55 MANUFACTURING PROCESSES

Ravi Janardan and Tony C. Woo

INTRODUCTION

This chapter surveys some recent work on the application of techniques from computational geometry to geometric problems arising in manufacturing processes such as layered manufacturing, mold design, and numerically controlled machining. Within each topic, we discuss problems that have benefited from the application of geometric techniques, and mention several other problems where such techniques could be used to advantage.

55.1 LAYERED MANUFACTURING

Layered Manufacturing (LM) is a relatively new technology which allows physical prototypes of 3D models to be built directly from their digital representations, using a "3D printer" attached to a computer [Jac92]. LM provides the designer with an additional level of physical verification and facilitates the early detection and correction of design flaws that may have gone unnoticed otherwise. The use of LM has proliferated into a wide variety of areas, including, among others, engineering (e.g., automotive and aerospace design), ergonomic product design (e.g., hand-held devices such as cell phones), medicine (e.g., prosthetics design and tissue engineering), and art (e.g., sculpture) [Cad02, Har01, KF97, Lev02, NLG02].

The basic principle underlying LM is simple: The digital model is oriented suitably and sliced into horizontal layers by a plane. The layers are transmitted over a network to a fabrication device, which "prints" them successively in the vertical direction, each layer on top of the previous one; thus the physical prototype is realized as a vertical stack of two-dimensional layers. The efficiency and accuracy of LM depends, in part, on the efficient solution of a number of geometric problems. For instance, the choice of the model's orientation determines the number of layers, the surface finish, and the quantity and location of temporary support structures. The problem of printing the layers efficiently reduces to covering the interior of a polygon with a collection of thin rectangles. Other problems include slicing the model efficiently and generating a compact representation of the support structures.

GLOSSARY

STL format: The model is assumed to be given as a surface triangulation. The format specifies the triangles by listing the coordinates of their vertices and the direction cosines of their unit outer normals. This is the *de facto* industry standard for LM; the name is derived from STereoLithography, one of the first

LM processes to be developed.

Model orientation: The rotation of the model from its default orientation in the STL file, prior to being sliced into horizontal layers and built in the vertical direction.

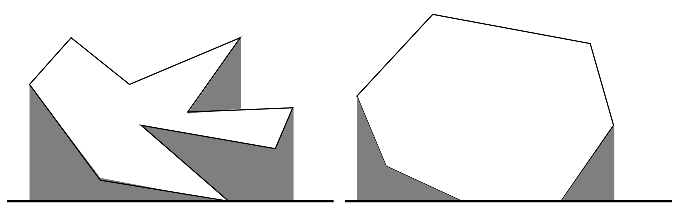
Stairstep error: Stairstep-shaped artifacts introduced on the model's facets due to discretization into layers (similar to antialiasing in computer graphics), which affect surface finish and accuracy. The stairstep error on a facet is the height of the stairstep perpendicular to the facet. It is a function of the (smaller) angle between the vertical direction and the facet's outer normal, hence of the model's orientation.

Supports: Temporary structures that are built simultaneously with the model to prop up layers that overhang previously-built layers; these are removed in a postprocessing step. Formally, for a model P, a point $p \in \mathbb{R}^3 \setminus \mathcal{P}$ is part of the supports if the upward-directed ray from p intersects $\mathcal{P}$; thus the membership of p in supports depends on the model's orientation. The supports form a collection of disjoint polyhedra. (Figure 55.1.1.)

Support requirements: Measured in two ways: The support volume is the total volume of the support polyhedra. The support contact-area is the area of that portion of the model's boundary that is in contact with supports. These should be minimized to reduce material costs, build time, and postprocessing time.

Hatching: The process of printing each layer (a polygon) by covering its interior with parallel rectangles of some small width; the width is a process parameter.

FIGURE 55.1.1
Support structures (shown shaded) for a nonconvex polygon (left) and a convex polygon (right). Illustration is in two dimensions for convenience.

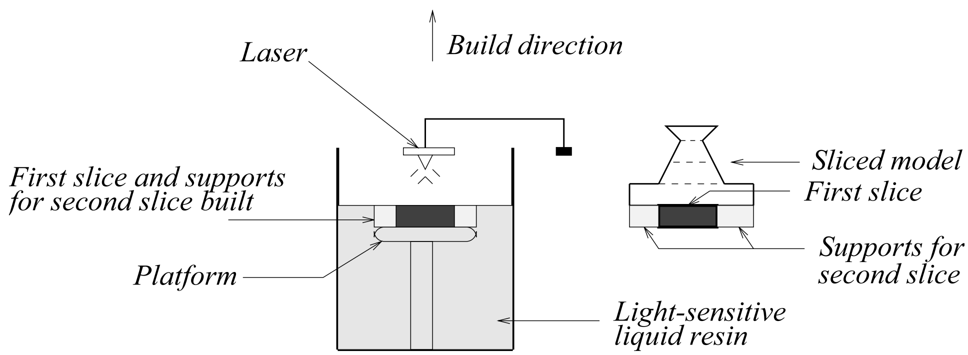


RESULTS

Figure 55.1.2 illustrates schematically a widely-used LM process called Stereolithography, where the printing of layers is achieved by having a laser trace out and hatch each layer on the surface of a liquid resin which hardens when exposed to light. After a layer is built, the platform is lowered by an amount equal to the layer thickness (on the order of a few thousandths of an inch) and the next layer is then built atop the previous one. The need for supports is ascertained beforehand, by analyzing the orientation and geometry of the model, and then generating and merging a description of the layers into the STL file for the model. Representative examples of other LM processes include Fused Deposition Modeling (where layers are printed by extruding and laying down molten plastic via a nozzle), Laminated Object Manufacturing (where the layers are cut out from sheets of adhesive-backed paper), and

3D Printing (where the layers are realized by outlining their shape via a binder fluid and then depositing a special powder onto it).

