

Finding Near-Optimal Build Orientations for Shape Deposition Manufacturing

Satyandra K. Gupta (skgupta@eng.umd.edu)
University of Maryland, College Park, USA

Qi Tian and Lee Weiss
Carnegie Mellon University, Pittsburgh, USA

Key words: CAD/CAM, NC machining, and layered manufacturing.

Abstract: This paper discusses a method for finding near-optimal build orientations for shape deposition manufacturing (SDM). SDM is a solid freeform fabrication process that allows the creation of complex shaped parts by the iterative application of numerically controlled material deposition and milling operations. The part build time in SDM process depends on the number of layers. And the number of layers depends on the build orientation. Therefore, finding build orientation is a crucial step in the process planning of SDM parts.

1. INTRODUCTION

Shape deposition manufacturing (SDM) is a solid freeform fabrication (SFF) process that allows the creation of complex shaped parts by the iterative application of numerically controlled material deposition and milling operations [5, 8, 14]. Parts are built up in layers of both part and sacrificial support materials. The basic concept of SDM is shown in Figure 1. In the SDM process, parts are decomposed into a set of layers and compacts that can be built in a single step. The compacts are deposited as near-net shapes that are then accurately machined to net shapes before depositing additional material. Use of support material eliminates need for fixturing. SDM is capable of producing parts with multi-material, functionally-graded material or embedded pre-fabricated components.

Furthermore, SDM can handle very large layer thickness and parts can be decomposed into true three-dimensional layers.

In other SFF processes, parts need to be decomposed to meet layer thickness and surface finish requirements. In SDM process, parts (and support material) are decomposed primarily to eliminate accessibility problem in machining. Geometric conditions that require part decomposition are described in detail in Section 2. Figure 2 shows an example of an SDM operation sequence.

The part build time in SDM and other SFF processes strongly depends on the number of layers. And the number of layers depends on the build orientation. Figure 3 shows how different build orientations can lead to different numbers of compacts and layers in SDM process. Usually, finding a suitable build orientation is the first step in the process planning of solid freeform fabrication processes. A number of optimization techniques have been developed to find optimal or near optimal build orientation for SFF processes [1, 4, 12, 13]. Because of significant differences between SDM and other SFF processes, existing techniques for identifying build orientation do not work for SDM.

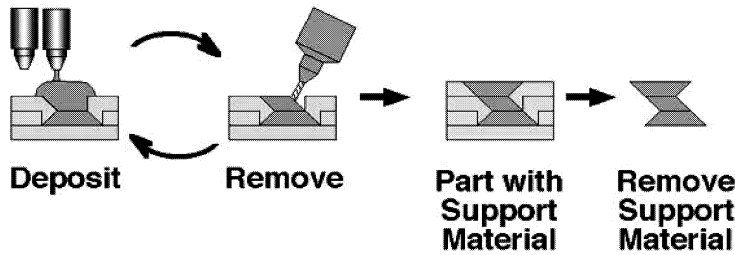


Figure 1. Shape Deposition Manufacturing

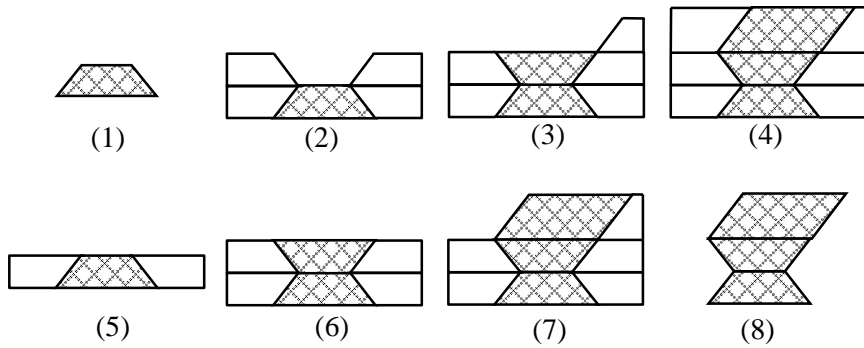


Figure 2. An Example Operation Sequence for SDM Process

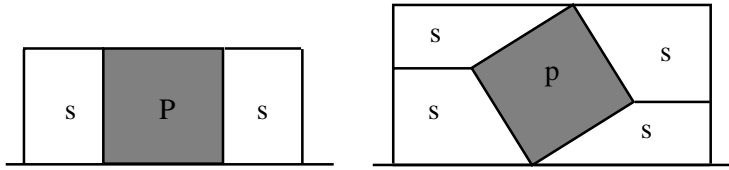


Figure 3. Effect of Build Direction on the Number of Compacts (P denotes part and S denotes support material)

In this paper, we present an algorithm for finding a near-optimal build orientation for SDM. Rest of the paper has been organized as follows. Section 2 presents background material and basic definitions. Section 3 presents our technical approach in detail. Finally, Section 4 presents concluding remarks and direction for the future work.

