10

A Note on Non-Manifold Object Sweeping

Benoit Touron, David Marcheix and Stefka Gueorguieva
University of Bordeaux I
Laboratoire Bordelais de Recherche en Informatique
351, Cours de la Libération
33405 Talence FRANCE
Email: {touron,marcheix,stefka}@labri.u-bordeaux.fr

Keywords
Geometric modeling, non-homogeneous object, non-manifold, topology, cellular complexes, generalized sweeping, boundary representation, Euler operators,

1 INTRODUCTION

Sweeping in a non-manifold environment is a more powerful modeling tool than its manifold counterpart. The interest towards non-manifold processing gives rise to active research in the field. Some of the proposed data structures are reported as internal representations of geometric modelers and are implemented for the realization of modeling operations. In particular with respect to non-manifold sweeping, basic ideas are given by (Ferrucci and Paoluzzi, 1993), (Paoluzzi, Bernardini, Cattani and Ferrucci, 1993), (Weiler, 1990). First due to the generality of the representation domain there are no restrictions either on the sweeping operands or on the resulting objects. Any combination of wireframes, shells and volumes could be processed. The "degenerate" cases of dimensionally non-homogeneous parts or self-intersection are also supported. Second non-manifold sweeping is a selective multi-step operation. Using traditional extrusion as an example, the elements are swept to a higher dimension. Thus a curve is swept to a shell and a face is transformed into a volume. In contrast, non-manifold sweeping allows elements to be swept in one of three ways: to a higher dimension (as the classic algorithm), to the same dimension or to remain in place. Moreover, just chosen portions of the object could be transformed while maintaining the connectivity between swept and unswept elements. Further, elements created in one step could be candidates to be swept in another step of the sweep operation. This level of generality is supported by the non-manifold environment in a quite natural and simple way.

This present work is an extension of the non-manifold sweeping algorithm proposed by (Weiler, 1990). The uniqueness of the discussed algorithm is the explicit handling of dimensionally non-homogeneous sweeps via topological technics. We propose to perform

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

a preprocessing analysis to locate sites where non-manifold conditions would appear and subdivide the initial object in such a way that each component would give rise to a manifold swept object. In this way, the result could be represented as a heap of manifolds and the topological structure at non-manifold points could be correctly calculated. The elaborated algorithm processes objects defined in terms of a non-manifold boundary representation called Radial model (Gueorguieva and Marcheix, 1994). The following steps are executed: first, elements to be swept are marked; then, sweep marks are propagated up-down the model hierarchy; next, object is subdivided into homogeneously marked regions; further, marked elements are extruded and swept objects for every region are constructed; finally, the result is validated in terms of the Radial model. All transformations are described as sequences of the appropriate topological operators (Marcheix and Gueorguieva, 1995). This guarantees the topological integrity of the model. A verification that the object is a cellular complex is also performed (see (Hoffmann, 1989)).

The paper is organized as follows. The non-manifold boundary representation in use is briefly discussed in Section 2. Basic definitions of the sweeping operation and the algorithm concept are introduced in Section 3. Finally, Section 4 addresses the implementation issues and the experimental results.

2 NON-MANIFOLD REPRESENTATION

The algorithm, presented in section 3, is implemented using the Radial model and a set of operators for manipulation of the Radial model given in (Gueorguieva and Marcheix, 1994) and (Marcheix and Gueorguieva, 1995). For the best understanding of the proposed method a brief description of the Radial model follows. However, it should be remarked that this method is representation independent and any non-manifold scheme could be used.

2.1 Radial Model

In terms of the Radial model, the solid is a cellular decomposition with an explicit encoding of the non-manifold conditions. The basic idea of the Radial model is to represent objects as generalized complexes such that the level of cellular decomposition is under user control. Formally an **object** is separated into simpler dimensionally homogeneous constituents of dimension i, $i = 0,..,3$. Each one is defined as a primitive and does not contain non-manifold points*. This decomposition is not unique and depends upon the user goal being either a detailed representation in terms of the finest complex cells† or a more rough subdivision. There could be different levels of refinement corresponding to different object parts. When different topological elements are piled up at the same geometric location, the primitives to which they belong are connected through a radial link‡. When the radial link does not define a non-manifold condition it could be destroyed without affecting boundary determinism. This leads to a macroscopic view of the part of interest.

