9

Solid Modeling as a Framework in Virtual Environments

M. Figueiredo, J. Teixeira
Grupo de Métodos e Sistemas Gráficos
Departamento de Matemática - FCTUC
Apartado 3008, 3000 COIMBRA
email: mauro/teixeira@mat.uc.pt

Abstract

The new technology of virtual environments provide to a visitor of a virtual world a high degree of immersion. On the other hand, the interaction techniques available in these virtual worlds are still rudimentary. Frequently, the visitor is enabled to grab 3D objects by gesturing, but unable to touch a face, edge or a vertex, neither edit its shape interactively.

In this paper we present the reasons that let us believe that boundary representation data structures should be integrated in virtual environments. They can be used not only to provide a fast rendering, but mainly to take advantage of the explicit geometrical and topological data of 3D models. This data enabled us to improve the interaction component of a virtual environment prototype by letting visitors to locally edit in real time the shape of virtual objects and, in this way, augmenting the visitor participation in a virtual world.

Keywords

Virtual environments, interaction techniques, solid modeling, boundary representations.

1 INTRODUCTION

Virtual environments try to provide to the user the illusion of being immersed and participating in a virtual scenario. To achieve these goals, the visitor is enabled to navigate inside a virtual world (that looks like a real one) and to interact with virtual objects.

Many experiments have been reported about successful prototypes that generate the illusion of immersion. Remaining problems to be solved, as lack of resolution, are mainly technological. In general, a high degree of immersion can be achieved by providing stereoscopic views of a given environment.

J. Rix et al. (eds.), *Virtual Prototyping*

The sensation of participation is currently generated by providing to the visitor a set of gestures to perform actions, such as grab, push and release. In this way, the participant is enabled to manipulate 3D objects in a direct format by grabbing, pushing or releasing them.

In some virtual environment systems, these 3D interaction techniques are only supported by bounding volumes. In this case, visitors manipulate 3D virtual objects by interaction with bounding boxes parallel to the coordinate axes. Thus, the visitor is unable to touch the surface of the virtual object and his actions upon the environment (grabbing, pushing, others) are inaccurate.

However, in many real tasks the user need to touch 3D objects or select its geometrical entities (vertices, edges and faces). For example, for tele-presence applications it is extremely important to let users accurately grab, push and position objects. In CAD, it is also important to select vertices, edges or faces of a 3D virtual object (for example to simulate the interactive distortion of object's surfaces done by a virtual hand).

This paper addresses these questions. We explain the approach that we have taken to enable the visitor to directly interact with the surface of virtual objects or to select its vertices, edges or faces, in a virtual environment.

2 REQUIREMENTS FOR GEOMETRICAL MODELS IN VIRTUAL ENVIRONMENTS

The representation of 3D objects' geometry is an important step in the graphical simulation of reality, since they have to describe real objects and provide the required data for the realistic simulation of the original entities and their behaviour.

The geometric modelling and its related representation schemes are already well studied (Baer, 79), (Requicha, 80), (Weiler, 86), (Mäntylä, 88): wire frame, surface, constructive solid geometry (CSG), manifold and non-manifold boundary representations (Breps). Each of these representations have pros and cons that characterise and enable them to be used in a specific application context: product modelling, virtual environments, etc. In order to distinguish them, Requicha (Requicha, 80) proposed a set of properties to characterise solid representation schemes: domain, validity, completeness, unambiguity, uniqueness, conciseness, efficiency, ease of creation.

These properties are extremely important in the geometric modelling area and should also be considered when studying and determining the representation scheme to integrate in a virtual environment. However, we have to emphasise that some of these properties are not equally important for a virtual environment application. For example, in a virtual space for arts it is not essential to ensure the geometrical validity of a representation, since the artist might be interested in the creation of surrealistic models.

In this case, we also have to study formal and practical properties of each representation that can enhance the three virtual environment components (Foley, 87): visualisation, behaviour and interaction.

Thus, a suitable representation scheme for a virtual space should include those properties studied by Requicha and the ones that will enable such representation to fulfil the virtual environment paradigm requirements:

- the creation of convincing and realistic environments (visualisation component);
- the intuitive interaction with visitors (interaction component);

- and the realistic simulation of objects' behaviour (behaviour component).

