

Research in Geometric Modeling*

Technical Report 83-7

Tony C. Woo

Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109

*Invited Paper, Conference on University Progress in
Computer Aided Engineering, Design and Manufacturing,
Provo Utah, April 1983.

This work was supported in part by AFOSR Grant F4920-82-C-0089.

0. Abstract

The objective of this paper is three-fold. First, we wish to describe a research program in Integrated Manufacturing and Robotics at the University of Michigan. Its activities cover a wide range of topics from materials processing to information process. Second, to illustrate the interaction in this multi-faceted research environment, two projects are described. One is an interface from geometric modeling to finite element analysis, and the other is an interface from geometric modeling to robotics. Third, to bring out the underlying design principles, we emphasize storage and run time efficiency of data structures and algorithms.

1. Introduction

In the interest of improving industrial productivity, a coordinated research program in manufacturing and robotics was initiated at the University of Michigan. The background of the program is described in this section. The body of the paper discusses two research projects that interweave design with engineering and manufacturing. These two projects result in the capability to generate a finite element solid mesh and to compute the gripping positions in assembly using robots. Section 2 discusses the approach we have taken in designing algorithms so that the end product is not just a program that works but one that is both time and storage efficient under varying conditions. Section 3 traces back to the design of a data structure that permits uniformity and efficiency. Section 4 and 5 uses the scenarios of finite elements and automatic assembly to illustrate analysis of worst case and average case time and storage complexities.

A Center for Robotics and Integrated Manufacturing (CRIM) has been established at the University of Michigan to conduct research and development in computer based automation of production processes. The Center began with an initial funding of \$1.9 million over a two-year period from the U-M Regents in October 1981. It is currently funded by a large number of industrial, foundation, and government sponsors including a three-year \$3.4 million grant from the Air Force Office of Scientific Research on Integrated Manufacturing and Robotics.

The Center is under the directorship of Professor Daniel Atkins and the participating faculty members come from six departments in the College of Engineering.

- Aerospace Engineering
- Electrical and Computer Engineering
- Industrial and Operations Engineering
- Materials and Metallurgical Engineering
- Mechanical Engineering and Applied Mechanics
- Naval Architecture and Marine Engineering

The research is organized under three divisions:

- . Design and Manufacturing
- . Management Systems, and
- . Robotics Systems.

The activities in the Design and Manufacturing division are concentrated in four areas:

- . Design and Manufacturing
 - * Manufacturing Processes,
 - * Control of Processes and Systems,
 - * Computer Aided Design, and
 - * Computer Aided Manufacturing.

The focuses the Management Systems division are:

- . Management Systems
 - * Human Factors and Organizational Aspects of Automation,
 - * Production Planning and Operations, and
 - * Management Information Systems.

The major categories in the Robotics Systems division are:

. Robotics Systems

- * Robot Control,
- * Sensing Systems,
- * Specialized Computer Architecture, and
- * High Level Language Development.

Though research in these areas are highly specialized, there exists a great deal of interaction among them. Consider, for example, optimization for Production Planning and Operations in the Management Systems division. It spans the control of manufacturing processes such as grinding and ECG, the structural analysis of such diverse designs are micro-Winchester disk drives and marine risers, and the computation of trajectories for robot arms. Another thread that runs across the divisional boundaries is the use of geometric information for mechanical components and systems as described in this paper.

2. Research Methodology

This paper shows two facets of an integrated design and manufacturing - the automatic generation of a solid, finite element mesh for engineering analysis and the computation for gripping positions in automatic assembly. Though these two projects can be described individually and in a procedural fashion, we wish to bring out the methodology in our research such that the rationale and the intents are clear.

Consider three phases of the life cycle of a geometric model in an integrated design and manufacturing environment.

- (1) data structure design
(complexity analysis)
- (2) algorithm development
(probabilistic modeling)
- (3) application interface

A desirable characteristic of a data structure in a modeling system is that it be storage and time efficient. But because the data structure is to serve a number of applications, it must also be fairly general. To overcome the contrasting nature of these two requirements, efficiency and generality, we took the top-down, back-tracing approach. We examine the applications of a modeler and describe the application algorithms at a high level. We gradually work our way down and back until we reach the level where data retrieval occurs. We find the common denominators of all the retrieval routines and design a data structure that responds to the queries efficiently. Techniques in the analysis of algorithms and computational complexity are used so that we would be able to tell how fast a certain algorithm would run and how much storage the data structure would take before the programs are written. Typically, we would have measures for the best and the worst case as a function of the complexity of the actual mechanical object.

