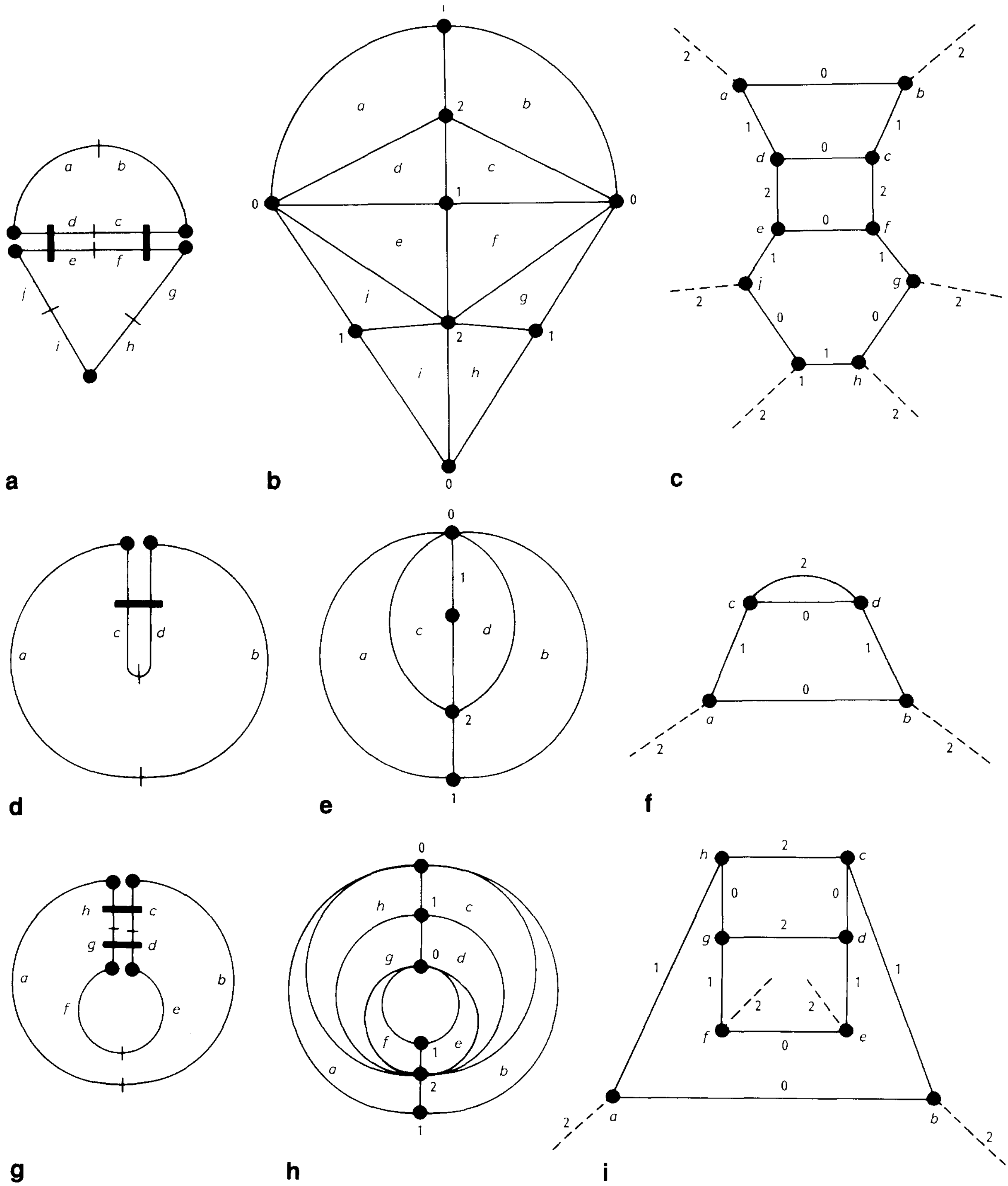


Figure 16. (a) 2-G-map G. (b) barycentric triangulation T of a subdivision S whose topology is represented by G: vertex labelled i in T is associated with each i-dimensional cell of S, and vice versa. Edge exists in T between two vertices labelled i and j if and only if $i \neq j$ and the corresponding i-dimensional and j-dimensional cells are incident in S. The faces of T (which are triangles) are defined in a similar manner (for more details, see Brisson[35]). (c) corresponding labelled graph L, deduced from T in the following manner: each node in l corresponds to a triangle of T, and vice versa. Edge labelled i joins two nodes if and only if corresponding triangles in T are adjacent along edge whose extremities are labelled j and k, with $i \neq j$ and $i \neq k$. Finally, we can deduce 2-G-map G from labelled graph L. Each dart of G corresponds to a node of L (and vice versa); two darts belongs to the same orbit of α_i if and only if the corresponding nodes are joined in L by an edge labelled i. (d)2-G-map G; (e) corresponding barycentric triangulation and (f) labelled graph. Notice that an edge is bent back on itself in G, i.e. $\alpha_0\alpha_2$ has fixed points, meaning that edge of subdivision is only incident to vertex. Informal definition of barycentric triangulation presented can be extended to take into account such 'degeneracies' (g) 2-G-map, where dart b exists, such that $\langle \alpha_0, \alpha_1 \rangle(b) \cap \langle \alpha_2 \rangle(b) \neq \{b\}$, and (h) its corresponding barycentric triangulation and (i) labelled graph

problem of the *geometric realization* of such topological models, i.e. the problem of *embedding*. It is clear that the definition of embedding models associated with ordered topological models is a real problem if the nature of the subdivisions of spaces which can be associated to these models is not known. However, different types of embedding models can be defined (as was described at the beginning of this paper). For instance, for 2-G-maps, a possible embedding consists in associating a point in Euclidean three-space to each orbit of $\langle \alpha_1 \rangle$, a segment to each orbit of $\langle \alpha_0 \rangle$ and a polygon to each orbit of $\langle \alpha_0, \alpha_1 \rangle$. Thus, *two* segments are associated with each edge of the 2-G-map, and more than one point is associated to each vertex of the 2-G-map (see, for instance, Figure 3). This is a *discontinuous* embedding. Another possible embedding consists in associating a point with each orbit of $\langle \alpha_1, \alpha_2 \rangle$ (i.e. to each vertex), a segment with each orbit of $\langle \alpha_0, \alpha_2 \rangle$ (i.e. to each edge) and a polygon with each orbit of $\langle \alpha_0, \alpha_1 \rangle$ (i.e. to each face). This defines a *continuous* embedding.

