ANL/ET/CP-93266

RECEIVED
JUL 01 1999
OSTI

Reverse Engineering: Algebraic Boundary Representations to Constructive Solid Geometry*

by

S. F. Buchele** and W. A. Ellingson

** Argonne National Laboratory Lab-Grad Student
Computer Sciences Department
The University of Texas at Austin
Austin, TX 78712

Energy Technology Division
Argonne National Laboratory
Argonne, IL 60439

June 1997

Distribution:

1. R. W. Weeks	5. K. Natesan
2. R. B. Poeppel	6. W. A. Ellingson
3. W. J. Shack	7. W. W. Schertz
4. A. C. Raptis	8. B. Baudino
5. R. A. Valentin	9. Authors
	10. D. Clayton
	11. D. Mingesz

Info Copies:
1. I&NDE Section
2. Ceramics Section
3. H. P. Engel

Paper submitted for presentation to the Eighth Annual Solid Freeform Fabrication Symposium, Austin, TX, August 11-13, 1997.

*Work supported by the U.S. Department of Energy, Energy Research, Office of Computational and Technology Research, Laboratory Technology Research.

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Key Words:

Reverse Engineering
Constructive Solid Geometry
CSG
Boundary Representations

Reverse Engineering: Algebraic Boundary Representations to Constructive Solid Geometry*

by

S. F. Buchele** and W. A. Ellingson

** Argonne National Laboratory Lab-Grad Student
Computer Sciences Department
The University of Texas at Austin
Austin, TX 78712

Energy Technology Division
Argonne National Laboratory
Argonne, IL 60439

Abstract

Recent advances in reverse engineering have focused on recovering a boundary representation (b-rep) of an object, often for integration with rapid prototyping. This boundary representation may be a 3-D point cloud, a triangulation of points, or piecewise algebraic or parametric surfaces. This paper presents work in progress to develop an algorithm to extend the current state of the art in reverse engineering of mechanical parts. This algorithm will take algebraic surface representations as input and will produce a constructive solid geometry (CSG) description that uses solid primitives such as rectangular block, pyramid, sphere, cylinder, and cone. The proposed algorithm will automatically generate a CSG solid model of a part given its algebraic b-rep, thus allowing direct input into a CAD system and subsequent CSG model generation.

1. Introduction

1.1 Solid Modeling

Solid modeling is the process of defining and manipulating unambiguous computer representations of physical solid objects that are specified using normalized or specific dimensions. Computer systems designed for this purpose are called solid modeling or CAD (Computer Aided Design) systems. A fundamental use of these systems is designing or engineering mechanical parts, although their use in other application areas, such as free-form sculpting and automated mesh generation, is currently growing.

Two basic types of CAD system architectures are in use today: (a) boundary representation (b-rep) modelers, which store objects' boundaries and other neighborhood and orientation information; and (b) constructive solid geometry (CSG) modelers, which construct objects from solid primitives by using the Boolean operations union, intersect, and difference; store CSG trees; and compute boundary representations when needed [3]. In practice, few CAD systems can be classified strictly as one type or the other, however. Sweep representations, in which a 2-D or 3-D object is swept along a 3-D trajectory, and spatial partitioning representations, in which a solid is decomposed into collections of adjoining cells or solids, are often included in CAD modelers as well [1,2]. Hybrid CAD systems, in which some combination of the above representations are used in tandem or conjunction with each other in a single CAD system, are the norm for current modeling systems. In particular, CAD systems for the design of mechanical objects provide a CSG interface, regardless of the underlying architecture(s) of the CAD system itself. For example, a CAD system with a Non-Uniform Rational B-Spline (NURBS) b-rep representation as its base architecture may provide a CSG-type input interface for the engineer. In this case, a part definition is entered as the union, intersection, and difference of instances of (properly parameterized) solid primitives. The boundary of the resulting CSG-specified solid object is then evaluated and represented internally in b-rep fashion. The initial CSG representation of the object may be stored internally as well. User interfaces that allow sweeps to be entered are also common in conjunction with CSG user interfaces, regardless of the underlying CAD architecture.

