Checking Boundary Non-Manifoldness of Solid Objects for Steel Construction

E. Tsegaye[1], *F. Werner*[1] *and C.A. Wüthrich*[2]

[1] *Faculty of Civil Engineering*
School of Architecture and Civil Engineering
HAB-Weimar - Universität -
Marienstr. 13
D-99423 Weimar, Germany.
E-mail: {tsegaye,werner}@bauing.hab-weimar.de

[2] *Faculty Informatics and Mathematics*
School of Architecture and Civil Engineering
HAB-Weimar - Universität -
Marienstr. 13
D-99423 Weimar, Germany.
E-mail: caw@informatik.hab-weimar.de

Abstract: One of the major limitations of most of the currently available commercial CAD systems is the fact that they lack the possibility of representing non-manifold solids. This leads to the consequence that big limitations are imposed on the types of solids representable in a system: for example, neither dangling faces are allowed, nor are solids obtained through one-dimensional soldering of steel profiles. This paper presents an algorithm directly derived from the definition of 2-manifoldness which performs local checks on the manifoldness of a solid around one of its vertices. The algorithm builds the face-vertex incidence matrix around a vertex of the solid, and deducts the number of angles topologically equivalent to the open disk incident into the vertex. The algorithm is based on the face-vertex representation of a solid, but can be easily extended to work on traditional edge-based or wing-edge data structures. The method has been developed for GT4Steel, a new CAD system for steel construction which allows to represent and manipulate non-manifold objects.

Keywords: Steel construction, data structures for CAD, non-manifold boundaries.

1. Introduction

For a long time have three-dimensional CAD systems helped the civil engineer in the design process, which starts from elementary forms, and generates increasingly complex objects that build the whole construction. In the course of time, additional features, like collision checks, computation of the volume, structural analysis, i.e. static analysis, stress and deformation analysis, have been added to CAD systems. One of the tasks of Computer Aided Design (CAD) in Civil Engineering is therefore not only the display of the construction elements, but also the analysis of the existing internal forces (momentum, shear force, axial tension force), and the display of the resulting deformations to the user. The functionality can also be extended to include checking of the topological and geometrical characteristics of the resulting object. In fact, for the manipulation of 3D objects, i.e. for their composition, intersection, or for the penetration of distinct objects, knowledge on such characteristics is of great value, since most problems occur where there are topological inconsistencies. For example, a common operation in steel construction is joining (soldering) a number of individual sections to reinforce and stabilize the construction. This process generates new sections having topological inconsistencies along the joining lines. Currently existing CAD systems are not able to handle these cases correctly, and explicitly forbid objects having topological inconsistencies [AUTO94, ALLP94, BOCA94]. However, the example shows that there are many real life cases where it is desirable to be able to handle in the system such solid objects.

All solid modellers implemented in current CAD systems can be roughly classified in two distinct categories. In Constructive Solid Geometry (CSG), objects are obtained as the result of additions and subtractions of simple solids, such as cubes, spheres, cylinders, pyramids etc. [ARBA90]. The problem of this approach is mainly two-fold: on one hand, the flexibility of such an approach is limited, because of the limited choice of basic objects; on the other hand, volume additions and subtractions are not closed operations, that is, it is possible to generate topologically inconsistent objects starting from simple ones. This bears the consequence that continuous checks on the topological consistency have to be performed after each operation to prevent the generation of non-consistent objects, leading to high computational costs. Moreover, no local manipulation on the solids is possible. In Boundary Representation (BRep) systems instead, the solid objects are represented by the surfaces representing their boundary. Since surfaces are composed by an infinite number of points, which is impossible to represent on a computer, the surface is subdivided into its vertices, edges and faces, and such constituting elements, called *simplices*, are used for the internal representation of the surface, which is called *complex* [LUO91]. The main drawback of such a representation is the very high amount of data that has to be used for an object: either every simplex must be present in the data structure, or long linking lists have to be used. Both these forms of representation are quite expensive either because many repetitions of the same data are present, or because each complex is composed by a long linked list of simpler simplices (e.g. a face is composed by its edges, which in turn are composed by vertex lists).

Most of the modern CAD systems try to combine the advantages of both the BRep and CSG approaches: such systems are called *hybrid systems* [BOYS82]. A hybrid solid modeler tries to use the most suitable of the two representations for the task given. However, these systems combine not only the advantages, but also the disadvantages of the representations. Conversions have to be made constantly from one representation into the other, and also here continuous checking of the topological consistency is required. Moreover, the coexistance of the two models induces even higher complexity of the data structure.