3 MODELS FOR THE REPRESENTATION OF VIRTUAL SCENES

In this section we discuss strengths and weakness of the most important representation scheme based on the set of properties proposed by Requicha (Requicha, 80). We will analyse wire frame, surface, constructive solid geometry, manifold and non-manifold boundary representation schemes to describe virtual scenes. Based on this stuff, we will show in the next section why manifold and non-manifold boundary representations are suitable and the most adequate choice for virtual environment applications.

3.1 Wire Frame Models

A wire frame model is a collection of lines, defining the edges of an object (figure 1). This representation can be used to generate object's drawings (2D wire frames), but a 3D wire frame model has no information about the surface boundary and this information cannot be generated from the geometry of vertices and edges, restricting the use of these models to certain applications. For example, 3D wire frame model have many drawbacks for product modelling (Goldman, 87), since he they can be ***impossible*** (figure 1-a), ***ambiguous*** (figure 1-b) or *incomplete*. The interpenetration of faces in a way that it makes impossible to represent a physical shape of a solid object and the multi-interpretation are two common problems of 3D wire frames.

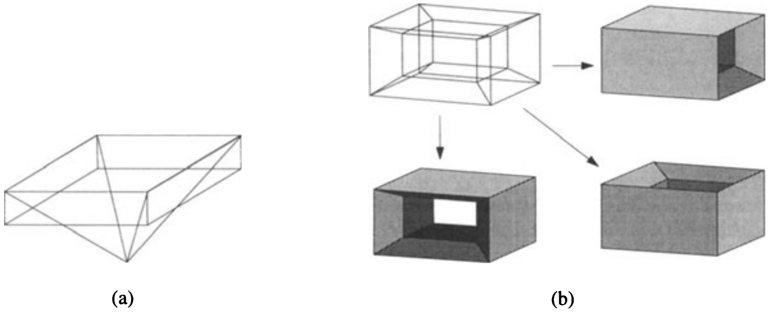

(a) (b)

Figure 1 (a) *Impossible* model (Mortenson, 85); (b) *ambiguous* and *incomplete* geometry (Goldman, 87).

3.2 Surface Models

In a surface modelling scheme, objects are represented as a collection of surface elements that describe the boundary of the object (figure 2). Several types of surface elements can be used: polygons (flat surfaces bounded by straight lines), analytical surfaces (for instance natural quadrics surfaces) and more general free-form surfaces (for example parametric surfaces, such as defined by the Bézier and B-spline method) (Foley, 90).

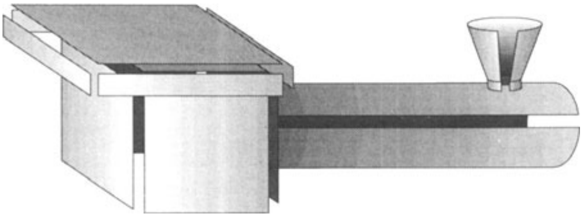


Figure 2 Surface model.

Surfaces are explicitly described in the computer model, providing some advantages over wire frame models for geometric modelling. With these representations interpenetration of faces can be identified, and the designer can be supported on preventing the creation of *impossible* objects. The introduction of faces removes the *ambiguity* from the wire frame models. Holes can be located easily and unambiguously, but not all objects which can be modelled with surfaces are realisable solid objects. A well known example is the Klein bottle. However, surface models could be *incomplete*, since there is no guarantee that the surfaces bound completely the model.

3.3 Solid Models

Wire frame and surface models do not guarantee the representation of *valid* physical solid objects, therefore these geometric modelers must rely on human assistance to supply missing data and clear inconsistencies, such as, ambiguity in wire frame models. Since complete and consistent 3D models are very important in many scientific and engineering applications, such as mechanical and civil fields (Requicha, 77), it has been widely identified that instead of representing drawings in a computer and incorporating their inherent limitations, computers should explicitly represent 3D objects as solids (Baer, 79], (Requicha, 80), (Mäntylä, 88).