*The representation of primitives follows the classic boundary method (see for example (Mäntylä,1988)).

†i-D cells,$i = 0,..,3$, are denoted resp. **vertex**, **edge**, **face**, **volume.**

‡A **radial vertex** defines the set of vertices of an object embedded in the same point of E^3. **Radial edges** and **radial faces** are defined by analogy.

The construction and the modification of the Radial model are performed using a kernel of topological operators presented in subsection 2.2.

2.2 Non-manifold boundary operators

A broadly used concept for construction, modification and maintenance of boundary representations are the so called Euler operators. They are considered as a kind of infrastructure with which more complex operations could be realized. The non-manifold sweeping algorithm, described in section 3, is a good example of using a set of non-manifold boundary operators for manipulation of objects defined in terms of the Radial model. They are classified into two groups: operators for cell creation and operators for the identification of cell faces. Their notations point out the effect of creation M (*Make*) and identification *MRad* (*Make Radial*) or *GRad* (*Glue Radial*) followed by the abbreviations of the concerned elements s, o, p, f, l, e, v respectively for scene[§], object, primitive, face, loop[¶], edge and vertex. The inverse operators are K (*Kill*) and *KRad* (*Kill Radial*) or *URad* (*Unglue Radial*).

Each d-cell is defined recursively in terms of its faces[‖]. Thus, d-cell construction starts with the definition of the i-cells, $0 \leq i < d$, that build up its boundary faces. Then the face adjacency relationships are filled up. The 0-cell, a vertex, is constructed with the operator *Mvpo*. Indeed, the most simple object is an isolated point that, following the Radial model, is defined as an object compound of a unique vertex primitive. The 1-cell, an edge, necessitates the creation of two 0-cells, one 1-cell and the initialization of the edge faces pointing to the newly created 0-cells. Moreover, each edge is added to an existing loop or provokes the creation of a new loop. In the first case *MeKpo* is applied, while in the second the *MelKpo* is used. The object and the primitive that are destroyed correspond to one of the 0-cell operands containing the boundary vertices. By analogy, for a 2-cell, first the boundary vertices and edges are created. Then, they are connected into a loop with *Mel* and finally a *Mf* initializes the face adjacency relationships and transforms the loop into a bondary face loop. The definition of a 3-cell, a volume, requires the construction of the corresponding i-cells, $0 \leq i < 3$, and their configuration as 3-cell boundaries with *Mvolume*.

Each operator has an inverse. Often, the converse logic is more complex. As for example, the inverse operator of *Mf* should distinguish between the cases when the face boundary is simply connected and when it is not. In the first case *Kf* is applied while in the second, *KfMpo* removes the internal boundaries transforming them into separate object components.

The identification of cell faces has a definition specific to the Radial model. When cell faces are identified the operators *GRadv*, *GRade*, *Gradf* are applied for the corresponding cell dimension. When this identification induces a non-manifold condition, the radial link permits its correct interpretation. Indeed, each element of a radial list has a specified orientation as a member of a given manifold component. In this way, the object interior

---

[§] A **scene** gives the list of all objects in the modeling space.

[¶] The **loop** is introduced in order to facilitate the manipulation of edge lists as wireframes or boundary contours of faces.

[‖] Following standard notations, *a face* is used to denote in one respect a boundary item of a cell and in the another a 2-cell. The context of use is clear enough to distinguish the two meanings.

is determined unambiguously in any point. In contrast, when no non-manifold condition occurs, the radial link could be destroyed with *KRadv*, *KRade*, *KRadf* and thus cells are merged into a single primitive.

The inverse operators *URadv*, *URade* and *URadf* (resp. *MRadv*, *MRade*, *Mradf* work in a similar way. This identification mechanism allows the representation of the object as a heap of primitives connected trough radial links thus preserving the internal structure while non-manifold conditions are described explicitly.