This points to problems concerning constraints on embedding models. For instance, what constraints must a discontinuous embedding model satisfy in order to define a continuous embedding? Some problems concerning constraints on embedding are well known (for instance, for the definition of solids[1]), although many algorithms have been studied, in particular for the control of self-intersections (see Casale[69], for example). As topological models and embedding models are independent, more than one embedding model may be associated with a same topological model, in a same geometric modeller. Other basic problems concerning embedding models have not yet been solved, but space does not permit a detailed discussion of these.

Operators

Many criteria may be defined when classifying the operators used in boundary representation. Two of the most important are the nature of the operators (topological, embedding or mixed); and the level (basic operator, high-level operator) and the degree of hierarchization of the operators.

Topological operators

Euler operators have been studied by Baumgart[15], and extended by many authors[14,16,17,19]. They are basic operators, used for handling topological models – for instance, creating a connected component, a face and a vertex, or an edge and a face, or an edge and a vertex, and their inverse operators (when modelling subdivisions of orientable surfaces without boundaries, the use of these operators agrees with the Euler formula for genus). This type of operator has been extended by Weiler[70] for modelling non-manifold objects (*sensu* Weiler[33]).

Other basic topological operators have been defined for handling ordered topological models. For instance, operators for 'inserting' and 'tying' darts are defined for handling 2-maps[71], and extended to *n*-G-maps[42]; 'make-edge' and 'splice' operators are defined for handling quad-edge data structures[illegible]; 'make-edge-facet', 'splice-edge' and 'splice-facet' operators are defined for handling facet-edge data structures[illegible]; a 'merging' operator is defined for handling pavings[illegible]; 'sewing' operators are defined for handling *n*-G-maps[11], etc.

A main criticism of boundary representation concerns the difficulty of creating and manipulating geometric objects[1,22,41,72]. Euler operators being low-level operators, numerous and non-standardized. For instance, basic operators used for constructive solid geometry (CSG), i.e. set operations (union, intersection, difference), are high-level operations in boundary representation (the results of operations are computed in boundary representation but not in CSG). Moreover, proving that a set of Euler operators is complete is not very easy[73]. Finally, embedding information is sometimes needed for a complete characterization of an operator[14,22].