After the programs are written, we find that the actual run-time performance varies within the bounds but it seldom reaches the best or the worst case. We also find that by re-sequencing the subprograms the overall performance improves or degrades. If we know what a typical set of geometric input look like, we can sequence the subprograms for optimal performance. But this is hardly the way to do things (despite its common practice.) There appear to be a need to model the stochastic nature of geometric objects. If we have such a model then we can say, for example, 65 per cent of the time the program will run at such and such a speed, without running the program to the extent that the sample satisfies the law of large numbers.

It is in this spirit that we develop our data structure, algorithms and interfaces. Though these three phases of the life cycle of a geometric model can be carried out sequentially or in parallel, we carry them out iteratively. The feedback provided between the stages are the analysis of time and storage complexity and the modeling of probalistic behavior. The former provides the bounds for the best and the worst case performance and the latter indicates the average case performance. We wish to illustrate efficiency using the scenario of integrating design data to engineering and manufacturing.

3. Geometric Modeling

As we make a distinction between source and object codes in programming, we distinguish the description of a geometric model from its representation. A description reflects the designer's view of a three-dimensional information is organized internally in the computer.

There are several geometric modeling systems at the University of Michigan including Synthavision [5], Build [3], Arch:Model [2] and 3DFORM [10]. They have different descriptive schema and internal organizations. Instead of enforcing a single language for input which might unfavorably bias certain application users, we design a data structure into which the various representations can be post processed.

Our data structure is efficient. Its storage requirement is $6E$ where E is the number of edges of the object. For example, if an object has 500 edges then it only takes 3,000 cells to store its complete topology of vertices, edges, faces and their relations. (We can refer to this kind of storage complexity as linear in E or simply linear.) The access time for the primitive inquiries in terms of which all procedures are written is linear in the worst case and constant in the best case. Significantly, we have been able to prove that the average performance takes only twice the amount of time it does for the best case [12]. Therefore, the accessing of the topological and geometric data, on the average, is also constant. For techniques of analysing data structures for geometric modeling, see [9].

4. Design and Manufacturing

Engineering analysis of a mechanical component or system requires input preparation. The effort required for input that is geometric in nature, as is the case with finite elements, is often quite substantial. As a result, computer-aided tools are developed that permit a designer to edit a mesh interactively. Though an improvement over the manual method, the interactive method has two shortcomings. First, automatic generation of a surface or solid mesh is limited to axialsymmetric objects (such as shafts) or 2 1/2D objects (such as beams and plates) that do not have "off-center" holes. To overcome this drawback, the user inserts and deletes nodes and elements. Second, the number of nodes is often very large. (If we assign 10 nodes per edge, a 500 edge object would have 5,000 nodes and several hundred elements.) User error is likely to occur in a complex situation.

Our approach to the automatic generation of finite elements is as follows. From the given geometry and topology in a model, we generate a small number of solid elements that completely fill the interior of the object. Since solid elements imply a surface mesh, it would satisfy users that require either a surface mesh or a solid mesh. Since the number of elements is small, the elements can be recursively subdivided into more elements depending on the loading conditions.

From the previously mentioned data structure, we developed an algorithm for generating tetrahedral, pentahedral and hexahedral elements for objects with or without holes [11]. In the best case, the algorithm runs in quadratic time, or $O(N^2)$ time, and in linear storage. In the worst case, it takes $O(N^2)$ time and quadratic storage. We do not yet have a measure for the average case.

Two items are worthy of notice. First the number of solid elements in a three-dimensional object is not unique. Consider, for example, a cube that can be subdivided either into five tetrahedra or six tetrahedra. Second, the minimal number of solid elements is exceedingly difficult to obtain [4]. It is the combination of these two factors that contribute to the possible worst case. Currently, we only have means to force the algorithm towards the lower bound.