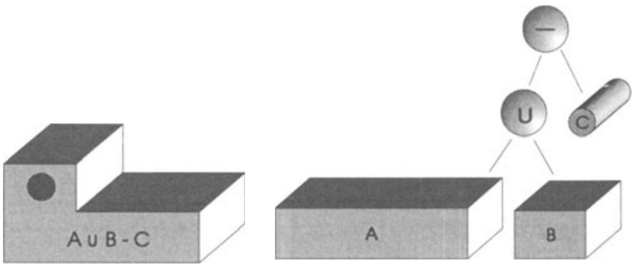
The major driving force for the development of solid modelling techniques has been the need to provide *complete* descriptions of the shape of physical solid objects, that means, representations which can answer arbitrary geometric questions automatically. These descriptions enable the computation of *geometrical* and *inertial* solid object's properties, *reliably* and *automatically* (Requicha, 80).

Next, we will concentrate on the three most important solid representation schemes: constructive solid geometry, manifold and non-manifold boundary representations.

3.3.1 Constructive Solid Geometry

Constructive solid geometry represents complex solids by applying Boolean operations and transformations on parametrized instances of solid primitives, such as block, sphere, cylinder and cone for example (Requicha, 77). Each solid primitive could be represented as a combination of half-spaces defining a bounded point set of $\mathbb{R}^3$. Thus, the constructive solid geometry representation can be said to be supported by a two-level scheme (Encarnação, 90): On the lower level, bounded volume primitives are defined on the basis of half-spaces; On the upper-level these primitives are combined by Boolean set operators.

The CSG defined object is internally represented as a binary tree, called CSG-tree (figure 3): The primitive solids are positioned at the leaves or terminal nodes; the internal or non-terminal nodes contain the Boolean operators of union, difference and intersection; transformation data for rigid-body motions, such as translation and/or rotation, can be stored both at the leave nodes as well as at the internal nodes. The internal nodes represent a solid defined by performing the transformations and the set operations of that node to its two subsolids indicated below it. The root node represents the resulting composite objects.



(a) (b)

Figure 3 a) Solid model; b) and the corresponding CSG tree.

Constructive solid geometry is a very powerful representational scheme and several researchers (Requicha, 80), (Mortenson, 85), (Mäntylä, 88) had already outlined its advantageous and problems: every CSG tree *unambiguously* models a physical solid, but they are not *unique* since in general it is possible to construct the same solid in other ways. However, if the primitive elements provided by a solid modelling system are valid bounded solids and the set operators are regularised, then the resulting solid models are guaranteed to be *valid* and *bounded.* This is an important property because it ensures that the validity of a CSG representation, based on bounded primitives, can be guaranteed by evaluating primitive leaf validity.

Nevertheless, CSG schemes produce *unevaluated* models, that is, they contain data that must be further processed in order to perform basic operations. For instance, for displaying and interaction with a solid model, details of the edges and faces of the object are required. Since these details are not explicitly present in it, the CSG representation must first be converted into a boundary representation, which can then be displayed with standard hidden-surface algorithms or used for the identification of interactions. This conversion known as *boundary evaluation* and it may be time consuming.

3.3.2 Manifold Boundary Representations

Boundary models represent a solid by describing *geometrically* its surface which is constructed as a closed boundary of surface elements, "faces" or "patches" (Baumgart, 74). In turn, planar faces can be represented by their bounding edges and vertices for example.

Additionally, it is provided a *topological* description of the connectivity and orientation of vertices, edges and faces. Topology specifies how bounding surfaces of a solid model are joined together.

Clearly, the boundary representation must satisfy certain rules in order to be able to represent physical solids and to reject boundaries that do not enclose volumes, as for example, the surface of a Klein bottle. The basic ideas of boundary representations are that the boundary should be *closed* (i.e. the boundary cannot have a boundary), *orientable* (the object must have a consistent inside and outside) manifold embedded in 3-space, and should *not self-intersect* (Baer, 79), (Hoffmann, 89). A two-manifold surface has the property that, around every one of its points, there exists a neighbourhood that is homeomorphic to the plane (Weiler, 86), that means, the surface exists in three-dimensional space but it is topological "flat" when the surface is examined closely enough in a small area around any given point.

Figure 4 illustrates a typical BRep data structure architecture.

