Solid Modeling and Geometric Reasoning for Design and Process Planning

Dana Nau,[2] Nicholas Ide,[3] Raghu Karinthi,[4] George Vanecek,[4] and Qiang Yang[4]

University of Maryland

Summary

This paper describes our work on the integration of techniques for solid modeling, geometric reasoning, and multi-goal planning, with application to computer-aided design and manufacturing. This work is being done with two long-term goals in mind: the development of a practical integrated system for designing metal parts and planning their manufacture, and the investigation of fundamental issues in representing and reasoning about three-dimensional objects. We believe this work will have utility not only for automated manufacturing, but also for other problems in design and multi-goal planning.

1. Introduction

One problem facing modern industry is the lack of a skilled labor force to produce machined parts as has been done in the past. In the near future, this problem may become acute for a number of manufacturing tasks. This has led to considerable interest in ways to automate various manufacturing tasks.

Our first work in this area was in the development of AI techniques for automated process selection. Since we believe that the rule-based approach used in most knowledge-based systems is not the most appropriate way to do process planning, we have developed a different approach, based on hierarchical abstraction. The implementation of this idea first resulted in SIPP, a process selection system written in Prolog, and later led to SIPS, a more sophisticated system written in Lisp. The evolution of SIPP and SIPS over the last several years have been described elsewhere [6,7,8,9,10,11,12], so SIPP and SIPS will not be described again here.

Recently we have increasingly become interested in integrating process planning with design and solid modeling, for two reasons. First, a good design system is essential to provide a decent interface to a process planning system. Second, there are process planning tasks which cannot be performed correctly without extensive interactions with a solid modeler. Our current work focuses on the following topics:

1. solid modeling techniques specifically suited for integration with automated reasoning systems such as process planning systems;
2. computer-aided design systems capable of reasoning about three-dimensional objects, both for use as a design aid and also for use in integrating design with process planning;
3. ways to reason about interacting features during design and planning.

These topics are discussed in Sections 2-4, respectively. Section 5 contains concluding remarks.

[1]This work has been supported in part by the following sources: an NSF Presidential Young Investigator Award to Dana Nau, NSF Grant NSFD CDR-85-00108 to the University of Maryland Systems Research Center, General Motors Research Laboratories, and Texas Instruments.

[2]Department of Computer Science, Institute for Advanced Computer Studies, and Systems Research Center.

[3]Current address: Century Computing, 1100 West St., Laurel, MD 20707.

[4]Department of Computer Science.

2. Solid Modeling

Most approaches to the integration of solid modeling with automated process planning have essentially involved using a geometric modeler as a front end to a process planning system. Two examples of this involve the use of SIPS as the process planning system: the interface produced at General Motors between SIPS and the MBF/X-Solid CAD system [12], and the interface being built at the National Bureau of Standards between SIPS and Unicad/Romulus [1]. Such interfaces make the process planning system more convenient to use, but in order to generate correct process plans for complex objects this approach is not sufficient. What processes can be used for some machinable feature—or whether the feature can even be made at all—may depend on geometric information not available solely from the descriptions of the features. To get this information will require the process planning system to interact extensively with the solid modeler during process planning.

For example, consider the task of drilling a hole in a flat surface. Although this is usually easy, it will be impossible if some other part of the object interferes with the tool trajectory. This condition can be recognized through the specification of geometric constraints and verification of these constraints through queries to a solid modeler. In more complex examples, the process planning system will need to make a large number of such queries.

Examples such as the one above can be handled by interfacing the process planning system to an existing solid modeler—and in fact, we have interfaced SIPS to the PADL-2 solid modeler for this purpose [4]. However, our experience at building this interface, as well as our experience with several other solid modelers, has led us to conclude that most existing solid modelers are not adequate for this purpose. One reason for this is that the primary focus guiding the development of most solid modelers has been the fact that they will be used by humans. Thus, much work has been done on efficient algorithms for operations such as rendering, but less attention has been paid to providing easy and efficient ways to answer queries, retrieve pieces of the objects being modeled, and make incremental changes.

The thorough integration of a solid modeler with a process planning system (or with various other automated systems) will require the ability to do several different kinds of solid modeling operations very quickly. Some of these operations include rotating or translating the solid, extracting bounding surfaces from it, and performing set operations such as union and intersection of solids. We believe that no existing approach to solid modeling can perform all of these tasks quickly and accurately enough.

In our opinion, the approach to solid modeling which comes the closest to fitting the above requirements is boundary representation. Using boundary representations, it is easy to do fast translations, rotations, and boundary extraction, but set operations are more time-consuming. Our approach is to enhance the capabilities of boundary representations, by developing fast algorithms for set operations.