FIGURE 55.1.2
The Stereolithography process.



Geometric problems in LM can be grouped loosely into three categories:

Choice of model orientation. Here the goal is to choose an orientation for the model that optimizes some design criterion (or to simply decide if the criterion can be satisfied). In [ABB$^+$97], $O(n)$-time algorithms are given for deciding whether an n-vertex polyhedron can be built without supports using two models of Stereolithography—one in which no layer can overhang the previous one, and another where some overhang, controlled by an angle parameter, is allowed. The classes of objects that can be built by these processes are also related to those buildable via NC-machining and casting. In [MJSG99], an algorithm is given to minimize the maximum stairstep error ([BB95]) over all facets of a polyhedron in $O(n \log n)$ time and to minimize the sum of the stairstep errors on all facets in $O(n^2)$ time; the first algorithm even allows facets to be weighted to indicate their relative importance with respect to surface finish. Also given are $O(n^2)$-time algorithms to minimize the volume and (independently) the contact-area of supports for a convex polyhedron. In [MJSS01], the preceding results are combined to reconcile simultaneously multiple design criteria, including support volume, contact-area, stairstep error, and number of layers. (Minimizing the number of layers is equivalent to finding the width of a polyhedron, and efficient solutions are known for this [HT88, SSMJ99].) Three formulations for reconciling the criteria are considered: optimizing the criteria sequentially, optimizing a weighted combination of the criteria, and allowing the criteria to meet designer-specified thresholds. The methods in [MJSG99, MJSS01] use well-known techniques from computational geometry, such as spherical arrangements, convex hulls, and Voronoi Diagrams, in conjunction with constrained optimization methods such as Lagrangian Multipliers. In [AD00], an approximation algorithm is given for minimizing the contact-area for a convex polyhedron. This method, based on computing approximate levels in a weighted arrangement of lines, runs in $O((n/\epsilon^3) \log^3 n)$ expected time and has an approximation ratio of $1 + \epsilon$, for any $\epsilon > 0$.

Optimization of supports for a nonconvex polyhedron is much harder due to the complicated structure of the supports. As Figure 55.1.1 illustrates, supports need not extend all the way to the platform, but may instead terminate on the model itself. Furthermore, only a fraction of a facet need be in contact with supports, unlike the convex case where either a facet is entirely in contact with supports or not in contact at all. In [MJS$^+$99], algorithms are given for the two-dimensional case, i.e., minimizing support area and contact-length. The approach is based on partitioning the unit-circle of directions into intervals and generating for each interval a formula for the support requirement of interest. The intervals are then scanned in order and the formula for each interval is updated incrementally from that of the previous interval and then optimized within the interval. The running time is $O(n \log n)$ plus the time to perform $O(n)$ minimizations of a certain polynomial of degree $\Theta(n)$. Heuristics for contact-area optimization are described in [AD95], where a subset of the facet normals of the convex hull of the model is used for choosing the orientation. For each orientation, the needed supports and their contact-area are computed approximately, and the best orientation is then output. No analysis of the quality of the approximation is provided. In earlier work, the problem of support optimization is addressed in [FF94], and heuristics are given in the context of an expert system.

Another design consideration in LM is to choose model orientations so that certain functionally-critical surfaces of the prototype (e.g., facets on gear teeth) are not in contact with supports, since the presence and subsequent removal of supports can affect surface finish and accuracy. In [SSJ$^+$00, SSJJ], an $O(n^2)$-time algorithm is are given to compute a description of all model orientations for which a prescribed facet is not in contact with supports. The related optimization problem of computing a description of all orientations for which the total area of facets not in contact with supports is maximized is solved in $O(n^4)$ time. These results are based on convex hulls, arrangements, and overlays of subdivision—all on the unit-sphere.

Fixed-orientation problems. Once an orientation has been chosen, several tasks remain. These include computing a description of any needed supports, slicing the model and supports, and deciding on how best to hatch the layers. In commercial software packages for LM, slicing and support generation are usually done in tandem. Specifically, as the model is sliced, the volume subtended under each layer is subtracted from that subtended by the layer above it; the result is the support between the two layers. Thus the supports are generated as a sequence of thin layers. In [Joh99], an alternative approach is pursued, where the goal is to generate a combinatorial description of the supports, as a collection of disjoint polyhedra. The algorithm is based on cylindrical decomposition [Mul93] and runs in $O(n^2 \log n)$ time.

Slicing algorithms used in LM are inefficient in that they compute from scratch the intersection of each slicing plane with the polyhedron, instead of taking advantage of the coherence that exists from layer to layer. This is due, in part, to the lack of any topological information in the STL format. In [MS99], an efficient and robust slicing algorithm is proposed. The algorithm builds a data structure based on a generalization of the well-known winged-edge structure [Bau75] and then uses the plane sweep paradigm to compute and update the layers incrementally, by taking advantage of the topological similarity between closely-spaced layers. A different perspective on slicing is taken in [DM94, KD96], where the focus is on slicing a model adaptively, with slices of variable thickness, so as to improve surface accuracy and to speed up the build time.

The hatching problem may be viewed as the two-dimensional analog of the model orientation problem. Here the goal is to find a common orientation of all the layer polygons (or, equivalently, a rotation of the model about the vertical axis) so that the total number of times the hatching tool (e.g., the laser in Stereolithography) meets the boundaries of all the polygons is minimized. This, in turn, minimizes the number of starts, stops, and direction changes of the tool and increases tool life. In [HJSS03], the problem is approximated as one of finding a direction in the plane that minimizes the sum of the lengths of the projections of all polygon edges in this direction. The latter problem is reduced to computing the width of a suitably defined convex polygon (see also [Sar99]). The overall running time is $O(n' \log n')$, where n' is the total number of number of polygon edges in all layers. On real-world STL models, the algorithm runs very fast and delivers solutions that are very close to the solution produced by an optimal, but much slower, algorithm [SSHJ02].