It is therefore necessary to point out a data representation form that minimizes the data quantity, and that is trimmed for the checking of the topological consistency of solids. It is highly desirable also to be able to represent in a system also topologically inconsistent solids, as mentioned before in the small example on steel construction.

This paper addresses exactly these two issues: we have developed and implemented a solid modeler called GT4Steel (geometric and topological modeler for steel construction) especially conceived for steel construction. Since the CSG model explicitly excludes solids having no topological consistency, and such solids occur often in steel construction, a BRep representation has been chosen. The new system is therefore based on BRep, but bears two main advantages against it: on one hand, the internal data structure is heavily reduced since the data model used is based on the classical vertex based model, and represents the faces and the vertices of a solid in an incidence matrix. On the other hand, the system allows to recognize non-manifold parts of an object, i.e. with objects having topological inconsistencies, and therefore allows the integration of such objects in the system, and their manipulation. This latter point is of great importance in our field of application. GT4Steel currently allows only planar faces, but is being extended to include free-form surfaces. The method proposed in this paper provides a practically oriented approach to the problem of manifoldness, and is geared towards practical use. For a consistent approach to non-manifold solids, we strongly suggest the reader to read [YAMA95].

The paper is structured as follows: Section 2 data structure models for BRep based solids, Section 3 is an introduction to solid topology, Section 4 presents the incidence matrix data structure and introduces a new algorithm for performing local checks on the topological manifoldness of a solid model. Finally, some concluding remarks and possible extensions are presented in Section 5.

2. The data structure in GT4Steel

From the point of view of the internal data structures of a system, the BRep representation of a polygonal solid is essentially surface based: a solid is described by means of its delimiting complexes, which are in turn described by their components, i.e. by the constituting simplices. All surfaces (polygons) are described through their normal vectors and through the ordered sequence (loop) of its vertices. Taking the example from the excellent book by Mäntyla on solid modeling [MANT88], to describe the parallelepiped shown in Figure 1, a typical solid modeler uses the following data structures: first of all, a list of vertices with their coordinates has to be built; the faces are then represented as the list of their vertices, as shown in Figure 2

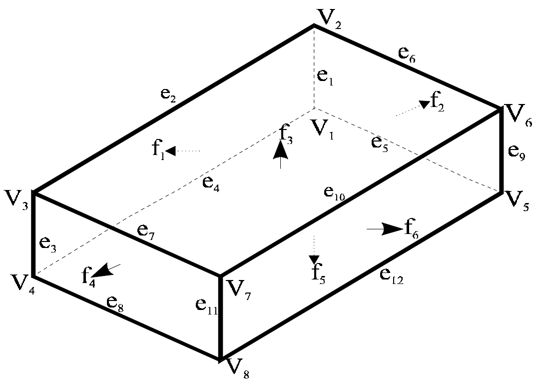

Figure 1:A parallelepiped solid

vertex	coordinates	faces	vertices
v_1	$x_1\ y_1\ z_1$	f_1	$v_1\ v_2\ v_3\ v_4$
v_2	$x_2\ y_2\ z_2$	f_2	$v_6\ v_2\ v_1\ v_5$
v_3	$x_3\ y_3\ z_3$	f_3	$v_7\ v_3\ v_2\ v_6$
v_4	$x_4\ y_4\ z_4$	f_4	$v_8\ v_4\ v_3\ v_7$
v_5	$x_5\ y_5\ z_5$	f_5	$v_5\ v_1\ v_4\ v_8$
v_6	$x_6\ y_6\ z_6$	f_6	$v_8\ v_7\ v_6\ v_5$
v_7	$x_7\ y_7\ z_7$		
v_8	$x_8\ y_8\ z_8$		

Figure 2: Vertex based data structure for simple BRep models

This data structure is called a *vertex-based* model; it is quite simple, but unsuited for manipulation purposes. Note that some information, such as face orientation, or the edges of the solid, is directly deductable from this table. However, the big disadvantage of this representation lies in the fact that edge information is not included: this is essential information, since it is needed to record neighbourhood relations between faces, and to accelerate computations, it is useful to include it in the data structure. This data structure, called *edge-based*, generates for the solid in Figure 1 the data structure scheme of Figure 3.