3 ALGORITHM FOR NON-MANIFOLD SWEEPING

3.1 Basic concepts

Following the terms used in (Weiler, 1990), two main entities define the sweeping operation: the generator and the director. The **generator** consists of the geometric elements to be swept to the new geometry which will result from the sweeping. The generator control variables are the generator shape and the generator membership. They refer to the ability to change the scale and the geometric shape as well as the content of the generator during repeated or continuous sweeps. The **director** is the geometric transformation function which is used to sweep the generator. The director geometry describes the actual transformation of the director while the director continuity refers to the continuity of the geometry resulting from the sweeping. In a non-manifold environment, both generator and director control variables are exploited. The algorithm proposed by Weiler (1990) takes as an input a model whose elements have been individually marked to be swept to higher or same dimension or not swept at all. In the following description these marks are denoted respectively 1, 0 and −1. The process starts with a check up of element marks: if the boundaries of each marked element are also marked to at least the same or greater dimension. Then through a down-up traversal, each element is examined and swept according to its individual mark. Due to this ordering, by the time when a higher dimensional element is processed all of its boundaries have been already swept, thus simplifying the definition of its image. However, the construction of the swept element is ambiguous when non-manifold conditions appear. See, for example, the interpretation of the shell sweep given in fig. 1.1 where the result of fig. 1.2 could correspond to all cases from fig. 1.3 to fig. 1.6. The non-manifold conditions are encoded by the radial connections created at the edge level but the underlying topological structures are not equivalent. Moreover, the down-up traversal does not take advantage of the local environment marks in order to detect the appearance of the non-manifold condition. Indeed, the determinative factor for the non-manifold occurence is the sweeping continuity of the higher dimensional element.

Our idea is to use a preprocessing step that subdivides the object in continuously marked regions. Each region is processed depending on its neighbours. Geometric discontinuity will appear in areas of non-homogeneity i.e. points where regions marked 1 are adjacent to regions marked 0 or −1. Thus, in points of geometric discontinuity the object is cut up into separated regions radially connected at the boundary of the discontinuity marked elements. In the next step, each region is swept according to its individual mark thus giving rise to the manifold swept component. Then, radial links are propagated between the newly created elements in correspondence with the radial adjacency of their predecessors.

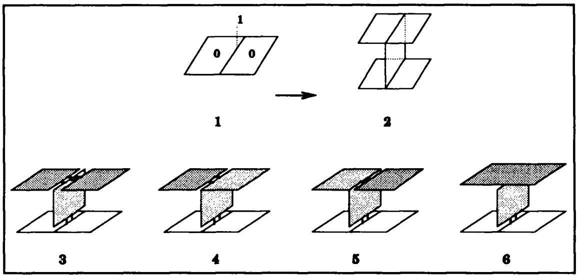

Figure 1 Shell sweep interpretation

Finally, the model is validated to make sure all auto-intersections are encoded through the proper topological adjacencies.

It should be remarked that no preliminary traversal is required if a global up-down traversal of the generator is carried out. We deal with two types of marks: the individual sweep mark and the mark inherited from the higher dimension. First, the elements are swept according to their inherited marks. This corresponds to the first step of the sweeping operation when the highest dimensional elements are transformed. Then, boundary elements whose individual sweep marks exceed their inherited marks are further processed. In the case when there are points of non-homogeneity, these are duplicated in radially connected elements swept according to the individual marks of the respective matching entities. Thus, different dimensional parts are radially connected at the proper boundary points maintaining the connectivity of the initial object. The whole algorithm is described in subsection 3.2.

3.2 Data structure and algorithm

The elaborated algorithm operates on objects defined in terms of the boundary non-manifold model. Along with this basic representation, a complimentary data structure is associated to each element.

. `NM`** `Geometric Model Entity.`

. `struct NM SweepData{ short` *Mark, inheritMark*; `Node` *SweepUp, SweepDown*; `sho` *flag*; }

The *Mark* controls the individual sweeping dimensionality. The *inheritMark* depends on the higher dimensional item. The mechanism that ensures the correspondance between an element on the generator shape at one location along the director and an element on the new generator shape at another location is provided by the *SweepUp* and *SweepDown* pointers denoting respectively the image and its predecessor. The *flag* is necessary for the model traversal.

**abbreviation for Non-Manifold

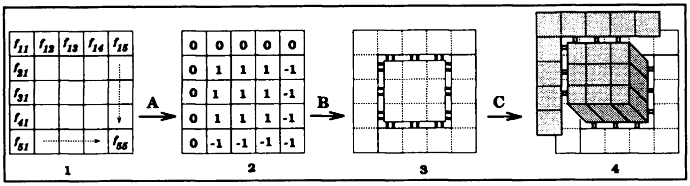

Figure 2 An example of NM shell sweep

The elaborated algorithm consists of four basic steps.

