Multi-criteria geometric optimization problems in Layered Manufacturing*

Jayanth Majhi† Ravi Janardan‡ Michiel Smid§ Jörg Schwerdt§

July 31, 2000

Abstract

In Layered Manufacturing, the choice of the build direction for the model influences several design criteria, including the number of layers, the volume and contact-area of the support structures, and the surface finish. These, in turn, impact the throughput and cost of the process. In this paper, efficient geometric algorithms are given to reconcile two or more of these criteria simultaneously, under three formulations of multi-criteria optimization: Finding a build direction which (i) optimizes the criteria sequentially, (ii) optimizes their weighted sum, or (iii) allows the criteria to meet designer-prescribed thresholds. The algorithms that involve "support volume" or "contact area" apply only to convex models, while the algorithms that involve "surface finish" and "number of layers" apply to any polyhedral model. Some of the latter algorithms have also been implemented and tested on real-world models obtained from industry.

The geometric techniques used in the paper include construction and searching of certain arrangements on the unit-sphere, three-dimensional convex hulls, Voronoi diagrams, point location, and hierarchical representations. Additionally, solutions are also provided, for the constrained versions of two geometric problems, namely polyhedron width and largest empty disk on the unit-sphere.

1 Introduction

Layered Manufacturing (LM) is an emerging technology which enables complex 3D parts to be built directly from their CAD models, as a stack of 2D layers. It is revolutionizing the field of CAD/CAM because it allows the designer to create rapidly a physical version of the CAD model (literally on the desktop) and to "feel and touch" it, thereby detecting and correcting flaws in the model early on in the design cycle. Moreover, it opens up the possibility of directly building functional parts composed of multiple materials—something that is not possible via conventional manufacturing methods.

---

*A preliminary version of this paper appears in *Proc. 14th ACM Symposium on Computational Geometry*, Minneapolis, MN, pp. 19–28, 1998.

†Synopsys Corp., 700 East Middlefield Road, Mountain View, CA 94043. E-mail: `majhi@synopsys.com` . This work was done while at the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, and was supported in part by NSF grant CCR-9712226.

‡Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A. E-mail: `janardan@cs.umn.edu` . Research supported in part by NSF grant CCR-9712226.

§Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, D-39106 Magdeburg, Germany. E-mail: `{michiel,schwerdt}@isg.cs.uni-magdeburg.de` .

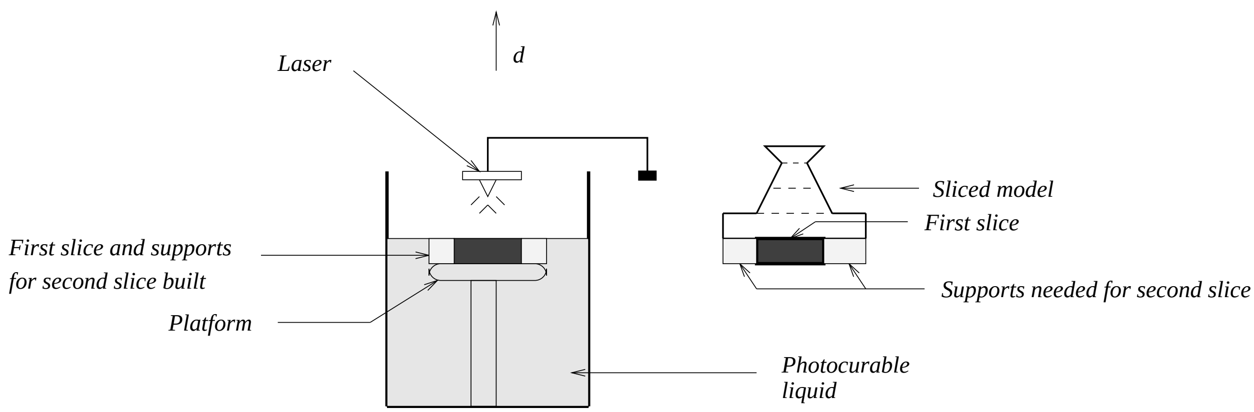

Figure 1: *The stereolithograpy apparatus.*

Figure 1 depicts a well-known LM process called *StereoLithography* [13]. The input is a surface triangulation of the CAD model in a format called STL. (The STL format is the *de facto* input standard in the LM industry; it consists of an unordered list of the triangles in the triangulation, along with their outward unit normals.) The model is oriented suitably, sliced by horizontal planes, and built vertically, as follows: The laser traces out the contour of each layer on the surface of the photocurable liquid and cross-hatches the interior, causing it to harden and attach to the previous layer. The platform is then lowered by an amount equal to the layer thickness and the next layer is built similarly atop the previous one. Simultaneously, *support structures* are also generated to prop up portions of the current layer that overhang the previous layer. Towards this end, the CAD model is analyzed beforehand and a description of the supports is generated and merged into the STL file. After it is built, the part is postprocessed to remove the supports and to improve the finish, which has a stair-stepped appearance due to the non-zero layer thickness. Figure 2 illustrates stair-stepping by showing the cross-section of a four-sided "cylinder" that is normal to the paper. Facets that are not parallel to the build direction **d** will experience stair-stepping.

1.1 Geometric issues in Layered Manufacturing

A key process-planning step in LM is choosing the model's orientation, or, equivalently, the *build direction*. This choice impacts, among other things, the number of layers, the surface finish, the quantity of supports, and the area of contact between supports and part. These, in turn, affect the speed, throughput, and cost of the process. For example, if the "cylinder" in Figure 2 is built in the indicated direction **d**, then it will have a stair-stepped finish and will require supports under facet $\overline{14}$ (along its entire length normal to the paper). If it is built normal to the paper, no supports are needed and there is no stair-stepping except possibly on the top and bottom facets (where we mean "top"/"bottom" w.r.t. the build direction); however, the number of layers will be very large. Presently, the build direction is chosen by the human operator, based on experience, so that the number of layers and the quantity of supports is "small" and the surface finish is "good". However, this approach is both time-consuming and prone to errors and there is an urgent need for computer algorithms that can automate this step. Indeed, the problems that we consider here are motivated,

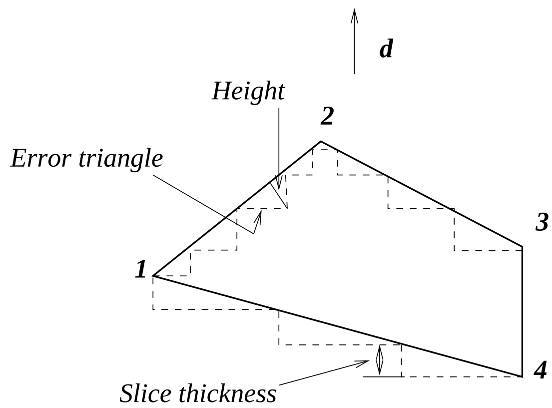

Figure 2: *Illustrating stair-stepping. The figure shows the cross-section of a 4-sided "cylinder". The cylinder is normal to the paper and has a uniform cross-section along its entire length. Facets that are not parallel to the build direction will have a stair-stepped finish.*

---

in part, by discussions with engineers at Stratasys, Inc.—a leading Minnesota-based manufacturer of LM machines.

