ACM Symposium on Solid Modeling and Applications (2004)
G. Elber, N. Patrikalakis, P. Brunet (Editors)

History Based Reactive Objects for Immersive CAD

Thomas Convard and Patrick Bourdot

LIMSI-CNRS, Université Paris XI
BP 133, F-91403
Orsay CEDEX (France)
Telephone: +33 (0)1.69.85.80.32, +33 (0)1.69.85.81.73
{convard,patrick.bourdot}@limsi.fr

Abstract

Virtual Environments (VE) allow direct 3D interaction, better perception of shapes and a feel of immersion, properties that are highly desirable for design tasks. Traditional CAD software extensively use WIMP interfaces (Windows, Icons, Menus and Pointing device), but these interaction models are not suited in VE. Moreover, during a design task, the use of dialog boxes, buttons, etc. deteriorates the user's focus on his work. However, to fully benefit from immersive interaction we need more reactive behavior from 3D objects. The objects data structures must provide efficient ways for real-time modification of the geometric definitions of solids via direct 3D interactions. We will present an approach that replaces the traditional editing of the construction history graph in parametric modelers. A description of data structures and algorithms that allow the user to implicitly modify the history of a solid through a direct 3D interaction on topological elements of the objects will be given. The techniques presented here are validated in a VE prototype, using the OpenCASCADE geometric kernel and a multimodal interface.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Three-Dimensional Graphics and Realism]: Virtual reality I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations

1. Introduction

Following the appearance of CAD software the design task has changed [Poi87]. The CAD activity is mainly focused on a logical process whereas a design task is also a combination of perceptive and sensorimotor actions. The interaction between the sensorimotor actions and the perceptive feedbacks allows designers to better express and communicate the forms and properties of the objects [Leb83]. Of course, CAD gets strong benefits from the computational power for complex geometric operations or constraint solving. Now, Virtual Environments (VE) offer new perspectives for design tasks. These interfaces give the user more realism and intuitive interactions. VE can help to get back the manual aspect of the design task to CAD software.

On current CAD software, most of editing commands are issued with the aid of widgets and alphanumeric data inputs. Only few selection commands are done with direct mouse selection. Previous researches in CAD related that multimodality could help avoiding utilization of widgets for the interface. For example, it has been shown that a user can be more productive using a CAD program when keyboard interactions are replaced by vocal ones [Mar89]. One of the strengths of Virtual Reality is the 3D interaction so that applications using this technology should limit the use of WIMP paradigms. To this end, we recently presented a multimodal Virtual Environment for CAD applications [CB03].

However, one of the drawbacks of current history-based modelers is how changes of design history are performed. In cases, users often edit an operation that is several steps upstream in the tree, and they must go back to the previous stage to see the result of their modification. For example, suppose we have a solid object being the result of profile extrusion, with several form features attached. If users modify the profile (moving points, changing dimensions), current software doesn't give feedback about the consequences of their gestures on the final solid. Creating design variants from an object is thus a tedious task.

To fully benefit from immersive interaction, as well as to improve the usability of history-based modelers, more reactive data structures are needed for objects. We mean by reactive, that the object could be modified in real-time by a direct interaction. This work provides a technique that allow implicit history editing of a solid by direct interaction on topological elements (vertices, edges, faces) of its boundary representation. This technique leads the CAD objects to be more reactive, allowing immersive design activity with current models and functionality of CAD software.

The rest of the paper is organized in the following manner. Section 2 presents related work on VE and CAD. In section 3, we describe our representation for objects. Section 4 explains the technique to obtain object reactivity. In section 5, we briefly describe our VE multimodal prototype. In section 6, we present open issues and we provide some possible solutions for them. Finally, we conclude and show future work.

2. Related Work

In the field of CAD, Virtual and Augmented Reality are especially used for virtual prototyping [dSZ99] or to manage parts assembly [BBMW02]. Nevertheless, in these cases, the objects are modeled with traditional CAD software and are exported into standard graphic object format to be used in VE.

However, since the beginning of Virtual Reality research has been done to bring geometric modeling in immersive environments.

Some results [CDG97] [KTK*97] [dAFS01] [CB03] focused on immersive interaction for a design task, since accurate interaction is one of the problems of VE for geometric modeling.