A major reason for these drawbacks is the following: classical Euler operators (see Braid *et al*[14], Mäntylä and Sulonen[16] and Ansaldi *et al*[17], for instance) handle different entities (vertex, edge, face, connected component, etc.). Control over the validity of the model is exerted through Euler operators: thus, the different entities must be manipulated simultaneously. Particularly, it is the case for operators defined for handling incidence graphs (cf. the remark in the previous section concerning consistency constraints on incidence graphs). This implies the definition of numerous basic operators, and difficulties when proving the completeness of these operators. Finally, it must be observed that no basic operators have been defined for manipulating incidence graphs in the general case of dimension *n* (however, see the basic operators defined by Weiler[70], in the case of dimension 3, and operators presented by Rossignac and O'Connor[57]).

On the other hand, basic operators defined for handling ordered topological models do not present such inconvenience[7,8,10,11,42,71]. They handle a single type of element (the basic element of the topological model); a small number (generally two or three, with inverse operators included) provides a complete set of operators; the proof of their completeness is (generally) easy, and they can easily be extended to higher dimensions[11,42]. These operators still remain non-standardized. Among these operators, some[7,8,10,29] seem to be higher-level operators than others[11,42,71]. It could be of great interest to study how to define these higher-level operators by the lower-level operators.

An example of a basic operator is the 'sewing' operator (presented above), defined for the manipulation of *n*-G-maps. Schematically, this basic operator consists in putting together *n*-dimensional cells along $(n-1)$-dimensional cells. It is easy to prove that this is sufficient for the construction of any *n*-G-map. However, it is possible to distinguish different cases of application of this operation, for instance in two dimensions, according to the variations of the characteristics associated with a 2-G-map. This type of distinction is one of the reasons for the multiplicity of Euler operators. The 'sewing' operator is very simple to define and to apply. A major reason for this advantage also constitutes a major drawback: this operator can only be applied to

n-G-maps *with boundaries*. One of the reasons for the complexity of some operators is the fact that they can only be applied to topological models which represent the topology of subdivisions of topological spaces *without boundaries* (see, for instance, merging[8], splice[7], splice-edge and splice-facet[10]).

Finally, it is clear that the use of low-level operators is tedious; moreover, it is sometimes hard to use low-level operators efficiently for creating or manipulating geometric objects. But this problem is not a characteristic of boundary representation (on the definition of high-level operators, see Braid[13], Braid *et al*[14], Varady and Pratt[18]). However, the definition of a basic set of low-level operators, used for defining high-level operators, is an important step which should not be omitted (cf. Dufourd[42,71], for instance).

Embedding and mixed operators

Numerous embedding and mixed operators have been defined – for instance, bending, tweaking chamfering, twisting[14,18,74]. Some operators are defined for a particular type of embedding model (for instance, operators for manipulating free-form surfaces) (cf. Barstels *et al*[67] and Barsky[75]; many works define 'topological' operators to handle free-form surfaces[76,77,78]), while other operators are more general (Boolean operations[20,23–26], deformation operations[79,80], etc.). See, in particular, the operators defined by Rossignac and O'Connor[37] for the manipulation of selective geometric complexes (SGCs) (cf. the previous section):

- *subdivision*, which makes two geometric complexes *compatible* with each other by *refining* them, i.e. by subdividing their cells (more precisely, two geometric complexes, A and B, are compatible if for each cell a of A and for each cell b of B, $a \cap b \neq \emptyset \Rightarrow a = b$);
- *selection*, which selects cells of one or more compatible geometric complexes;
- *simplification*, which, by deleting or merging cells produces a simpler SGC.

These operators have been extensively studied by many authors, and it is not intended to discuss their advantages and drawbacks here. It is important to retain consistency with boundary representation logic, i.e. to distinguish between embedding operators and mixed operators. This means that basic mixed operators should be expressed by using topological operators and embedding operators (mixed operators being the highest-level operators in a geometric modeller). As said above, the department of Computer Science at Strasbourg is engaged in developing software for modelling subdivisions of surfaces and subdivisions of three-dimensional spaces, in which this hierarchization is maintained.

CONCLUSION

Boundary representation has evolved considerably in recent years. Such methods are now used not only in CAD but also, for instance, in computational geometry. This is one of the main reasons why topological models used in boundary representation have changed, and the set of objects which can be modelled by boundary representation methods has been considerably extended from subdivisions of orientable surfaces without boundaries, to subdivisions of nD topological spaces, orientable or non-orientable, with or without boundaries.