As illustrated above, a single build direction will, in general, not optimize multiple design criteria simultaneously. Thus it is important to find a build direction which reconciles multiple criteria in a meaningful way. In this paper we formulate and solve several such multi-criteria optimization problems.[1]

1.2 Some design criteria for LM

Throughout the paper, we denote by $\mathcal{P}$ the polyhedron that we wish to build and by n the number of vertices in $\mathcal{P}$.[2] We let $\mathbf{d}$ denote the build direction. The criteria of interest include:

STAIR-STEP ERROR (CRITERION $\mathcal{C}_{str}$): The degree of stair-stepping on a facet, f, depends on the angle between the facet normal and $\mathbf{d}$, and it can be mitigated by a suitable choice of $\mathbf{d}$. In [4], the notion of an *error-triangle* for a facet is introduced as a way of quantifying stair-stepping (Figure 2). For a direction $\mathbf{d}$, we define the *stair-step error*, $\mathcal{C}_{str}(\mathbf{d})$, as the maximum height of the error triangle, taken over all facets.

VOLUME OF SUPPORTS (CRITERION $\mathcal{C}_{vol}$): The quantity of supports used affects the build time and the cost of raw materials. If $\mathcal{P}$ is convex, then the support volume, $\mathcal{C}_{vol}(\mathbf{d})$, for direction $\mathbf{d}$ is the volume of the region enclosed between $\mathcal{P}$ and the platform, i.e., the volume of the polyhedron which is bounded from below by the platform and from above by the *back facets* of $\mathcal{P}$ w.r.t. $\mathbf{d}$, i.e., the facets whose outward normals make an angle greater than $\pi/2$ radians with $\mathbf{d}$. If $\mathcal{P}$ is non-convex, then the supports for some facets may actually be attached to other facets instead of to the platform, and hence the structure of the supports is more complex.

[1] Besides the build direction, process planning in LM is also affected by other factors, such as, for instance, the STL format which is redundant, error-prone, and devoid of topological information, and hence not very conducive to efficient processing. However, we do not address these issues here.

[2] Note that because of the STL representation, the input is polyhedral.

CONTACT-AREA OF SUPPORTS (CRITERION $\mathcal{C}_{area}$): For direction $\mathbf{d}$, we define $\mathcal{C}_{area}(\mathbf{d})$ as the area of $\mathcal{P}$'s surface that is in contact with the supports. This area affects the postprocessing time, since the supports that "stick" to $\mathcal{P}$ must be removed, and the parts of $\mathcal{P}$ that were in contact with supports must be further processed (e.g., sanded) to improve their finish. If $\mathcal{P}$ is convex, then $\mathcal{C}_{area}(\mathbf{d})$ is the total area of the back facets (including the area of facets that rest on the platform). If $\mathcal{P}$ is non-convex, then $\mathcal{C}_{area}(\mathbf{d})$ is the total surface area that is in contact with the supports. (It includes the areas of all back facets and portions of the other facets.)

NUMBER OF LAYERS (CRITERION $\mathcal{C}_{wid}$): As layer thickness is often measured in thousandths of an inch, the number of slices can run into the thousands if the part is oriented along its longest dimension. This can increase the build time considerably. If the layer thickness is fixed, then for a given build direction $\mathbf{d}$, the number of layers is proportional to the *width of $\mathcal{P}$ in direction* $\mathbf{d}$, i.e., the smallest distance between two parallel planes that are normal to $\mathbf{d}$ and which enclose $\mathcal{P}$. We denote this by $\mathcal{C}_{wid}(\mathbf{d})$.

In previous work [16], we show how each of the first three criteria can be optimized, when considered independently. Specifically, we show how to find a $\mathbf{d}$ which minimizes $\mathcal{C}_{str}(\mathbf{d})$ in time $O(n \log n)$, for any polyhedron $\mathcal{P}$, and how to find a $\mathbf{d}$ which minimizes $\mathcal{C}_{vol}(\mathbf{d})$ or $\mathcal{C}_{area}(\mathbf{d})$ in $O(n^2)$ time for convex $\mathcal{P}$. Finally, a $\mathbf{d}$ minimizing $\mathcal{C}_{wid}(\mathbf{d})$ can be computed in $O(n^2)$ time [12].[3]

We close this discussion by noting that in considering the criteria $\mathcal{C}_{str}$, $\mathcal{C}_{vol}$, and $\mathcal{C}_{area}$, we have assumed implicitly that all facets are equally important. In practice, however, it is possible that certain facets are more important functionally than others, so the designer may require that these be safeguarded by, for instance, not allowing supports to touch them. We address this problem in a separate paper [19].

1.3 Formulation of multi-criteria optimization problems in LM

We investigate our problem under three formulations of multi-criteria optimization. We describe these below in the context of two generic criteria, $\mathcal{C}_1$ and $\mathcal{C}_2$, but we emphasize that our methods extend easily to more than two criteria.

THRESHOLD FORMULATION: Find a build direction $\mathbf{d}$ such that $\mathcal{C}_1(\mathbf{d}) \leq \rho_1$ and $\mathcal{C}_2(\mathbf{d}) \leq \rho_2$, where $\rho_1 \geq 0$ and $\rho_2 \geq 0$ are designer-specified real numbers.

For instance, if $\mathcal{C}_1$ and $\mathcal{C}_2$ correspond to support volume and contact-area, respectively, then the designer could insist that the support volume be a fraction (e.g., 50%) of the part's volume and the contact-area be a fraction (e.g., 25%) of the part's surface area. Then ρ_1 and ρ_2 would be the appropriate fraction times the part's volume or surface area, respectively.

WEIGHTED FORMULATION: Find a build direction such that $w_1\mathcal{C}_1 + w_2\mathcal{C}_2$ is minimum, where w_1 and w_2 are designer-specified real numbers reflecting the relative importance of the two criteria.

SEQUENTIAL FORMULATION: Among all build directions minimizing $\mathcal{C}_1$, find one minimizing $\mathcal{C}_2$.

1.4 Contributions of this paper

We provide a suite of algorithms that provably optimize different combinations of the above-mentioned criteria under the three formulations. To our knowledge, these are the first such results

[3] A faster (randomized) algorithm, which runs in roughly $O(n^{1.5})$ time is known [1] but it is not well-suited for our purposes here.

for this problem. These algorithms incorporate our earlier algorithms for single-criterion optimization [16] as building blocks, and are modular in the sense that solutions for different combinations of criteria can be derived quickly by combining the appropriate single-criterion algorithms. We have also implemented some of our algorithms and tested them on a variety of STL files obtained from industry.