2. BACKGROUND

Face and Edge Classification: As shown in Figure 4, faces of the part are classified into three categories. Let us assume that V is the normal vector of the face and Z is the candidate orientation direction, then

1. F is an undercut face if $V.Z < 0$.
2. F is a non-undercut face if $V.Z > 0$.
3. F is a boundary face if $V.Z = 0$.

As shown in Figure 5, edges of the part are classified into four categories according to the categories of its two connected faces. If $F1$ and $F2$ are the two connected faces of edge E , then

1. E is a *transition* edge if $F1$ and $F2$ are of opposite attributes, i.e., if $F1$ is an undercut face and $F2$ is a non-undercut one or vice versa.
2. E is a *non-transition* edge if $F1$ and $F2$ are of the same attributes, i.e., if both $F1$ and $F2$ are undercuts or non-undercuts.
3. E is a *boundary-undercut* edge if $F1$ is a boundary face and $F2$ is an undercut face or vice versa.
4. E is a *boundary-non-undercut* edge if $F1$ is a boundary face and $F2$ is a non-undercut face or vice versa.

A *transition loop* is the loop formed by connecting the transition edges into an open or closed loop. A *transition loop* can consist of one edge.

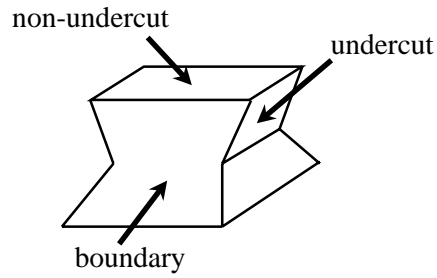


Figure 4. Face Classification

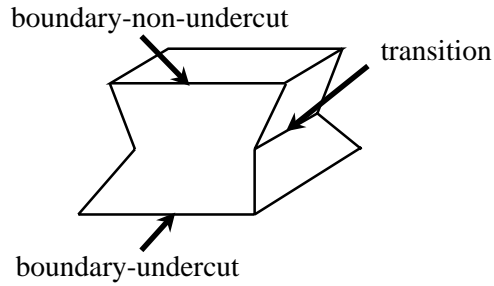


Figure 5. Edge Classification

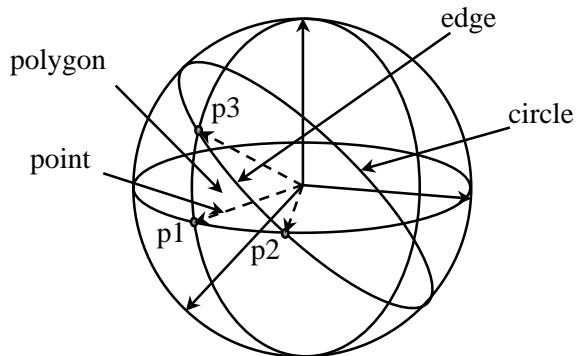


Figure 6. Spherical Geometry Basics

Review of Definitions from Spherical Geometry: Our algorithms are based on earlier work by Woo *et al.* in the area of spherical geometry [2, 16]. Few basic definitions from spherical geometry area are described below (please refer to Figure 6 for graphical interpretation).

1. A *point* p is a unit vector which is denoted by a tuple $(x1, x2, x3)$.
2. A *circle* is determined by the intersection of the unit sphere with a plane, if the plane contains the center of the sphere, it is a *great*

circle. Great circles are denoted by unit vectors corresponding to normal vectors of planes.

3. A *spherical edge* is formed by two points on a great circle.
4. A *spherical polygon* P is a closed contour on sphere of ordered connected edges which don't intersect with each other.