Most of the immersive approaches to geometric modeling are based on object representations that are favorable with a 3D interaction. They use models that allow object creation and modification with direct manipulation [DBW*00]: voxels grid [LWLO02], function representation [Sav00], freeform surfaces drawing [USV96], simple polyhedral primitives [BDHO92] or more complex multiresolution mesh [GEL00]. All these results can only be used for early design stage of product lifecycle. Only few attempts have be done in the field of part design, for example Zhong et al. [ZMWM02] tried to bring a new data model suited for Virtual Reality with interactive constraint recognition.

3. Object Representation

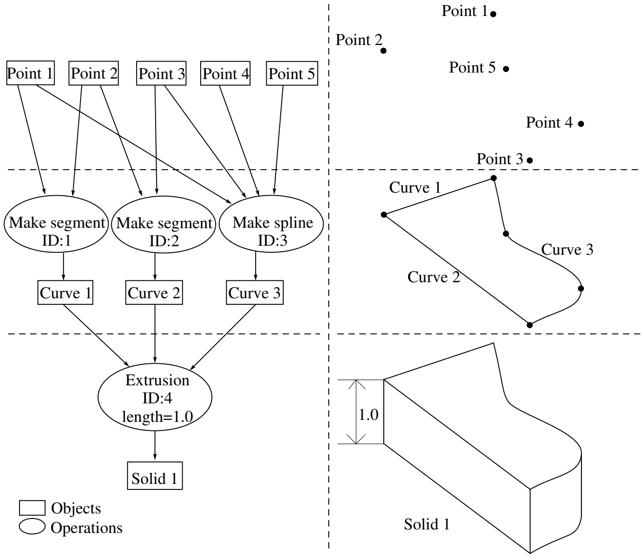

Figure 1: *Construction history graph of a solid produced by extruding a profile*

Here we describe the data structure to represent the objects. This data structure is composed of two models common in many CAD systems:

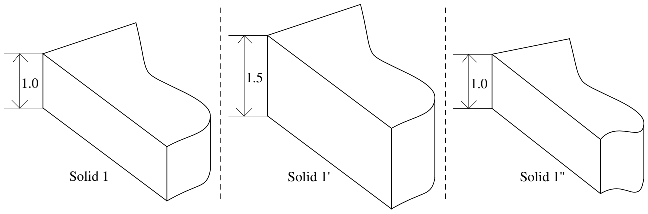

Figure 2: *Design variants from a same construction history graph*

- A *construction history model*, which is a graph representing the sequence of modeling operations used to create the object, and their results.
- A *boundary representation* (B-Rep), an evaluation of the construction history model.

A B-Rep model is attached to each node of the history tree, representing the geometric result of its operation. In practice, a mesh model is also associated to each B-Rep for displaying and collision detection, necessary for the interaction.

The construction history graph is derived from Constructive Solid Geometry tree, sometime also called feature dependency graph [HJ93] or design feature history. It is an implicit representation of the parametric definition of the object. The B-Rep is used for several operations needing topology referencing, i.e. form feature attachment [SM95]. Additional information can be added to the object nodes, such as object location or material features. In standard history-based modelers, the construction history is generally described by a binary tree with solids as object nodes, whereas in our model it is a graph where object nodes can also be primitives like points and curves (see Figure 1). Boolean operations nodes are extended with curve creation nodes, profile extrusion nodes, and form features attachment nodes.

Any object enables geometric transformation by changing parameters of operations or the locations of objects in the construction history graph. In Figure 2 we can see several object variants: one is obtained by giving a new parameter to the extrusion (Figure 2 solid $1'$), another by moving points of the profile (Figure 2 solid $1''$). The three solids have the same construction history graph. Moreover, to replay completely the design history, a persistent naming management is necessary [Hof93]. Several methods and formalisms have been proposed [RS98] to solve this problem. In our case we use topological as well as geometrical information, close to the approach of Wu et al. [WZZZ01].

In a standard system, users have to edit a tree representation with widgets and alphanumeric values inputs to modify the construction history of their objects. Our goal is to allow this modification without WIMP interaction, using the strength of VE with direct 3D interaction and multimodality: to implicitly edit the construction history graph only with users manipulation.