1.2 Reverse Engineering

Reverse engineering (or geometry recovery) of a part (mechanical, biomedical, etc.) is the general process of recovering a model of a physical object from information obtained by some type of sensing technique. These sensing techniques include, but are not limited to, computed tomographic imaging (CAT scans), nuclear magnetic resonance imaging (MRI scans), laser rangefinder scans, stereoscopic sensing, and coordinate measuring machine (CMM) sensing. In some cases, the process of recovering a model of the part is enhanced by using an existing model of an as-designed part and altering the existing model on the basis of information from the sensing device. In many other cases, however, an original model of the part does not exist. Many currently manufactured parts predate the existence or prevalence of CAD systems, which are a relatively new development. In addition, many parts have been modified in use and no record exists of these modifications. In these cases, it is advantageous to derive a CAD model of an existing part.

Most current advances in reverse engineering of mechanical parts have focused on recovering a b-rep of the part under study. Recovered boundary representations of the surfaces of the part can vary from a triangulation of points to piecewise parametric surfaces (common representations include parametric spline surfaces and NURBS surfaces) to piecewise algebraic surfaces. For reverse engineering, a triangulation (or tesselation) of points is often not sufficiently detailed to fully recover the original geometry of an object. Piecewise parametric surfaces, while providing a high level of detail, can fail to recover the original geometry of the object due to inherent limitations of parametric representations. Consider for example a circular cylinder. A typical b-rep of a cylinder describes the surfaces on the boundary of the cylinder; two planar surfaces at the top and bottom, and a parametric surface wrapped around the cylinder. Anyone who wishes to recover specific quantitative information from this object desires information such as the height and diameter of the cylinder. Such parameters are not readily available from a parametric boundary representation of the cylinder [3].

Most mechanical objects are composed of well-defined primitives, and mechanical designers often think in terms of primitives when describing an object. Recovering a bounded solid CSG representation of an object does provide information about the object that is directly applicable to the way an engineer would model the object in the usual forward engineering sense. Figure 1 is a diagram of a CSG tree of a mechanical part (a portion of an injection mold), indicating the constructive process inherent in the engineering process. For the purpose of reverse engineering, a CSG representation of a solid is a "user friendly" form for the geometric data, because it is easier than a boundary representation for an engineer to visualize and manipulate and could be entered into CSG, b-rep, and hybrid modeling CAD systems through a CSG user interface. The Initial Graphics Exchange Specification (IGES), a standard representation for interchange of CAD data, includes a specification for CSG trees as an exchange data format. This is important for portability of any CSG file.

1.3 Previous Work

Extensive research has already been conducted in computing a b-rep of a solid from sensed data, but that work is not discussed here because of space limitations. The work described in this paper focuses on the development of a CSG-type solid model. The importance of CSG in rapid prototyping, an area related to reverse engineering, has been discussed by Crawford [4].

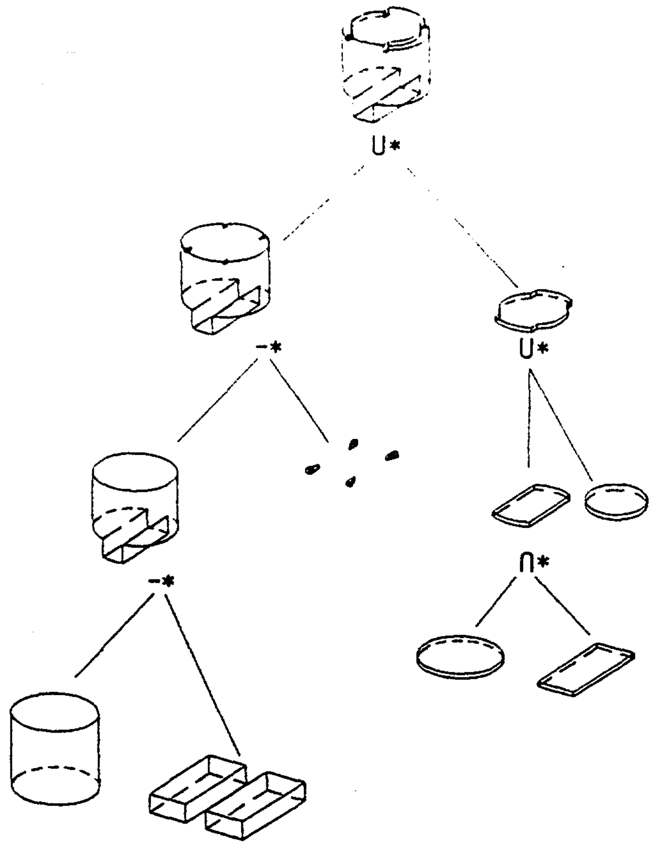

Fig. 1. Diagram of a CSG tree of a mechanical object.