(A) Mark elements to be swept. This initial stage is the marking, when each element is allocated a *Mark* that indicates the element sweeping dimension.

(B) Subdivide object into regions with homogeneous sweeping continuity. At this stage of the algorithm the initial object is subdivided into fields of uniform sweeping continuity. This is done through the creation of radial links in points where faces marked 1 are adjacent to faces marked 0 or -1 as shown in fig.2. By anology, for the wireframe object the subdivision is illustrated in fig.4.

(C) Sweep regions. The sweeping operation starts with an up-down traversal of models parts. Shells, wires and vertices are all defined as separate primitives. For a given level of the model hierarchy, the processing begins with elements marked 1 down to elements that are not transformed at all. This order is a result of the observation that each element should be swept with at least the same sweeping dimension as the element to whose boundary it belongs. Consequently, analysing neighbouring sweep marks enables the detection of points where the sweeping will create geometric discontinuity and thus necessitating the creation of radial links. First we will consider the sweep operation on shells.

Regions marked 1. At this stage of the treatment fields of the shell marked 1 have already been separated from the rest of the shell through radial edge links as illustrated in fig.2. These regions will be swept into volume components. For each face of such a region we examine its neighbours. If the face is internal to the region i.e. the marks of all adjacent faces are 1 (see f_{33} in fig.2.), then the face is extruded into a face of the boundary of the swept volume. If the face has a neighbour marked 0 then the radial link is propagated between the volume and the image of the face marked 0. Otherwise, no more processing is needed. Faces could have no neighbours at all (see the first and last row, column f_{1j}, f_{5j},f_{i1} , f_{i5} in fig.2.). Then no adjacencies have to be maintained.

Regions marked 0. Let a face belong to a field marked 0. This field is swept into a shell. Then if the face is internal to the field it is extruded into a face adjacent to the images of the neighbours of the initial face. If the face has neighbours marked 1 then its image is radially connected to the corresponding volumes. See for example f_{21} in fig.2. that is adjacent to f_{22}. They are linked trough a radial edge and this radial connection is propagated between their images (the shell image and the swept volume). For adjacent faces marked -1 no further control is needed.

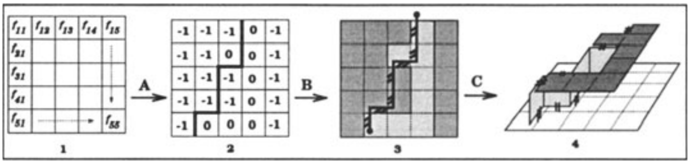

Figure 3 NM Sweep of boundary edge sequence

Regions marked -1. These parts of the model are not subject to transformations and the adjaceny relationships are established at stage B of the algorithm.

Boundary edge sequences. At this step we should process boundary edges whose individual sweep marks have a superior value to their inherited marks. That means the sequences of edges marked 1 on regions marked 0 or -1, and edges marked 0 on fields marked -1. In the first case, sweeping will create a non-manifold condition between the shell produced by the edge sequence and the initial region (see fig.3.). Thus the edge sequence is duplicated in a radially connected edge sequence that is swept into a face strip. The radial links are propagated between the strip and the image of the initial region. In the second case, the edge marked 0 sequence is transformed into a wireframe that is possibly connected to the rest of the object if its extremities are adjacent to marked elements.

Boundary vertices. For vertex sweeping the marks of adjacent edges should be also taken into account. For each vertex the neighbouring edges and faces are analysed. The remaining elements to be swept are vertices with individual marks exceeding the inherited marks. There could be vertices marked 1 on fields marked 0 or -1 and vertices marked 0 on regions marked -1. In the first case, if there is at least one adjacent edge marked 1 then the vertex has been already processed in a previous stage. Otherwise the vertex sweeping will result in a non-manifold condition. That is why it is duplicated in a radially connected vertex that is swept into an edge. When the vertex has no neighbour edge marked 0 then the processing is over. If the vertex has just one adjacent edge marked 0 then the swept edge is connected to the edge image. Otherwise another radial connection at vertex level regroups the swept edge and the images of the vertex neighbour elements. In the second case, the vertex marked 0 is swept into an isolated vertex primitive.