In the course of solving the multi-criteria problems, we also provide algorithms for solving the constrained versions of two fundamental geometric problems, namely, computing the *width* of a polyhedron in three-dimensions (i.e., the smallest separation between two parallel planes that sandwich the polyhedron) and computing the *largest empty disk* for a set of point-sites on the unit-sphere, $\mathbb{S}^2$ (i.e., the largest spherical disk that does not contain any site in its interior). The problems are constrained in the sense that the search space (in the former case, the set of directions realizing the width, and, in the latter case, the center of the largest empty disk) is restricted to a polygonal region on $\mathbb{S}^2$. Previously, only the unconstrained width problem and the planar version of the largest empty disk problem had been investigated [12, 18]. We establish new geometric characterizations for these constrained problems, which may be of independent interest.

Our results also demonstrate the applicability of a number of techniques from computational geometry, including construction and searching of various arrangements on the unit-sphere, three-dimensional convex hulls, Voronoi diagrams, point location, and hierarchical representations of polyhedra. Additionally, we also employ methods from continuous optimization such as Lagrange Multipliers.

A limitation of our work is that the solutions involving "support volume" or "contact area" as a criterion are applicable only to convex polyhedra. We are investigating these problems for non-convex polyhedra. We note that our solutions for "stair-step error" and "number of layers" are applicable to any polyhedron, even one with holes—and these are the ones that we have implemented.

1.5 Prior work

While there has been a fair amount of work done on single-criterion optimization for LM (e.g., [2, 3, 11, 15, 16, 17]), the only work that we have come across for multi-criteria optimization in LM is [7], which considers two criteria—accuracy and build time—under the sequential formulation. (Reference [7] also mentions two technical reports [14, 20] that appear to consider related problems.) The solution proposed in [7] is not optimal since it restricts the search space to the set of facet normals, which are then checked exhaustively. (By contrast, our solutions here are all provably optimal.)

The rest of the paper is organized as follows: In Section 2, we review briefly our earlier results on single-criterion optimization. In Section 3, we describe our new results for constrained geometric problems. In Section 4, we show how to combine these results to solve the multi-criteria problems of interest. We describe our experimental results in Section 5, and conclude in Section 6.

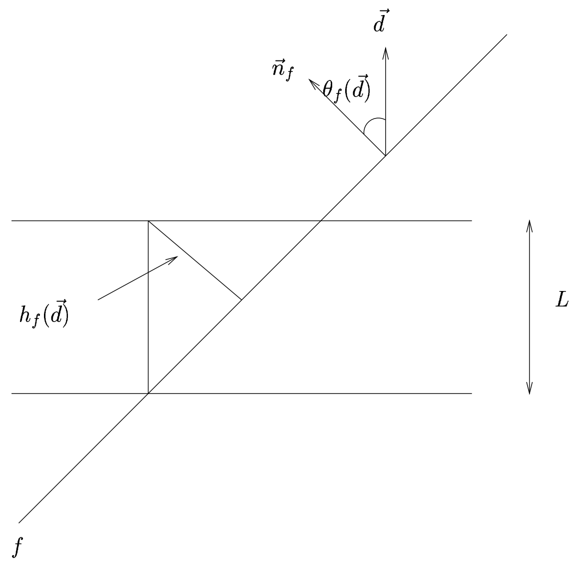

Figure 3: *The error triangle height for a facet f.*

2 Single-criterion optimization—an overview

2.1 Minimizing stair-step error

Given a polyhedron $\mathcal{P}$, we wish to find a direction $\mathbf{d}$ which minimizes the maximum error-triangle height, $\mathcal{C}_{str}(\mathbf{d})$. Let $t_f(\mathbf{d})$ denote the error triangle for facet $f \in \mathcal{P}$ and let $h_f(\mathbf{d})$ denote the height of $t_f(\mathbf{d})$. If $\mathbf{n}_f$ is the outward unit-normal to f, it is easy to see that $h_f(\mathbf{d}) = L\cos\theta_f(\mathbf{d})$, where L is the layer thickness and $\theta_f(\mathbf{d})$ is the smaller of two angles, one between $\mathbf{d}$ and $\mathbf{n}_f$ and the other between $\mathbf{d}$ and $-\mathbf{n}_f$. (See Figure 3.) We seek a direction $\mathbf{d}$ such that $\mathcal{C}_{str}(\mathbf{d}) = \max_f L\cos\theta_f(\mathbf{d})$ is minimized, i.e., $\min_f \theta_f(\mathbf{d})$ is maximized.

Consider the set $S = \{\mathbf{n}_f \cap \mathbb{S}^2, -\mathbf{n}_f \cap \mathbb{S}^2 | f \text{ is a facet of } \mathcal{P}\}$ of $O(n)$ *sites* on $\mathbb{S}^2$. Our problem is to find a point $\mathbf{d}$ on $\mathbb{S}^2$ such that the minimum angle between it and the sites is maximized. Let us define a *disk* on $\mathbb{S}^2$, with *center* $\mathbf{d}$ and *radius* θ, as the set of all points on $\mathbb{S}^2$ that are at a distance of at most θ from $\mathbf{d}$, as measured along the surface of $\mathbb{S}^2$. Clearly, our problem is equivalent to finding the center $\mathbf{d}$ of the largest-radius disk whose interior does not contain any site, i.e., a *largest empty disk*. It can be shown that the desired center is a vertex of *Vor(S)*—the Voronoi diagram of S on $\mathbb{S}^2$. (Equivalently, the desired center is the normal from the origin to a facet of the convex hull of S [16].) Thus our problem can be solved in $O(n \log n)$ time.

Let R be a face of *Vor(S)* and let $\mathbf{n}_R$ be the site of S associated with it (i.e., any point in R's interior is closer to $\mathbf{n}_R$ than to any other site). Then, for any $\mathbf{d} \in R$, the largest empty disk with its center at $\mathbf{d}$ contains $\mathbf{n}_R$ on its boundary. Hence, if $\theta_R(\mathbf{d})$ is the angle between $\mathbf{n}_R$ and $\mathbf{d}$, then the stair-step error for any $\mathbf{d} \in R$ is given by $L\cos\theta_R(\mathbf{d}) = L(\mathbf{n}_R \cdot \mathbf{d})$, which is a linear function of the form $A_R x + B_R y + C_R z$, where $\mathbf{d} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ and A_R, B_R, and C_R are constants depending only on $\mathbf{n}_R$ and L. We will use this fact in Section 4.2.