When set operations are done on solid objects represented using boundary representation, the usual approach is to check each edge of one object against every edge of the other object. This results in a cost worse than $O(n^2)$. We have developed a faster algorithm based on the divide-and-conquer paradigm, in the form of non-regular decomposition of space [13]. This results in an average-case performance which has empirically been found to be $O(n \log n)$, where n is the total number of edges of both the input and the output. We have implemented a modeling system using non-regular decomposition for the representation and manipulation of two-dimensional polygons [14]. We are extending our algorithm to handle three-dimensional objects containing both flat and curved surfaces, and we are currently building a three-dimensional solid modeler using this approach.

3. Reasoning about Features

One of the greatest problems facing the manufacturing industry today is the differences in product description in various segments of the industry. Many tools created for aiding the design and the manufacturing processes seperately, but the problem is how to provide automatic integration of these tools.

CAD-generated objects can be defined in terms of the complete geometry of the part. The descriptions contain the faces, edges and vertices making up the part. For the purpose of manufacturing, the geometry and topology are the same, but the meaning associated with this geometric structure is different, and dictates a change in the description. An object which, to the designer, is a block minus a cylinder, is to the manufacturing engineer a block with a hole and certain tolerances.

One proposed way to handle this incompatibility is *automated feature extraction*, which consists of automating the task of determining the manufacturing features of a part from its geometry. This is an extremely difficult process, and the reader is referred to [3,5] for a discussion of the complexities involved. Some of the tougher problems include (1) inferring faces needed to describe the machining operations that do not appear in the CAD description, and (2) extracting a feature which intersects or otherwise interacts with other features, without disturbing those other features.

Another approach is *design by features*, in which the user builds a solid model of an object by specifying directly its "manufacturing features." For example, one might start with a model of a piece of metal stock, and modify it by adding holes, slots, pockets, and other machinable features. One problem with design by features is that it requires a significant change in the way a feature is designed. Traditionally, a designer designs a part for functionality, and a process engineer determines which are the manufacturable features are. However, design by features places the designer under the constraints of not merely having to design for functionality, but at the same time specify all of the manufacturable features as part of the geometry—a task which the designer is not normally qualified to do.

To overcome this problem, it would be desirable to allow the designer to use not manufacturing features, but instead "design features," which may not correspond directly to manufacturing operations, but which make sense to the designer. This would require the system to translate the design features into manufacturing features after the design of the part was completed. With an intelligently chosen set of available features and ways for combining them, this should be less complicated than extracting manufacturing features from an ordinary solid model.

Given a definition of a part as a combination of design features, there may be several possible ways to translate the part into a collection of machinable features. Different translations of the same object could result in very different process plans for that object, with different costs. For example, if a wide slot bisects a pocket, it may lead to a cheaper plan if the bisected pocket is considered to be two separate machinable features rather than just one. However, if the slot is narrow, it may be better to consider the pocket to be a single feature.

We intend to develop a system for feature-based design and analysis, with the ability to reason about interactions among the features in order to make good decisions about how to translate design features into machinable features. This system will make extensive use of the solid modeler described in Section 2, and will provide information about feature interactions for use by the process planning system (see Section 4).

4. Reasoning about Interacting Features

The SIPS process selection system works well when the plans for the various features are independent. However, the problem becomes much more complicated when one tries to handle interactions among features (for example, see [2,15]).

For example, consider an object containing two holes `h1` and `h2`, both having the same diameter and the same machining tolerances. Suppose `h1` can be created by either twist drilling or spade drilling. Then the least costly way to make `h1` is twist drilling. If the depth of `h2` is sufficiently large, `h2` may require spade drilling rather than twist drilling. In this case, the cheapest way to make the entire object is to use spade drilling for both `h1` and `h2` in order to avoid a tool change—even though spade drilling would not be the cheapest way to make `h1` if `h1` were the only hole being made.

The problem described above can be characterized as a problem in multiple-goal planning, with the restriction that all interactions among the actions in the plans should be expressible in terms of partial ordering constraints, identity constraints, and the possibility of "merging" various actions [15]. In the case of process planning, each feature represents a separate goal, and merging corresponds to saving set-up or tool-change costs by performing two operations at the same time (such as the two twist-drilling actions mentioned above). In such problems, finding an overall plan to achieve all of the goals consists of selecting from among alternate plans for each of the goals and then merging certain of the actions.

As one might expect, the problem of finding an optimal overall plan is NP-hard, but it is possible to develop efficient approximation algorithms for this problem (i.e., algorithms which will produce results that are close to optimal, with reasonably fast average-case performance) [15]. We are developing such algorithms, and intend to develop them further. This will provide a way to produce process plans that take feature interactions into account.