Decomposition problems. LM processes generally view the model as a single, monolithic unit. An alternative approach is to decompose the model into a small number of pieces, build the individual pieces, and then glue them back together. This allows large models to be built in parallel on multiple machines (or even simultaneously on the same machine) and also reduces the build time. Moreover, the support requirements of the decomposed parts is usually lower than that of the original. This decomposition-based approach is pursued in [IJM$^+$02], where it is shown how to decompose, with a plane, a convex or nonconvex polyhedron in a given orientation into a user-specified number of pieces so that the support volume or contact-area is minimized. The algorithms run in $O(n \log n)$ and $O(n^2 \log n)$ time for convex and nonconvex models, respectively, and are based on cylindrical decomposition and space sweep. In related work [FM01], it has been shown that the problem of deciding whether a polyhedron of genus zero or a polygon with holes can be decomposed into k terrains (hence built with zero supports) is NP-complete; here k is part of the input. In [IJS02], it is shown how to decompose, with a line, a polygon into two smaller polygons such that each is a terrain in the direction normal to the line; the algorithm runs in $O(n \log n)$ or $O(n^2 \log n)$ time, depending on whether or not both terrains have their "base" edge on the dividing line (see also [ABB$^+$97, RR94] for related work).

Besides the problems described above, a (necessarily incomplete) list of other related work on LM includes: automatic repair of STL files [Bøh95, Bar97]; elimination of support structures for a class of models by selectively thickening the walls of the model [AD98]; the design of a complete software front-end for the LM pipeline, from digital model import, to model repair, to batch scheduling of multiple models [BK98]; new modeling techniques for LM based on voxels [CMP95] and on analytic surfaces such as quadrics [FK96]; and investigation of alternatives to the STL format in LM [KD97, DKPS98].

OPEN PROBLEMS

1. Support optimization for nonconvex polyhedra is a challenging problem and an optimal solution remains elusive. Specifically, given a nonconvex polyhedron, $\mathcal{P}$, the goal is to compute an orientation which minimizes the support volume or (independently) the contact-area when $\mathcal{P}$ is built in that orientation. Extending the method in [MJS$^+$99] to three dimensions appears difficult

and expensive, so a new approach may be needed. Also of interest would be simple and efficient approximation algorithms.

2. The decomposition algorithm in [IJM$^+$02] assumes that an orientation is given for the model and then proceeds to find a decomposition which minimizes the support requirements. A natural extension of this is to find an optimal (or near-optimal) decomposition over all possible orientations. Similarly, for the hatching problem, it would be interesting to design an algorithm which computes an optimal or near-optimal hatching direction over all possible orientations of the model.

3. Although the STL format is the *de facto* industry standard for model representation in LM, it is plagued with many problems. It introduces an approximation error when used to represent smooth surfaces, it lacks useful topological information, it is highly redundant and error-prone, and it is very voluminous for surfaces of high curvature. As mentioned earlier, alternatives to STL have been investigated [KD97, DKPS98, CMP95, FK96] recently. In particular, in [FK96] a representation based on quadric surfaces has been considered. It has been shown empirically that for the tasks of slicing and filling in the layers using equidistant offset curves, this analytic representation is superior to STL both in accuracy and in computation time and memory requirements. A natural extension of this work would be to investigate the effect of such representations on other LM tasks such as support generation and minimization, reduction of stairstep error, layer minimization, and hatching.

55.2 MOLD DESIGN

Casting and injection molding processes are used extensively to mass-manufacture a wide variety of products. A key step here is the design of the mold from a digital model of the part, since this affects both the speed of the process and the quality of the finished part. For instance, how the model is decomposed into pieces to make the mold halves determines the number of undercuts in the mold: the greater the number of undercuts, the slower the de-molding process. As another example, the location of venting holes on the mold and the choice of pouring direction determine the extent of air pockets created during mold filling; this ultimately affects the strength and finish of the product.

GLOSSARY

Mold: A cavity in the shape of the part to be manufactured into which molten metal is poured. It consists of two mating parts called ***mold halves***. Once the metal has hardened, the mold halves are pulled apart in opposite directions (i.e., ***de-molded***) and the part is removed.

Undercut: Any point p on a part's surface such that the outward normal at p makes an angle greater than 90° with the de-molding direction for the mold half containing p. Generally, a group of such points forms a recess or projection in the part that prevents easy de-molding.

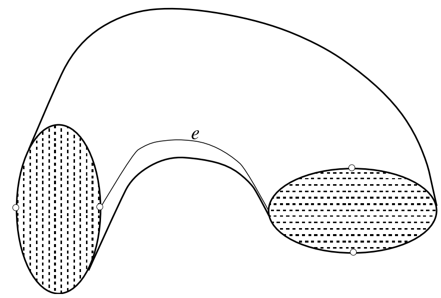


FIGURE 55.2.1
A parting line (e) for the exhaust manifold of an automobile.

Parting line: A continuous closed curve on the surface of the part that defines the two halves; thus it also defines the profile of the contact surface between the two mold halves (Figure 55.2.1).

CH(*P*)*:* The convex hull of a polyhedron P.

Dent: For a polyhedron P, a connected component of $\mathrm{CH}(P) \setminus P$.

Fillability: The ability to fill a mold from a given pouring direction without creating air pockets. This is a function of the mold geometry, the pouring direction, and the location of air-venting holes.

Part decomposition: The process of dividing a part into smaller pieces and making mold halves for these that satisfy certain optimization criteria.