Shell sweeping is over once all the element marks have been processed. In the above discussion we detailed shell primitive sweeping. Treatment of wireframe primitive is done by analogy. The wireframe primitive is subdivided into uniformly marked edge sequences. The sequences marked 1 are separated from the rest of the primitive trough radial links at vertex level. Then, they are extruded into face strips. The analysis of the adjacency sectors is simple as long as the continuous marked regions are edge sequences and the connection between regions are points. Depending on the boundary points the radial links are propagated to the swept elements as shown in fig.4.

The last case when just an isolated vertex primitive is swept is trivial. The result is either a swept edge or an isolated vertex primitive according to the sweeping continuity. No adjacency relationships should be adjusted. Finally, let us examine a complex of primitives.

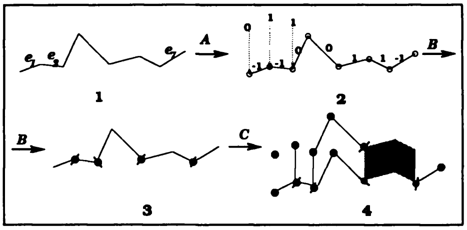

Figure 4 An example of NM Wireframe sweeping

In fact, that is a set of volume, shell, wireframe and vertex primitives[††] radially connected at boundary of potential non-manifold conditions. The sweeping operation is applied to each primitive. Radial links are propagated between corresponding elements. It should be noted that at the subdivision step different choices could be done thus resulting in different swept objects. However, the invariant of the algorithm that the object is a heap of manifold parts is preserved. Some of the subdivisions could be redundant. Some of the radial connections could be destroyed if they do not encode a non-manifold condition. This postprocessing is performed at the validation step.

(D) Validate final model. The validation step is the more general step common to all operations that implies as direct or as side effect modifications in the geometric model. For the complex based representational schemes one should ensure that the underlying stucture is a cellular one i.e. that cells do not intersect in internal points and the boundary intersections are encoded trough proper topology adjacency relationships. In the present algorithm we stick to the method proposed in (Hoffmann, 1989).

As mentioned in section 2, each solid is described as a sequence of non-manifold operators. Consequently three additional operators are introduced to mark faces ($Markf$), edges ($Marke$), and vertices ($Markv$), according to the desired sweeping operation. The subdivision, the region sweep and the validation steps are also done using the corresponding NM operators. In this way the topological integrity of the resulting Radial model is ensured.

4 IMPLEMENTATION RESULTS AND CONCLUSION REMARKS

Despite the great interest in non-manifold modeling and the common use of sweeping as a basic construction technique there is still a lack of thorough understanding of the impact of the non-manifold topological domain. The novelty of the proposed algorithm is the explicit handling of dimensionally non-homogeneous sweeps via topological technics. The presented algorithm is actually implemented in a solid modeler based on a non-manifold

[††]no volume component sweep is considered

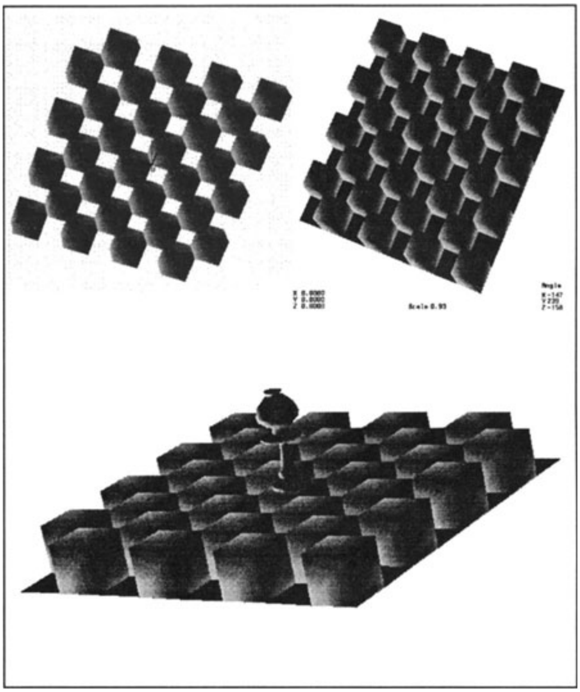

Figure 5

boundary representation called Radial model. It is running on SGI Indy with all graphical interface written on GL.