4. Implicit Editing of Design History

Our approach consists in replacing editing of the history tree by direct interaction on the solid shape. This section explains the details of our process. We describe it for objects resulting of linear extrusion and regularized Boolean operations. However, the same scheme can be used for additionnal operations such as sweeping and form feature attachment.

4.1. Reactivity Scheme

The principle of our method is:

1. The user selects a topological element of the solid's B-Rep (vertices, edges and faces) and ask for a solid modification.
2. Depending on the selected topological element, one or more targets are identified, as we will see in section 4.2. Targets might be parameters of operations or objects in the construction history model.
3. For each target, manipulation constraints are also identified: for example if the target is a point in a profile, the constraint would be to allow only manipulation in the supporting plane of the profile.
4. The user's manipulation (a variation of position and orientation from the previous iteration) is processed. The targets are modified according to the constraints and the manipulation.
5. The model is reevaluated: a new B-Rep is produced.
6. The selected topological element is identified in the new B-Rep, thanks to a persistent naming process, to allow visual feedback to the user.

The steps 3 to 6 are repeated so that the solid is continuously modified with direct manipulation. The step 2 doesn't have to be repeated because the history model isn't changed during the process, only parameters and primitives are affected, hence the targets are the same for each iteration. The step 6 is not used in geometric deformation process. Nevertheless, if this step is omitted, there is no possible feedback on the initial selected entity where the B-Rep is regenerated. Conversely, using step 6 to have an active feedback on the initially selected entity helps the user to continue reactive manipulation.

During the manipulation, the selected topological element could disappear due to consequent topological changes in the reevaluated B-Rep. This does not affect the process because the targets and the constraints have been identified in an initialization stage (steps 2 and 3 in the list). The user can continue his interaction with the solid.

4.2. Target Identification

In order to identify the target from a topological element, constructive information (face splitting history, intersection generated from which operation...) must be attached to the B-Rep elements. These informations come from the labeling process and are taken into account also for our persistent naming scheme.

After an operation, topological entities of the resulting solid model are labeled depending on their role by the operation. The label is an expression containing at least the identifier of the operation, which created the entity. If an entity was present in the B-Rep before the operation, it keeps the exact same label. These labels are not unique, for example, a face could have been split in two during the operation or several edges can come from the same intersection of two faces after a Boolean operation.

4.2.1. Labeling topological elements

In this section, we give the example of labeling for the extrusion operation and the regularized Boolean operations. The labeling process for the different operations is as following:

Extrusion: This operation takes a set of connected curves forming a closed profile, has a length parameter, and results in a solid.

- Vertices on the bottom and top faces of the solid are labeled $VB(ID_{pnt}, ID_{op})$ and $VT(ID_{pnt}, ID_{op})$ respectively, where ID_{pnt} is the identifier of the point in the profile that leads to the vertex and ID_{op} is the extrude operation identifier.
- Edges on the bottom and top faces of the solid are labeled $EB(ID_{curve}, ID_{op})$ and $ET(ID_{curve}, ID_{op})$ respectively, where ID_{curve} is the identifier of the curve in the profile that leads to the edge.
- Edges created from the extrusion of vertices are labeled $EX(ID_{pnt}, ID_{op})$ where ID_{pnt} is the identifier of point in the profile that leads to the extruded vertex.
- Faces created from the extrusion of edges are labeled $FX(ID_{curve}, ID_{op})$, where ID_{curve} is the identifier of the curve in the profile that leads to the extruded edge.
- The bottom and top faces of the solid are labeled $FB(ID_{op})$ and $FT(ID_{op})$ respectively.

Boolean operations: These operations take two solids and the result is a solid. The order in which the two solid operands are given is important for the cut operation and is significant for the labeling purpose.