RESULTS

Geometric problems in mold design generally fall into two categories.

Fillability problems. These are concerned with questions such as whether a mold can be filled from a given pouring direction without creating air pockets, and finding a pouring direction that eliminates air pockets using the smallest number of venting holes. In [BvKT98], several results are presented including: (a) deciding in $O(n)$ time whether an n-vertex polyhedron can be filled from a given pouring gate in a given direction without creating air pockets; (b) enumerating in $O(n^2)$ time all pouring directions that permit such a fill; (c) computing in $O(n^2)$ time a pouring direction which minimizes the number of air pockets; and (d) characterizing classes of polyhedra according to their fillability. The two-dimensional counterparts of these problems are solved in [BT95], with running time $O(n)$ for the decision problem and $O(n \log n)$ for the enumeration and optimization problems. Similar questions are also addressed in [FM93] for different mold-filling strategies and different types of filling material (ranging from gas to liquid to solid).

Part decomposition. This refers to the problem of "cutting" the digital model into smaller pieces and making mold halves for these that meet certain optimization criteria. For instance, how can a 3D part P be divided into two such that the parting line is as "flat" as possible? As noted in the mold-design literature, the flatter the parting line, the more cost-effective and accurate the mold. While the notion of flatness has not been quantified in the literature, it is generally taken to mean that the parting line should lie as nearly in a plane as possible. Although a parting line that lies completely in a plane can always be produced by intersecting P with a plane, this can create undercuts, even if P is a convex polyhedron (Figure 55.2.2).

The problem of computing a flattest undercut-free parting line for an n-vertex

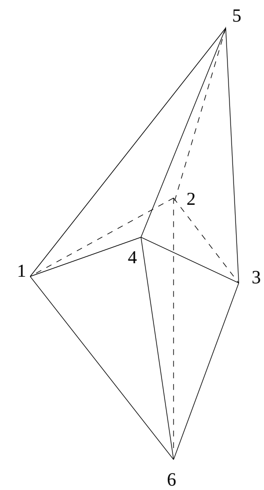


FIGURE 55.2.2
An octahedron that cannot be divided by a plane into two halves without creating undercuts. For example, the plane containing vertices 1, 2, *and* 3 *creates a projection under the chain* 1–4–3. *Undercuts can be avoided by choosing the parting line to be* 1–2–3–4–1 *(or* 2–5–4–6–2*), but this is no longer in a plane. (From* [MGJ99]*, with permission.)*

convex polyhedron P is considered in [MGJ99]. That such a line always exists is clear—simply take the boundary, $L(\vec{d})$, of P, as viewed along lines of sight parallel to any direction $\vec{d}$. The ***flatness***, $\rho(\vec{d})$, of $L(\vec{d})$ is defined in [MGJ99] as the sum of the squares of the projected lengths of the segments of $L(\vec{d})$, where the projection is onto a plane normal to $\vec{d}$, divided by the sum of the squares of the lengths of the segments of $L(\vec{d})$. Thus, $\rho(\vec{d}) \leq 1$, with equality holding if and only if $L(\vec{d})$ lies in a plane. An $O(n^2)$-time algorithm is given to compute a direction $\vec{d}$ that maximizes $\rho(\vec{d})$. The algorithm blends together geometric techniques such as visibility cones, arrangements, and shortest paths in a simple polygon, with methods from continuous optimization. Algorithms are also given for optimizing other measures of flatness. These include (a) finding a direction which maximizes the flatness criterion defined above, but uses segment lengths rather than squared lengths; and (b) finding a direction which minimizes the width of the parting line, where, for any direction $\vec{d}$, the width of the parting line $L(\vec{d})$ is defined to be the smallest separation between two parallel planes normal to $\vec{d}$ that enclose $L(\vec{d})$.

In [BBvK97] the problem of deciding if a given n-vertex polyhedron can be parted by a single plane without creating undercuts is addressed. For an n-vertex nonconvex (resp. convex) polyhedron, where the cast parts are to be removed by translation in mutually-opposing directions, the bounds are $O(n^{3/2+\epsilon})$ time and $O(n^{3/2+\epsilon})$ space (resp. $O(n \log^2 n)$ time and $O(n)$ space), where $\epsilon > 0$ is an arbitrarily small constant. A related result is presented in [AdB$^+$02], where it is shown that, for an n-vertex polyhedron, all directions that admit an undercut-free parting line (for cast removal in mutually opposing directions) can be computed in $O(n^4)$ time. This is shown to be optimal in the worst case by demonstrating a polyhedron which admits $\Omega(n^4)$ such directions. Finally, in [CCW93a], an $O(nd \log d)$-time algorithm is given to compute a pair of opposing directions maximizing the number of visible dents in an n-vertex polyhedron with d dents. This minimizes the number of undercuts; however, the method does not yield a parting line.

Other related work includes decomposition of two-dimensional molds [RR94], identification of criteria other than parting line shape and number of undercuts [RS90], and heuristics for computing a de-molding direction without too many undercuts [HT92].

OPEN PROBLEMS

1. It is unlikely that the $O(n^2)$-time algorithm in [BvKT98] for minimizing the number of air-venting holes can be improved (in view of the 3Sum-Hard-based lower bound). However, can a significantly faster algorithm be devised that approximates the minimum number of air-venting holes to within a constant factor?
2. The goals of maximizing the flatness of the parting line and minimizing the number of undercuts are usually at odds. Often, however, meeting specified thresholds suffices: for instance, given parameters u_0 and ρ_0, design an efficient algorithm to find a parting line with at most u_0 undercuts and flatness at least ρ_0.
3. A polyhedron P is ***1-castable*** if it can be parted by a plane without creating undercuts. The results in [BBvK97] allow one to decide 1-castability efficiently. However, there exist polyhedra that are not 1-castable (Figure 55.2.3). To extend the class of polyhedra that can be cast with planes, call a polyhedron P ***2-castable*** if there is a plane h such that the polyhedra $P \cap h^+$ and $P \cap h^-$ are both 1-castable. (Here h^+ and h^- denote the two halfspaces of h.) Give efficient algorithms to decide 2-castability and characterize the class of 2-castable polyhedra.

