9

Classification of boundary representations for manifold and non-manifold topology

Yasushi Yamaguchi
The University of Tokyo
Department of Graphics and Computer Sciences
3-8-1, Komaba, Meguro-ku, Tokyo 153, Japan
Telephone: +81-3-5454-6848, Telefax: +81-3-5454-6845
E-mail: yama@graco.c.u-tokyo.ac.jp

Abstract

Since boundary representations for non-manifold topology have been studied aiming at several different points, their domains are some how different. This paper introduces the classification of the boundary representations for both manifold and non-manifold topology based on their neighborhood constraints. According to the classification data structure for each model is proposed in this paper.

Keywords

Boundary representation, non-manifold topology, r-set solid, manifold solid

1 INTRODUCTION

Most of the current geometric modeling systems using boundary representations are based on manifold topology as their internal representations. A manifold solid is a solid such that any point on its boundary is homeomorphic to a two-dimensional disk. This property is quite important for defining its data structure as well as the operations to manipulate them. However manifold topology has some limitations. Firstly, it is not closed for regularized Boolean operations. A regularized Boolean operation may yield a non-manifold result as shown in Figure 1. The edge drawn with a bold line in the right figure is a non-manifold edge, because it has four adjacent faces. Secondly, a solid model can basically represent a closed volume and cannot represent internal structures such as an object consisting of several materials. Thirdly it is capable of representing only one closed volume and not able to handle mixed dimensional models consisting of volumes, surfaces, and curves in the three dimensional space. Non-manifold topology models are expected to be a major stream of boundary representations in order to solve these issues.

There have been done many researches regarding non-manifold topology. Though the

M. J. Pratt et al. (eds.), *Product Modeling for Computer Integrated Design and Manufacture*

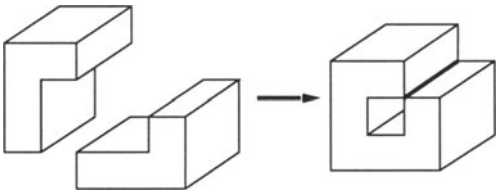

Figure 1 A non-manifold object generated by a regularized Boolean operation.

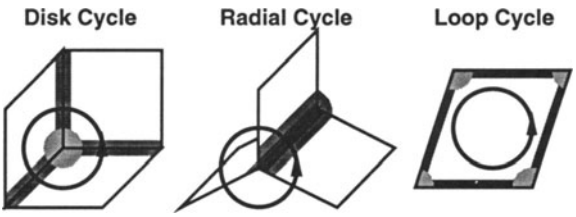

Figure 2 Three kinds of cycles.

domains of those studies are somehow different because the term, non-manifold topology, only indicates that the boundary is not 2-manifold but no more. Some of them are aimed at manipulating r-sets (Desaulniers and Stewart, 1992) (Higashi, Yatomi, Mizutani, and Murabata, 1993), some are for representing internal structures as well as mixed-dimensional models (Weiler, 1986) (Gursoz, Choi, and Prinz, 1990), some are capable of representing multi-dimensional models (Brisson, 1989) (Lienhardt, 1989) (Rossignac, O'Connor, 1990). The purpose of this study is to classify topology models in the three dimensional space in terms of boundary representations. This classification is crucial for designing data structures. The topology models are classified with their neighborhoods so that they require different data structures for representing their adjacency relationships.

In Section 2, topological constraints on boundaries and neighborhoods are observed. Section 3 discusses the classification of boundary representation models based on the observations in Section 2. Three different data structures are proposed in Section 4 based on the topological constraints. Section 5 concludes this paper.

2 CONSTRAINTS ON TOPOLOGY

The fundamental concept of this study is a coupling entity (Yamaguchi, 1995) which stands for the adjacency relationship among primitive entities of point sets, i.e., vertices,

Table 1 Coupling entities.