edge	vertices	vertex	coordinates	faces	edges
e_1	$v_1\ v_2$	v_1	$x_1\ y_1\ z_1$	f_1	$e_1\ e_2\ e_3\ e_4$
e_2	$v_2\ v_3$	v_2	$x_2\ y_2\ z_2$	f_2	$e_9\ e_6\ e_1\ e_5$
e_3	$v_3\ v_4$	v_3	$x_3\ y_3\ z_3$	f_3	$e_{10}\ e_7\ e_2\ e_6$
e_4	$v_4\ v_1$	v_4	$x_4\ y_4\ z_4$	f_4	$e_{11}\ e_8\ e_3\ e_7$
e_5	$v_1\ v_5$	v_5	$x_5\ y_5\ z_5$	f_5	$e_{12}\ e_5\ e_4\ e_8$
e_6	$v_2\ v_6$	v_6	$x_6\ y_6\ z_6$	f_6	$e_{12}\ e_{11}\ e_{10}\ e_9$
e_7	$v_3\ v_7$	v_7	$x_7\ y_7\ z_7$		
e_8	$v_4\ v_8$	v_8	$x_8\ y_8\ z_8$		
e_9	$v_5\ v_6$				
e_{10}	$v_6\ v_7$				
e_{11}	$v_7\ v_8$				
e_{12}	$v_8\ v_5$				

Figure 3: Edge-based data structure.

Many popular solid modeling systems expand further this data structure to include face-face neighbourhood information and by representing the loop information in edge nodes. The new data structure is called wing-edge data structure, and is shown in Figure 4. Here, edge information is completed by adding to the edge data structure the two-edges ncw and nccw leaving the endpoint of an edge.

edge	vstart	vend	ncw	nccw	vertex	coordinates	face	first edge	sign
e_1	v_1	v_2	e_2	e_5	v_1	x_1 y_1 z_1	f_1	e_1	+
e_2	v_2	v_3	e_3	e_6	v_2	x_2 y_2 z_2	f_2	e_9	+
e_3	v_3	v_4	e_4	e_7	v_3	x_3 y_3 z_3	f_3	e_6	+
e_4	v_4	v_1	e_1	e_8	v_4	x_4 y_4 z_4	f_4	e_7	+
e_5	v_1	v_5	e_9	e_4	v_5	x_5 y_5 z_5	f_5	e_{12}	+
e_6	v_2	v_6	e_{10}	e_1	v_6	x_6 y_6 z_6	f_6	e_9	-
e_7	v_3	v_7	e_{11}	e_2	v_7	x_7 y_7 z_7			
e_8	v_4	v_8	e_{12}	e_3	v_8	x_8 y_8 z_8			
e_9	v_5	v_6	e_6	e_{12}					
e_{10}	v_6	v_7	e_7	e_9					
e_{11}	v_7	v_8	e_8	e_{10}					
e_{12}	v_8	v_5	e_5	e_{11}					

Figure 4: Wing-edge data structure.

Note that this data structure is possible only at vertices where the solid is topologically consistent, i.e. at points which are the intersections of three polygons, since the outgoing edges are not necessarily two where, for instance, there is a face hanging off the solid.
In GT4Steel we have chosen not to use the winged-edge structure, since the restriction on the type of solids used was too heavy for this special application field. In fact, the implementation uses the vertex based structure, although the algorithm presented below can be extended to edge-based data structures.

3. On the topology of solids

In the former sections, the concept of topological inconsistency has been mentioned. This section presents a sufficient condition often used in CAD systems for checking the acceptability of a solid for its processing in the system, and the formal definition of a manifold solid. For a more detailed introduction to topology issues the reader is encouraged to read [DUBR85, BOUR80]

Consider a set X, and a set A of subsets of X (called *open sets* of X) such that

i) X and $\emptyset \in X$
ii) the union of any collection of elements of A is an element of A
iii) the intersection of a finite number of elements of A is an element of A

is called a *topology* in X.

In the set topological space X, the *neighbourhood* of a subset SÍ X is any subset of X containing an open set B containing S. A base for the topology in the topological space X is any set W of open sets of X such that every open set of X is the union of sets belonging to W. For example, in the Euclidean space E^n, the open spheres centered in a point p, i.e. the sets N(p,r) of the form $N(p,r)=\{x\in E^n: d(x,p)<r\}$, constitute a basis for the usual topology and are the most commonly used neighbourhoods of the point p.