At the advent of modern CAD systems, a duality developed between CSG and b-rep representations. Algorithms were quickly developed and optimized for the CSG-to-b-rep conversion process. The problem of b-rep-to-CSG was considered too difficult and unnecessary for general-purpose implementation. In recent years, however, progress has been made in the b-rep-to-CSG conversion process.

In 2-D, Peterson [5] presented a solution to the problem of finding a halfspace CSG representation for the interior of a closed curve. Peterson approached the problem from a reverse engineering viewpoint, recognizing that many 3-D mechanical objects are sweeps of a closed 2-D curve along an orthogonal axis. Also in 2-D, Vossler [6] presented a solution to finding a CSG representation for the interior of a closed curve using bounded primitives, such as rectangle, circle, chorded circle, and right triangle. In 2-D, therefore, the problem is considered solved for cases with modest (and realistic) limitations, for both halfspace and bounded solid CSG.

Limited attempts have been made to date to construct a solid-primitive 3-D CSG representation from a point cloud or a b-rep of an object [7-10]. These works do not extend to general 3-D objects. For example, in Lin and Chen's work [7], all planar surfaces are assumed to be part of a cube. A further assumption of Lin and Chen is that each primitive can act as a subtractor no more than once, which in general prohibits objects that contain more than one subtraction operator on any path from the root to a leaf in the CSG tree representing the object. The work of Woo [8] and Tang and Woo [9,10] does not scale to nonpolyhedral objects due to the use of the convex hull operator.

The general 3-D b-rep-to-CSG conversion has only recently been tackled [11-13]; these pioneering works deal only with halfspace CSG representations, however. A halfspace CSG tree is a CSG tree in which the leaves of the tree are halfspaces. In the present work, a CSG tree is used in which the leaves of the tree are bounded solid primitives (the typical case in CAD systems with a CSG interface) and such a CSG tree is defined as a bounded solid CSG tree (e.g., that shown in Fig. 1).

Shapiro and Vossler [11] first presented a solution to recovering a halfspace CSG tree for 2-D objects. They also presented a solution to recovering a halfspace CSG tree of polyhedral solids [12]. Shapiro and Vossler followed up this work with steps to extend the work to include solids bounded by quadric surfaces [13]. The solution technique employed by Shapiro and Vossler is cumbersome and necessitates extensive CSG tree simplification. However, their method [13] of adding separating halfspaces to a solid object to make the object describable is of significant importance.

2. Problem Description

2.1 Background

CAD systems typically use regularized sets and regularized set operations. This avoids problems with manipulating regions of space with empty interiors that are not meaningful in a solid modeling context. The regularization of a set X, reg (X), is the closure of the interior of the set X. Thus, the regularization of a set with empty interior, such as a line in 2-D or a surface in 3-D, is the empty set. The regularized set operations $\cup^*$ (union), $\cap^*$ (intersection), $-^*$ (difference), and $^{-*}$ (complement) produce a regularized set as the result of the operation on the regularized set operands and are defined as follows:

$$a \cup^* b = reg(a \cup b), \tag{1}$$

$$a \cap^* b = reg(a \cap b), \tag{2}$$

$$a -^* b = reg(a - b), \tag{3}$$

$$\overline{a}^* = reg(\overline{a}). \tag{4}$$

A halfspace ψ of R^3 is a set of the form $\psi = \{(x,y,z): g(x,y,z) \geq 0\}$ for some function $g: R^3 \rightarrow R$. The work herein utilizes quadric functions, that is, polynomial functions of degree 2. The zeros of the function map a surface in 3-D. A surface, $S_\psi = \{(x,y,z): g(x,y,z) = 0\}$, *induces* the two halfspaces $\psi = \{(x,y,z): g(x,y,z) \geq 0\}$ and $\overline{\psi} = \{(x,y,z): g(x,y,z) \leq 0\}$. It can be shown that for the surfaces under consideration in this work, namely planar, spherical, cylindrical, and conical surfaces, the halfspaces induced by these surfaces are regularized sets. A solid will be defined as a non-empty regularized subset of R^3. A natural halfspace of a solid is defined to be a halfspace induced by a surface in the b-rep of the solid.

