Representing Dimensions, Tolerances, and Features in MCAE Systems

David C. Gossard, Robert P. Zuffante, and Hiroshi Sakurai
Massachusetts Institute of Technology

We present a method for explicitly representing dimensions, tolerances, and geometric features in solid models. The method combines CSG and boundary representations in a graph structure called an object graph. Dimensions are represented by a relative position operator. The method can automatically translate changes in dimensional values into corresponding changes in geometry and topology. The representation provides an important foundation for higher level application programs to automate the redesign of assemblies and to automate tolerance analysis and synthesis. We implemented a prototype interactive polyhedral modeler based on this representation.

The role of dimensions

Dimensions, such as appear on a drawing or sketch, are natural descriptors of geometry. They reflect important design or manufacturing considerations and are the most logical and appropriate "control points" through which to alter a component's geometry.

The term dimension-driven geometry (DDG) refers to...
Figure 3. (a) Object before change in dimension B. (b) Object after change—fixed topology. (c) Object after change—variable topology.

Figure 4. Relative position operator.

Figure 5. Relative position operators implemented.

Figure 6. Dimensional annotation.

The representation

A new representation was developed. The representational scheme combines CSG and boundary representations (B-reps) with a new concept: the relative position operator.

Representing dimensions as relative position operators

Dimensions are explicitly represented by a relative position operator (RPO). The RPO uses a single scalar parameter (the dimension's nominal value) to move one
tions. One of the main objectives of the effort has been to perfect methods by which engineers and designers can sketch parts quickly without concern for precise geometry. Then, as dimensional values are refined, they can request that the system automatically compute the exact geometry. Our initial approach was to constrain a boundary representation (B-rep); i.e., to represent dimensions as constraints on the permissible location of particular geometric entities on the boundary of the solid. These (dimensional) constraints collectively define a set of nonlinear equations

\[ F(D,X) = 0 \]  

where

\[ F = (f_1, f_2, ..., f_n) \] vector of constraint equation residuals

\[ D = (d_1, d_2, ..., d_n) \] vector of dimension values

\[ X = (x_1, x_2, ..., x_n) \] vector of vertex coordinates

that relate the component's dimensions D to its geometry X. A simple example is shown in Figure 1.

The set of constraint equations represents a family of parts which share a common topology and arrangement of dimensions. Any member of the family, i.e., any particular component geometry, can be obtained by specifying a set of dimensional values and numerically solving the equations. Given an initial estimate of the geometry, a new estimate is obtained from the following modified Newton-Raphson method:

\[ J_i \cdot dx_i = -F_i \]  

\[ x_{i+1} = x_i + dx_i \]

where

\[ J_i = \text{matrix of partial derivatives of constraint equations} \]

\[ dx_i = \text{vector of changes in } x \]

\[ F_i = \text{vector of residuals} \]

\[ x_i = \text{current estimate of geometry} \]

Iterative solution produces results such as those shown in Figure 2. Details of the method can be found in Light and Gossard, which summarizes earlier work found in several places. Although promising, the approach presented a number of problems. These included large computational requirements, nonuniqueness of the resulting geometry, user difficulty in creating and maintaining a consistent set of constraints, and limitation to a fixed topology.

Several researchers made significant progress on these problems. Lin reduced the computational requirements by isolating the minimum subset of the constraints affected by a given dimensional change. He also applied the approach to 3D components. Light used row-and-column operations on the Jacobian to detect overconstrained, underconstrained, and incorrectly constrained situations in the dimensioning scheme for 2D components.

**Variable versus fixed topology**

One of the fundamental limitations of the variational geometry approach, however, is that it can only gener-
Figure 3. (a) Object before change in dimension B. (b) Object after change—fixed topology. (c) Object after change—variable topology.

Figure 4. Relative position operator.

Figure 5. Relative position operators implemented.

Figure 6. Dimensional annotation.

We thought the ability to compute geometries automatically with differing topology from a single representation was desirable. That was one of the principal motivations for the work described in this article.

The representation

A new representation was developed. The representational scheme combines CSG and boundary representations (B-reps) with a new concept: the relative position operator.

Representing dimensions as relative position operators

Dimensions are explicitly represented by a relative position operator (RPO). The RPO uses a single scalar parameter (the dimension's nominal value) to move one
The object graph

The object graph consists of nodes and branches, and may contain loops. The nodes are operators, while the branches point to evaluated B-reps of objects. Objects need not be solid. The topmost (root) branch of the graph points to the B-rep of the complete object. Intermediate branches point to B-reps of intermediate objects. Terminal branches point to infinite half-spaces denoted $S_i$.

Figure 7 shows the object graph for the same block-and-slot with a slightly different arrangement of dimensions. Notice that from the root, the graph separates into two major branches. The one on the left represents the block, and the one on the right represents the slot. The slot is a geometric feature, consisting of a set of three faces and an associated set of dimensional constraints.

Figure 7. Object graph for a cube with a slot.

The object graph consists of nodes and branches, and may contain loops. The nodes are operators, while the branches point to evaluated B-reps of objects. Objects need not be solid. The topmost (root) branch of the graph points to the B-rep of the complete object. Intermediate branches point to B-reps of intermediate objects. Terminal branches point to infinite half-spaces denoted $S_i$.