2.2 Minimizing contact-area of supports

For convex $\mathcal{P}$, a facet $f \in \mathcal{P}$ needs support for direction $\mathbf{d}$ iff it is a back facet. Thus, all directions for which f needs support can be represented on $\mathbb{S}^2$ by an open hemisphere, h_f, with center $-\mathbf{n}_f$.

Let $\mathcal{A}_{area}$ be the arrangement of the great circles bounding all h_f's. If R is any face of $\mathcal{A}_{area}$, then for all $\mathbf{d} \in R$, the same facets of $\mathcal{P}$ require support, and so the total contact area, $\mathcal{C}_{area}(\mathbf{d})$, for all such $\mathbf{d}$ is a constant $\mathcal{C}_{area}(R)$. Thus we need to find a face, R, of $\mathcal{A}_{area}$ for which $\mathcal{C}_{area}(R)$ is minimum. In [16], we show how to do this in $O(n^2)$ time using a planar (topological) sweep algorithm [10].

2.3 Minimizing volume of supports

For convex $\mathcal{P}$, the combinatorial structure of the supports for a direction $\mathbf{d}$ is determined by the back facets and by the vertex of $\mathcal{P}$ that is farthest away in direction $-\mathbf{d}$. (This is the vertex touching the platform when $\mathcal{P}$ is built along $\mathbf{d}$; we call it the *extreme vertex* of $\mathcal{P}$ w.r.t. $\mathbf{d}$.)

Let $\mathcal{A}_{vol}$ be an arrangement on $\mathbb{S}^2$ with the property that, for any face $R \in \mathcal{A}_{vol}$, the back facets and the extreme vertex are the same for any $\mathbf{d} \in R$. Thus, the combinatorial structure of the supports (but not their volume) is invariant within R. $\mathcal{A}_{vol}$ can be computed as the overlay of two arrangements of great arcs, $\mathcal{A}_{area}$ and $\mathcal{A}_{ext}$: $\mathcal{A}_{area}$ (defined in Section 2.2) is an arrangement of great arcs, each of whose faces represents all directions for which the back facets of $\mathcal{P}$ are invariant. $\mathcal{A}_{ext}$ is an arrangement of great arcs, each of whose faces represents all directions for which the extreme vertex of $\mathcal{P}$ is invariant. This arrangement was introduced in [5] and we refer the reader to that source for more details.

As proved in [16], for any $\mathbf{d} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ in R, the support volume is of the form $Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz$. In [16], we find the best $\mathbf{d}$ within R by using the Lagrange Multipliers method, subject to the constraints $x^2 + y^2 + z^2 = 1$ and $a_i x + b_i y + c_i z \geq 0$ for each great arc r_i bounding R. (Here $A, B, \ldots, F$ and a_i, b_i, and c_i are constants.) The volume formula is updated incrementally and efficiently by visiting the faces in a depth-first order. The algorithm runs in $O(n^2)$ time.

2.4 Minimizing the number of layers

Let $\mathcal{P}$ be a polyhedron and let $CH(\mathcal{P})$ be its convex hull. Vertices u and v of $CH(\mathcal{P})$ are an *antipodal vertex-vertex pair* (*VV pair*) if $\mathcal{P}$ can be contained between parallel planes passing through u and v. *Antipodal vertex-edge (VE), vertex-facet (VF), and edge-edge (EE) pairs* are defined similarly. (Antipodal edge-facet and facet-facet pairs are subsumed by antipodal vertex-facet pairs, and hence not considered.) In [12], it is shown that a direction $\mathbf{d}$ minimizing the width $\mathcal{C}_{wid}(\mathbf{d})$ must be perpendicular to the parallel planes for a VF or an EE pair. The VF and EE antipodal pairs can be determined as follows [6]:

The upper and lower hulls of $CH(\mathcal{P})$ are mapped to two subdivisions, $\mathcal{A}_u$ and $\mathcal{A}_l$, on $\mathbb{S}^2$. The vertices of these subdivisions are in 1–1 correspondence with the facet normals of $CH(\mathcal{P})$, and two vertices are joined by a great arc if the corresponding facets share an edge. The faces of the subdivisions are in 1–1 correspondence with the vertices of $CH(\mathcal{P})$. The VF pairs are obtained from $\mathcal{A}_u$ and $\mathcal{A}_l$ by locating each vertex of $\mathcal{A}_u$ in a face of $\mathcal{A}_l$, and vice versa. The EE pairs are obtained by computing the overlay, $\mathcal{A}_{wid}$, of $\mathcal{A}_u$ and $\mathcal{A}_l$ and taking the edge intersections. The running time is $O(n^2)$.

For later reference, we note the following properties of $\mathcal{A}_{wid}$: The vertices correspond to directions that are normal to the parallel supporting planes of antipodal VF or EE pairs. The interior of any edge (resp., face) corresponds to directions that are normal to the parallel supporting planes of antipodal VE (resp., VV) pairs. Consequently, we have the following observation: Let E be any *element* of $\mathcal{A}_{wid}$, i.e., a vertex, edge interior, or face interior, and let f_1 and f_2 be any of the following pairs of *features* of $CH(\mathcal{P})$: two vertices, two edges, a vertex and a facet, or a vertex and an edge. Then, for any $\mathbf{d} = \mathbf{x}i + \mathbf{y}j + \mathbf{z}k$ in E, the same two features of $CH(\mathcal{P})$ determine $\mathcal{C}_{wid}(\mathbf{d})$. Moreover, let $\mathbf{u_1}$ and $\mathbf{u_2}$ be the position vectors of any points on f_1 and f_2 respectively. Then for any $\mathbf{d} \in E$, $\mathcal{C}_{wid}(\mathbf{d}) = (\mathbf{u_1} - \mathbf{u_2}) \cdot \mathbf{d}$ (assuming $\mathbf{u}_1$ is above $\mathbf{u_2}$ w.r.t. $\mathbf{d}$). Thus, $\mathcal{C}_{wid}(\mathbf{d})$ is a linear function of the form $A_E x + B_E y + C_E z$, where A_E, B_E, and C_E depend only on E. We will use this fact in Section 4.1.

3 Solving geometric problems in a constrained domain

We now discuss how to solve the width, stair-step error, and support volume problems when $\mathbf{d}$ is restricted to a polygonal region on $\mathbb{S}^2$. These results will be used as subroutines in our multi-criteria algorithms in Section 4. We begin with a utility lemma:

Lemma 3.1 *(i) Let $\mathcal{A}$ and $\mathcal{B}$ be arrangements of $O(n)$ great arcs each on $\mathbb{S}^2$. Then the overlay, $\mathcal{C}$, of $\mathcal{A}$ and $\mathcal{B}$ can be computed in $O(n^2)$ time.*

(ii) Let R be a convex polygon on $\mathbb{S}^2$ composed of great and/or small arcs and let $\mathcal{A}$ be an arrangement of $O(n)$ great arcs on $\mathbb{S}^2$. Then the overlay, $\mathcal{A}'$, of R and $\mathcal{A}$, restricted to the region R, has size $O(n^2 + |R|)$ and can be computed in $O(n^2 + n\log|R| + |R|)$ time, where $|R|$ denotes the number of vertices of R.