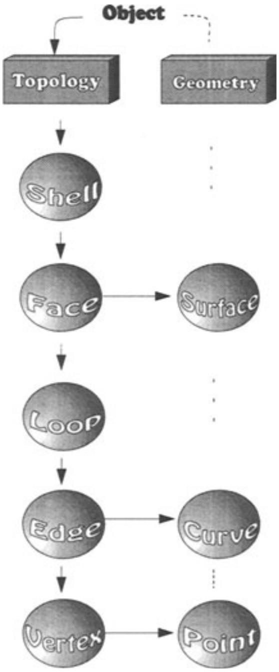


Figure 4 BRep data structure.

The solid object representation is defined as an hierarchical data structure of topological entities with shell, face, loop, edge and vertex nodes. Entities are linked by means of pointers, for example, from a face to each of its bounding edges, from an edge to its two ending vertices, and so on. This description that specifies vertices, edges and faces abstractly, and indicates their incidence and adjacencies relations is the ***topological information***. Geometric information is attached to each of the three object types: *vertices* are defined by coordinate triples; *edges* in general are represented by a parametric equation; the portion of the curve that forms the edge is defined by its two ending vertices; each *face* in the solid model lies on a single planar, quadratic, toroidal or parametric surface that supports it and its bounding edges. All these data, about the geometry of the entities of a solid model, we call the ***geometry*** of a boundary model.

A boundary representation scheme is *valid* if it defines a closed, orientable and not self-intersect boundary in order to guarantee that the model is representing a solid object. Boundary representations are *complete* and *unique* if the Brep data structure includes enough data about: (i) adjacency relationships between topological entities (vertices, edges, loops, faces, shell); (ii) orientation and closed properties (Gomes, 92). These properties are extremely important to those concerned in the design and implementation of a solid modeler. In addition, boundary representations provide *explicit representations* for the geometry of faces and edges and for the relations between them, which are quite useful for visual and interactive operations: the computer can automatically, by using the surface geometry, generate realistic pictures of the objects represented from any desired point of view in real time.

3.3.3 Non-Manifold Boundary Representations

Boundary based solid modelling techniques have found wide applications. However, conventionally they are restricted to representing only two-manifold domains. This disables to represent such conditions as two surfaces touching at a single point, two distinct enclosed volumes sharing a face as a common boundary, and a wire edge emanating from a point on a surface (figure 5). These conditions are known in the geometric modelling field as non-manifold conditions (Weiler, 86). Furthermore, common modelling operations, such as the Boolean operations can produce non-manifold results, and therefore not representable under manifold representations, even with strictly two-manifold input.

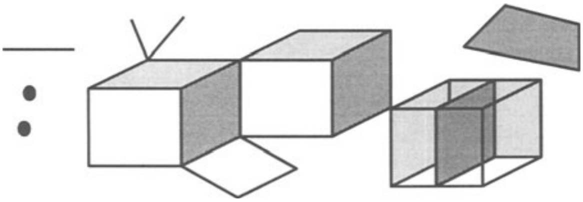


Figure 5 Non-manifold topology (Gursoz, 90).

Research on non-manifold surface topology led to the development of a new representation that expresses the non-manifold topology in order to expand the representational domain to cover such cases. Whereas manifold based solid modeler either give up some adjacency data whenever a non-manifold data occurs, treat it as a special case, or simply fail to perform the operation, in a non-manifold topology, we can represent all possible adjacencies among the basic topological elements (Hoffmann, 89).

Non-manifold geometric modelling domain encompasses both manifold and non-manifold conditions and is therefore quite general. An edge might bound no faces (wire edge), one face (lamina face), two faces (manifold edge) or more then two faces (non-manifold edge). A major advantage introduced by non-manifold boundary representations is the fact that they provide a single unified representation for any combination of wire frame, surface and solid modelling forms (Weiler, 86). In this way, these geometric modelling approaches can exist under the same representation scheme in pure or hybrid form. This uniformity offers significant advantages to the staging and delivery of geometric modelling systems as well as providing enhanced functionality and simplicity.