Schematically, two main classes of topological model used in boundary representation may be distinguished. The first is incidence graphs[17,35–38], defined using $(n+1)$ distinct basic elements (the $n+1$ types of cell, in dimension n). These models are sufficient for modelling the topology of subdivisions, but their constraints of consistency seem difficult to define. Control over consistency is exerted through operators which are applied to these models (or, by constructing a particular model, when given a mathematical definition of the subdivision). Moreover, manipulating simultaneously $n+1$ basic entities involves the definition of numerous basic operators, even in the two-dimensional case. Such operators seem to be difficult to define in the general case of n dimensions (no thorough study about this problem is known). Finally, the computation of some important topological properties (orientability, for instance) directly on a data structure deduced from an incidence graph is not easy.

The second class of topological model is ordered topological models[5–12,29,30], which use a single type of basic elements, on which element functions act. Data structures can be easily deduced from these models, and the consistency constraints on these data structures can be directly deduced from the definition of the models. Control is not only exerted through operators but also directly from the definition itself of the model. Numerous topological properties can be directly computed using the model, or using a data structure deduced from the model (classification in dimension 2; in the general case of dimension n, number of boundaries, orientability, duality, etc.). For each model, a small set of basic operators is enough to create and manipulate the model; this simplifies the definition, processing, control and hierarchization of higher-level operators. Finally, it seems important to highlight the fact that order models are based on the same ideas, and that it is possible to show that these models are equivalent (with respect to dimension and orientability). No ordered topological model is known to have been defined for modelling the topology of general CW-complexes (see the approach presented by Weiler[33] for the particular case of dimension 3).

The evolution of boundary representation has also involved embedding models. Topological models (initially embedded using planar faces and straight edges, for modelling subdivisions of orientable surfaces without boundaries) are associated with embedding models using parametric surfaces, quadrics, and free-form surfaces, and often with more than one single embedding model (this is an advantage of boundary representation[6,18]). But the control of the embedding is still an unsolved problem, to which new approaches are being studied (cf. Bertrand[27], for instance).

Concerning operators (topological, embedding and mixed), it is important carefully to define basic operators, and a hierarchization of higher-level operators based upon these basic operators – see, for instance, Dufourd[42], who proposes an algebraic functional specification of n-G-maps and n-maps, presents several implementations, in particular one by rapid prototyping using logic programming; and studies morphological aspects, i.e. embedding and photometric aspects, with two-dimensional examples. It is also important to maintain boundary representation logic, i.e. to define mixed operators as high-level operators, expressed by using topological operators and embedding operators.

ACKNOWLEDGEMENTS

I wish to thank the members of the Strasbourg 3D Modelling Group, in particular J. Françon and J.-F. Dufourd, for their encouragement and assistance. Thanks to D. Viaud, for reviewing the English version of this paper. Special thanks to the reviewers of the paper, for many helpful remarks.

This paper has been completed with financial support from the PRC Mathématique et Informatique.

REFERENCES

1 **Requicha, A** 'Representations for rigid solids: theory, methods and systems' *Computing Surveys* Vol 12 No 4 (1980) pp 437–464

2 **Baer, A, Eastman, C and Henrion, M** 'Geometric modelling: a survey' *Comput.-Aided Des.* Vol 11 No 5 (September 1979) pp 253–272

3 **Weiler, K** 'Topology as a framework for solid modeling' *Proc. Graphics Interface '84* Ottawa, Canada (May 1984) pp 53–56

4 **Hillyard, R** 'The BUILD group of solid modelers' *Comput. Graph. Appl.* Vol 2 No 2 (March 1982) pp 43–52

5 **Baumgart, B** 'A polyhedron representation for computer vision' *Proc. AFIPS Nat. Conf.* 44 (1975) pp 589–596

6 **Weiler, K** 'Edge-based data structures for solid modeling in curved-surface environments' *Comput. Graph. Appl.* Vol 5 No 1 (1985) pp 21–40

7 **Guibas, L and Stolfi, J** 'Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams' *ACM Trans. Graph.* Vol 4 No 2 (1985) pp 74–123

8 **Spehner, J-C** 'Merging in maps and pavings', Research Report no 48, Laboratoire de Mathématique et Informatique, Université de Haute-Alsace, Mulhouse, France (to appear in *Theoretical Comput. Sci.*)

