INTEGRATING CURVES AND SURFACES IN B-REP SOLID MODELLERS USING TOPOLOGICAL INFORMATION

Wang CONGLI¹, Fabio K. Takase¹ and Marcos de S. G. TSUZUKI¹
¹University of São Paulo, Brazil

ABSTRACT: Historically, the representation techniques are classified in two groups: Solid Modelling and Geometric Modelling. Recently, in some commercial CAD systems, where Geometric Modelling techniques are combined with Solid Modelling techniques to represent computer models, the final model is not assured to be a valid Solid Model. In this article a new data structure will be defined to support the representation of curves and surfaces in B-Rep solid models. This new data structure will support synchronization between the geometry of the Geometric Model and an approximated polyhedral Solid Model. An approximated polyhedral Solid Model is obtained allowing the calculation of mass properties using several existent algorithms in the literature. And it will be possible to reduce the approximated polyhedral for a minimum representation whenever necessary. The face, in commercial Solid Modelers, has two functions: it represents the boundary of the solid and represents the geometrical shape of the contour. Faces with three, five or more sides are not commonly supported by commercial CAD systems. Usually, those faces are divided in a number of four sided faces. That fact is an inconsistency between topology and geometry, as the number of sides in a face is defined by topology and geometry usually "forces" a face to have four sides. The control points from curves and surfaces are associated to vertices in the B-Rep data structure. This association allows the determination of several adjacency relationships information among curves, surfaces, vertices, edges and faces without the necessity of numerical comparison, as the information is explicitly stored in the data structure.

Keywords: Solid Modeling, Geometric Modeling
1. INTRODUCTION
Solid modelling and geometric modelling have been developed separately as pointed by Farin [6]. It is very common to associate polyhedral solids to solid modelling; and curves and surfaces to geometric modelling. Polyhedral models have some disadvantages. First, the significant topological features of the model are obscured - for instance, a simple cylindrical hole is represented by many faces. Second, the geometry of the model is approximated - the level of geometric accuracy depends on the number of faces used to approximate each curved face. In the other hand, geometric models do not have adjacency information, also known as topological information. In the literature we have very few proposals to integrate these two topics. Toriya and Chiyokura [14] developed a curved solid modeler at Ricoh, and they considered curves as attributes of edges and surfaces as attributes of faces. Mäntylä [13] described a similar approach used in the GWB solid modeler developed at Helsinki University of Technology. Turner [15] proposed a method to generate polyhedral solids from geometrical information, where the geometry is processed as an attribute of faces and edges. The support to both technologies solid modelling and geometric modelling is a must according to the actual technological needing. For some purposes the polyhedral representation is necessary, it assures that the solid is closed with coherent orientation, then it is possible to calculate mass properties among other properties. The exact representation of curves and surfaces is very necessary to have precise access to the exact shape representation.
In this work, we will propose a data structure to integrate both technologies: solid modelling and geometric modelling. In this integration, topological information will be available among curves and surfaces. It will be proposed a synchronization method where it will be possible to generate a polyhedral solid as an approximation for a free form surface, using the development operator. And it will be possible to reduce the approximated polyhedral solid model to a minimum representation whenever necessary using the shrink operator.

2. GEOMETRIC MODELING
Curves and surfaces have a very important role in engineering design and manufacturing. Chiyokura and Kimura [5] have shown that faces with three, five and six sides can be created in a solid when rounding operations are done [4,7,12,16,17,18]. Hosaka and Kimura [8] demonstrated the algebraic expressions to represent such free form surfaces. However, this is not the common approach, as the polynomial expression becomes very complex. In the literature, we found a better approach where a face with n-sides can be transformed into n faces with four sides. Figure 1 shows an
example where a face with three sides is transformed in three faces with four sides. This technique has a collateral effect that is the creation of three extra points \((P_{3,0}, P_{2,0}, \text{and} P_{1,0})\) in the boundary of the original face. This way, the number of sides of the adjacent faces is increased by one. We call such special situation as T-node.

Chiyokura and Kimura [4] realized a post processing to define the geometrical model based on the solid model. This way, the solid model is not affected by the additional vertices. This process is shown in Figure 2. Figure 2.(a) shows the original B-Rep solid model with some curved edges. Figure 2.(b) shows the post-processed surface model, with the creation of three extra four sided surfaces. Figure 2.(c) shows the generated mesh, where the faces were divided in a 4x4 fashion. It is possible to observe that the mesh does not fit at some curves, generating holes. Some vertices from one surface’s border does not fit the vertices of the adjacent surface’s border.

When adjacent information among surfaces and curves is introduced, those holes can be eliminated.

3. SOLID MODELLING

B-Rep solid models emerged from the polyhedral models used in computer graphics for representing objects and scenes for hidden line and surface removal. They can be viewed as enhanced graphical models that attempt to overcome the problems of graphical models by including a complete description of the bounding surface of the object. There are three primitive entities face, edge and vertex, and the geometric information attached to them forms the basic constituents of B-Rep models. In addition to geometric information such as face equation and vertex coordinates, a B-Rep model must also represent how the faces, edges and vertices are related to each other.