	Vertex	Edge	Face	Region
Vertex	***	end	fan	corner
Edge	end	***	blade	wedge
Face	fan	blade	***	side
Region	corner	wedge	side	***

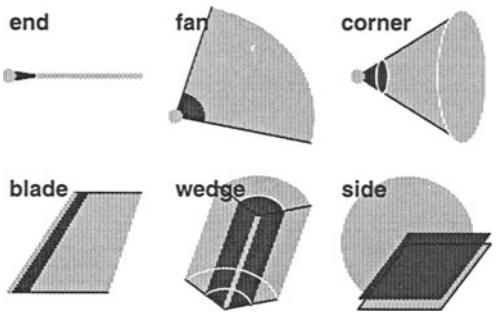

Figure 3 Coupling entities.

edges, faces and regions. Let us start with explaining how it works. A basic requirement of boundary representation is ordering information. It is known that there are three kinds of cycles in a three dimensional space. Figure 2 illustrates the three cycles. A disk cycle stands for a cycle consisting of faces and edges around a vertex. A loop cycle is a cycle of edges and vertices along the boundary of a face. A radial cycle is a cycle of regions and faces radially existing along an edge. It is natural to think that a radial cycle of the non-manifold edge in Figure 1 consists of four portions of adjacent regions as well as four adjacent faces, but the edge is in fact adjacent to only two regions. At local sight it looks as there are four adjacent regions, but in a global sense there are only two regions, the inside and outside of the solid. The coupling entities allow us to distinguish those portions. There are six types of coupling entities according to the pair of point sets as shown in table 1. Both boundaries and neighborhoods of point sets are represented in terms of coupling entities. Figure 3 illustrates the coupling entities.

2.1 Constraints on boundaries

In this section the constraints on the boundaries of primitive entities are going to be observed. Since a vertex has no boundary, no constraint exists on the vertex boundary. The boundaries of an edge are its two ends. Two distinct ends are expected even if the

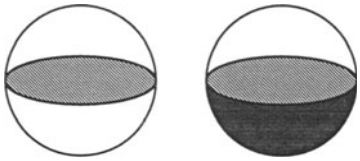

Figure 4 Neighborhoods of face points.

both ending vertices of an edge are the same, because there exist two individual adjacency relationships. Therefore the number of ends is twice the number of edges:

$$N_{\text{end}} = 2N_{\text{Edge}}. \tag{1}$$

Here N_{x} denotes the number of entities of type x.

The boundary of a face, namely a loop, consists of blades and fans standing for adjacency relationships to edges and vertices respectively. While a loop is being traced, blades and fans appear in turn composing a loop cycle. Thus, the number of blades is the same as that of fans:

$$N_{\text{blade}} = N_{\text{fan}}. \tag{2}$$

The boundary of a region, i.e., a shell, consisting of corners, wedges, and sides can be seen as 2-manifold. The following constraint on the numbers of the entities composing region boundaries are obtained:

$$(N_{\text{corner}} - N_{\text{a\_disk}}) - N_{\text{wedge}} + (N_{\text{side}} - 2N_{\text{a\_Loop}}) = 2(N_{\text{Shell}} - N_{\text{c\_Cycle}}). \tag{3}$$

Here, a *c_Cycle*, an abbreviation of "cut Cycle", is a cycle on a boundary that is not contractable to a point. An *a_Loop*, which is an abbreviation of "additional Loop", stands for a loop which is not a primary loop of its face. If two loops belong to the same face, either of them is treated as a primary loop and the other as an a_Loop. An *a_disk* is a secondary disk in a corner like an a_Loop on a face.

2.2 Constraints on neighborhoods

This section observes the constraints on the neighborhoods of the primitive entities. There is no particular constraint on the neighborhood of a region, because every point in a region is homeomorphic to a 3-manifold. Any neighborhood of a point in a region consists of the points of itself.

The neighborhoods of points on faces are illustrated in Figure 4. The left figure shows a neighborhood of a face point in a cell decomposition model. It is homeomorphic to a two-dimensional disk without a boundary and separates its neighborhood into two portions of

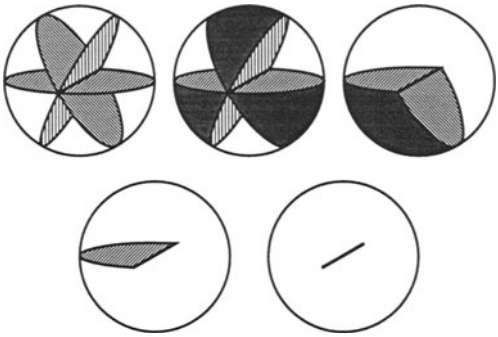

Figure 5 Neighborhoods of edge points.