A point p is called *accumulation point* of a set S, if any neighbourhood of p contains at least one point of S. A set C is called closed if its complement is an open set. Given a subset S of X, its *boundary* is the set of accumulation points both of S and of its complement. Given a subset S of X, its *closure* is the set S plus its accumulation points. The interior of a set S is the set of points of S that are accumulation points of S but not of its complement. The exterior points are points of S that are accumulation points only for the complement of S, but not accumulation points for S.

A *map f*: B®B' from a set B to a set B' is a relation associating points of B to points of B'. A map is said to be *injective* if every point of B has an image in B', *surjective* if each point of B' has at least one counterimage in B. It is said to be *bijective* or a *one-to-one* mapping if it is both injective and surjective.
A map *f*: X®Y of a topological space X into the topological space Y is said to be *continuous* if the inverse image of any open subset of Y is an open set in X. A mapping is said to be *homeomorphic* if it is continuous and one to one. If the mapping is locally homeomorphic but not necessarily a bijection then it is said to be an *immersion*.

Two sets are said to be *homeomorphic* if there is a homeomorphic map between them. Two homeomorphic sets are considered topologically equivalent. Consider the Euclidean space E^n, and consider a set S: S is said to be *n-manifold* if each of its points has a neighbourhood in S that is homeomorphic to an open set in E^n. For example, a single point is 0-manifold, since it is homeomorphic to a single point; the border lines of a polygon are 1-manifold since they are homeomorphic to a line; an open portion of a surface is 2-manifold since it is homeomorhic to an open disk, and the interior cube in the three-dimensional space is 3-manifold since it is homeomorhic to an open sphere. In other words, the number n represents the inherent dimensionality of the set in question. For a polyhedron, its boundary must be two-manifold in all of ist points to separate its interior part from its exterior.

In common CAD systems, the topological consistency of solids is checked through the Euler-Poincaré formula. For polyhedra this formula relates the number v of vertices, the number e of edges and the number f of faces of the surface S of a solid as

$$v-e+f=c(S).$$

The constant c(S) on the right hand of the equation is called the Euler characteristic of the solid. For a simple polyhedron, i.e. for a solid having connected faces and no holes, c(S)=2. From the theory of homology it is known that the Euler characteristic can be expressed as

$$c=h_0-h_1+h_2,$$

where h_0, $h_1$and h_2 are called the Betti numbers of the solid. Here, h_0 is the number of connected components of the solid, h_1 is called the connectivity number of the surface, is equal to the largest number of closed curves that can be drawn on the surface of the solid without disconnecting it, which in turn is dependent on the number of holes in the 3D solid, while h_2 reveals the orientability of the solid surface, and is the number of hollows enclosed by two-dimensional surfaces. Intuitively, the m-th Betti number indicates the number of holes surrounded by m-dimensional loops in the object.

However, checking the solid through Euler's formula is sometimes not sufficient, especially for practically oriented work since the formula works only on solids having
a 2-manifold surface. For example, for the solid depicted in Figure 5, we have f=6, e=12 and v=8, which leads to f-e+v=2 as in the case of a simple polyhedron. Euler's formula gives us no information on the underlying topological characteristics of the solid.

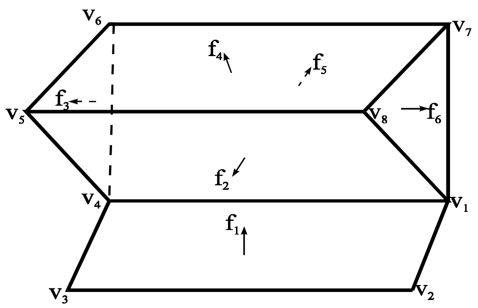

Figure 5: A non-2-manifold complex

Since also cave solids are often used in steel construction, we will deal from now on with complexes that are defined through planar facets which might or not enclose 3-manifold volumes. The purpose of this study will be therefore to study the 2-manifoldness of the defining surfaces to deduce information on the 3-manifoldness of the enclosed solid.

In the next section, we are going to introduce a new method for checking the manifoldness of a solid which has the great advantage of being both local and of working on a generic BRep defined complex.