A solid Γ is said to be *describable* by halfspaces $\Psi = \{\psi_1, \ldots, \psi_N\}$ if there exists a (halfspace) CSG tree representing the solid, such that each leaf of the tree is an element of the set $\left\{\psi_1, \overline{\psi_1}^* \ldots, \psi_N, \overline{\psi_N}^*\right\}$ [11,12].

2.2 Approach

In our approach, a mechanical object is represented as a bounded solid, a regularized, non-empty, bounded subset of R^3. The subset of quadric surfaces that is allowable to bound the solid contains the surfaces of the bounded solid CSG primitives to be used as primitives in the CSG representation. Primitives to be used consist of a rectangular solid, a pyramid, a sphere, a cylinder, and a cone. Therefore, the allowable surfaces in the boundary representation of the solid are planes, spherical surfaces, cylindrical surfaces, and conical surfaces (surfaces bounding cones), although other nondegenerate quadric surfaces of a single sheet could be used as well. It is assumed that the input bounding surfaces are piecewise algebraic surfaces in implicit form (e.g., a spherical surface in the form $x^2 + y^2 + z^2 - r^2 = 0$), with an additional representation of the boundary of each surface patch. Our approach seeks to produce a bounded solid CSG tree of Γ consisting of the operations union, intersection, and difference operating on the canonical primitives rectangular solid, pyramid, sphere, cylinder, and cone, if the solid Γ is describable by its natural halfspaces. If the solid Γ is not describable by its natural halfspaces, this condition will be detected and the attempt will be terminated.*

*An extension would be to use the work of Shapiro and Vossler [13] to add separating half-spaces to make the solid describable.

This approach is shown schematically in Fig. 2. A quadric binary space partitioning tree (BSP tree) is used as an intermediate representation, which facilitates the process. A (typical) BSP tree is a binary tree used to represent arbitrary polyhedra, in which internal nodes are planes and leaf nodes represent homogeneous regions of space called "in" cells and "out" cells [1]. The definition of a BSP tree is extended to a *quadric BSP tree*, in which internal nodes are no longer limited to be planes, but can also be quadric surfaces such as the spherical, cylindrical, and conical surfaces we will use. Thus, in a quadric BSP tree, each internal node is a quadric surface with two child pointers, one for each side of the halfspace induced by the surface. Just as in a typical BSP tree, if either child halfspace is subdivided further, then it is the root of a subtree; if the halfspace is homogeneous with respect to the solid, then it is a leaf node, classified as either an "in" cell or an "out" cell. Figure 3 shows an example of a 2-D solid and a BSP tree representation of the solid.

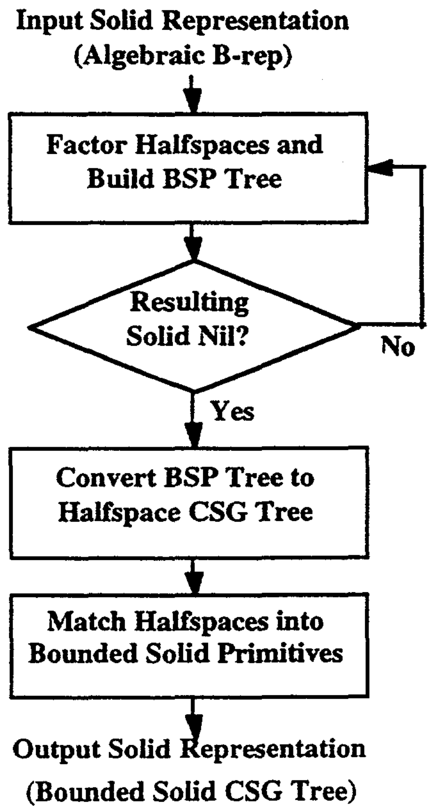

Fig. 2. Flow chart of approach.