Proof We prove part (i) as follows: Let a be any great arc of $\mathcal{A}$ or $\mathcal{B}$ and let G_a be the great circle containing it. We add the G_a's one by one to the initially empty arrangement. We compute the intersections of each G_a with the current arrangement and retain only those that also belong to a. By the zone theorem [8], which is also applicable to great circles, due to central projection, this takes $O(n)$ time per G_a, hence $O(n^2)$ time in all.

The proof of part (ii) is as follows: $\mathcal{A}'$ has three types of vertices: (a) vertices of $\mathcal{A}$ inside/on R, (b) vertices of R, and (c) intersections between arcs of $\mathcal{A}$ and arcs of R. The number of vertices of type (a) is $O(n^2)$ and the number of vertices of type (b) is $O(|R|)$. The number of vertices of type (c) is $O(n)$ since, by convexity, each great arc of $\mathcal{A}$ can intersect R at most twice. Therefore, $\mathcal{A}'$ has size $O(n^2 + |R|)$. To compute it, we start with the polygon R, and add the great circles, G_a, corresponding to the great arcs, a, of $\mathcal{A}$ one at a time. We compute the at most two intersections of the current G_a and R in $O(\log|R|)$ time using binary search on R (this yields the type (c) vertices). We then walk along G_a through the part of the current arrangement that is in the interior of R, from one intersection point to the other, and compute the intersection of G_a with previously added great circles (type (a) vertices). We retain only those intersections that are also on the great arc a. By the zone theorem [8], applied to G_a and the great circles, this takes $O(n)$ time per G_a. The time bound follows. ∎

3.1 The constrained width problem

The problem here is to minimize $\mathcal{C}_{wid}(\mathbf{d})$ when $\mathbf{d}$ is restricted to lie within a convex polygonal region, R, on $\mathbb{S}^2$, bounded by great arcs. Note that the antipodal VF or EE pair which minimizes the width of $\mathcal{P}$ in the unconstrained case may not yield a direction lying in R. Indeed the minimum width may now be determined by some other type of antipodal pair, as the following lemma shows.

Lemma 3.2 *Let $\mathcal{P}$ be a polyhedron and let R be a convex polygon on $\mathbb{S}^2$ which is bounded by great arcs. A width-minimizing direction, $\mathbf{d}$, lying in R satisfies one of the following:*

(i) *it is perpendicular to the parallel supporting planes associated with an antipodal VF or EE pair of $CH(\mathcal{P})$ and lies in the interior of R, or in the interior of an edge of R, or coincides with a vertex of R,*

(ii) *it is perpendicular to the parallel supporting planes associated with an antipodal VE pair of $CH(\mathcal{P})$ and lies in the interior of an edge of R or coincides with a vertex of R,*

(iii) *it is perpendicular to the parallel supporting planes associated with an antipodal VV pair of $CH(\mathcal{P})$ and coincides with a vertex of R.*

Proof Any direction in R must lie either in the interior of R, or in the interior of an edge of R, or must coincide with a vertex of R. Moreover, any such direction must be perpendicular to the parallel supporting planes associated with an antipodal VF, or VE, or EE, or VV pair of $CH(\mathcal{P})$. The only possibilities for $\mathbf{d}$, other than those given by the lemma, are that (a) it lies in the interior of R and is perpendicular to the parallel supporting planes of an antipodal VE or VV pair, or (b) it lies in the interior of an edge of R and is perpendicular to the parallel supporting planes of an antipodal VV pair.

Consider such a direction $\mathbf{d}$. Suppose that case (a) holds and suppose that the antipodal pair in question is a VE pair. Thus, one of the parallel supporting planes, h_1, contains an edge, e, of $CH(\mathcal{P})$, and the other plane, h_2, contains a vertex, u, of $CH(\mathcal{P})$. Direction $\mathbf{d}$ is perpendicular to h_1 and h_2 and is in R's interior; without loss of generality assume $\mathbf{d}$ coincides with the normal of h_1 that is directed outwards from the region enclosed by h_1 and h_2. If we rotate h_1 and h_2 infinitesimally about e in either direction, such that they remain parallel and contain e and u, then the outward normal of h_1 moves along a great arc and remains in R's interior. As shown in [12], one of the two rotations will yield a width less than $\mathcal{C}_{wid}(\mathbf{d})$—a contradiction. A similar discussion applies for a VV pair in case (a) or case (b). (In case (b), we can rotate the supporting planes such that the width-reducing direction stays in the interior of the edge in question, which yields a contradiction). ∎

From Lemma 3.2 and the properties of $\mathcal{A}_{wid}$ in Section 2.4, it follows that the optimal direction for the constrained minimum width problem is (1) a vertex of R, or (2) a vertex of $\mathcal{A}_{wid}$ lying in the interior of R or in the interior of an edge of R, or (3) the intersection of the interior of an edge of $\mathcal{A}_{wid}$ with the interior of an edge of R.

Theorem 3.1 *A width-minimizing direction for an n-vertex polyhedron, where the directions are restricted to a convex polygon R on $\mathbb{S}^2$ composed of great arcs, can be found in $O(n^2+n\log|R|+|R|)$ time.*

Proof The number of candidate directions of types (1)–(3) listed above are, respectively, $O(|R|)$, $O(n^2)$, and $O(n)$. These points can be identified by constructing the overlay of R and $\mathcal{A}_{wid}$ in $O(n^2+n\log|R|+|R|)$ time (Lemma 3.1(ii)), which implies the same time bound for the constrained width problem. ∎

Remark 3.1 This method can be extended to regions R composed of small arcs. The directions given by Lemma 3.2 remain candidates. In addition, we need to consider each direction that corresponds to a VV pair lying in the interior of a small arc of R. (For a VV pair, (u, v), a width-reducing direction is obtained by passing any pair of parallel lines, l_u and l_v, through u and v, and rotating the supporting planes about l_u and l_v—see [12]. This rotation yields a great arc, not a small arc. Thus, if a VV pair lies in the interior of a small arc then it may not be possible to find a rotation which reduces the width. Hence, a width-minimizing direction could lie in the interior of a small arc.)