5. Manufacturing and Assembly

In pursuit of finding the characteristics of an "average" three-dimensional object, we have taken one step towards that goal. While the long range research interest is to be able to assess average run-time performance of geometric programs, the work reported here has an immediate application in automatic gripping for assembly [7].

Consider a scenario in which a computer-controlled camera directs a manipulator in an automatic assembly environment. A key task is for the vision system to recognize what component the projective image corresponds to

and where the gripping points are. Though there are infinitely many projections possible, an object can assume only one of the finitely many "stable orientations" [1, 6]. We are interested in not only the stable orientations but also their probability distribution in terms of likelihoods. With a likelihood ranking of the silhouettes of a certain component, the vision system can perform more efficiently in searching and recognition.

Recognizing the infeasibility of deterministic solutions to the problem, we constructed a state transition model for determining the probabilities of a stable orientations. Each state corresponds to a stable face on the convex hull of the object. The probability of each state is computed by summing the probabilities of the neighboring states multiplied by the conditional transition probability. The model therefore simulates the rolling of an object on a plane. This model is initiated by an a priori probability distribution of the states that simulates dropping. The input to the model are the initial energy (kinetic and potential) of the object, the roughness (friction) and the concavity of the ground plane. The determination of the various probabilities are discussed in [8].

Acknowledgement

The author acknowledges the support of his colleagues at Michigan and the sponsorship of AFOSR. Two participating members are particularly outstanding. Tim Thomasma contributed to the finite element interfacing work. Jan Wolter implemented the probabilistic model and meticulously carried out the actual experiment of throwing n -sided dice.

References

- [1] Boothroyd, G. and C. Ho, "Natural Resting Aspects of Parts for Automatic Handling", ASME Paper (76-WA/Prod 40), December 1976.
- [2] Borkin, H., J. McIntosh and J. Turner, "The Development of Three-Dimensional Spatial Modeling Techniques for the Construction Planning of Nuclear Power Plants", ACM SIGGRAPH, Vol. 12, No. 3, August 1978, pp. 341-347.
- [3] Cary, C. A. G. "BUILD User's Guide", CAD Group Document No. 102, University of Cambridge, Computer Laboratory, Cambridge, U.K., November 1979.
- [4] Cottle, R. W. "Minimal Triangulation of the 4-Cube", Discrete Mathematics, Vol. 40, No. 1, June 1982, pp. 25-30.
- [5] MAGI, "Synthavision User's Manual", Mathematical Applications Group, Inc. Elmsford, New York, January 1982.
- [6] Wesley, M. A., T. Lozano-Perez, L. I. Lieberman, M. A. Lavin, D. D. Grossman, "A Geometric Modeling Systems for Automated Mechanical Assembly", IBM Journal Research Development, Vol. 24, No. 1, January 1980, pp. 64-74.
- [7] Wolter, J., T. C. Woo and R. A. Volz, "Gripping Positions for 3D Objects", IEEE Industrial Applications Society Annual Meeting, San Francisco, Calif., October 1982.
- [8] Wolter, J., T. C. Woo, and R. A. Volz, "Probabilistic Distribution of Stable Orientations of 3D Objects", also available as Technical Report 83-5, Department of Industrial and Operations Engineering, The University of Michigan Ann Arbor, Michigan 48109, February 1983.
- [9] Woo, T. C., "Data Structures for Topological Inquiries", to appear in IEEE Computer Graphics and Applications, also available as Technical Report 83-2, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109, January 1983.

- [10] Woo, T. C. and R. F. Poshadlo, "A Feature Oriented Design Modification System", Proc. First Annual Conference on Computer Graphics in CAD/CAM Systems, Cambridge Mass. April 1979.
- [11] Woo, T. C. and T. Thomasma, "An Algorithm for Generating Solid Elements in Objects with Holes", to appear in Journal of Computers and Structures, also available as Technical Report 82-5, Department of Industrial and Operations Engineering, Ann Arbor, Michigan 48109, April 1982.
- [12] Woo, T. C. and J. Wolter, "A Constant Average Time Linear Storage Data Structure for 3D Objects" also available as Technical Report 83-4, Department of Industrial and Operations Engineering, Ann Arbor, Michigan 48109, February 1983.

UNIVERSITY OF MICHIGAN



3 9015 03994 8610