According to Mäntylä [13], it is customary to bundle all information of the geometry of the entities under the term geometry of a boundary model, and, similarly information of their interconnections under the term topology. It is
possible to say that the topology is a glue that tie together the geometry. This topological information can be derived through some numerical techniques; however, the computational cost is very high and numerical precision problems can arise.

The first B-Rep based solid modeler was implemented using the winged edge data structure [1]. However, during the past twenty years several data structures have been proposed to implement the B-Rep representation. The proposed data structures have in common the fact that they are based on the edge. In a curved environment where it is possible to an edge to be adjacent to two identical faces or to two identical vertices, the access to the topological information must be made with special care. Weiler [19] adapted the winged-edge data structure to incorporate a mechanism to access the topological information even in such a curved environment. With the objective of algorithm simplification, mainly in the determination of the circuit of edges surround a face, the half-edge entity was created. It was observed that the edge in the original winged-edge data structure had two main functions: represent the circuit of edges surround the face and to represent the real edge. The algorithm to determine the circuit of edges surrounds a face was very complex with several rules. Some researchers observed that separating these two functions, the algorithm can become much simpler. This way, in the modern solid modelers we have one entity to represent the edge itself and another entity to represent the circuit of edges surrounds the face, usually called as halfedge (see Figure 3). A loop represents a sequence os halfedges. A face must have one external loop defining its contour, and can have zero or more internal loops defining holes of a face, like protrusions or depressions.

### 3.1 Euler Operators

The Euler Operators were proposed to simplify the manipulation of the B-Rep data structure. The main concept behind the Euler Operators, is the construction of the B-Rep solid model incrementally, step by step, such that the complexity details of the data structure will be encapsulated by the operators. We will differentiate the internal loops from the external loops calling the internal loops as rings and the external loops will be called by face. For historical causes, the Euler Operators are defined using the following convention:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>make</td>
</tr>
<tr>
<td>K</td>
<td>kill</td>
</tr>
<tr>
<td>V</td>
<td>vertex</td>
</tr>
<tr>
<td>E</td>
<td>edge</td>
</tr>
<tr>
<td>F</td>
<td>face</td>
</tr>
<tr>
<td>S</td>
<td>solid</td>
</tr>
<tr>
<td>H</td>
<td>hole</td>
</tr>
<tr>
<td>R</td>
<td>ring</td>
</tr>
</tbody>
</table>

For example, the MEV operator is translated by Make Edge and Vertex, where an edge and a vertex are created. It is possible to demonstrate that a set of five operators is sufficient to create...
any solid according to the B-Rep data structure presented in Figure 2. Several Euler Operators can be defined; however, considerations of modularity and independence created few variations found in the literature. The set of Euler Operators chosen for this work is described below [13]:

1. **MVSF (Make Vertex Solid Face)**: that is the first operator that must be applied to initiate a solid.
2. **MEV (Make Edge Vertex)**: it inserts an edge and a vertex.
3. **MEF (Make Edge Face)**: it divides a face creating a new edge.
4. **KEMR (Kill Edge Make Ring)**: it divides the contour of a face in two components deleting a bridge edge.
5. **KFMRH (Kill Face Make Ring Hole)**: it glues two faces such that the boundary of the first face becomes the boundary of the second face.

Associated with those five Euler Operators we have five complementary Euler Operators.

1. **KVSF (Kill Vertex Solid Face)**: that operator is the last operator that must be applied when removing a solid.
2. **KEV (Kill Edge Vertex)**: it kills an edge and a vertex.

3. **KEF (Kill Edge Face)**: it kills a face removing an edge.
4. **MEKR (Make Edge Kill Ring)**: it composes two boundaries of a face creating a bridge edge.
5. **MFKRH (Make Face Kill Ring Hole)**: it creates a new face from the boundary of another face.

During the construction of B-Rep model using the Euler Operators, the topology is maintained valid. At the end of a sequence of Euler Operators it is assumed that the geometry of the solid is correct; however, during the intermediary stages it is not possible to maintain the topology and the geometry consistent. As a consequence, the Euler Operators are not safe operators, and they must be collectionated in sequences with a proper meaning. Perhaps this is why, no solid modeler support Euler Operators as Functions to the end user.

4. **PROPOSED DATA STRUCTURE**

Mäntylä [13] and Chiyokura [14] represented curves as edge’s attributes. However, whe a T-node is explicitly represented in the data structure, the curve information must be associated to the halfedge element and not to the edge element. Figure 4 shows that a curve can be associated to more than on halfedge.
This way, a surface is defined by four curves, independently from the number of halfedges in its contour. Our proposed data structure is shown in Figure 5, and it is possible to observe that the surface’s and curve’s control points are associated to the B-Rep solid model’s vertices. B-Rep solid modelers data structure store the relationships among the primitives elements: vertex, edge and face. This way, it is possible to obtain all the nine types of adjacency relationship. As control points are associated to vertices, this type of information is extended to curves and surfaces. This will solve problems like the determination if the curves surrounding a vertex have a common normal vector, or the surfaces surrounding an edge have $C^1$ continuity. This type of information can be derived by numerical comparison, but this is not the best approach. Usually, a curve is represented as an attribute of the edge, then it should be enough to determine the edges surrounding a vertex and consequently the associated curve can be determined. However, it is necessary to group the control points. Usually, this process is done by numerical comparison.