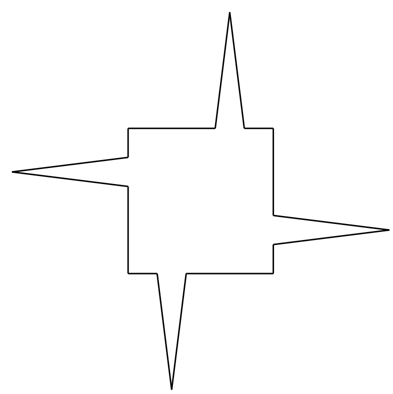


FIGURE 55.2.3
Cross-sectional view of a polyhedron that is not 1-castable. The cross section tapers along the length of the polyhedron to a point and then expands again, so that the polyhedron consists of a "double pyramid." Any casting plane will create an undercut at one (or more) of the spikes or at some of the slanted facets corresponding to the horizontal and vertical segments in the cross section.

55.3 NUMERICALLY CONTROLLED MACHINING

The dominant machining process today is *numerically controlled (NC) machining*, where parts are manufactured under computer control based on information extracted from a digital model. Examples of NC-machines include milling machines and lathes. Typical questions of interest concern accessibility of the tool to the part and generation of toolpaths that satisfy certain optimization criteria.

GLOSSARY

Degrees of freedom (dof): The types of motion permitted of an NC-machine. Specified as a combination of translation and (full or partial) rotation with respect to the coordinate axes.

Visibility map (or VMap): The set of points on the unit sphere representing the directions along which a tool can approach (or "see"; cf. Chapter 28) *all* points on the surface in question without being blocked by other portions of the part. The VMap is a function of the surface geometry and the geometry of the cutting tool, and is in practice usually representable as a (spherical) polygon formed by the intersection of a certain set of hemispheres [GWT94]. For instance, the VMap of a plane is the hemisphere whose pole is the normal to the plane, the VMap of a half-cylinder is a half-great circle, the VMap of a hemisphere is a point, and the VMap of a dent in a polyhedron is the intersection of the set of hemispheres determined by the normals to the dent's faces.

Pocket: A region bounded by one or more closed curves, which delineates the area on the part from which material must be removed.

Spherical band of width b: The set of all points on the unit sphere that are at a distance of at most b on either side of a great circle, where the distance is measured along a great circle arc.

Part setup: The process of dismounting a part, and re-calibrating and re-mounting it in a new orientation on the worktable of an NC-machine.

Direction-parallel pocket machining: A machining discipline where the tool is constrained to stay within a pocket and, moreover, always moves from left to right with respect to a chosen reference line.

Zigzag pocket machining: Similar to direction-parallel machining, except that the tool moves from left to right, then right to left, and so on.

Contour-parallel pocket machining: The tool is constrained to move along a sequence of closed paths that are parallel to the pocket's contour.

RESULTS

Two important parameters of an NC-machine are the dof of the machine and the type of cutting tool. The dof include translation along the principal coordinate directions (***3-axis machine***), plus rotation of the worktable about one axis (***4-axis machine***), plus partial swivel of the tool about a second axis (***5-axis machine***). The dof determines global motion of the tool. For example, in a 4-axis machine, the directions in which the tool can move can be represented on the unit sphere as a great circle whose normal is the rotational axis of the worktable. In a 5-axis machine, if the tool can swivel by $\pm b/2$ radians, then the tool motion directions are given by a spherical band of width b, where the great circle associated with the band is as in the 4-axis case. Cutters are classified, according to the maximum angle θ that they can tilt from the local surface normal, as: ***flat-end*** ($\theta = 0$ radians), ***fillet-end*** ($\theta < \pi/2$ radians), and ***ball-end*** ($\theta = \pi/2$ radians). Thus the cutter geometry determines local motion of the tool: a flat-end cutter can approach a point p on a surface only along the surface normal at p, while a ball-end cutter can approach p along any direction lying within the hemisphere with pole p.

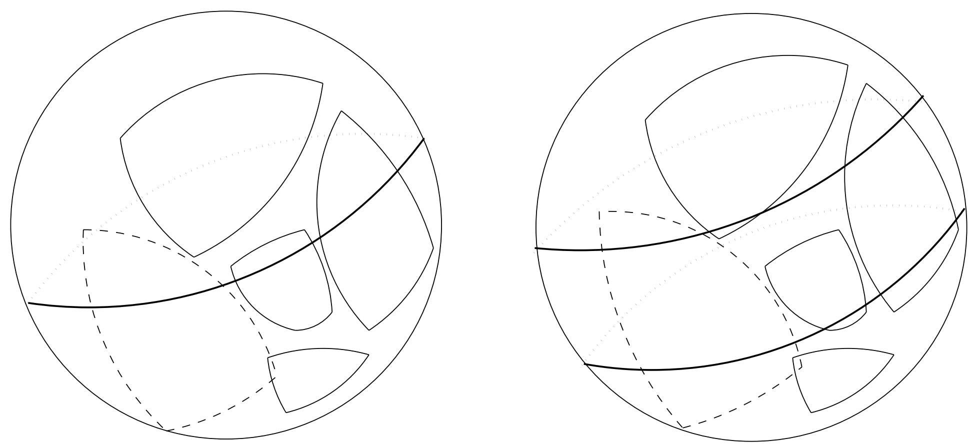
Part orientation. In order to machine a surface on a part, the tool must be able to approach (or see) every point on the surface without being blocked by other portions of the part. For a given orientation of the part on the machine's worktable, only a subset of the surfaces that need to be machined might be so visible to the tool. Therefore, after each such set of visible surfaces has been machined, a part setup is performed to bring a new set of surfaces into view. However, part setup can be quite time-consuming in relation to the actual machining time (hours versus minutes, sometimes). This motivates the following problem. Given the part geometry and the machine parameters, compute a sequence of part orientations that minimizes the number of setups. Unfortunately, this problem is NP-hard, and so attention has focused on obtaining efficient algorithms that approximate closely the minimum number of setups.