4 A SUITABLE REPRESENTATION SCHEME FOR VIRTUAL ENVIRONMENTS

Polygonal representation schemes (commonly used in virtual environment's prototypes) are perfectly adequate to generate the illusion of immersion on a virtual scenario. However, they do not provide complete geometrical and topological data required to enable the visitor to perform actions as those described above: touching object's surface, selecting its geometrical entities, identifying the neighbouring geometrical entities. In fact, there are many data structures that can generate data for real time realistic visualisation, but are unable to provide complete data for the interaction and behaviour component of a virtual environment.

Our goal was to provide more advanced interaction capabilities in a virtual environment to enhance the visitor's participation involvement, but the polygonal representational scheme that we were using was too restrictive. Therefore, we should review the representations schemes presented in section 3 and discuss their properties in the specific context of the virtual environment paradigm to find out a suitable representation that provide all the required data for this paradigm.

We can start with the wire frame models. The visual component of a virtual environment undertakes an important part in providing to the visitor the illusion of immersion in a virtual world by the creation of photo-realistic scenes. However, the lack of facet information in the wire frame model implies that there is no capability for the automatic generation of realistic shaded images that make visualisation so much easier.

Furthermore, wire frame models do not provide the required data for the interaction and behaviour component. It is the object's boundary that interact with any other objects in an environment. But, in a wire frame model we cannot locate its boundary and therefore, in a wire frame virtual world we cannot identify any interactions between virtual objects (figure 6). On the other hand, we cannot simulate physical object's behaviour. These models do not provide the required data to compute volume properties, such as weight and mass, and in this case, wire frame virtual models do not follow gravity laws, for instance.

Thus, we can say that wire frame models have drawbacks that make them unuseful in a virtual environment.

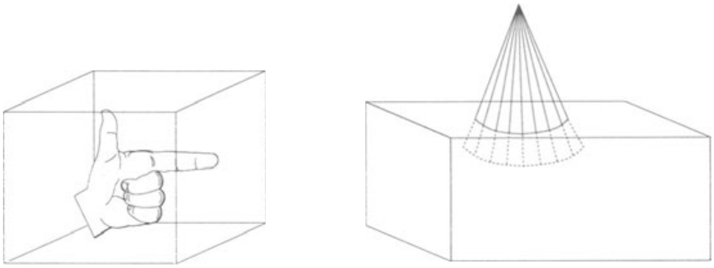
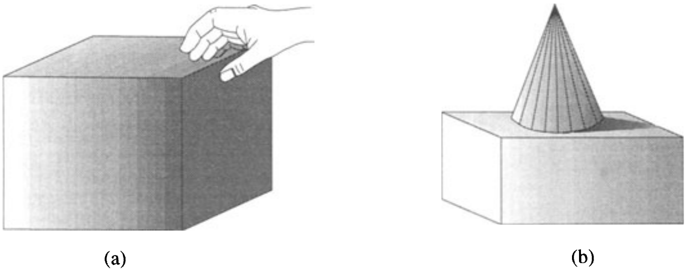


Figure 6 No object interactions identified in a wire frame virtual world.

For the surface models, we have to find out, as we have done above, if these representations provide the means to generate to the visitor the illusion of immersion in a three-dimensional space. In fact, surface models describe the solid object's geometry (faces, edges, vertices) required for the visible-surface determination, illumination and shading algorithms and, in this way, they can be used to generate realistic 3D scenes. In particular, polygon modelling schemes, initially developed for rendering (Mortenson, 85), describe the surface geometry by a cross-referenced list of vertices, edges and planar faces, enabling the creation of photo-realistic images and in real time. Thus, we can understand the massive use of polygon representations in virtual environment toolkits, since they provide the means to present to the visitor the illusion of a world that does not exist out of his perception.

The availability of surface's geometry in object's models introduces remarkable advantages for the interactive component of a virtual world. In generic terms, the surface can be used to find out the intersection of two objects. This will suffice for the identification of interactions between virtual objects (figure 7). Certainly, it is a powerful feature, auspicious for the recognition of grabbing or pushing operations upon virtual objects enabling the visitor to interact with 3D models through a hand cursor (figure 7-a). In the same way, the identification of interactions can be used to prevent two objects of sharing the same spatial region (figure 7-b).