In our approach, the input is currently an algebraic b-rep of the solid Γ and a null BSP tree. Surfaces are systematically factored from the b-rep of the object, and once factored, the surfaces and their induced halfspaces are added to the BSP tree representation. As the surfaces are factored, additional information in the form of *special faces* is added to the data structure representing the solid Γ (partially factored), to facilitate the building of a valid BSP tree representation.

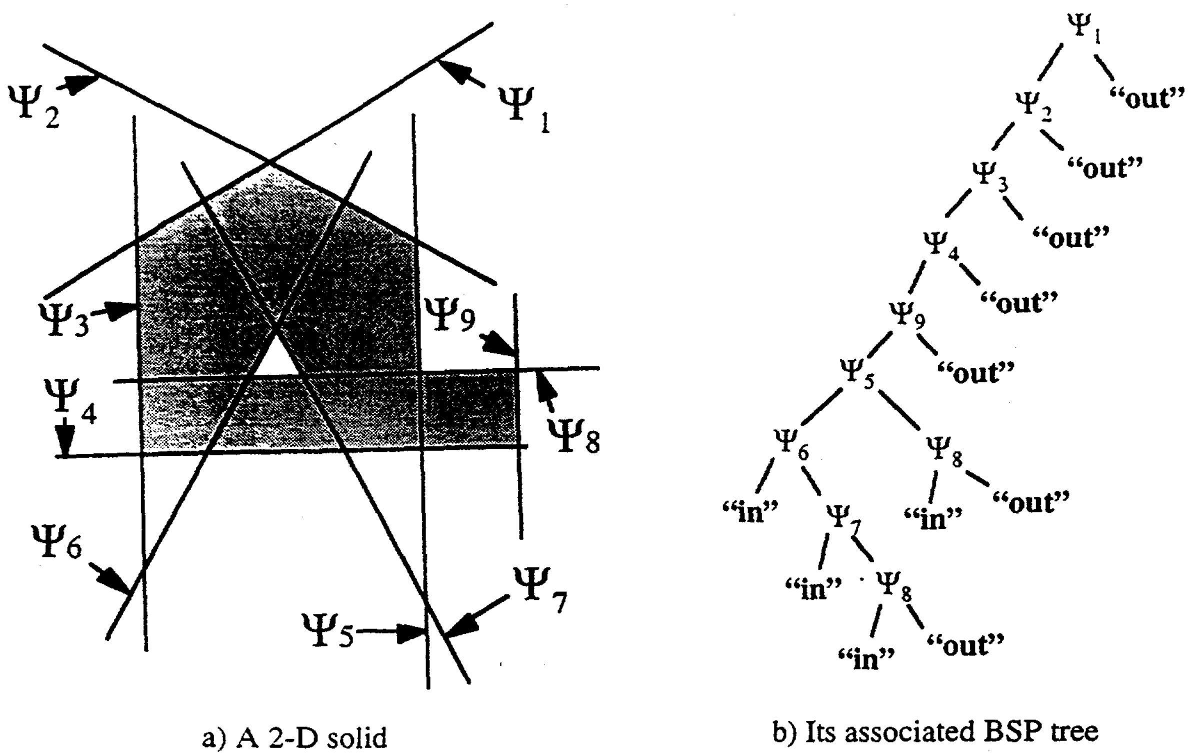

Fig. 3. BSP Example: a) Diagram of a 2-D bounded solid (shaded region is interior of bounded solid); b) BSP tree representation of solid shown in a).

When there are no more unfactored surfaces in the b-rep of the solid, the BSP tree representation is complete. The BSP tree is converted to a halfspace CSG tree, performing some simplification techniques as a part of this process. Further simplification may be performed after this operation to the halfspace CSG tree itself, if desired. The last step is to match the halfspaces into solid primitives and construct the bounded solid CSG tree.