To handle such directions, we divide the boundary of R into segments such that within a segment, the same VV pair determines the width of $\mathcal{P}$. By the properties of $\mathcal{A}_{wid}$ (Section 2.4), each such segment lies in the interior of a face of $\mathcal{A}_{wid}$. The division of R into segments can be accomplished by first intersecting each great circle containing an arc $a \in \mathcal{A}_{wid}$ with R and retaining only those intersections that are on a. This gives $O(n)$ intersections. We sort the intersections on each edge of R to obtain the desired segments. This takes $O(|R| + n \log n)$ time. As noted in Section 2.4, the width within a segment, E, is a linear function $A_E x + B_E y + C_E z$. We find the minimum of this function by solving an optimization problem under the constraint that the direction must lie within the segment. Specifically, let s be such a segment with endpoints p and q, and let o be the center of the small circle, C, corresponding to s. Let H be the plane such that $C = H \cap \mathbb{S}^2$. Let $\mathbf{d} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ be the desired direction. Then the optimization problem is of the form:

Minimize $f(x, y, z) = A_E x + B_E y + C_E z$

Subject to $x^2 + y^2 + z^2 = 1$ ($\mathbf{d}$ lies on $\mathbb{S}^2$)
$ax + by + cz = d$ ($\mathbf{d}$ lies on H)
$a'x + b'y + c'z \geq 0$ ($\mathbf{d}$ and s lie on the same side of the plane defined by the points p, o and the origin)
$a''x + b''y + c''z \geq 0$ ($\mathbf{d}$ and s lie on the same side of the plane defined by the points q, o and the origin)

Thus, the four constraints ensure that $\mathbf{d}$ lies on s. This problem can be solved using the method of Lagrange Multipliers in constant time. Coupled with Theorem 3.1 we have:

Theorem 3.2 *A width-minimizing direction for an n-vertex polyhedron, where the directions are restricted to a convex polygon R on* $\mathbb{S}^2$ *composed of great and/or small arcs, can be found in* $O(n^2 + n \log |R| + |R|)$ *time.* ∎

3.2 The constrained largest empty disk problem on $\mathbb{S}^2$

Let S be a set of n sites on $\mathbb{S}^2$ and let R be a convex polygon on $\mathbb{S}^2$ which is bounded by great and/or small arcs. We wish to find a largest disk, D, on $\mathbb{S}^2$ which contains no site in its interior and whose center lies in R. The unconstrained problem can be solved in $O(n \log n)$ time using the Voronoi diagram, *Vor(S)* [18]. The following lemma characterizes the optimum center for the constrained problem.

Lemma 3.3 *The center,* $\mathbf{d}$, *of the largest empty disk, D, for a set, S, of sites on* $\mathbb{S}^2$, *where* $\mathbf{d}$ *is restricted to a convex polygon, R, composed of great and/or small arcs, satisfies one of the following:*

(i) *it coincides with a vertex of Vor(S) lying in the interior or on the boundary of R,*

(ii) *it is the intersection of an edge of Vor(S) with the boundary of R,*

(iii) *it coincides with a vertex of R,*

(iv) *it lies in the interior of an edge, $r \in R$ and in the interior of a face, $F \in Vor(S)$, and is the farthest point on $r \cap F$ from the site p_F associated with F.*

Proof Assume, for a contradiction, that $\mathbf{d}$ does not satisfy conditions (i)–(iv) of the lemma. Then the only possibilities for $\mathbf{d}$ are: (a) it is in the interior of a face, F, of *Vor(S)* and in the interior of R, or (b) in the interior of a face, F, of *Vor(S)* and in the interior of an edge, r, of R and it is not the farthest point in the interior of $r \cap F$ from the site associated with F, or (c) it is in the interior of an edge, e, of *Vor(S)* and in the interior of R.

Consider case (a). Let p_F be the site of S associated with F. Thus, p_F is the only site on the boundary of D. Since $\mathbf{d}$ is in the interior of both R and F, there is a point $\mathbf{d}' \in R$, on the great circle through p_F and $\mathbf{d}$, such that $\mathbf{d}'$ is further away from p_F than is $\mathbf{d}$; furthermore, the disk centered at $\mathbf{d}'$ is empty. This implies that $\mathbf{d}'$ is the center of an empty disk which is larger than D—a contradiction.

Next, consider case (b). Let p_F be the site associated with F. Then, there is another point $\mathbf{d}'$ in the interior of $r \cap F$ such that its distance from p_F is larger than the distance between $\mathbf{d}$ and p_F. Therefore, $\mathbf{d}'$ yields a larger empty disk than D.

Finally, consider case (c). Let p_1 and p_2 be the sites whose Voronoi faces share e. The great arc through p_1 and p_2 is perpendicular to (the great circle through) e. Moreover, p_1 and p_2 are the only sites on D's boundary. Clearly, there is a point $\mathbf{d}'$ on e, which is on one side of $\mathbf{d}$ or the other, such that $\mathbf{d}'$ is further away from p_1 and p_2 than is $\mathbf{d}$. Again, $\mathbf{d}'$ yields a larger empty disk than D. ∎

The following lemma is useful in identifying directions $\mathbf{d}$ that satisfy condition (iv) in Lemma 3.3.

Lemma 3.4 *Let r be an arc of a small/great circle and let p be a site on $\mathbb{S}^2$. Let C be the small/great circle containing r. Let G be the great circle determined by the intersection of $\mathbb{S}^2$ with the plane which passes through p and the origin and is perpendicular to the plane containing C. Then the point on r that is furthest from p is either an endpoint of r or an intersection point of G and r.*

Proof We first prove the result for the case where r is a small arc. Let o_C and r_C be the center and radius of C, respectively. We may assume, without loss of generality, that the plane containing C is perpendicular to the positive z-axis and that $p = (p_x, 0, p_z)$, where $p_x \geq 0$. Thus G is in the xz-plane. (See Figure 4. To avoid clutter, the arc r itself is not shown.) Of the two intersection points between G and C, let q be the one closer to p. Let h be the height of C above the xy-plane. Then q has coordinates $(r_C, 0, h)$. Let t be any point on C and let it make an angle α with $\overline{oq}$. The coordinates of t are $(r_C \cos\alpha, r_C \sin\alpha, h)$. The square of the Euclidean distance between p and t is $(r_C \cos\alpha - p_x)^2 + (r_C \sin\alpha)^2 + (h - p_z)^2 = {p_x}^2 + {r_C}^2 + (h - p_z)^2 - 2p_x r_C \cos\alpha$. Thus, as α increases from 0 to π (resp. from π to 2π), the Euclidean distance between p and t increases (resp., decreases) monotonically (or remains constant if $p_x = 0$). Clearly, the same is true of the distance between p and t as measured along $\mathbb{S}^2$.

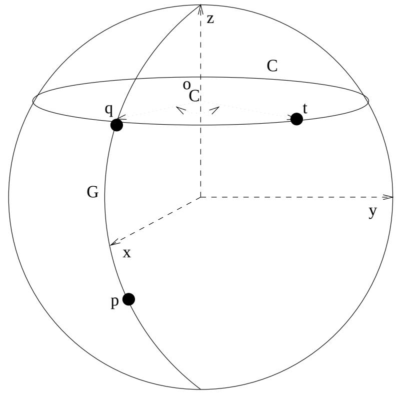

Figure 4: Finding the point on the small circle, C, which is furthest away from p.