- Vertices, edges and faces coming from the two original solids inherit the same labels.
- Vertices resulting from the merge of two vertices are labeled $VM(V_1, V_2, ID_{op})$, where V_1 and V_2 are the labels of the welding vertices of the two solids and ID_{op} is the Boolean operation identifier.
- Edges resulting from the merge of two edges are labeled $EM(E_1, E_2, ID_{op})$, where E_1 and E_2 are the labels of the merging edges of the two solids.
- Vertices resulting from edge-face intersections are labeled $VI(E, F, S, ID_{op})$, where E and F are the labels of the intersecting edge and face respectively, S is a Boolean value which is 1 if the edge comes from the first solid parameter of the Boolean operation 0 otherwise. The information S allows us to keep track of which solid provided the intersecting edge and which solid provided the face.
- Edges resulting from two face intersections are labeled $EI(F_1, F_2, ID_{op})$, where F_1 and F_2 are the labels of the intersecting faces.
- Faces resulting from the merge of two faces are labeled $FM(F_1, F_2, ID_{op})$, where F_1 and F_2 are the labels of the merging faces.

4.2.2. Selecting the targets and the constraints

From labels of the B-Rep entities of each node of a construction history graph, it becomes possible to manage the selection of the targets. Different rules can be considered and would reflect different interaction strategies. Actually, the selection of the target is an essential task in the reactivity process because it should lead to an intuitive behavior of the solid modeler. Rules we adopted for selecting the targets depending on the selected entity's label are listed

#	Label	Targets
1	$VB(ID_{pnt},ID_{op})$	• point ID_{pnt} in the profile • length parameter of the extrude operation ID_{op} • the position of the resulting solid
2	$VT(ID_{pnt},ID_{op})$	• point ID_{pnt} in the profile • length parameter of the extrude operation ID_{op}
3	$EB(ID_{curve},ID_{op})$	• points in the profile defining the curve ID_{curve} • length parameter of the extrude operation ID_{op} • the position of the resulting solid
4	$ET(ID_{curve},ID_{op})$	• points in the profile defining the curve ID_{curve} • length parameter of the extrude operation ID_{op}
5	$EX(ID_{pnt},ID_{op})$	• point ID_{pnt} in the profile
6	$FX(ID_{curve},ID_{op})$	• points in the profile defining the curve ID_{curve}
7	$FB(ID_{op})$	• length parameter of the extrude operation ID_{op} and the location of the resulting solid
8	$FT(ID_{op})$	• length parameter of the extrude operation ID_{op}
9	$EM(V_1,V_2,ID_{op})$ $VM(E_1,E_2,ID_{op})$ $FM(F_1,F_2,ID_{op})$ $VI(E,F,S,ID_{op})$ $EI(F_1,F_2,ID_{op})$	• location of the second solid of Boolean operation ID_{op}

Table 1: *Example of targets selection rules*

in Table 1. For example, the following are the rules applied while editing the shape in Figure 3:

- If vertex labeled $VB(6,10)$ is selected, then according to the rule 1 the targets will be the position of point 6 in the profile, the length parameter of the extrusion operation 10 and the location of solid 2 (cut object in Figure 3).
- If edge labeled $EI(FX(4,5),FX(5,10),11)$ is selected, then according to the rule 9 the target will be the location of solid 2.
- If face labeled $FT(5)$ is selected, then according to the rule 8 the target will be the length parameter of the extrusion operation 5.

We represent the user's manipulation as a variation of position and orientation. This matrix must be transformed into targets reference frames because in our construction history graph, each object node has location information.

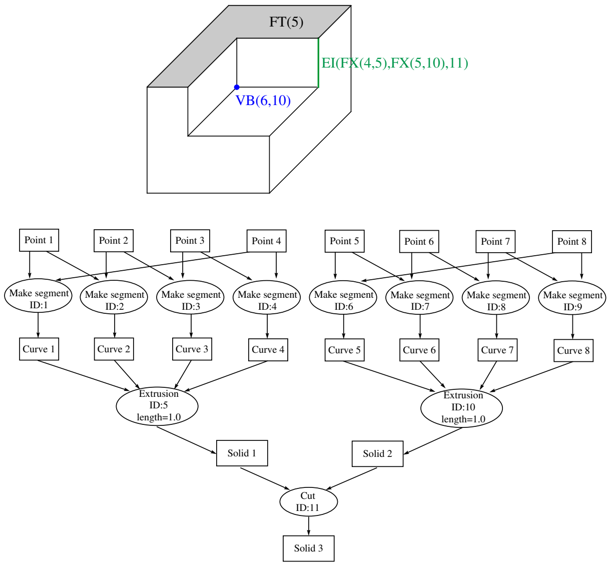

Figure 3: *Example of a solid and its Construction History Graph for target selection*

During the stage of target selection, manipulation constraints are also identified. These constraints reflect the degree of freedom allowed to the user in order to produce a valid modification of this target. The constraints depend on the targets.