4. Checking local manifoldness

In order to be able to handle in a CAD system non-2-manifold solids, a method has to be developed which first of all is capable of recognizing potentially non-manifold vertices, and which is then able to check locally the manifoldness of a vertex. Both purposes can be achieved by representing the data in a vertex-face incidence matrix. In this matrix, each row represents a face, while each column represents a vertex of the solid object. In its simplest form each matrix element m_{ij} is set to one if the j-th vertex v_j belongs to the face f_i of the solid, while it is set to zero otherwise. For the solid of Figure 1, therefore, the related incidence matrix is the following:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
f_1	1	1	1	1				
f_2	1	1			1	1		
f_3		1	1			1	1	
f_4			1	1			1	1
f_5	1			1	1			1
f_6					1	1	1	1
$n_F(V_j)$	3	3	3	3	3	3	3	3

Table 1: The incidence-matrix

Note that this kind of representation does not allow solids with dangling edges. In the last row of the table, the sum $n_F(v_j)$ of the total number of faces incident in the corresponding vertex is computed. This is simply the total sum of the elements of the column. In the case of a vertex where the solid the surface of which is locally 2-manifold and where its adjacent faces are non-parallel, $n_F(v_j)$ must be at least three, since there must be at least three polygons incident into a vertex to build a neighbourhood on the surface of the solid which is homeomorphic to a open set of the plane. If the interiors of all polygons are mutually non-intersecting, then all vertices for which $n_F(v_j)=3$ are 2-manifold. Local 2-manifoldness has to be therefore checked in the surroundings of all vertices for which $n_F(v_j)\neq 3$. Table 2 shows the incidence matrix of the non-regular solid of Figure 5: here the vertices v_1, v_2,, v_3 and v_4 have to be checked. For these vertices, therefore, the topological structure around the vertex has to be reconstructed to point out whether or not there exists a local homeomorphism with an open set of the plane.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
f_1	1	1	1	1				
f_2	1			1	1			1
f_3				1	1	1		
f_4					1	1	1	1
f_5	1			1		1	1	
f_6	1						1	1
$n_F(V_j)$	4	1	1	4	3	3	3	3

Table 2: The incidence matrix for the solid in Figure 5

Performing the reconstruction of the topological structure of a neighbourhood of a point V_α is a relatively straightforward task. All that is required is to gradually reconstruct the portions of plane incident in the point in question which are delimited by two edges of the solid, and to check whether their chaining builds a set homeomorphic to the open disk. This is a task easily archieved on the incidence matrix. In fact, there will be an edge $V_N V_M$ shared by the two faces f_m and f_n when two vertices V_N and V_M are both present in the rows corresponding to f_m and f_n. To prevent potential problems, we assume hereafter that all adjacent polygons share in the data structure the same endpoints of the common edges, i.e. that whenever two adjacent polygons share a portion of an edge all the endpoints of this edge appear in the incidence matrix in both rows corresponding to the adjacent polygons. For example, for the configuration shown in Figure 6 all the vertices v_3, v_5 and v_2 must appear in the incidence matrix in the lines corresponding to both faces F_1 and F_2.