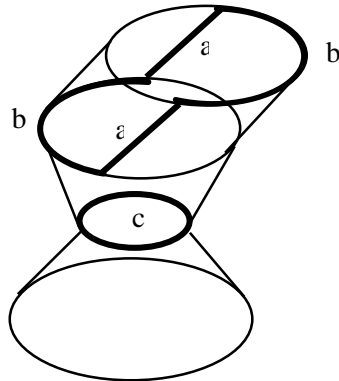


Figure 7. Silhouette Edges

Conditions Requiring Part Decomposition: Parts need to be partitioned into layers in order to eliminate accessibility problems in machining. As described in Ramaswami *et al.*'s work [10, 11], to eliminate accessibility problems, various compacts (basic building blocks of SDM process) need to satisfy the following conditions:

Condition 1: No concave silhouette loops (edges) are allowed in a compact.

Condition 2: No self-intersections of the projected silhouette loop of the compact are allowed.

Condition 3: No cyclic ordering of the silhouette loops of the compacts are allowed.

Whenever a compact does not satisfy any of the above three conditions, it needs to be decomposed. Each decomposition leads to extra layers. Below, we have reproduced definitions of silhouette edges and loops from Ramaswami *et al.* [10, 11]:

Silhouette edges are those curves on the surface of the model along which $N(u,v) \cdot Z = 0$. Silhouette edges serve as the boundary between undercut and non-undercut portions of the surface. As shown in Figure 7, there are three kinds of *Silhouette edges*:

a: convex surface silhouette edge

b: convex boundary silhouette edge
c: concave boundary silhouette edge

Silhouette loops are the loops formed by connecting the silhouette edges into a closed contour. A silhouette loop can consist of both concave and convex silhouette edges.

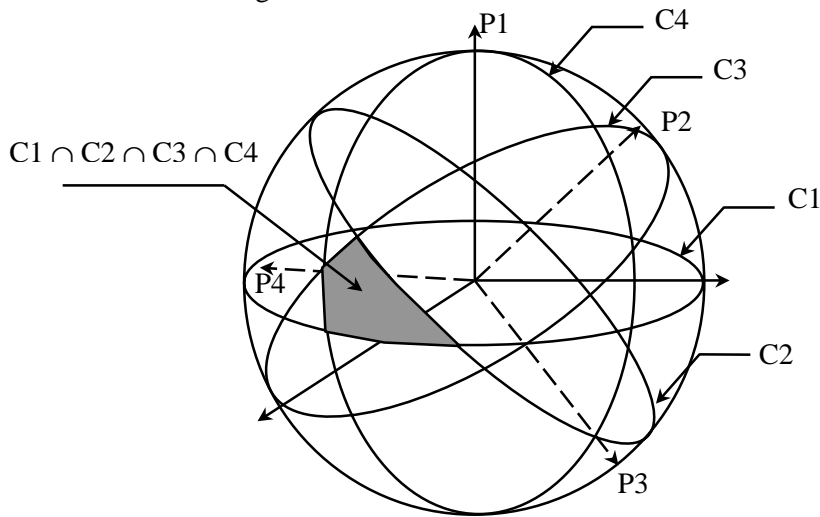


Figure 8. Spherical Polygon Formation

3. TECHNICAL APPROACH

The build time of SDM process is mainly determined by the number of layers and compacts to be built. The number of layers and compacts depends on the build orientation for the SDM process. For other SFF processes, build direction also has a great impact on surface finish. In SDM process, every surface is machined. Therefore, we only need to consider the effect of build orientation on the build time.

Currently we only consider part decomposition because of Condition 1 (described in Section 2). In order to find the optimal build orientation direction, we need to define a cost function that estimates the goodness of a build orientation. We have chosen a cost function that closely relates to the build time. Our cost function is based on the estimates of the number of layers needed to build the part in a given orientation direction. In general, the build time depends upon the actual number and shapes of compacts. Therefore, our cost function is an approximation of the build time. Usually horizontal partitioning is considered better than inclined partitioning.

Therefore, each non-horizontal partition is considered more expensive than the horizontal partition. We reflect this in our cost function by taking *weighted summation* of layers, where non-horizontal layers are multiplied by a factor greater than one.

Due to accessibility problems in machining whenever we encounter a concave transition edge, we need to partition the part. Under certain special conditions, boundary edges can also act as transition edges. On the other hand, convex transition edges require partitioning of the support material. We connect various transition edges into planar loops and we base our estimates of number of layers on the number of transition loops. Non-horizontal transition loops contribute more than the horizontal loops to the cost function.