regions that correspond to sides. The following equation implies this constraint that each face has two sides:

$$N_{\text{side}} = 2N_{\text{Face}}. \tag{4}$$

The right figure shows a face neighborhood of a solid model. The main difference from the left figure is the distinction of sides. Since every face of a solid separates an inside region from its outside, one of its two sides corresponds to the inside and the other corresponds to the outside. Every face has two types of sides, an inside side and an outside side. This constraint is represented with the following equation:

$$N_{\text{i\_side}} = N_{\text{o\_side}}. \tag{5}$$

Any entities related to regions in a solid model can be classified into two types, inside and outside. An inside entity and an outside entity of type x will be denoted as i_x and o_x respectively. It is obvious that the numbers of each entity fulfill the following equation:

$$N_{\text{i\_x}} + N_{\text{o\_x}} = N_{\text{x}}.$$

The neighborhoods of edge points are shown in Figure 5. The upper left figure shows a neighborhood of an edge in a cell decomposition model. The edge is adjacent to faces and regions in turn. Thus there are the same number of blades and wedges radially existing around the edge:

$$N_{\text{blade}} = N_{\text{wedge}}. \tag{6}$$

The upper middle figure illustrates an edge neighborhood of an r-set solid model. Two types of wedges, i.e., inside wedges and outside wedges, mutually appear around the edge

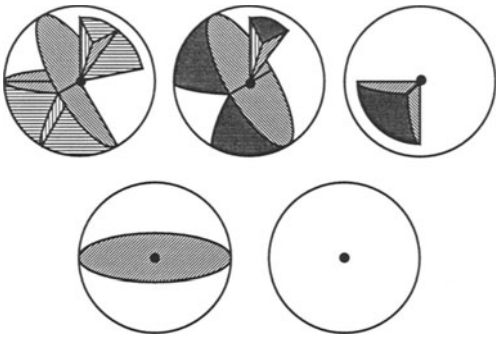

Figure 6 Neighborhoods of vertex points.

in turn. Edge neighborhoods of a solid model must fulfill the additional condition that the number of inside wedges and that of outside wedges are the same:

$$N_{\text{i\_wedge}} = N_{\text{o\_wedge}}. \tag{7}$$

The upper right figure shows an edge neighborhood of a manifold solid model. Because of the definition of a manifold solid every edge is adjacent to two faces and separates its inside from its outside. Therefore the edge neighborhoods of a manifold solid consists of two blades, an inside wedge and an outside wedge. This constraint can be described with the following equation:

$$N_{\text{i\_wedge}} = N_{\text{o\_wedge}} = N_{\text{Edge}}. \tag{8}$$

The lower left figure illustrates a neighborhood of a boundary edge in a surface model, i.e., an open shell edge. The neighborhood consists of a blade and a wedge. It is important that this type of neighborhood satisfies the constraint of Equation (6):

$$N_{\text{blade}} = N_{\text{wedge}} = 1.$$

This indicates that the neighborhood of the boundary edges in a surface model can be categorized into the same class as a cell decomposition model. The lower right figure shows a neighborhood of an edge in a wireframe model, i.e., a dangling edge. The neighborhood of the edge does not satisfy Equation (6), because this edge is adjacent to a region but no face. In other words a dangling edge neighborhood must be categorized into a different class from a cell decomposition model.

Figure 6 shows neighborhoods of vertices. The upper left figure illustrates a neighborhood of a vertex in a cell decomposition model. By taking an intersection with a sphere of an infinitesimal radius, a graph on the sphere satisfying Euler-Poincaré formula is ob-

tained. Thus the constraints on the vertex neighborhood is identical to the Euler-Poincaré formula on a sphere as below:

$$N_{\text{end}} - N_{\text{fan}} + (N_{\text{corner}} - N_{\text{a\_disk}}) = 2N_{\text{Vertex}}. \tag{9}$$

The upper middle figure shows a vertex neighborhood of an r-set solid model. The main characteristic of this class is that the graph on a sphere can be colored with only two colors, i.e., inside and outside, though this characteristic itself can be deduced from Equation (7). The upper right figure shows a vertex neighborhood of a manifold solid model. The neighborhood which is homeomorphic to a two dimensional disk has two corners without any a_disks. Vertex neighborhoods of a manifold solid model must fulfill the following equations in addition to the equation (9).

$$\begin{aligned} N_{\text{a\_disk}} &= 0, \\ N_{\text{i\_corner}} &= N_{\text{o\_corner}} = N_{\text{Vertex}}. \end{aligned} \tag{10}$$

Any vertex neighborhood fulfills Equation (9) if the intersection with an infinitesimal sphere forms an ordinary graph. The lower two figures show the cases that the intersection does not form a ordinary graph on a sphere. The lower left figure illustrates the neighborhood of an isolated vertex on a face consisting of two corners and a fan. The intersection which is a single circular arc without any node does not form a graph. Thus the neighborhood of an isolated vertex on a face does not satisfy Equation (9) and has a different class of neighborhood from a cell decomposition model. The lower right figure illustrates the neighborhood of an isolated vertex in a region. Its neighborhood is simply a corner adjacent to the surrounding region. The neighborhood of an isolated vertex in a region has also a different class of neighborhood.