3. Conclusions

A new approach that uses 3-D CSG has been described to allow an advancement in reverse engineering. In particular, this method will allow algebraic surface representations (that can be obtained from a 3-D point cloud data set) to be used as input to the 3-D CSG algorithm, which can be used to provide direct input into any CAD package. This research is an extension of the work of Shapiro and Vossler to allow for bounded solid CSG conversion from a b-rep of an object. The approach is expected to be more intuitive and computationally efficient.

Theoretical and algorithmic development of this process is continuing. In particular, computational analysis theory, halfspace CSG tree simplification, and heuristics needed to construct the bounded solid CSG tree have yet to be fully investigated. Implementation, beginning with the b-rep to BSP procedure, is commencing.

Possible extensions to this work may facilitate its usefulness in the areas of reverse engineering and solid modeling. First, the possibility of including sweep representations as primitives in the bounded solid CSG tree would increase the applicability of the process, because sweep representations are often combined with CSG interfaces in mechanical CAD systems. It is currently unknown if parametric rather than algebraic representations of the quadric surfaces (or sweeps) may be used in practice; this would also increase the applicability of the procedure because most commercially available point-cloud-to-b-rep software outputs parametric surface representations. Converting parametric to algebraic representations has been discussed [14], but there may exist fundamental problems of numerically unstable computations.

The work described here is part of a contribution to the early stages of general-purpose 3-D b-rep-to-CSG conversion. The advantages of the proposed algorithm in the area of reverse engineering are in providing a more intuitive model representation of a mechanical part than that provided by existing b-rep recovery methods.

References

[1] J. D. Foley, A. van Dam, S. F. Feiner, and J. F. Hughes. Computer Graphics: Principle and Practice. Addison-Wesley, 1990.

[2] M. E. Mortenson. Geometric Modeling. John Wiley & Sons, 1985.

[3] A. A. G. Requicha and J. R. Rossignac. *Solid Modeling and Beyond*, IEEE Computer Graphics and Applications, Sept. 1992, pp. 31-44.

[4] R. H. Crawford. *Computer Aspects of Solid Freeform Fabrication: Geometry, Process Control, and Design*, Proceedings of the 1993 Solid Freeform Fabrication Symposium, Aug. 1993, pp. 102-112.

[5] D. P. Peterson. *Boundary to Constructive Solid Geometry Mappings: A Focus on 2D Issues*. Computer Aided Design, Vol. 18, No. 1, Jan./Feb. 1986, pp. 3-14.

[6] D. L. Vossler. *Sweep-to-CSG Conversion Using Pattern Recognition Techniques*, IEEE Computer Graphics and Applications, Vol. 5, No. 8, Aug. 1985, pp. 61-68.

[7] W. Lin and T. Chen, *CSG-Based Object Recognition Using Range Images*, Proceedings of the 9th International Conference on Pattern Recognition, Rome, Italy, Nov. 1988, pp. 99-103.

[8] T. C. Woo, *Feature Extraction by Volume Decomposition*, Proceedings on CAD/CAM Technology in Mechanical Engineering, Cambridge, MA, 1982, pp. 76-94.

[9] K. Tang and T. Woo, *Algorithmic Aspects of Alternating Sum of Volumes: Part 1: Data Structure and Difference Operation*, Computer Aided Design, Vol. 23, No. 5, June 1991, pp. 357-366.

[10] K. Tang and T. Woo, *Algorithmic Aspects of Alternating Sum of Volumes: Part 2: Nonconvergence and its Remedy*, Computer Aided Design, Vol. 23, No. 6, July 1991, pp. 435-443.

[11] V. Shapiro and D. L. Vossler, *Efficient CSG Representations of Two-Dimensional Solids*, Transactions of the ASME, Vol. 113, Sept. 1991, pp. 292-305.

[12] V. Shapiro and D. L. Vossler, *Construction and Optimization of CSG Representations*, Computer Aided Design, Vol. 23, No. 1, 1991, pp. 4-20.

[13] V. Shapiro and D. L. Vossler, *Separation for Boundary to CSG Conversion*, ACM Transactions on Graphics, Vol. 12, No. 1, Jan. 1993, pp. 25-55.

[14] D. Manocha and J. F. Canny, *The Implicit Representation of Rational Parametric Surfaces*, Journal of Scientific Computation, Vol. 15, No. 2, 1992, pp. 485-510.