- Points on a profile: Let $\overrightarrow{U}$ and $\overrightarrow{V}$ be two orthogonal unit vectors of the profile's plane, and the vector $\overrightarrow{T}$, the translationnal part of the user's manipulation in the reference frame of the profile. Each point is translated by vector $\overrightarrow{D}=(\overrightarrow{T}.\overrightarrow{U})\overrightarrow{U}+(\overrightarrow{T}.\overrightarrow{V})\overrightarrow{V}$.
- Length parameter of an extrusion: Let unit vector $\overrightarrow{N}$ the normal of the profile plane, $\overrightarrow{N}.\overrightarrow{T}$ is added to the length parameter of the extrusion.
- Position of a solid: For the cases $VT(ID_{pnt},ID_{op})$ and $EB(ID_{curve},ID_{op})$, the solid is translated by vector $(\overrightarrow{T}.\overrightarrow{N})\overrightarrow{N}$ where $\overrightarrow{N}$ is the normal of the profile plane.
- Location of a solid: In this case, the solid can be solidarized with the user's interaction allowing 6 DoF manipulations.

Constraints can also be used for feedback to guide the user during the modification. For example, when the target is a single point in a profile, the manipulation is constrained to the profile plane. At the user interface level, additionnal constraints can be added. For example to guide the user with snapping.

4.3. Example

In the Figure 4 and 5 we show an example of object reactivity. The user edits the parametric history-based definition of solid objects by selecting and moving their vertices, edges and faces.

In the Figure 4, the solid is the result of a cut Boolean operation between a solid produced by extrusion and a second solid produced by a revolution, then a fillet form feature is added to an edge. The user modifies the position of two points in the former profile of the cut object by dragging the selected edge, the boolean operation is automatically re-executed. In the Figure 5, the solid is the result a

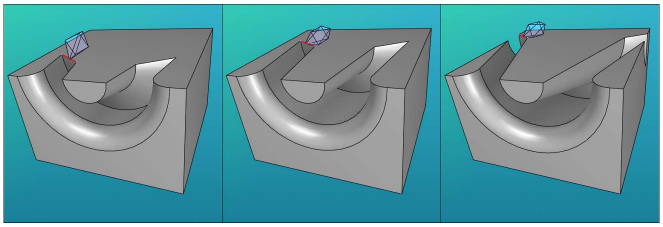

Figure 4: *Object reactivity, the user modifies the position of two points in the former profile of the cut object by dragging the red edge.*

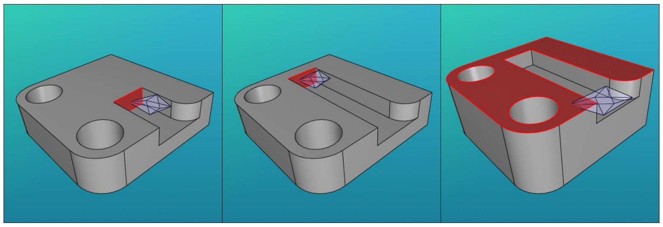

Figure 5: *Object reactivity, the user modifies the length parameter of extrusion operations by dragging the red faces.*

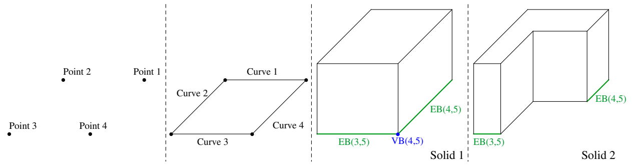

Figure 6: *Point 4 has no resulting topological entity in the B-Rep of Solid 2.*

cut operations between two solids created by extrusion, then a fillet is added to an edge. The user modifies extrusion length of the cutting object, and of the cut object, by dragging faces, again the boolean operation is re-executed. The visual feedback of the selected topological element (the edge hilighted in red on Figure 4 and the faces in Figure 5) is possible because of the persistent naming in step 6 (see section 4.1). However, as we have already recalled in section 3, it is the presistent naming that allows correct replay of the form feature constructions included in a history construction graph (such as the edge rounding on Figure 4).