The examples on fig. 5 is generated using a NM shell sweep. Calculations and model generation are quasi instantaneous. These results are more significant when we compare them with the same object generation through other construction technics like *CSG* or primitive instaciation. Further experiments on algorithm performance are in progress.

REFERENCES

Ferrucci,V. and Paoluzzi,A. (1993). Extrusion and boundary evaluation for multidimensional polyhedra. *Computer- Aided- Design*, **23**, 1, 40 – 50.

Crocker,G.A. and W.F.Reinke (1991). An editable non-manifold boundary representation. *IEEE ComputerGraphics&Applications.*, March, 39 – 51.

Desaulniers,H. and N.F. Stewart (1992). An extension of manifold boundary representa-

tion to the r-sets. *ACM Transactions on Graphics*, **11**, 1, 40 – 60.
Gueorguieva, S. and Marcheix, D. (1994). Non-Manifold Boundary Representation for Solid Modeling. *Proc. of the International Computer Symposium.* National Chiao Tung University, Hsinchu, Taiwan.
Gursoz,E.L. and Choi,Y. and Prinz,B. (1990). Vertex-based representation of non-manifold boundaries. *Geometric modeling for product engineering.* Wozny,M.J, Turner,J.U. and Preiss,K.,Edts., Elsevier Sci, North Holland.
Hoffmann, Ch.M. (1989). Geometric and Solid Modeling: An Introduction. Morgan Kaufmann.
Lang,V. and Lienhardt, P. (1994). Modélisation géométrique à l'aide d'ensembles simpliciaux. *AFIG'94, Toulouse.* Association Française d'Informatique Graphique.
Lienhardt, P. (1991). Topological models for boundary representation: A comparison with n- dimensional generalized maps. *Computer-Aided-Design.*, **21**, 1, 59 – 82.
Mäntylä, M. (1988). Geometric and Solid Modeling: An introduction. Computer Science Press.
Marcheix, D. and Gueorguieva, S. (1995). Topological Operators for Non-Manifold Modeling. *Proc. of the Third International Conference in Central Europe on Computer Graphics and Visualisation 95.* University of West Bohemia, Plzen, Czech Republic.
Masuda,H. (1993). Topological operators and boolean operations for complex-based non-manifold geometric models. *Computer-Aided-Design*, **25**, 2, 119 – 129.
Paoluzzi,A., Bernardini,F., Cattani,C. and Ferrucci,V. (1993) Dimension-independent modeling with simplicial complexes. *ACM Transactions on Graphics* , **12**, 1, 56 – 102.
Rossignac, J.R. and O'Connor, M.A. (1989). SGC: A dimension- independent- model for pointsets with internal structure and incomplete boundaries. *Research rapport RC14340.* IBM T.J. Watson Research Center, NY 10598.
Rossignac, J.R. and Requicha, A.A.G. (1991). Constructive non-regularized geometry. *Computer-Aided-Design* , **23**, 1, 21 – 32.
Stroud,I. (1992). Modelling with degenerate objects. *Computer-Aided-Design*, **22**, 6, 344 – 351.
Takala,T. (1992). A taxonomy on geometric and topological models. *Computer graphics and topological models.* Falcidieno,B., Herman,I. and Pienovi,C., Edts., Springer Verlag.
Weiler, K. (1988). The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modeling. *IFIP'88 Geometric Modeling for CAD Applications.* Wozny,M.J., McLaughlin,H.W and Encarnaçao,J.L.,Edts., Elsevier Sci, North Holland.
Weiler, K. (1988). Boundary graph operators for non-manifold geometric modeling topology. *IFIP'88 Geometric Modeling for CAD Applications.* Wozny,M.J., McLaughlin,H.W and Encarnaçao,J.L., Edts., Elsevier Sci, North Holland.
Weiler, K. (1990). Generalized sweep operations in the non-manifold environment. *Geometric modeling for product engineering.* Wozny,M.J, Turner,J.U. and Preiss,K.,Edts., Elsevier Sci, North Holland.
Wu, S.T. (1992). Non-manifold data models: implementational issues. *Proc of MICAD'92*, 37 – 56.
Yamaguchi,Y. and Kimura,F. (1995). Non-manifold topology based on coupling entities. *IEEE Computer Graphics & Applications*, March, 42 – 50.