9 **Arquès, D and Koch, P** 'Modélisation de solides par les pavages' *Proc. Pixim '89* Paris, France (September 1989) pp 47–61

10 **Dobkin, D and Laszlo, M** 'Primitives for the manipulation of three-dimensional subdivisions' *Proc. 3rd Symp. Computational Geometry*, Waterloo, Ontario, Canada, ACM (June 1987) pp 86–99

11 **Lienhardt, P** 'Subdivisions of n-dimensional spaces and n-dimensional generalized maps' *Proc. 5th ACM Symp. Computational Geometry*, Saarbrücken Germany (June 1989) pp 228–236

12 **Brisson, E** 'Representing geometric structures in d dimensions: topology and order' *Proc. 5th ACM Symp. Computational Geometry*, Saarbrücken, Germany (June 1989) pp 218–227

13 **Braid, I** 'The synthesis of solids bounded by many faces' *Commun. ACM* Vol 18 No 4 (April 1975) pp 209–216

14 **Braid, I, Hillyard, R and Stroud, I** 'Stepwise construction of polyhedra in geometric modelling' in *Mathematical Methods in Computer Graphics and Design*, Academic Press, London, UK (1980) pp 123–141

15 **Baumgart, B** 'Geometric modeling for computer vision', Memo AIM-249, Stanford Artificial Intelligence Laboratory, Stanford University, CA, USA (October 1974)

16 **Mäntylä, M and Sulonen, R** 'GWB: a solid modeler with Euler operators' *Comput. Graph. Appl.* Vol 2 No 7 (1982) pp 17–31

17 **Ansaldi, S, De Floriani, L and Falcidieno, B** 'Geometric modeling of solid objects by using a face adjacency graph representation' *Comput. Graph.* Vol 19 No 3 (1985) pp 131–139

18 **Varady, T and Pratt, M** 'Design techniques for the definition of solid objects with free-form geometry' *Comput. Aided Geometric Des.* Vol 1 (1984) pp 207–225

19 **Mäntylä, M** 'Computational topology: a study of topological manipulations and interrogations in computer graphics and geometric modeling' *Acta Polytechnica Scandinavia* No 37 (1983)

20 **Mäntylä, M** 'Boolean operations of 2-manifolds through vertex neighbourhood classification' *Trans. Graph.* Vol 5 No 1 (1986) pp 1–29

21 **Lienhardt, P** 'Free-form surfaces modeling by evolution simulation' *Proc. Eurographics '88* Nice, France (September 1988) pp 327–341

22 **Takala, T** 'Geometric boundary modeling without topological data structures' *Proc. Eurographics '86* Lisbon, Portugal (August 1986) pp 115–128

23 **Segal, M and Sequin, C** 'Partitioning polyhedral objects into nonintersecting parts' *Comput. Graph. Appl.* Vol 8 No 1 (January 1988) pp 53–65

24 **Laidlaw, D and Hugues, J** 'Constructive solid geometry for polyhedral objects' *Comput. Graph.* Vol 20 No 4 (August 1986) pp 161–170

25 **Bieri, H and Nef, W** 'Elementary set operations with d-dimensional polyhedra' *Proc. CG '88* Würzburg, Germany (March 1988) (in *Lecture Notes in Computer Science* No 333)

26 **Putnam, L and Subrahmanyam, P** 'Boolean operations on n-dimensional objects' *Comput. Graph. Appl.* Vol 6 No 6 (1986) pp 43–51

27 **Bertrand, Y** 'Spécification algéberique et prototypage logique d'un noyau de modeleur 3D basé sur les cartes' *Research Report no 90–06*, Département d'Informatique, Université Louis Pasteur, Strasbourg, France (March 1990)

28 **Miller, A** 'Architectural issues in solid modelers' *Comput. Graph. Appl.* Vol 9 No 5 (September 1989) pp 72–87

29 **Tutte, W** *Graph Theory*, Encyclopedia of Mathematics and its Applications, Vol 21, Cambridge University Press, USA (1984)

30 **Bryant, R and Singerman, D** 'Foundations of the theory of maps on surfaces with boundaries' *Quart. J. Math. Oxford* Vol 2 No 36 (1985) pp 17–41

31 **Chiyokura, H** *Solid Modelling with Designbase* Addison-Wesley (1988)

32 **Mäntylä, M** *An Introduction to Solid Modeling* Computer Science Press (1988)