5. Multimodal VE Prototype

We have implemented the techniques presented here in an immersive CAD prototype. For geometric modeling we use the kernel of OpenCASCADE®. However, our immersive CAD software only requires the geometric modeling and tessellation functionalities of OpenCASCADE® libraries. Our prototype runs our own persistent naming scheme and replaces the standard WIMP user interface by a multimodal immersive one.

The display is composed of two stereoscopic retroprojected screen walls for visual feedbacks. We use a Flock of Birds^{TM} 6 degree of freedom (DoF) motion capture system to track the users hands and head. The head is tracked to allow correct stereoscopy computation and the hands are tracked for interaction purpose. Selection and manipulation tasks are done with a Wand^{TM} pointing device on the dominant hand and a 5DT® data glove on the slave hand. Commands are issued by the user with vocal events, so a speech recognition system is also connected to the immersive application. All devices and displays are managed by the EVI3d platform [TBBB02].

In this prototype, users can sketch 2D profiles, extrude or revolve them to obtain solids. Users can also manipulate solids either with direct manipulation or constraint based translations and rotations. Solids can be modified by attaching them form features or doing Boolean operations. Object reactivity allows to transform the solid history by selecting a topological entity, asking for the implicit editing mode and manipulating the entity with the Wand.

The interaction of this prototype is fully multimodal and immersive, no widget are used.

6. Open Issues

Some parameter operation or primitive may not have resulting topological entities in the B-Rep of the current object. For example in Figure 6, the point 4 can't be selected as a target. A solution to this problem could be to extend the rules with multiple selection, for example to allow the selection of the point 4 as target when the selection is composed of the edges $EB(3,5)$ and $EB(4,5)$. Another possibility could be to improve the Boolean operation labeling with form feature recognition information [HPR00].

7. Conclusion and future work

We presented in this paper an approach for implicit modification of design history by direct 3D interaction on the shapes of objects. This technique is one possible way to provide reactive object in CAD. Such real-time deformation on CAD objects is necessary for some design tasks (part design for example). It allows more intuitive modification of solids, such as is suitable for immersive modeling.

Real-time feedback are currently limited by the speed of the geometric kernel and the power of the computer. Currently, we are able to replay only several Boolean and features operations in real-time, depending on the complexity of the geometry. This computational cost is an inherent problem on history-based modelers when replaying long history sequences.

The labeling and targets selection methods proposed here for extrusion and Boolean operations can be generalized to sweeping, revolution and feature attachment operations. For example, we can imagine a rule for modifying the radius of an edge rounding operation by a direct selection and drag on the resulting face of the operation.

Haptic feedback could be added to the VE prototype to benefit from the constraints identified during the reactivity process. For example, we could constraint the user manipulation according to the identified DoF of the manipulated targets.

The critical step that makes the link between the interaction and the implicit editing is the targets and constraints identification step.

Keeping the same mechanism, other rules can be defined to allow a different deformation interaction. We are also working on the identified limitations decribed in section 6.

References

[BBMW02] Biermann P., B.Jung, M.Latoschik, Wachsmuth I.: Virtuelle Werkstatt: A Platform for Multimodal Assembly in VR. In *VRIC '02, 4th Virtual Reality International Conference* (Laval, France, 2002). 2

[BDHO92] Butterworth J., Davidson A., Hench S., Olano T.: 3DM: A Three Dimensional Modeler Using a Head-Mounted Display. In *ACM Symposium on Interactive 3D Graphics* (1992), pp. 134–138. 2

[CB03] Convard T., Bourdot P.: A Multimodal Approach For Computer Aided Design In Immersive Environments. In *Virtual Concept '03* (Biarritz, France, 2003). 1, 2

[CDG97] Chu C.-C. P., Dani T. H., Gadh R.: Multi-Sensory User Interface for a Virtual-Reality-Based Computer Aided Design System. *Computer-Aided Design*, 10 (1997), 709–725. 2

[dAFS01] de Amicis R., Fiorentino M., Stork A.: Parametric Interaction for CAD application in Virtual Reality Environment. In *Proceedings XII ADM International Conference* (sept 2001). 2