3 CLASSIFICATION OF BOUNDARY REPRESENTATIONS

According to the above mentioned observations, boundary representations which might be commonly used for representing shapes in three dimensional space can be classified into six classes, a cell decomposition model, an r-set solid model, a manifold solid model, a dangling edge, an isolated vertex on a face, and an isolated vertex in a region.

- Cell decomposition model.
  This class of representation handles the topology that fulfills Equations (1), (2), (3), (4), (6), and (9). This model is capable of representing several regions as well as open shells which usually appear in a surface model. It is a generic model for representing point sets in a three dimensional space. The following two models are eligible for introducing further constraints.
- R-set solid model.
  This class of representation handles the topology that fulfills Equations (1), (2), (3), (4), (5), (6), (7), and (9). This model is capable of representing an r-set solid which is a single connected closed point set with a certain volume in a three dimensional space. It allows coincident edges and vertices that might be generated by regularized Boolean

set operations. A face whose two sides are adjacent to the same region is prohibited in this model.

- Manifold solid model.
  This class of representation handles the topology that fulfills Equations (1), (2), (3), (4), (5), (6), (7), (8), (9), and (10). This model can represent a manifold solid. Any point on its boundary is homeomorphic to a two-dimensional disk.
- Dangling edge.
  Dangling edges which appear in a wireframe model do not satisfy Equation (6).
- Isolated vertex on a face.
  Isolated vertices on faces which can appear any faces even in a manifold solid do not satisfy Equation (9).
- Isolated vertex in a region.
  Isolated vertices in regions also do not satisfy Equation (9).

Due to the fulfillment of the constraint Equations (1) through (10), there exist the hierarchical inclusion relations among the first three models as below:

cell decomposition model $\supset$ r-set solid model $\supset$ manifold solid model.

On the contrary the last three classes of entities are exceptional cases rather than topology models. Although they can appear in the first three models as well as a mixed dimensional model, they cannot be handled in uniform manners. Since those exceptional cases were already discussed in the previously presented paper (Yamaguchi, 1995), this paper concentrates on the first three classes of representations.

4 DATA STRUCTURES FOR BOUNDARY REPRESENTATIONS

The boundary representations can be mainly classified into three models. The data structures for those classes of boundary representation models are discussed in this section.

4.1 Feather data structure for a cell decomposition model

A cell decomposition model can be represented by a feather data structure (Yamaguchi, 1995). Two Equations, (2) and (6), indicate that the numbers of wedges, blades, and fans are all the same and there must be some correspondence among them. Those coupling entities can be represented by a pair of pseudo-coupling entities named *feather*. Figure 7 illustrates the data structure. There is no explicit entities for a wedge, a blade, or a fan. The substitute entity named feather takes a role in representing adjacency relationships.

A certain set of pointers among feathers are necessary for representing the cyclic ordering information. Feathers are the substitutes of three coupling entities, fans, blades, and wedges, the mating pointers which point to mates of those entities are crucial. Of course our first objective is to represent the three kinds of cycles, loop cycles, radial cycles, and disk cycles, that can be traversed in two different manners, either clockwise or counter clockwise. Thus each feather may have nine types of pointers pointing to other feathers, an FM (Fan Mate) pointer, a BM (Blade Mate) pointer, a WM (Wedge Mate) pointer, a CCD (Counter Clockwise Disk) pointer, a CD (Clockwise Disk) pointer, a CCR

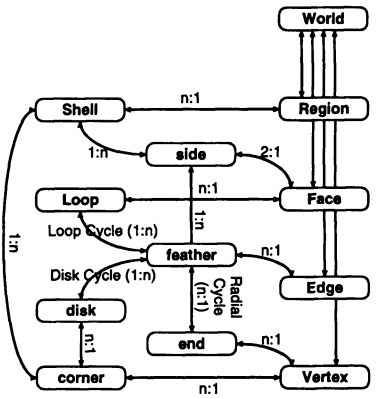

Figure 7 Feather data structure for a cell decomposition model.