The idea behind our algorithm to find a near-optimal build orientation is as follows. An edge in the part may be a transition edge or a non-transition edge based on the build direction. We currently take faceted part as input, the facetting accuracy can be adjusted by changing the precision of the facetting parameters. We consider a unit sphere that represents all possible orientation direction. The basic idea is to split the boundary of the unit sphere into three regions: (1) transition region where the edge is a transition edge; (2) boundary region, where the edge is a boundary edge; and (3) non-transition region, where the edge is a non-transition edge. In practice, we accomplish this by first characterizing the boundary of unit sphere for each face. Each planar face is an undercut face on one hemisphere and a non-undercut on the other hemisphere. These hemispherical regions are defined by the direction normal of various faces. Based on the classification of the connecting faces of an edge, various regions for the edge can be computed by performing simple boolean operations.

Let us consider the set of spherical polygons formed by taking the intersection of all great circles corresponding to various direction normals in the part. We can classify each edge as a transition or a non-transition edge within these spherical polygons. Once every edge has been classified within spherical polygons, we can form transition loops for each spherical polygon. The number of transition loops for a spherical polygon gives us an estimate of the number of required layers for a direction selected from that spherical polygon. There are only polynomial numbers of spherical polygons, so selecting build orientation can be simply accomplished by examining various polygons. Within a spherical polygon, we select a direction such that the maximum transition loops are horizontal. Main steps of our build orientation algorithm are described below:

Step 1. Generate spherical polygons for the faceted part.

Step 2. Classify various part edges on spherical polygons based on two faces forming the edge.

Step 3. Find the best build orientation for the part by finding best build orientations for various polygons. To find best orientation for various polygons, we do the following:

1. We connect all the transition edges into separate transition loops. Each single transition edge is also treated as a transition loop and is stored in a transition loop list.
2. After forming all the transition loops, we compute the value of the cost function for all the polygon vertices and some inner points where the normal vector of the transition loops are within the polygon.

Generating Spherical Polygons: We have taken a direct approach to intersect the spherical circles and form spherical polygons, our approach for generating spherical polygons consists of the following main steps:

Step 1. Collect face normal set F from the input faceted part. Each face normal F_i should be unique in the set.

Step 2. Form a set of great circles C from the face normal set F .

Step 3. Intersect all the great circles in pair-wise manner. Save all the intersections as a list for each great circle c_i .

Step 4. Sort the intersection list for each great circle c_i . Sorting is done by first projecting all the intersections on a great circle onto the x-y plane, then comparing their relative angle with the X-axis.

Step 5. Form spherical polygons in the following manner. Start from an unaccessed edge. Try to form a polygon in a counterclockwise manner. Repeat until all the edges on the great circles are accessed twice which means no polygons are left.

Figure 8 shows four great circles and a spherical polygon formed by taking intersection of these great circles.

Classifying Edges on Spherical Polygon: When all the spherical polygons are formed, we can proceed to classify various part edges on spherical polygons. The goal of this step is to record the attributes of all the edges with respect to each polygon, i.e., which edge is transition, non-transition or boundary edges on which spherical polygon.

We can uniquely classify various edges for spherical polygon because of the following reason. Spherical polygons are formed by taking intersection

of all possible face normals in the part. Therefore, edge characteristics do not change within a spherical polygon (i.e, for all possible directions within this spherical polygon, edges will either be transition or non-transition). This allows us to treat all directions within a spherical polygon uniformly.

Finding Near-Optimal Build Orientation: Once we have generated spherical polygons from face normals, and classified various edges on spherical polygons, we proceed with determining build orientation. In general a part can be decomposed using non-planar sections. But, currently we only deal with planar transition loops, i.e., all the edges of a transition loop reside on the same plane.

Using our cost function, we can compute the cost of various candidate orientations. Therefore, by comparing cost of two candidate orientations, we can determine which one is better. We observe that, generally speaking, vertices on a spherical polygon are better candidates than the interior points of spherical polygon for minimizing the cost function. There are, however, exceptions to this observation: interior points along which transition loops are horizontal lead to horizontal partitioning and therefore reduced cost. By applying this criteria to various candidate points within a spherical polygon, we are able to find an optimal orientation for the polygon. So when we have done these comparisons for various spherical polygons, we can find an optimal orientation as per our cost function.