33 **Weiler, K** 'The radial edge structure: a topological representation for non-manifold geometric boundary modeling' *Proc. IFIP WG 5.2 Working Conference* Rensselaerville, U.S.A. (May 1986). Also in *Geometric Modeling for CAD Applications*, Elsevier Science (1988) pp 3–36

34 **Lienhardt, P** 'Extension of the notion of map and subdivisions of a three-dimensional space' *Proc. 5th Symp. Theoretical Aspects of Cojputer Science* Bordeaux, France (February 1988) (in *Lecture Notes in Computer Science no 294* pp 301–311)

35 **Brisson, E** 'Representing geometric structures in *d* dimensions: topology and order' *Technical Report 88-11-07* Department of Computer Science, University of Washington, Seattle, USA (November 1988)

36 **Edelsbrunner, H** *Algorithms in Combinatorial Geometry* Springer-Verlag (1987)

37 **Rossignac, J and O'Connor, M** 'SGC: a dimension-independent model for pointsets with internal structures and incomplete boundaries' *Proc. 1988 IFIP/INSF Workshop on Geometric Modelling* Rensselaerville, USA (September 1988). Also in **Wozny, M, Turner, J and Preiss, K (Eds)** *Geometric Modeling for Product Engineering* North-Holland (1989)

38 **Sobhanpanah, C** 'Extension of a boundary representation technique for the description of *n*-dimensional polytopes' *Comput. and Graph.* Vol 13 No 1 (1989) pp 17–23

39 **Lienhardt, P** '*n*-dimensional generalized topological maps' *Research Report R89-10*, Département d'Informatique, Université Louis Pasteur, Strasbourg, France (December 1989)

40 **Connolly, M** 'An application of algebraic topology to solid modeling in molecular biology' *The Visual Computer* Vol 3 (1987) pp 72–81

41 **Requicha, A and Voelker, H** 'Solid modeling: a historical summary and contemporary assessment' *Comput. Graph. Appl.* Vol 2 No 2 (March 1982) pp 9–24

42 **Dufourd, J-F** 'Formal specification of topological subdivisions using hypermaps' to appear in *Comput.-Aided Des.* Vol 23 No 2 (1991)

43 **Edmonds, J** 'A combinatorial representation for polyhedral surfaces' *Notices Amer. Math. Soc.* Vol 7 (1960)

44 **Jacques, A** 'Constellations et graphes topologiques' *Colloque Math. Soc. Janos Bolyai* (1970) pp 657–672

45 **Cori, R** 'Un code pour les graphes planaires et ses applications' *Astérique* No 27 (1975)

46 **Vince, A** 'Combinatorial Maps' *J. Combinatorial Theory B* No 34 (1983) pp 1–21

47 **Vince, A** 'Regular combinatorial maps' *J. Combinatorial Theory B* No 35 (1983) pp 256–277

48 **Michelucci, D and Gangnet, M** 'Saisie de plans à partir de tracés à main levée' *Proc. Micad '84* Paris, France (February 1984) pp 96–110

49 **Braquelaire, J-P and Guitton, P** 'A model for image structuration' *Proc. Computer Graphics International '88* Geneva, Switzerland (May 1988)

50 **Baudelaire, P and Gangnet, M** 'Planar Maps: an Interaction Paradigm for Graphic Design' *Proc. CHI '89* (May 1989) pp 313–318

51 **Michelucci, D and Peroche, B** 'Représentation interactive d'arbres CSG' *Proc. Micad '87* Paris, France (February 1987) pp 535–548

52 **Berge, C** *Graphes et hypergraphes* Dunod, Paris, France (1970)

53 **Lienhardt, P** 'Subdivisions of surfaces and generalized maps' *Proc. Eurographics '89* Hamburg, Germany (September 1989) pp 439–452

54 **James, L** 'Maps and hypermaps: operations and symmetry', PhD thesis, Department of Mathematics, University of Southampton, UK (August 1985)

55 **Griffiths, H-B** *Surfaces* Cambridge University Press (1981)

56 **Lienhardt, P** 'Simplex *n*-G-maps and barycentric triangulation of *n*-G-maps' *Research Report R90–08* Département d'Informatique, Université Louis Pasteur, Strasbourg, France (April 1990). Also presented at the 2nd Canadian Conference on Computational Geometry, Ottawa, Canada (August 1990)