(a) (b)

Figure 7 (a) In a surfaces' virtual world the visitor can grab virtual objects; (b) and the system can simulate objects' impenetrability.

On the other hand, we have to remember that it is users' responsibility to warrant the creation of a closed surface model, with a finite volume. Therefore, it is clear that incomplete models of solid objects can coexist in some cases and cannot be used to determine algorithmically its mass properties. This insufficiency leads to a virtual environment where its virtual models cannot algorithmically simulate the behaviour of their real counterpart objects.

In the end, we can succinctly say that surface models provide proper data for the generation of convincing interactive 3D environments. Nevertheless, they cannot be considered the best choice for virtual environment applications. In fact, some faults stand out immediately when we want to use them for behavioural simulation and in more advanced interactive features.

Constructive Solid Geometry schemes provide *valid* and *complete* solid models which can be used in virtual environments, especially to determine object's volume and mass properties, providing a preliminary evaluation of its performance and for the behavioural simulation of 3D virtual objects.

Nevertheless, CSG schemes produce *unevaluated* models, that is, they contain data that must be further processed in order to perform basic operations. For instance, in displaying and for interaction with a solid model, details of the edges and faces of the object are required. Since these details are not explicitly present in it, the CSG representation must first be converted into a boundary representation, which can then be displayed with standard hidden-surface algorithms or used for the identification of interactions. This conversion known as *boundary evaluation* and it may be time consuming. Thus, CSG representation cannot maintain in real time basic operations for a virtual environment and therefore with no practical interest for the moment for this paradigm.

Boundary representations provide *explicit representations* for the geometry of faces and edges and for the relations between them, which are quite useful for the visual and interactive components of a virtual environment. The computer can automatically generate realistic pictures of the objects represented from any desired point of view and in real time using the surface geometry. In this way, boundary models provide to the visitor the illusion of immersion in a real world.

In the same way, the boundary data available in a Brep model can be used for the identification of interactions, in particular for: (i) letting the visitor to grab, push or release virtual objects; and (ii) preventing two virtual objects of sharing the same region.

On the other hand, the availability of the geometrical data of faces, edges and vertices, makes it possible to find out precisely when two virtual objects are colliding (Figueiredo, 93). In fact, the collision detection can be implemented using bounding boxes (to filter out pairs of faces that cannot intersect, speeding up this process) and calculating the cross sections of the two objects. In this way, we can let the visitor to touch the surface of a virtual object and grab it with his finger tips (figure 8).

In addition, boundary models have unique advantages quite important for the geometric modelling field, since they explicitly represent topological adjacency of a solid model, which are also send back into the virtual environment paradigm.

Topological data is relevant in the resolution of certain problems automatically, such as, interactive manipulation of the shape of a solid object. Consider, for example, a can described by a collection of surface patches (figure 9). A designer then decides to alter the shape of one of these patches. If a boundary is affected, then adjacent patches must also be changed; otherwise these panels will not join evenly and they will separate and tear. The problem for the computer is to fix these adjacent patches automatically or at very least to cue the designer to the location of the problem patches. If topology is not available directly into the geometric model, the computer will be unable to determine connectedness and juxtaposition and therefore it will be unable to solve this question.

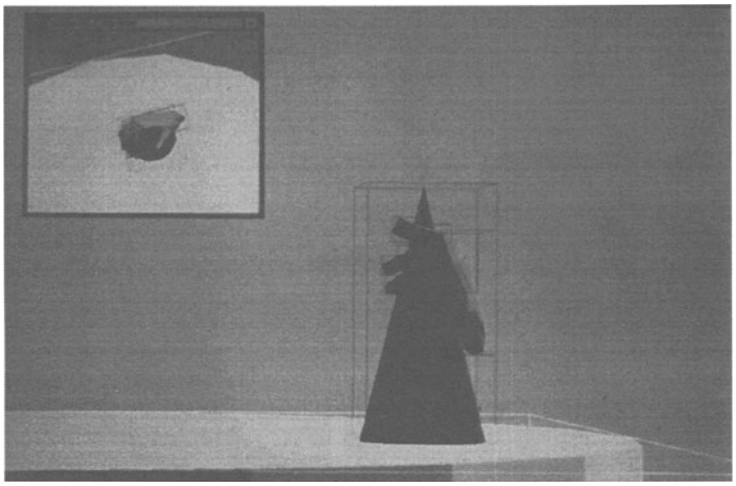


Figure 8 The visitor is enabled to touch the virtual cone's surface.