The main steps for finding an optimal build orientation for a given spherical polygon P are as follows:

Step 1. Find the transition edge sets E for the polygon P .

Step 2. Form transition loop set L from the transition edge set E .

Step 3. Form the set of candidate points C_p by including all vertices of P and adding inner points along which a transition loops is horizontal.

Step 4. Compute costs for various candidate points and select the best orientation O_b for set C_p .

Implementation and Examples: This algorithm has been implemented using C++ programming language. For geometric modeling and transformation we have used ACIS geometric kernel [3]. For basic data structures we have used LEDA (a Library of Efficient Data types and Algorithms from Max Planck Institute) [7]. For graphical interface we have used Open Inventor graphics library (available from Silicon Graphics, Inc) [15].

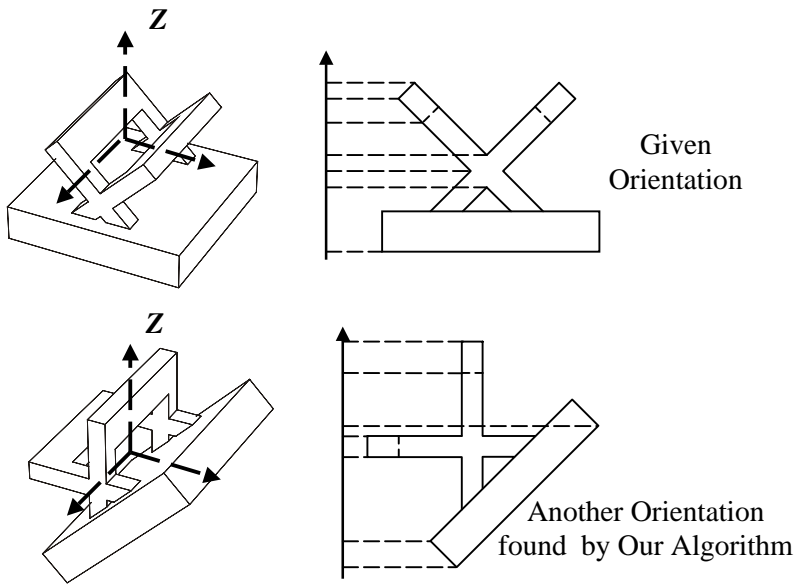


Figure 9. Example A

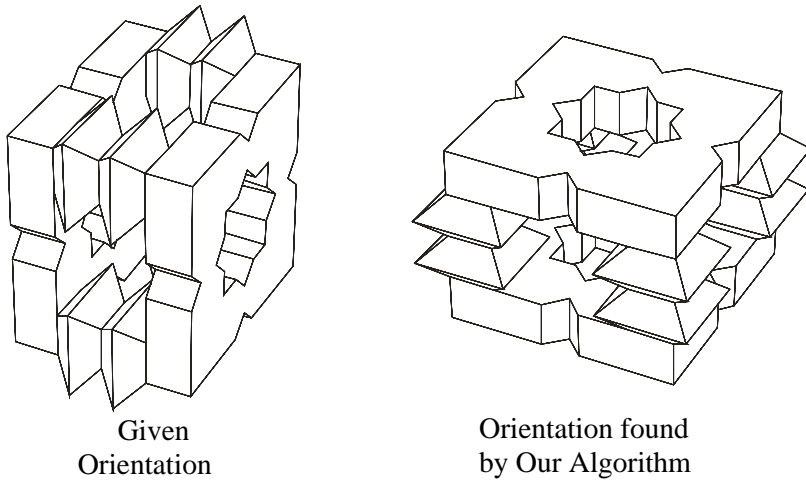


Figure 10. Example B

Figures 9 and 10 show two examples. For the first example, we identified two candidate build orientations. One is the same as the given part, the other is in the direction that has been generated by rotating the given part about X -axis by -45 degree. For the second example, the build direction has been generated by rotating this part about Y -axis by -90 degree.