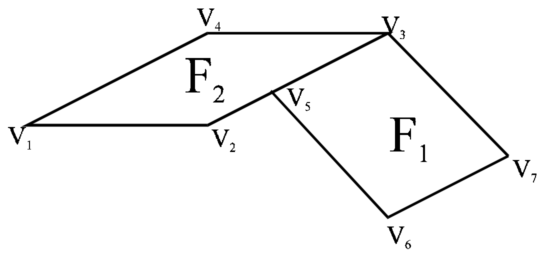

Figure 6: Two adjacent polygons have to share entire edges

From the incidence matrix the submatrix of all faces incident in V_α is extracted. This submatrix is the local incidence matrix $LM(V_\alpha)$ in V_α. Note that in each row of $LM(V_\alpha)$ the position corresponding to V_α is always set to one. To obtain the edges which delimit the portions of a plane incident in the point V_α, it is sufficient to look in the local incidence matrix for the presence of common vertices (beside V_α) in two different faces, i.e. on two different rows. These faces will build an edge between them which will end in V_α. To obtain all edges incident in V_α thus, the matrix $LM(V_\alpha)$ has to be scanned for all vertices V_b appearing in two or more rows. These endpoints are the second endpoints of the edges $V_\alpha V_b$ of the solid which are incident in V_α. Note that each such edge $V_\alpha V_b$ belongs to at least two faces, but may belong to more faces. The list L of all edges $V_\alpha V_b$ is then constructed, and for each of its edges e the list L_e of all faces incident into it. Whenever a face appears in two different L_e lists, say in L_{e_1} and L_{e_2}, the corresponding edges share this common face $F_{e_1 e_2}$. Such a face $F_{e_1 e_2}$ builds therefore around V_α a planar non-zero angle. Let $\varphi_{e_1 e_2}$ be such angle. The measure of this angle is irrelevant for checking local manifoldness, and so is the order in which edges appear as indices of the notation of the angles: what is important is the fact that such a planar angle exists. Consider the list F of all such angles $\varphi_{e_i e_j}$: starting from an arbitrary member of this list, it is now possible to build a list of consecutive angles, i.e. of angles of F sharing a common edge $F_1=\{ \varphi_{e_{\kappa 1} e_{\kappa 2}}, \varphi_{e_{\kappa 2} e_{\kappa 3}}, \ldots \}$. Whenever one edge is added to the list F_1, it is removed from the list F. If the starting edge e_{κ_1} is reached, the list F_1 is closed, and a new list F_2 is opened. This process continues until all angles of F are exhausted. Each list F_m represents a set homeomorphic to the open disk. The surface of the solid is 2-manifold in V_α if there is exactly one closed sublist F_1. If there are more closed sublists, then the point is the joining point of more 2-manifold solids in V_α. If there are non-closed sublists, then at least one of the faces entering in V_α is a dangling face. The algorithm presented above will be called the *incidence-matrix algorithm*. Figure 7 illustrates the incidence matrix algorithm.

Note that due to the fact that the vertex-face matrix does not contain edge information, and thus is not capable to represent dangling edges, the proposed algorithm cannot recognize dangling edges in V_α. In order to be able to represent dangling edges, an edge based data structure has to be used. Here dangling edges can be easily recognized directly in the data matrix. Note that the incidence matrix algorithm can be also easily adapted to work starting from an edge-based data structure.

- Build the incidence matrix
- Choose vertices v_α such that $n_F(v_\alpha)\neq 3$
- FOR each v_α:
    - build the local incidence matrix $LM(v_\alpha)$
    - Search in $LM(v_\alpha)$ for vertices v_α appearing in two different lines: $v_\alpha v_\beta$ build incident edges in v_α.
    - Build list of edges L
    - For each edge e in L, build list L_e of faces incident in it
    - Search for faces apperaring in different L_e: such faces build around v_α a non-zero angle $\varphi_{e_i e_j}$
    - Build list Φ of all angles $\varphi_{e_i e_j}$
    - p:=0
    - UNTIL Φ is empty REPEAT
        - Build new list Φ_p.
        - choose one element $\varphi_{e_{\kappa_1} e_{\kappa_2}}$ of Φ, add it to Φ_p and delete it from Φ.
        - UNTIL edge e_{κ_1} is reached REPEAT
            - Choose consecutive $\varphi_{e_{\kappa_2} e_{\kappa_3}}$, add it to Φ_p and delete it from Φ.

          ENDUNTIL
        - p=p+1
    - ENDUNTIL
    - The neighbourhood of v_α is homeomorphic to (p-1) open disks
- ENDFOR

Figure 7: Pseudocode for the incidence matrix algorithm.

5. Conclusions

In this paper, we have presented a practically oriented algorithm for performing local checks on the manifoldness of a solid. The algorithm works on the vertex-face incidence matrix of a solid, and is able to recognize the topological configuration around a vertex, i.e. to distinguish in the single vertices of the solid whether there are dangling faces hanging at that vertex, or whether the vertex is a joining point of two or more manifold solids. The algorithm is relatively simple and proceeds in a constructive way by trying to build a homeomorphism between the vertex neighbourhood and one or more open disks, without making any assumptions on the orientation of the faces of the solid. In this sense, it is a direct derivation of the definition of manifoldness. The new algorithm is being implemented in a CAD system for steel construction.

The new method allows easy checks of the local topology, and can be easily used for acquiring local knowledge on the topological configuration of a solid, so as to speed up intersection tests between non-manifold solids.

Further research is currently being performed in two directions: on one hand, the algorithm is being extended to work on solids having non-linear (i.e. curved) boundaries. On the other hand, fast algorithms for performing the usual boolean operations on non-manifold objects are being studied so as to extend the class of usable solids in a CAD system.

6. Acknowledgement

The authors wish to thank Prof. Beucke of the Hochschule für Architektur und Bauwesen in Weimar for his helpful comments in the revision process of this paper.