A natural approach is a greedy heuristic which finds repeatedly a part orientation that allows access to the maximum number of as-yet-unmachined surfaces [CCW93b, GJM$^+$96]. Suppose, for example, that a 4-axis machine equipped with a ball-end cutter is used. Assume further that the VMaps for the part's surfaces are available; for a ball-end cutter, the VMaps are intersections of certain hemispheres and can be computed as described in [GWT94]. Recall that each VMap represents the directions along which every point on the corresponding surface can be seen by the tool. Therefore, to find an orientation in which the maximum number of surfaces can be seen is equivalent to finding a great circle, C, that intersects the maximum number of VMaps. (Here, C represents the directions in which the tool can move in a 4-axis machine.) Similarly, for a 5-axis (resp. 3-axis) machine, the problem is to find a spherical band B of width b (resp. a point P) that intersects the maximum number of VMaps (Figure 55.3.1).

Given m VMaps with a total of n vertices, this problem is solved in [CCW93b] in $O(nm \log m)$ time and $O(nm)$ space for a 3- and 4-axis machine equipped with a ball-end cutter. In [GJM$^+$96], the time bound is improved to $O(n^2)$ in the worst case—when $m = \Theta(n)$—and, moreover, an $O(nm \log m)$-time and $O(nm)$-space algorithm is given for 5-axis machines. These results are based on geometric duality, topological sweep (Section 24.4), and properties concerning intersections and covering of polygons on the unit sphere. In [GJM$^+$96], an $O(n^2 + nm \log m)$-time and $O(nm)$-space algorithm is also given for fillet-end tools on 4- and 5-axis machines. All of these results imply an $O(\log m)$-approximation to the minimum number of setups, via the well-known approximation result for the set-cover problem.

Tool paths. A related problem is that of generating tool paths that meet certain optimization criteria, given the pocket geometry, the tool size and geometry, and a machining discipline such as direction-parallel machining, zigzag machining, or contour-parallel machining. The optimization criteria include minimizing the total length traveled by the tool, minimizing the number of ***tool retractions*** (i.e., the number of times the tool is lifted off the workpiece), and minimizing the number of times any point is machined by the tool. (This problem bears similarities to the hatching problem discussed earlier.) In [AHS00], a zigzag pocket machining algorithm is given and it is proved that the number of retractions is at most $5r + 6h$ for a pocket with $h \geq 0$ holes, where r is the minimum number of retractions. Moreover, no point is machined more than once. (Experiments in [AHS00] indicate a better approximation factor of 1.5.) The approach is based on constructing and processing a so-called machining graph. The algorithm runs in $O(n)$ time, where n is the number of vertices in the machining graph. (Here n is a function of the

FIGURE 55.3.1
A great circle for a 4-axis machine (a) *and a spherical band for a 5-axis machine* (b) *intersecting a set of VMaps. (From* [GJM+96]*, with permission.)*



(a) (b)

pocket geometry and the tool size.)

In [AFM00], the following related optimization problem is shown to be NP-hard: Given a polygonal pocket of size n and a tool represented by a unit disk or a square, find a closed path of minimum length that visits every point of the pocket at least once. It is shown, however, that one can compute a path that is at most a constant times longer than a shortest path in time $O(n \log n)$.

Heuristics have also been investigated for other tool-path generation problems—see, for instance, the references cited in [AHS00]. However, no approximation bounds have been proved.

OPEN PROBLEMS

1. The type of visibility considered in the part setup problem is between two points (one being the tool and the other being a point on the part's surface) along a straight line. Characterizations of such VMaps and efficient algorithms are given in [GWT94]. Give characterizations and efficient algorithms for VMaps under point-point visibility along circular trajectories (e.g., as produced by the rotary joints of a robot arm) or along parabolic trajectories (e.g., as executed by droplets under gravity in vapor deposition processes). Also of interest are segment-segment and plane-plane visibility along straight line trajectories.

2. Consider an augmented 4-axis (resp. 5-axis) machine, where the worktable can rotate fully (resp. tilt by $\pi/2$ radians) about a second axis. In the greedy framework described earlier, this reduces to finding a pair of orthogonal great circles (resp. spherical bands) that intersect the maximum number of VMaps. No algorithms are known for this problem.

3. Prove that the zigzag pocket machining problem that calls for the minimum number of retractions and requires that no pocket point is machined more than once ([AHS00]) is NP-hard, or provide a polynomial-time algorithm.

4. Investigate tool-path generation problems for contour-parallel machining and provide provably good approximation algorithms.

55.4 OTHER TOPICS

Besides the three representative topics that we have addressed, there are other areas for fruitful interaction between computational geometry and manufacturing. These include: design of mechanisms and linkages (Section 48.1); geometric constraint systems (Section 56.3); tolerancing of machined parts; interpretation and reconstruction of engineering drawings, assembly and disassembly of components (Section 48.3); geometric software for manufacturing applications, process planning and simulation, mesh generation (Section 25.4); VLSI design and layout, and vision, robotics (Chapter 48); geometric modeling issues relevant to manufacturing (Chapter 53 and 56); and geometric problems arising in other manufacturing processes such as bending, forming, welding, forging, etc.