Suppose that the endpoints of r make angles θ_1 and θ_2, respectively, with $\overline{oq}$. Clearly, if $\pi \notin [\theta_1, \theta_2]$, then the maximum distance between p and r is attained at one of the endpoints of r; otherwise, it is attained at the point $G \cap r$ for which $\alpha = \pi$.

The above discussion carries through unchanged if r is a great arc. (Note that in this case our assumption about the orientation of C implies that it is the equator.) ∎

The above discussion leads to the following result for finding a constrained largest empty disk.

Theorem 3.3 *For a set, S, of n sites on $\mathbb{S}^2$, a largest empty disk whose center is constrained to lie in a convex polygonal region R composed of great and/or small arcs can be found in time $O(n \log n + n \log |R| + |R| \log n)$.*

Proof We compute *Vor(S)* and preprocess it for point location queries; this takes $O(n \log n)$ time. There are $O(n)$, $O(n)$, and $O(|R|)$ candidate directions of types (i)–(iii) in Lemma 3.3. (There are only $O(n)$ candidate points of type (ii) since each great arc of *Vor(S)* can intersect R at most twice.) These can be identified in $O(n \log |R|)$ time. (For type (i) and type (ii) points we use binary search on R.) For type (i) and (ii) points, the corresponding empty caps can be identified in constant time apiece since they are determined by the sites associated with the Voronoi faces that share the Voronoi vertex (in type (i)) or Voronoi edge (in type (ii)). For type (iii), we do a point location query in *Vor(S)* for each vertex of R; this takes $O(|R| \log n)$ time.

To determine points of type (iv), we divide the boundary of R into segments such that each segment lies completely within a face of *Vor(S)*. This yields $O(n + |R|)$ segments. Let r be such a segment lying in face F and let p_F be the site associated with F. We need to find the point on r which is furthest from p_F. This point can be found in $O(1)$ time using Lemma 3.4. Thus, the candidate points of type (iv) can be found in $O(n + |R|)$ total time. The time bound follows. ∎

3.3 The constrained minimum support volume problem

We wish to find a **d** which minimizes the volume of supports required by a convex polyhedron $\mathcal{P}$ when **d** is restricted to lie in a convex polygon R whose boundary consists of great arcs and/or small arcs. The idea is to use the support volume minimization algorithm given in [16] on only those faces of $\mathcal{A}_{vol}$ (or portions thereof) that are also in R. Towards this end, we compute the overlay of R and $\mathcal{A}_{vol}$ in $O(n^2 + n \log |R| + |R|)$ time (Lemma 3.1(ii)). The arrangement of interest to us is the part of $\mathcal{A}_{vol}$ which is inside R. This has size $O(n^2 + |R|)$. Thus, the support volume minimization algorithm runs in $O(n^2 + |R|)$ time [16].

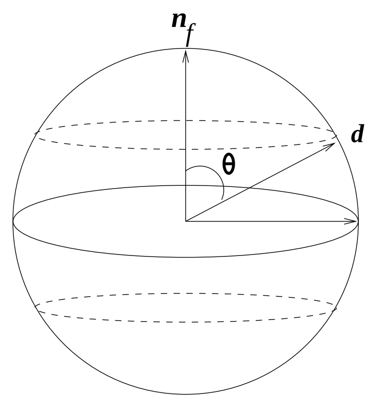

Figure 5: *The band corresponding to a facet f.*

Theorem 3.4 *A support volume-minimizing direction for a convex n-vertex polyhedron, where the directions are constrained to lie inside a convex polygonal region R on $\mathbb{S}^2$ composed of great and/or small arcs, can be found in $O(n^2 + n \log |R| + |R|)$ time.* ∎

4 Computing a build direction to reconcile multiple criteria

In this section we combine the results established in previous sections to solve several multi-criteria optimization problems arising in LM. Throughout, the polyhedron $\mathcal{P}$ is assumed to be convex if one of the criteria is "support volume" or "contact-area"; otherwise, $\mathcal{P}$ can be non-convex.

4.1 Threshold formulation

Recall from Section 1.3 that our problem is to find a direction $\mathbf{d}$ such that $\mathcal{C}_1(\mathbf{d}) \leq \rho_1$ and $\mathcal{C}_2(\mathbf{d}) \leq \rho_2$.

STAIR-STEP ERROR AND WIDTH: Suppose that $\mathcal{C}_1 = \mathcal{C}_{str}$ and $\mathcal{C}_2$ is $\mathcal{C}_{wid}$. We compute the directions $\mathbf{d}$ for which $\mathcal{C}_{str}(\mathbf{d}) \leq \rho_1$ as follows: We require that $\mathcal{C}_{str}(\mathbf{d}) = \max_f L \cos\theta_f(\mathbf{d})$ be at most ρ_1. That is, $L\cos\theta_f(\mathbf{d}) \leq \rho_1$ for all facets f, i.e., $\theta_f(\mathbf{d}) \geq \cos^{-1}(\rho_1/L)$ for all f. Let $\theta = \cos^{-1}(\rho_1/L)$. For each facet f, the directions $\mathbf{d}$ such that $\theta_f(\mathbf{d}) \geq \theta$ can be represented by the complement of the union of two open disks of radius θ and centers $\mathbf{n}_f$ and $-\mathbf{n}_f$. This complement, which we call a *band*, consists of two parallel small circles at a distance of $\pi/2 - \theta$ on either side of the great circle defined by $\mathbf{n}_f$ (Figure 5). Thus, the desired set of directions $\mathbf{d}$ for which $\mathcal{C}_{str}(\mathbf{d}) \leq \rho_1$ is the intersection of such bands for all facets $f \in \mathcal{P}$. This is a set of regions bounded by arcs of small circles. The total size of the regions is $O(n)$ and they can be computed in time $O(n \log n)$. To see this, consider, for each band, the two halfspaces whose bounding planes pass through the small circles of the band and which enclose it. The intersection of these halfspaces is a convex polyhedron of size $O(n)$. Then the desired regions are the intersections of this polyhedron with the boundary of $\mathbb{S}^2$. These have total size $O(n)$ because each vertex of a region is the intersection of an edge of the polyhedron with the sphere and there can be at most two such intersections per edge. The intersection of the halfspaces can be computed in time $O(n \log n)$ [18] and the resulting convex polyhedron can be intersected with $\mathbb{S}^2$ in $O(n)$ additional time.

We now solve the constrained minimum width problem within each region using Theorem 3.2 and check if the resulting width meets the threshold ρ_2. Since the total size of the regions is $O(n)$, and the regions are disjoint, the total time is $O(n^2)$.