Figure 7 shows the object graph for the same block-and-slot with a slightly different arrangement of dimensions. Notice that from the root, the graph separates into two major branches. The one on the left represents the block, and the one on the right represents the slot. The slot is a geometric feature, consisting of a set of three faces and an associated set of dimensional constraints.

The object graph consists of nodes and branches, and may contain loops. The nodes are operators, while the branches point to evaluated B-reps of objects. Objects need not be solid. The topmost (root) branch of the graph points to the B-rep of the complete object. Intermediate branches point to B-reps of intermediate objects. Terminal branches point to infinite half-spaces denoted $S_i$.

Figure 7 shows the object graph for the same block-and-slot with a slightly different arrangement of dimensions. Notice that from the root, the graph separates into two major branches. The one on the left represents the block, and the one on the right represents the slot. The slot is a geometric feature, consisting of a set of three faces and an associated set of dimensional constraints.
All nodes are of degree three, i.e., three branches impinge upon each node. Each node operates on boundary representations of two objects and produces a third. The operations are closed; any resultant object can be used as an operand by another operator. The nodes include set operators as well as relative position operators. The set operators implemented are the conventional regularized set operators of union, intersection, and difference.

**Evaluation**

The object graph is evaluated from the bottom up. When a dimensional change is made by the user, the object graph is reevaluated. Since B-reps of intermediate objects are retained, only the path from the changed relative position operator to the root branch need be reevaluated. This partial reevaluation helps maintain and maximize the speed and interactivity of the system.

**Results**

A polyhedral modeler based on the representational scheme described above was implemented in C on a Silicon Graphics IRIS 3030 workstation. The appropriate RPO is created whenever the user interactively specifies a dimensional constraint between two faces and is displayed to the user as a directed arrow. Figure 8a shows a block with a slot. The width of the slot is defined by dimension B (40.0), and the location of the slot is defined by dimension A (30.0). Figure 8b shows the result of increasing the value of dimension A to 50.0. Figure 8c shows the change in topology that occurs when the value of dimension A ≥ C − B.

Figures 9a, 9b, and 9c show the result when the same block is dimensioned differently. As before, dimension A (30.0) locates the left face of the slot. Dimension B (70.0), however, now locates the right face of the slot. The width of the slot, B − A, is not dimensioned explicitly. As before, Figure 9b shows the result when the value of dimension A is increased to 50.0. Notice in Figure 9c, where the value of dimension A has increased so that A ≥ B, that the slot “disappears.”
Advantages of the method

The method is considerably more robust than the original variational geometry approach.

Robustness

All problems associated with numerical solutions (e.g., stability, convergence) have been eliminated. Perhaps more importantly, underconstrained objects can be evaluated by the method. Redundancies and conflicts in overconstrained objects can be identified with graph-traversal methods.

Efficiency

The method is efficient because only the branches in the graph affected by a given change need be reevaluated.

Future work

One area of future work will be to represent assembly relationships explicitly, especially contact between mating surfaces.

Assembly relationships

The assembly data structure proposed by Lee and Gossard will be refined and extended to automatically generate chains of dimensions around loops of contact within mechanical assemblies. Figures 10 and 11 show some preliminary work with a model of a child’s puzzle in the disassembled and assembled conditions, respectively.

Automated tolerance synthesis and analysis

Another area of future work will be to develop algorithms to determine automatically the optimal distribution of tolerances around a dimension chain that minimizes manufacturing cost.

Automated redesign of assemblies

Another area of future work will be to adapt computational methods for automated redesign of assemblies developed by Chang, Pabon, and Gallagher to work with this representational scheme.

March 1988
Acknowledgments

We would like to gratefully acknowledge the financial support of the Control Data Corporation.

References


The authors can be contacted through David C. Gossard at the Computer Aided Design Laboratory, Dept. of Mechanical Engineering, MIT, Cambridge, MA 02139.

David C. Gossard is an associate professor with tenure in the Design, Systems and Controls Division of the Mechanical Engineering Department at MIT. He is director of the Computer Aided Design Laboratory where he conducts research on advanced CAE systems for mechanical design. His current research interests include geometric modeling, computational tools for conceptual design, artificial intelligence, and engineering application of neural networks.

Gossard received his BSME and MSME degrees from Purdue University in 1968 and 1970 respectively. He received his PhD from MIT in 1975. He is a member of IEEE, AAAI, and ASME.

Robert Zuffante is software engineer for Aries Technology, Inc. He received his BS and MS in mechanical engineering from MIT. His MS degree focused on a solid modeler suitable for feature-based design. Current research activities include localized fillet and chamfer operations on solids modeled with boundary representation, and incorporation of localized unary operations in a constructive solid geometry tree.

Hiroshi Sakurai is a doctoral candidate in mechanical engineering at MIT. His research interests are solid modeling, computer graphics, artificial intelligence, and their application to computer-aided design and manufacturing. He received his BS degree in ocean engineering from the University of Tokyo and his MS in mechanical engineering from MIT. He is a student member of IEEE.