57 **Agoston, M** *Algebraic topology: a first course* Marcel Dekker, New York, USA (1976)

58 **Dufourd, J-F, Gross, C and Spehner, J-C** 'A digitization algorithm for the entry of planar maps' *Proc. Computer Graphics International '89* Leeds, UK (1989)

59 **Ansaldi, S and Falcidieno, B** 'Form feature representation and recognition in a hierarchical boundary model' *Proc. IFIP WG 5.2 Working Conference* Rensselaerville, USA (May 1986). Also in **Wozny, M J, McLaughlin, H W and Encarnacão, J L (Eds)** *Geometric Modeling for CAD Applications*, North-Holland (1988) pp 137–159

60 **Ansaldi, S, De Floriani, L and Falcidieno, B** 'An edge–face relational scheme for boundary representations' *Comput. Graph. Forum* Vol 4 (1985) pp 319–332

61 **Arquès, D and Koch, P** 'Pavages tridimensionnels' *Bigre + Globule* Nos 61–62 (April 1989) pp 5–15

62 **Arquès, D and Jacques, I** 'Implantation et plongement de solides effeuillables' *Journées Groplan* Strasbourg, France (December 1989) *Bigre + Globule* No 67 pp 112–123

63 **Munkres, J** *Elements of Algebraic Topology* Addison-Wesley (1984)

64 **Lienhardt, P** 'Topological models for boundary representation: a survey' *Research Report R90–02* Département d'Informatique, Université Louis Pasteur, Strasbourg, France (February 1990)

65 **Woo, T** 'A combinatorial analysis of boundary data structure schemata' *Comput. Graph. Appl.* Vol 5 No 3 (March 1985) pp 19–27

66 **Gangnet, M, Hervé, J-C, Pudet, T and Van Thong, J-M** 'Incremental computation of planar maps' *Comput. Graph.* Vol 23 No 3 (July 1989) pp 345–354

67 **Barstels, R, Beatty, J and Barsky, B** *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling* Morgan Kaufmann Publishers, Los Altos, CA, USA (1987)

68 **Mortenson, M** *Geometric Modeling* John Wiley & Sons, New York, USA (1985)

69 **Casale, M** 'Free-form solid modeling with trimmed surface patches' *Comput. Graph. Appl.* Vol 17 No 1 (January 1987) pp 33–43

70 **Weiler, K** 'Boundary graph operators for non-manifold geometric modeling topology representations' *Proc. IFIP WG 5.2 Working Conference* Rensselaerville, USA (May 1986). Also in **Wozny, M J, McLaughlin, H W and Encarnacão, J L (Eds)** *Geometric Modeling for CAD Applications* North-Holland (1988) pp 37–66

71 **Dufourd, J-F** 'A topological map-based kernel for polyhedron modellers: algebraic specification and logic prototyping' *Proc. Eurographics '89* Hamburg, Germany (September 1989) pp 301–312

72 **Requicha, A and Voelker, H** 'Solid modeling: current status and research directions' *Comput. Graph. Appl.* Vol 3 No 7 (October 1983) pp 25–37

73 **Mäntylä, M** 'A note on the modeling space of Euler operators' *Comput. Vision, Graph. Image Process.* Vol 26 No 1 (1984) pp 45–60

74 **Post, F and Klok, F** 'Deformations of sweep objects in solid modeling' *Proc. Eurographics '86* Lisbon, Portugal (August 1986) pp 103–113

75 **Barsky, B** *Computer Graphics and Geometric Modeling using Beta-Splines* Springer-Verlag (1988)

76 **Gascuel, M-P** 'Welding and pinching spline surfaces: new methods for interactive creation of complex objects and automatic fleshing of skeletons' *Proc. Graphics Interface '89* (June 1989)

77 **Bechmann, D** 'Définition et manipulation de primitives surfaciques de forme libre', PhD Thesis, Département d'Informatique, Université Louis Pasteur, Strasbourg, France (June 1989)

78 **Bechmann, D and Sequeira, J** 'Computing a free-form patch from its boundary characteristics' *Proc. Pixim '89* Paris, France (September 1989) pp 93–108

79 **Barr, A** 'Global and local deformations of solid primitives' *Comput. Graph.* Vol 18 No 3 (July 1984) pp 21–30

80 **Sederberg, T and Parry, S** 'Free form deformations of solid geometric models' *Comput. Graph.* Vol 20 No 4 (August 1986) pp 151–160