5. Summary and Conclusions

This paper describes our work on the integration of techniques for design, geometric reasoning, and multi-goal planning, with application to computer-aided design and manufacturing. Our work focuses on the following tasks:

1. Knowledge representation and reasoning techniques for process planning. We believe that the rule-based approach normally used in knowledge-based systems is not the best approach to use in process planning. Instead, we have developed an approach based hierarchical abstraction, and implemented it in the SIPS process planning system.

2. Algorithms and data structures for solid modeling. We feel that existing solid modelers are inadequate for the kinds of interactions required for thorough integration with automated process planning systems, and we are addressing this issue by developing a new approach to solid modeling which we believe will satisfy the necessary requirements.

3. Ways to extract and reason about features and feature interactions. We believe that if a design-by-features system is to be made convenient to the designer, it is unrealistic to force the designer to design using manufacturing features. Thus, it will still be necessary to extract the manufacturing features from the model produced by the designer. However, we also believe that this task can be made less complicated than the task of extracting manufacturing features from an ordinary (non-feature-based) geometric model. We are developing techniques to handle this problem.

4. Ways to reason about feature interactions and their effects on the resulting plans. We have been developing fast algorithms to handle optimization in multi-goal planning problems, and intend to use these algorithms to handle feature interactions in process planning.

This work is being done with two long-term goals in mind: the development of a practical integrated system for designing metal parts and planning their manufacture, and the investigation of fundamental issues in representing and reasoning about three-dimensional objects. We believe this work will have utility not only for automated manufacturing, but also for other problems in design and multi-goal planning.

References

[1] P. Brown and S. Ray, "Research Issues in Process Planning at the National Bureau of Standards," *Proc. 19th CIRP International Seminar on Manufacturing Systems*, June 1987, pp. 111-119.

[2] C. Hayes, "Using Goal Interactions to Guide Planning," *Proc. AAAI-87*, 1987, 224-228.

[3] M. Henderson, "Extraction of Feature Information from Three Dimensional CAD Data," Ph.D. Dissertation, Purdue University, May 1984.

[4] N. Ide, *Integration of Process Planning and Solid Modeling through Design by Features*, Master's thesis, Computer Science Department, University of Maryland, College Park, 1987.

[5] L. Kyprianou, "Shape Classification in Computer-Aided Design," Ph.D. Dissertation, Cambridge University, July 1980.

[6] D. Nau and T. Chang, "A Knowledge-Based Approach to Generative Process Planning," *Production Engineering Conference at ASME Winter Annual Meeting*, Miami Beach, Nov. 1985, 65-71.

[7] D. Nau and T. Chang, "Hierarchical Representation of Problem-Solving Knowledge in a Frame-Based Process Planning System," *Jour. Intelligent Systems* 1:1, 1986, pp. 29-44.

[8] D. Nau and M. Gray, "SIPS: An Application of Hierarchical Knowledge Clustering to Process Planning," *Symposium on Integrated and Intelligent Manufacturing at ASME Winter Annual Meeting*, Anaheim, CA, Dec. 1986, pp. 219-225.

[9] D. Nau and M. Gray, "Hierarchical Knowledge Clustering: A Way to Represent and Use Problem-Solving Knowledge," *in* J. Hendler, *Expert Systems: The User Interface*, Ablex, 1987, 81-98.

[10] D. Nau and M. Luce, "Knowledge Representation and Reasoning Techniques for Process Planning: Extending SIPS to do Tool Selection," *Proc. 19th CIRP International Seminar on Manufacturing Systems*, June 1987, pp. 91-98.

[11] D. Nau, "Hierarchical Abstraction for Process Planning" *Second Internat. Conf. Applications of Artificial Intelligence in Engineering*, 1987.

[12] D. Nau, "Automated Process Planning Using Hierarchical Abstraction," Award winner, Texas Instruments 1987 Call for Papers on Industrial Automation, *Texas Instruments Technical Journal*, Winter 1987, 39-46.

[13] G. Vanecek and D. Nau, "Computing Geometric Boolean Operations by Input Directed Decomposition," Tech. Report, 1987.

[14] G. Vanecek, Jr. and D. Nau, "Non-Regular Decomposition: An Efficient Approach for Solving the Polygon Intersection Problem," *Symposium on Integrated and Intelligent Manufacturing at ASME Winter Annual Meeting*, 1987.

[15] Q. Yang, D. Nau, and J. Hendler, "Planning for Multiple Goals with Limited Interactions," submitted for publication, 1988.

Mechanism Design