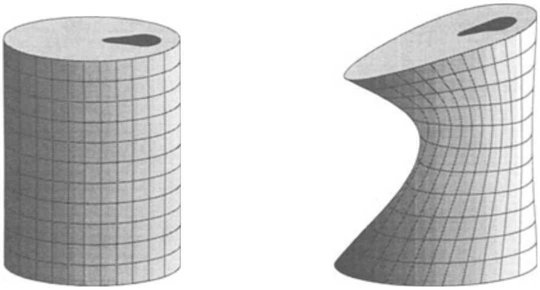


Figure 9 Patches manipulation.

Thus, a quite convenient advantage of a BRep is that it provides explicit topological data that can be used in general to enhance the interaction capabilities in a virtual environment.

We can use adjacency data in the implementation of intuitive 3D interaction techniques for selecting geometrical identities (e.g., faces, edges, vertices) and for local operations. For example, as presented in figure 10 the topological data enables the visitor to select all the edges of a face by selecting the face directly with a finger. Then, we can edit interactively the surface of the virtual object (figure 11-a-b), using the topological data to guarantee that all patches maintain jointly.

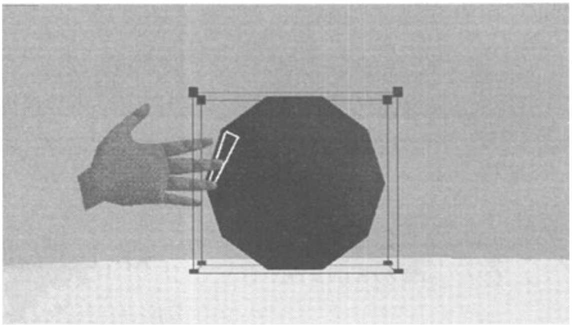
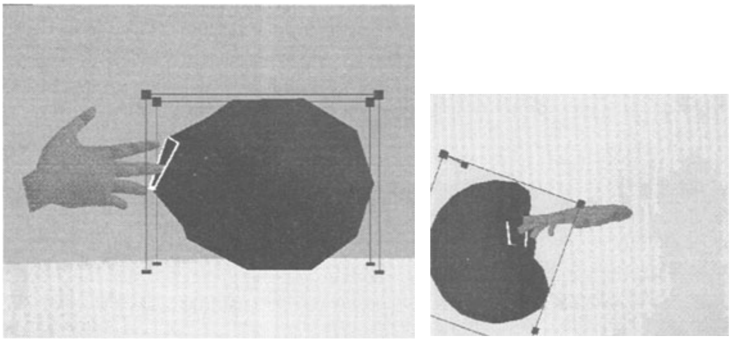


Figure 10 Face and its boundary edges are selected.

In fact, in a virtual environment where virtual objects are represented by Brep models, the visitor is enabled to interactively edit the model's shape with a 3D cursor (virtual hand). In this way, it allows the modification of a localised region of the data structure in an efficient manner and with greater naturality. For example, the geometry associated with a single face can be redefined and the result evaluated quickly. In this way, a Brep data structure can contribute to augment visitor's capabilities under a virtual environment enabling the implementation of intuitive 3D interaction techniques to let the visitor touch the surface of 3D objects, or select geometrical entities or performing local operations.



(a) (b)

Figure 11 The visitor interactively edit the shape of a 3D object via a hand cursor.

5 CONCLUSIONS

We have shown that boundary representation schemes not only have important formal properties, but they also have practical characteristics that are suitable for the goals of the virtual environment paradigm; they guarantee *valid* and *complete* representations of solid object's boundary and provide *geometrical* and *topological* data *explicitly*. Thus, the visual, interaction and behaviour components of a virtual environment can be improved, improving the overall experience of a visitor.

To conclude, it is presented a table that compares the representation schemes described above and clarifies the advantages introduced by Brep models. We should emphasise, that *validity*, *completeness* and *explicit* data of the boundary surface of an object, are extremely important properties that make boundary models suitable for virtual environments.