5. CURVES AND HALFEDGES
The work presented by Mäntylä [13], Toriya and Chiyokura [14] and Turner [15] showed the same approach to represent the geometry of curves and surfaces as an attribute of edges and faces. Thus, each edge has a pointer to a record specifying the exact mathematical representation of the underlying curve. This data structure gives full compatibility with the original half-edge data structure. It is possible to draw the solid traversing the list of edges, it is possible to obtain the adjacency information directly from the data structure, and etc. Turner [15] proposed that a curve can be approximated by a number of edges; however, in some situations it is necessary to increase the number of approximating edges or even decrease such number. Figure 6 shows an example where a cylinder is initially created with four edges approximating. In the middle, each curve is associated to only one edge and in the last slide each curve is approximated by eight edges. We need just two operators: one to compress the memory used to represent the curve, and another to approximate the curve by a number of edges. Using our proposed data structure, such operators can be easily implemented using sequences of Euler operators.

![Figure 6. Mechanism showing how to increase the number of approximating edges.](image_url)
The development operator will approximate the curve by a number of line segments, as shown in Figure 6. Every curve has a start and an end, turning necessary to know which vertex, from the associated original edge, corresponds to the starting control point and which vertex corresponds to the ending control point. Let’s suppose that our curve is a Bézier curve expressed as follows:

$$ Q(t) = \sum_{i=0}^{n} B_i^n(t)P_i. $$  \hfill (1)

The starting control point is $P_0$ and the ending control point is $P_n$. The curve can be evaluated at any position $0 \leq t \leq 1$. In our data structure, the information of which is the vertex associated to the starting control point of a given curve is explicitly available. Then, the development operator is easily implemented as a sequence of MEV Euler Operators with the coordinate of the new vertex.

Each approximating halfedge has a pointer to the original curve attribute, showing that it was originated from the same geometry. When the compress operator is applied it is necessary to traverse the circuit of half-edges surround the face searching for edges with the same geometric attribute, then a KEV Euler Operator is applied. It will remain only one edge associated to such geometric attribute.

6. SURFACES AND FACES
We can extend the concept presented for curves to surfaces. Several faces can point to the same surface, showing that it originated from the same geometry. It is necessary to know which vertex in the B-Rep data structure corresponds to a given control point. This way it is possible to define an algorithm based on MEV and MEF Euler Operators, as shown in Figure 7, that implements the development operator for a surface. The compress operator will remove...
edges with adjacent faces associated to the same surface tag using the KEF Euler Operator, and edges adjacent to the same surface at both sides using the KEV Euler Operator.

7. TOPOLOGY AMONG CURVES AND SURFACES
The explicitly representation of the association between control points and vertices allows the supply of several adjacency relationship information without any numerical comparison. We can enumerate seven new adjacency relationships:

- **V<C>**: which curves are adjacent to a given vertex. Figure 8 shows an example of V<C> adjacency relationship. Given a vertex it is possible to determine its adjacent halfedges and consequently its adjacent curves;
- **V<S>**: which surfaces are adjacent to a given vertex. (see Figure 9);
- **C<V>**: which vertices are adjacent to a given curve (see Figure 10);
- **C<C>**: which curves are adjacent to a given curve (see Figure 11);
- **C<S>**: which surfaces are adjacent to a given curve (see Figure 12);
- **S<V>**: which vertices are adjacent to a given surface (see Figure 13);
- **S<C>**: which curves are adjacent to a given surface (see Figure 14);
- **S<S>**: which surfaces are adjacent to a given surface.

Such information is very important to check the presence of continuity in the geometry of the solid model.

8. CONCLUSIONS
We have shown that polyhedral approximations
may be synchronized to geometrical representation. The basis of the approach is a routine that calculate the polyhedral approximation and another to compress the data structure removing all approximating entities. We resulted in simplified topological structure where a face does not interfere in its neighboring faces when geometry is associated.

ACKNOWLEDGMENTS
The second author was partially supported by CNPq.

REFERENCES

ABOUT THE AUTHORS

1. Wang Congli, Master, is a Dr candidate at the Department of Mechatronics and Mechanical System Engineering, School of Engineering, University of São Paulo, Brazil. His research interests are in CAD/CAM, Computer Graphics and Computational Geometry.

2. Marcos de S. G. Tsuzuki, Ph.D., is an Associate Professor in the Department of Mechatronics and Mechanical System Engineering, School of Engineering, University of São Paulo, Brazil. His research interests are in CAD/CAM, Computer Graphics and Computational Geometry. He can be reached by e-mail: mtsuzuki@usp.br.

3. Fabio K. Takase, PhD, is a Lecturer in the Department of Mechatronics and Mechanical System Engineering, School of Engineering, University of São Paulo, Brazil. His research interests are in CAD/CAM, Computer Graphics and Computational Geometry. He can be reached by e-mail: fktakase@usp.br.