(Counter Clockwise Radial) pointer, a CR (Clockwise Radial) pointer, a CCL (Counter Clockwise Loop) pointer, and a CL (Clockwise Loop) pointer. There are the following relations among those pointers:

$$
\begin{aligned}
x &= \mathrm{CCD}(\mathrm{CD}(x)) = \mathrm{CD}(\mathrm{CCD}(x)),\\
x &= \mathrm{CCR}(\mathrm{CR}(x)) = \mathrm{CR}(\mathrm{CCR}(x)),\\
x &= \mathrm{CCL}(\mathrm{CL}(x)) = \mathrm{CL}(\mathrm{CCL}(x)),\\
\mathrm{CCD}(x) &= \mathrm{CCR}(\mathrm{FM}(x)),\\
\mathrm{CCR}(x) &= \mathrm{WM}(\mathrm{BM}(x)),\\
\mathrm{CCL}(x) &= \mathrm{FM}(\mathrm{BM}(x)).
\end{aligned}
$$

The above six equations indicate dependencies among the nine types of pointers. Therefore three independent pointers are essential for representing all of those relations. Three pointers, FM, BM and WM, are recommendable due to the several reasons such as sufficiency and efficiency.

4.2 Halfwedge data structure for an r-set solid model

An r-set solid model can be represented by a halfwedge data structure. Since it is unnecessary to represent regions explicitly, the data structure has no region data entity. Instead the model requires the distinction of the inside and the outside. The entities associated with regions, such as shells, corners, disks, and wedges, must correspond to either the inside or the outside. A face has asymmetric sides, an inside side and an outside side, which causes asymmetry of a blade and a fan. A pseudo-coupling entity named *halfwedge* stands for a blade and a fan as well as a half of both inside and outside wedges. Figure 8 illustrates the data structure.

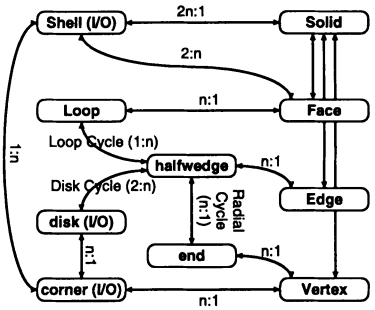

Figure 8 Halfwedge data structure for an r-set solid model.

Since a halfwedge itself represents a fan and a blade it has no longer an FM pointer and a BM pointer. Instead it must be capable of distinguishing the inside and the outside of wedges and disks. Therefore a halfwedge has ten types of pointers pointing to each other, an IM (Inside wedge Mate) pointer, an OM (Outside wedge Mate) pointer, a CCI (Counter Clockwise Inside disk) pointer, a CI (Clockwise Inside disk) pointer, a CCO (Counter Clockwise Outside disk) pointer, a CO (Clockwise Outside disk) pointer, a CCR (Counter Clockwise Radial) pointer, a CR (Clockwise Radial) pointer, a CCL (Counter Clockwise Loop) pointer, and a CL (Clockwise Loop) pointer. There are the following relations among those pointers:

$$
\begin{aligned}
x = \mathrm{CCI}(\mathrm{CI}(x)) &= \mathrm{CI}(\mathrm{CCI}(x)),\\
x = \mathrm{CCO}(\mathrm{CO}(x)) &= \mathrm{CO}(\mathrm{CCO}(x)),\\
x = \mathrm{CCR}(\mathrm{CR}(x)) &= \mathrm{CR}(\mathrm{CCR}(x)),\\
x = \mathrm{CCL}(\mathrm{CL}(x)) &= \mathrm{CL}(\mathrm{CCL}(x)),\\
\mathrm{CCR}(x) &= \mathrm{OM}(\mathrm{IM}(x)),\\
\mathrm{CCL}(x) &= \mathrm{CCI}(\mathrm{IM}(x)),\\
\mathrm{CCL}(x) &= \mathrm{CO}(\mathrm{OM}(x)).
\end{aligned}
$$

The above seven equations indicate dependencies among the ten types of pointers. Three independent sets of pointers are essential for representing those relations. Three pointers, CCI, CCO and CL, are one of those independent set of pointers. Four pointers, IM, OM, CCL, and CL, are also useful for implementing a system.