[DBW*00] Deisinger J., Blach R., Wesche G., Breining R., Simon A.: Towards Immersive Modeling - Challenges and Recommendations: A Workshop Analyzing the Needs of Designers. In *Eurographics Workshop on Virtual Environments (EGVE00)* (Amsterdam, The Netherlands, June 2000). 2

[dSZ99] de Sá A. G., Zachmann G.: Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes. *Computers and Graphics*, 3 (1999), 389–403. 2

[GEL00] Gregory A. D., Ehmann S. A., Lin M. C.: inTouch: Interactive Multiresolution Modeling and 3D Painting with a Haptic Interface. In *IEEE Virtual Reality Conference* (New Brunswick, New Jersey, March 2000), pp. 45–52. 2

[HJ93] Hoffmann C. M., Juan R.: Erep, an editable high-level representation for geometric design. *Geometric Modeling for Product Realization* (1993), 129–164. 2

[Hof93] Hoffmann C. M.: On the Semantics of Generative Geometry Representations. In *19th ASME Design Automation Conference* (New York, NY, USA, 1993), pp. 411–419. 2

[HPR00] Han J.-H., Pratt M., Regli W.: Manufacturing Feature Recognition from Solid Models: A Status Report. *IEEE Transactions on Robotics and Automation 16*, 6 (December 2000), 782–796. 5

[KTK*97] Kiyokawa K., Takemura H., Katayma Y., Iawasa H., Yokoya N.: VLEGO: A Simple Two-Handed Modelling Environment Based on Toy Blocks. In *Proceedings VRST'97* (1997), ACM New York, pp. 27–34. 2

[Leb83] Lebahar J. C.: *Le dessin d'architecte : simulation graphique et réduction d'incertitude*. Editions Parenthèses, 1983. 1

[LWLO02] Lu Y.-H., Wang W.-T., Liang R.-H., Ouhyoung M.: Virtual Sculptor: A Feature Preserving Haptic Modeling System. In *ACM International Workshop on Immersive Telepresence (ITP2002)* (Juan Les Pins, France, Dec 2002). 2

[Mar89] Martin G.: The utility of speech input in user-computer interfaces. *International Journal of Man-Machine Studies* (1989), 355–375. 1

[Poi87] Poitou J. P.: *STEREO: Etude d'un système tridimensionnel en relief par ordinateur.* Tech. rep., CNRS, November 1987. 1

[RS98] Raghothama S., Shapiro V.: Boundary Representation Deformation in Parametric Solid Modeling. *ACM Transactions on Graphics 17*, 4 (October 1998), 259–286. 2

[Sav00] Savchenko V.: 3-D Geometric Modeller with Haptic Feedback: engraving simulation. In *2nd Phantom Users Research Symposium* (Zurich, Switzerland, July 2000), pp. 35–42. 2

[SM95] Shah J. J., Mäntylä M.: *Parametric and Feature-Based CAD/CAM: Concepts, Techniques and Applications*. John Wiley & Sons, Inc., New York, NY, USA, 1995. 2

[TBBB02] Touraine D., Bourdot P., Bellik Y., Bolot L.: A framework to manage multimodal fusion of events for advanced interactions within Virtual Environments. In *International EUROGRAPHICS Workshop on Virtual Environments* (Barcelona (Spain), May 2002), EGVE 2002. 5

[USV96] Usoh M., Slater M., Vassilev T. I.: Collaborative Geometrical Modelling in Immersive Virtual Environments. In *Proceedings of the Eurographics workshop on Virtual environments and scientific visualization '96* (Monte Carlo, Monaco, January 1996), pp. 111–120. 2

[WZZZ01] Wu J., Zhang T., Zhang W., Zhou J.: A Face Based Mechanism for Naming, Recording and Retrieving Topological Entities. *CAD 33*, 10 (September 2001), 687–698. 2

[ZMWM02] Zhong Y., Mueller-Wittig W., Ma W.: A Model Representation for Solid Modelling in A Virtual Reality Environment. In *2002 International Conference on Shape Modeling and Applications (SMI 2002)* (BANFF, Alberta, Canada, May 2002), IEEE Computer Society. 2