Table 1 Representation schemes' classification procedure for virtual environments.

Models	Valid	Unique	Complete	Explicit data	
				Geometry	Topology
Wire frame					
Surface				✓	
CSG	✓		✓		
BRep	✓	✓	✓	✓	✓

ACKNOWLEDGEMENTS

We would like to thank Klaus Böhm, Volker Kühn for allowing the use of GIVEN (Böhm, 92) in our experimental work. We also would like to thank to Rosario for the support in the implementation of part of this work.

REFERENCES

Baer, A., Eastman, C. and Henrion, M. (1979) Geometric Modeling: a Survey. *Computer Aided Design*, **11**, **5**, 253-272.

Baumgart, B.G. (1974) *Geometric Modeling for Computer Vision.* PhD Thesis, Stanford University, Palo Alto.

Böhm, K., Hübner, W. and Väänänen, K. (1992) GIVEN: Gesture Driven Interactions in Virtual Environments, a Toolkit Approach to 3D Interactions. *Proc. of the Interfaces to Real and Virtual Worlds Conference*, Montepelier, 243-254.

Encarnação, J.L., Lindner, R., Schlechtendahl, E.G. (1990) *Computer Aided Design — Fundamentals and Systems Architectures.* 2.ª ed., Springer-Verlag, Berlin.

Figueiredo, M., Böhm, K. and Teixeira, J. (1993) Advanced Interaction Techniques in Virtual Environments. *Computer & Graphics,* **17**, **6**, 651-661.

Foley, J.D. (1987) Interfaces for Advanced Computing. *Scientific American*, **257**, **4**, 126-135.

Foley, J., Dam, A., Feiner, S.K., and Hughes, J.F. (1990) *Computer Graphics — Principles and Practice.* 2.ª ed., Addison-Wesley Publishing Company, Massachusetts.

Goldman, R. (1987) The Role of Surfaces in Solid Modeling, in *Geometric Modeling: Algorithms and New Trends* (ed. Gerald Farin), SIAM, 69-90.

Gomes, A. (1992) *Modelos Algébricos de Sólidos e Morfologia.* Master Thesis, Coimbra.

Gursoz, E.L. and Prinz, F.B. (1990) A Point Set Approach in Geometric Modeling, in *Advanced Geometric Modeling for Enginnering Applications* (ed. F.L. Krause, H. Jansen), Elsevier Science Publishers B. V., 73-88.

Hoffmann, C.M. (1989) *Geometric and Solid Modeling — An Introduction.* Morgan Kaufmann Publishers, San Mateo, California.

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

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

Requicha, A. (1977) *Mathematical Models of Rigid Solid Objects.* Production Automation Project, Technical Memorandum 28, University of Rochester, New York.

Requicha, A. (1980) Representations for Rigid Solids: Theory, Methods, and Systems. *ACM Computing Surveys*, **12**, **4**, 437-464.

Weiler, K.J. (1986) *Topological Structures for Geometric Modeling*, PhD Thesis Rensselaer Polytechnic Institute, Troy, New York.

BIOGRAPHY

Mauro Figueiredo is an Assistant at University of Coimbra. His research interests are in virtual environments, 3D interaction techniques, cooperative work, solid modeling and user interfaces.

Figueiredo received his degree in Computer Science from University of Coimbra in 1990 and his MS in Industrial Automation from University of Coimbra in 1994.

José Carlos Teixeira is an Auxiliary Professor at the University of Coimbra in the areas of Computer Graphics and Geometric Modelling, head of its Computer Graphics Research Group and President of the CCG/ZGDV Executive Board. His main research interests are in Geometric Modelling, Virtual Environments, CSCW and new Interaction Techniques.

Responsible and involved in different European and Portuguese Projects, is member of the Editorial Board of the journal "Computer & Graphics" (Pergamon Press), founding member of the WG 5.10 on Computer Graphics of IFIP TC 5, President of the EUROGRAPHICS Portuguese Chapter and head of the Portuguese Technical Committee for Standardisation on Computer Graphics - CT 109. He is a member of EUROGRAPHICS, ACM, ACM-Sigraph and IEEE.