4.3 Halfedge data structure for a manifold solid model

A manifold solid model can be represented by a halfedge data structure (Mäntylä, 1988). Equation (8) indicates that an edge has only two wedges corresponding to the inside and the outside of a solid. A pair of pseudo-coupling entities named *halfedge* are eligible for representing adjacency relationships of an edge. Furthermore the inside structure of a

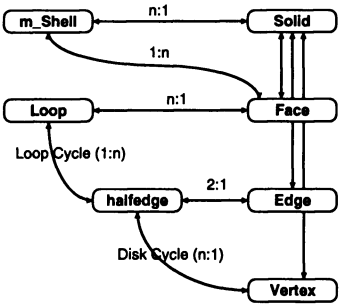

Figure 9 Halfedge data structure for a manifold solid model.

manifold solid is completely identical to its outside structure. In other words, there is one-to-one correspondence between inside shells and outside shells. Therefore the data structure requires only one integrated shell named *m_shell*, which is an abbreviation of "manifold shell", standing for both an inside shell and an outside shell. Figure 9 illustrates the data structure. It has neither corners nor disks, because each vertex has exactly one inside corner and one outside corner without any a_disks.

Since its outside and its inside are completely symmetric, four cyclic pointers, CCI, CI, CCO, and CO, merged into two cyclic pointers. Other two cyclic pointers, CCR and CR, are no longer necessary, because a radial cycle makes no sense. An IM pointer and an OM pointer are merged into one mate pointer, because every edge consists of two halfedges. A halfedge could have five types of pointers pointing to other halfedges, an EM (Edge Mate) pointer, a CCD (Counter Clockwise Disk) pointer, a CD (Clockwise Disk) pointer, a CCL (Counter Clockwise Loop) pointer, and a CL (Clockwise Loop) pointer. There are the following relations among those pointers:

$$
\begin{aligned}
x &= \mathrm{CCD}(\mathrm{CD}(x)) = \mathrm{CD}(\mathrm{CCD}(x)), \\
x &= \mathrm{CCL}(\mathrm{CL}(x)) = \mathrm{CL}(\mathrm{CCL}(x)), \\
\mathrm{CCL}(x) &= \mathrm{CD}(\mathrm{EM}(x)).
\end{aligned}
$$

The above three equations indicate dependencies among the five types of pointers. Two independent pointers are essential for representing those relations. Two pointers, CD and CL, are one of those independent pair of pointers. Three pointers, EM, CL, and CCL is also useful for implementing a system.

5 CONCLUSIONS

This paper introduced a classification of boundary representation models. The classification has been done based on the neighborhood constraints on each topology model. The following conclusions are obtained:

- Equations of topological constraints, (1) through (10), characterize all classes of boundary representation models.
- Three main data structures, a feather data structure, a halfwedge data structure, and a halfedge data structure, are discussed.

The classification of boundary representation models are very important for exchanging the topological data between different geometric modeling systems.

REFERENCES

E. Brisson. Representing geometric structures in d dimensions: Topology and order. In *5th Symposium on Computational Geometry*, pages 218–227. A.C.M., 1989.

H. Desaulniers and N. F. Stewart. An extension of manifold boundary representations to the r-sets. *ACM Transactions on Graphics*, 11(1):40–60, 1992.

E. L. Gursoz, Y. Choi, and F. B. Prinz. Vertex-based representation of non-manifold boundaries. In M. J. Wozny, J. U. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 107–130. North-Holland, 1990.

M. Higashi, H. Yatomi, Y. Mizutani, and S. Murabata. Unified geometric modeling by non-manifold shell operation. In J. Rossignac, J. Turner, and G. Allen, editors, *Proceedings of Second Symposium on Solid Modeling and Applications*, pages 75–84. A.C.M., 1993.

P. Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *5th Symposium on Computational Geometry*, pages 228–236. A.C.M., 1989.

M. Mäntylä. *An Introduction to Solid Modeling.* Computer Science Press, 1988.

J. R. Rossignac and M. A. O'Connor. SGC: A dimension-independent model for pointsets with internal structures and incomplete boundaries. In M. J. Wozny, J. U. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 145–180. North-Holland, 1990.

K. J. Weiler. *Topological Structures for Geometric Modeling.* PhD thesis, Rensselaer Polytechnic Institute, 1986.

Y. Yamaguchi and F. Kimura. Representation and operations for non-manifold topology based on coupling entities. *Computer Graphics and Applications*, 15(1):42–50, January 95.