4. CONCLUSIONS

Summary: In this paper, we have described an algorithm for finding near optimal build orientation for Shaped Deposition Manufacturing. The advantages of our algorithm compared with other algorithms for finding orientations are as follows. First, our algorithm is robust and complete. By robust we mean our algorithm is not sensitive to any initial values set by the users. Our algorithm does not need any pre-specified values to start as common in most other SFF orientation algorithms. By complete we mean our algorithm is able to search all the candidate orientations so that near-optimal or optimal orientation can be found. Another feature of our algorithm is that it handles the build orientation problem in a divide-and-conquer manner. We partition the unit sphere (representing all the candidate directions) into smaller spherical polygons, identify the best orientation within every spherical polygon, and assemble solutions from various polygons to find the final build orientation.

Current Limitations and Future Plans: Our approach currently has the following limitations.

1. Only parts with planar surfaces are considered.
2. Part decomposition based on (1) self-intersections of the projected silhouette loop, and (2) cyclic ordering of the silhouette loops are not considered.

We are working towards relaxing these limitations. We are also developing heuristics for estimating upper bound and lower bound on the cost estimates for a given spherical polygon. This will allow us to develop branch and bound search techniques to quickly find the near-optimal orientation direction without having to examine all spherical polygons.

Acknowledgments: This research has been supported by a subcontract from Stanford University on an NSF Grant MIP-9617994.

REFERENCES

1. Seth Allen and Deba. Dutta. On the computation of part orientation using support structures in layered manufacturing. In H.L. Marcus et al., editor, Solid Freeform Fabrication Symposium 1994, pages 259--269, University of Texas, Austin, 1994.
2. Lin Chen, Shuo-Yan Chou, and Tony C. Woo. Separating and intersecting spherical polygons: Computing machinability on three-, four-, and five-axis numerically controlled machines. *ACM Transactions on Graphics*, 12(4):305--326, October 1993.
3. Jonathan Corney. *ACIS by Example*. Edinburgh, 1994.
4. Dietmar Frank and Georges. Fadel. Preferred direction of build for rapid prototyping processes. In A.J. Lightman R.P. Chartoff and J.A. Schenk, editors, *Proceedings of the Fifth International Conference on Rapid Prototyping*, pages 191--200, University of Dayton, June 1994.
5. J.W. Kietzman, A.G. Cooper, L.E. Weiss, L.Schultz, J.L. Lombardi, and F.B. Prinz. Layered manufacturing material issues for sdm of polymers and ceramics. In *Proceedings of the Solid Freeform Fabrication Symposium*, Austin, Texas, August 1997. University of Texas at Austin.
6. A. Marsan and D. Dutta. A survey of process planning techniques for layered manufacturing. In *Proceedings of the 1997 ASME Design Technical Conferences*, Sacramento, CA, September 1997.
7. Max-Planck Institute of Information, 66123 Saarbrucken, Germany. *The LEDA User Manual*, 1994. Release 3.4.1.
8. R. Merz, F.B. Prinz, K. Ramaswami, M. Terk, and L. Weiss. Shape deposition manufacturing. In *Proceedings of the Solid Freeform Fabrication Symposium*, pages 1--8, Austin, Texas, August 8-10 1994. University of Texas at Austin.
9. Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.
10. K. Ramaswami. *Process Planning for Shape Deposition Manufacturing*. PhD thesis, Stanford University, Stanford, CA, USA, 1997.
11. K. Ramaswami, Y. Yamaguchi, and F.B. Prinz. Spatial partitioning of solids for solid freeform fabrication. In *proceedings of the Fourth ACM/SIGGRAPH Symposium on Solid Modeling and Applications*. ACM/SIGGRAPH, May 1997.
12. P. N. Sreeram and Debasish Dutta. Determination of optimal orientation based on variable slicing thickness in layered manufacturing. In *Proceedings of the 1995 ASME Winter Annual Conference*, San Francisco, CA, November 1995.
13. David C. Thompson and Richard . Crawford. Optimizing part quality with orientation. In H. L. Marcus et al., editor, *Solid Freeform Fabrication Symposium 1995*, University of Texas, Austin, August 1995.
14. L.E. Weiss, R. Merz, F.B. Prinz, G. Neplotnik, P. Padmanabhan, L. Schultz, K. Ramaswami. Shape Deposition Manufacturing of Heterogeneous Structures. *SME Journal of Manufacturing Systems*, 16(4):239--248, 1997.
15. Josie Wernecke. *The Inventor Mentor*. Addison-Wesley, 1994.
16. Tony C. Woo. Computational geometry on the sphere with application to automated machining. *Journal of Mechanical Design*, 114:288--295, June 1992.