55.5 SOURCES AND RELATED MATERIAL

FURTHER READING

The following contain additional discussion and references related to the topics in this chapter.

[Bos95, Maj98]: Provide good expositions of the application of computational geometry techniques to problems in molding, casting, and layered manufacturing.

[Woo94]: Discusses various kinds of visibility in the context of different manufacturing processes.

[Hel91]: Contains a detailed discussion of the application of geometric techniques to problems in pocket machining.

[Bra86]: A good general reference on a variety of design and manufacturing processes, including casting, molding, forging, stamping, machining, etc.

RELATED CHAPTERS

Chapter 24: Arrangements
Chapter 28: Visibility
Chapter 29: Geometric reconstruction problems
Chapter 48: Robotics
Chapter 53: Splines and geometric modeling
Chapter 56: Solid modeling

REFERENCES

[ABB+97] B. Asberg, G. Blanco, P. Bose, J. Garcia-Lopez, M. Overmars, G. Toussaint, G. Wilfong, and B. Zhu. Feasibility of design in stereolithography. *Algorithmica*, 19:61–83, 1997.

[AD95] S. Allen and D. Dutta. Determination and evaluation of support structures in layered manufacturing. *J. Design and Manufacturing*, 5:153–162, 1995.

[AD98] S. Allen and D. Dutta. Wall thickness control in layered manufacturing for surfaces with closed slices. *Comput. Geom. Theory Appl.*, 10:223–238, 1998.

[AD00] P. Agarwal and P. Desikan. Approximation algorithms for layered manufacturing. In *Proc. 11th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 528–537, 2000.

[AdB+02] H-K. Ahn, M. de Berg, P. Bose, S.W. Cheng, D. Halperin, J. Matoušek, and O. Schwartzkopf. Separating an object from its cast. *Comput. Aided Design*, 34:547–559, 2002.

[AFM00] E. Arkin, S. Fekete, and J. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom. Theory Appl.*, pages 25–50, 2000.

[AHS00] E. Arkin, M. Held, and C. Smith. Optimization problems related to zigzag pocket machining. *Algorithmica*, 26:197–236, 2000.

[Bar97] G. Barequet. Using geometric hashing to repair CAD models. *IEEE Computational Science & Engineering*, 4:22–28, 1997.

[Bau75] B.G. Baumgart. A polyhedron representation for computer vision. In *Proc. AFIPS National Computer Conf.*, volume 44, pages 589–596, 1975.

[BB95] M. Bablani and A. Bagchi. Quantification of errors in rapid prototyping processes and determination of preferred orientation of parts. In *Transactions of the 23rd North American Manufacturing Research Conf.*, 1995.

[BBvK97] P. Bose, D. Bremner, and M. van Kreveld. Determining the castability of simple polyhedra. *Algorithmica*, 19(1–2):84–113, 1997.

[BK98] G. Barequet and Y. Kaplan. A data front-end for layered manufacturing. *Comput. Aided Design*, 30:231–243, 1998.

[Bøh95] J.H. Bøhn. Removing zero-volume parts from CAD models for layered manufacturing. *IEEE Computer Graphics and Applications*, 15:27–34, 1995.

[Bos95] P. Bose. *Geometric and computational aspects of manufacturing processes.* Ph.D. thesis, School of Computer Science, McGill Univ., Montréal, Canada, 1995.

[Bra86] J.G. Bralla. *Handbook of product design for manufacturing.* McGraw–Hill Book Company, 1986.

[BT95] P. Bose and G. Toussaint. Geometric and computational aspects of gravity casting. *Comput. Aided Design*, 27:455–464, 1995.

[BvKT98] P. Bose, M. van Kreveld, and G. Toussaint. Filling polyhedral molds. *Comput. Aided Design*, 30:245–254, 1998.

[Cad02] CADCAM Net, 2002. `http://www.cadcamnet.com/Sections/rapid%20prototyping/Applications.htm`.

[CCW93a] L-L. Chen, S-Y. Chou, and T. Woo. Parting directions for mould and die design. *Comput. Aided Design*, 25:762–768, 1993.

[CCW93b] L-L. Chen, S-Y. Chou, and T. Woo. Separating and intersecting spherical polygons: Computing machinability on three-, four-, and five-axis numerically controlled machines. *ACM Trans. Graph.*, 12:305–326, 1993.

[CMP95] V. Chandru, S. Manohar, and C. Prakash. Voxel-based modeling for layered manugacturing. *IEEE Computer Graphics and Applications*, 15:42–47, 1995.

[DKPS98] D. Dutta, V. Kumar, M. Pratt, and R. Sriram. Towards STEP-based data transfer in Layered Manufacturing. In *Proc. 10th Internat. Conf. PROLOMAT*, 1998.

[DM94] A. Dolenc and I. Mäkelä. Slicing procedures for layered manufacturing techniques. *Comput. Aided Design*, 26:119–126, 1994.

[FF94] D.F. and G. Fadel. Preferred direction of build for rapid prototyping processes. In *Proc. 5th Internat. Conf. Rapid Prototyping*, pages 191–200, 1994.

[FK96] R. Farouki and T. König. Computational methods for rapid prototyping of analytic solid models. *Rapid Prototyping Journal*, 2:41–48, 1996.

[FM93] S. Fekete and J. Mitchell. Geometric aspects of injection molding. Workshop Geometric and Computational Aspects of Injection Molding, Bellairs Research Institute, 1993.

[FM01] S. Fekete and J. Mitchell. Terrain decomposition and layered manufacturing. *Internat. J. Computational Geometry and Applications*, 11:647–668, 2001.