7. References

[ARBA90] Arbab, F., ”Set Models and Boolean Operations for Solids and Assemblies”. *IEEE Computer Graphics and Applications,* Vol. 10, No. 6, 1990, PP76-86.
[ALLP94] *Allplan Handbuch*, Nemetschek Programmsystem GmbH, Munich, Germany, 1994.
[AUTO94] *Autocad Release 13 Benutzerhandbuch,* Autodesk Development B.V., Neuchatel, Switzerland, 1994.
[BAUM75] Baumgarten, B.G., ”A Polyhedron Representation for Computer Vision” *AFIP Conference Proceeding*, N.Y, Vol. 44, 1975, pp. 589-596.
[BOCA94] *BOCAD-3D Benutzerhandbuch*, Bocad Software GmbH, Bochum, 1994.
[BOUR80] Bourbaki, N., *General Topology*, Springer Verlag, Berlin, 1980.
[BOYS82] Boyse, J.W. and Gilchrist, J.E., ”GMSolid: Interactive Modeling for Design and Analysis of Solids”. *IEEE Computer Graphics and Applications,* Vol. 2, No. 2, 1982, PP27-40.
[DESA92] Desaulniers, H., and Stewart, N.F., ”An Extension of manifold Boundary Representation to the r-Sets”. *ACM Transactions on Graphics*, Vol.11, No 1, 1992, PP 40-46.

[DUBR84] Dubrovnin, B.A., Fomenko, A.T. and Novikov, S.P., *Modern Geometry: Methods and Applications*, Springer Verlag, Berlin, 1984.

[GRUS90] Gursoz, E.L., Choi, Y. and Prinz, F.B., ”Vertex-based Representation of Non-Manifold Boundary”, in *Geometric Modeling of Pruduct Engineering* , Wozny M.J., Turner J.u. and Preiss K. (eds), IFIP WG, North -Holland, 1990, PP. 107-130.

[GURS91] Gursoz, E.L., Choi, Y. and Prinz, F.B., ”Boolean Set Operations on Non-Manifold Boundary Representation Objects”, *Computer-Aided-Design* Vol 23, No 1, 1991, PP. 33-39.

[LUO91] Luo, Y. and Luka´cs, G., ”A Boundary Representation for Form Features and Non-Manifold Solid Objects”. In *Proceedings of the first ACM/SIGGRAPH Symposium on Solid Modeling Foundations and CAD /CAM Applications.* USA Austin,Texas 1991,PP45-60.

[MÄNT88] Mäntylä, M., ”An Intruductin to Solid Modeling ” *Computer Science Press,* Maryland, 1988.

[REQU80] Requicha, A.A.G., ”Representations for Rigid Solids: Theory, Methods and Systems”, *ACM Computing Surveys*, Vol. 12, No. 4, 1980.

[REQU85] Requicha, A.A.G. and Voelcker, H.B., ” Boolean Operations in Solid Modeling: Boundary Evaluation and Merging Algorthms”. *IEEE Computer Graphics and Applications,* Vol. 73, No. 1, 1985, PP30-44.

[PAOL93] Paoluzzi, A., Bernardini, F., Cattani C. and Ferrucci, V., ”Dimension independent modelling with simplicial complexes” *ACM Transactions on Graphics*, Vol.12, 1993, PP. 56-102

[SHAP94] Shapiro, V., ”Real Functions for representation of rigid solids” *Computer Aided Geometric Design,* Vol. 11, 1994, PP. 153-175.

[WEIL85] Weiler, K., ”Edge-Based Data Structures for Solid Modeling in Curved- Surface Enviroments”. *IEEE Computer Graphics and Applications,* Vol. 5, No. 1, 1985, PP. 21-40.

[WEIL88a] Weiler, K., ”The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modeling”. In Wozny M.J., McLaughlin H.J., Encarnaçao J.L. (eds), *Geometric Modeling for CAD Applications*, Elsevier Science Publishers, North Holland, 1988, PP. 3-36.

[WEIL88b] Weiler, K., ”Boundary Graph Operators for Non-manifold Geometric Modeling Topology Representations”. In Wozny M.J., McLaughlin H.J., Encarnaçao J.L. (eds), *Geometric Modeling for CAD Applications,* North -Holland, 1988, pp. 37-66.

[YAMA95] Yamaguchi, Y. and Kimura, F., ”Nonmanifold Topology Based Coupling Entities”. *IEEE Computer Graphics and Applications,* Vol. 15, No. 1, 1995, PP. 42-50.