[GJM+96] P. Gupta, R. Janardan, J. Majhi, and T. Woo. Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC-machining. *Comput. Aided Design*, 28:577–587, 1996.

[GWT94] J. Gan, T. Woo, and K. Tang. Spherical maps: Their construction, properties, and approximation. *J. Mech. Design*, 116:357–363, 1994.

[Har01] G. Hart. Rapid prototyping of geometric models, 2001. `http://www.georgehart.com/cccg/rpgm.html`. Invited talk at 13th Canad. Conf. Comput. Geom., Waterloo, Canada, August 2001.

[Hel91] M. Held. *On the Computational Geometry of Pocket Machining.* LNCS, volume 500. Springer–Verlag, 1991.

[HJSS03] M. Hon, R. Janardan, J. Schwerdt, and M. Smid. Minimizing the total projection of a set of vectors, with applications to layered manufacturing. *Comput. Aided Design*, 35:57–68, 2003.

[HT88] M.E. Houle and G.T. Toussaint. Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10:761–765, 1988.

[HT92] K.C. Hui and S.T. Tan. Mould design with sweep operations–a heuristic search approach. *Comput. Aided Design*, 24:81–91, 1992.

[IJM+02] I. Ilinkin, R. Janardan, J. Majhi, J. Schwerdt, M. Smid, and R. Sriram. A decomposition-based approach to layered manufacturing. *Comput. Geom. Theory Appl.*, 23:117–151, 2002.

[IJS02] I. Ilinkin, R. Janardan, and M. Smid. Terrain polygon decomposition with application to layered manufacturing. In *Proc. 8th Internat. Comput. Combin. Conf.*, LNCS 2387, pages 381–390, 2002.

[Jac92] P. Jacobs. *Rapid Prototyping & Manufacturing: Fundamentals of StereoLithography.* McGraw-Hill, 1992.

[Joh99] E. Johnson. Support generation for three-dimensional layered manufacturing. Master's project report, Dept. of CS&E, Univ. Minnesota, Minneapolis, MN, 1999.

[KD96] P. Kulkarni and D. Dutta. An accurate slicing procedure for layered manufacturing. *Comput. Aided Design*, 28:683–697, 1996.

[KD97] V. Kumar and D. Dutta. An assessment of data formats for layered manufacturing. *Advances in Engineering Software*, 28:151–164, 1997.

[KF97] C.C. Kai and L.K. Fai. *Rapid Prototyping: Principles and applications in manufacturing*. J. Wiley & Sons, 1997.

[Lev02] W. Leventon. Synthetic skin. *IEEE Spectrum*, pages 28–33, December 2002.

[Maj98] J. Majhi. *Geometric methods in computer-aided design and manufacturing*. Ph.D. thesis, Dept. of Comput. Sci. & Eng. Univ. Minnesota, Minneapolis, MN, 1998.

[MGJ99] J. Majhi, P. Gupta, and R. Janardan. Computing a flattest, undercut-free parting line for a convex polyhedron, with application to mold design. *Comput. Geom. Theory Appl.*, 13:229–252, 1999.

[MJS+99] J. Majhi, R. Janardan, J. Schwerdt, M. Smid, and P. Gupta. Minimizing support structures and trapped area in two-dimensional layered manufacturing. *Comput. Geom. Theory Appl.*, 12:241–267, 1999.

[MJSG99] J. Majhi, R. Janardan, M. Smid, and P. Gupta. On some geometric optimization problems in layered manufacturing. *Comput. Geom. Theory Appl.*, 12:219–239, 1999.

[MJSS01] J. Majhi, R. Janardan, J. Schwerdt, and M. Smid. Multi-criteria geometric optimization problems in layered manufacturing. *Internat. J. Mathematical Algorithms*, 2:201–225, 2001.

[MS99] S. McMains and C. Séquin. A coherent sweep plane slicer for layered manufacturing. In *Proc. 5th Annu. ACM Sympos. Solid Modeling and Applications*, pages 285–295, 1999.

[Mul93] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice-Hall, 1993.

[NLG02] P. Ng, P. Lee, and J. Goh. Prosthetic sockets fabrication using rapid prototyping technology. *Rapid Prototyping Journal*, 8:53–59, 2002.

[RR94] A. Rosenbloom and D. Rappaport. Moldable and castable polygons. *Comput. Geom. Theory Appl.*, 4:219–233, 1994.

[RS90] B. Ravi and M.N. Srinivasan. Decision criteria for computer-aided parting surface design. *Comput. Aided Design*, 22:11–18, 1990.

[Sar99] S. Sarma. The crossing function and its application to zig-zag tool paths. *Comput. Aided Design*, 31:881–890, 1999.

[SSHJ02] J. Schwerdt, M. Smid, M. Hon, and R. Janardan. Computing an optimal hatching direction in layered manufacturing. *Internat. J. Comput. Math.*, 79:1067–1081, 2002.

[SSJ+00] J. Schwerdt, M. Smid, R. Janardan, E. Johnson, and J. Majhi. Protecting critical facets in layered manufacturing. *Comput. Geom. Theory Appl.*, 16:187–210, 2000.

[SSJJ] J. Schwerdt, M. Smid, R. Janardan, and E. Johnson. Protecting critical facets in layered manufacturing: implementation and experimental results. *Comput. Aided Design*, 35:647–657, 2003.

[SSMJ99] J. Schwerdt, M. Smid, J. Majhi, and R. Janardan. Computing the width of a three-dimensional point-set: an experimental study. *ACM J. Experimental Algorithmics*, volume 4, Art. 8, 1999.

[Woo94] T. Woo. Visibility maps and spherical algorithms. *Comput. Aided Design*, 26:6–16, 1994.