A similar approach works if $\mathcal{C}_2$ is $\mathcal{C}_{vol}$ except that we solve the constrained minimum support volume problem in each region using Theorem 3.4.

Theorem 4.1 *For an n-vertex polyhedron $\mathcal{P}$, the threshold formulation for the pair $(\mathcal{C}_{str}, \mathcal{C}_{wid})$ can be solved in $O(n^2)$ time. If $\mathcal{P}$ is convex, then the problem for the pair $(\mathcal{C}_{str}, \mathcal{C}_{vol})$ can also be solved in $O(n^2)$ time.* ∎

<u>Support contact-area and width:</u> Suppose that $\mathcal{C}_1 = \mathcal{C}_{area}$ and $\mathcal{C}_2 = \mathcal{C}_{wid}$. We construct $\mathcal{A}_{area}$ and compute $\mathcal{C}_{area}(R)$ for each face R in $O(n^2)$ time (Section 2.2). For each face R such that $\mathcal{C}_{area}(R) \leq \rho_1$, we compute the direction $\mathbf{d}_R \in R$ which minimizes $\mathcal{C}_2$. If $\mathcal{C}_2(\mathbf{d}_R) \leq \rho_2$, then $\mathbf{d}_R$ satisfies both thresholds; if $\mathcal{C}_2(\mathbf{d}_R) > \rho_2$ for all R such that $\mathcal{C}_{area}(R) \leq \rho_1$, then no direction exists which satisfies both thresholds.

Minimizing $\mathcal{C}_2$ over R is merely the constrained minimum width problem (Theorem 3.1). A straightforward approach is to apply this theorem to each candidate face R in turn, which could take $O(n^3)$ time. However, there is a more efficient approach based on the observation that the candidate faces all come from the same arrangement, i.e., $\mathcal{A}_{area}$.

We compute the overlay, $\mathcal{A}$, of $\mathcal{A}_{area}$ and $\mathcal{A}_{wid}$. $\mathcal{A}$ has size $O(n^2)$ and can be computed in $O(n^2)$ time (Lemma 3.1(i)). $\mathcal{A}$ has three types of vertices: (i) vertices of $\mathcal{A}_{area}$, (ii) vertices of $\mathcal{A}_{wid}$, and (iii) intersections between arcs of $\mathcal{A}_{area}$ and $\mathcal{A}_{wid}$. This coupled with Lemma 3.2 implies that the directions $\mathbf{d}_R$ of interest, taken over all candidate faces, R, of $\mathcal{A}_{area}$, are the vertices of $\mathcal{A}$ that are in the interior or on the boundary of R. We can identify these vertices as follows:

Any region $R \in \mathcal{A}_{area}$ is divided into one or more regions R' in $\mathcal{A}$. Thus the candidate directions are boundary vertices of R'. As seen in Section 2.4, for any build direction $\mathbf{d} = \mathbf{x}i + \mathbf{y}j + \mathbf{z}k$ in R', the width is a linear function of x, y, and z, with the coefficients depending only on R'; we denote the function for the width in R' by $\mathcal{C}_{wid}(R')$. Furthermore, for any $\mathbf{d}$ in R', the contact-area is a constant; we denote this by $\mathcal{C}_{area}(R')$.

We visit all the faces of $\mathcal{A}$ in the order given by a depth-first search of the dual graph of $\mathcal{A}$. As we visit each face, $R' \in \mathcal{A}$, we update the value of $\mathcal{C}_{area}(R')$ and also the formula for $\mathcal{C}_{wid}(R')$. We can update these quantities incrementally, in $O(1)$ time, using the corresponding quantities from the previous face visited in the depth-first search. For each face R' such that $\mathcal{C}_{area}(R') \leq \rho_1$, we also compute $\mathcal{C}_{wid}(\mathbf{v})$ for each vertex, $\mathbf{v}$, on the boundary of R'. If $\mathcal{C}_{wid}(\mathbf{v}) \leq \rho_2$, then $\mathbf{v}$ satisfies both thresholds; if $\mathcal{C}_{wid}(\mathbf{v}) > \rho_2$ for all such $\mathbf{v}$, then no direction exists which satisfies both thresholds.

A similar approach works if $\mathcal{C}_2 = \mathcal{C}_{str}$ or $\mathcal{C}_2 = \mathcal{C}_{vol}$.

Theorem 4.2 *For an n-vertex convex polyhedron $\mathcal{P}$, the threshold formulation for pairs $(\mathcal{C}_{area}, \mathcal{C}_{wid})$, $(\mathcal{C}_{area}, \mathcal{C}_{str})$, and $(\mathcal{C}_{area}, \mathcal{C}_{vol})$ can be solved in $O(n^2)$ time.* ∎

4.2 Weighted formulation

Recall that here we wish to find a build direction such that $w_1\mathcal{C}_1 + w_2\mathcal{C}_2$ is minimum.

<u>Stair-step error and width:</u> Suppose that $\mathcal{C}_1 = \mathcal{C}_{str}$ and $\mathcal{C}_2 = \mathcal{C}_{wid}$. Let S be the set of sites on $\mathbb{S}^2$ corresponding to the facet normals of $\mathcal{P}$ and their negations, as defined in Section 2.1. We construct $Vor(S)$ and $\mathcal{A}_{wid}$, and compute their overlay, $\mathcal{A}$, in $O(n^2)$ time. Then we know (from Sections 2.1 and 2.4) that within a face, R, of $\mathcal{A}$, the formula for the stair-step error is of the form

$\mathcal{C}_{str} = A_R x + B_R y + C_R z$ and the formula for the width is of the form $\mathcal{C}_{wid} = A'_R x + B'_R y + C'_R z$. Therefore, within R, we need to minimize $w_1(A_R x + B_R y + C_R z) + w_2(A'_R x + B'_R y + C'_R z)$, which is again a linear function. This problem is similar to the optimization problem described in Section 2.3 (except that the objective function is now linear), and it can be solved in $O(n^2)$ total time. The formula can also be updated in a total time of $O(n^2)$ as we visit all the faces of $\mathcal{A}$ in a depth-first order.

We can use this approach to solve for other criteria combinations also. Note that $\mathcal{C}_{area}$ is a constant within a face of $\mathcal{A}_{area}$, while $\mathcal{C}_{vol}$ is a quadratic function within a face of $\mathcal{A}_{vol}$ (Section 2.3).

Theorem 4.3 *For an n-vertex polyhedron $\mathcal{P}$, the weighted formulation for pair $(\mathcal{C}_{str}, \mathcal{C}_{wid})$ can be solved in $O(n^2)$ time. If $\mathcal{P}$ is convex, then the weighted formulation for the other pairs can also be solved in $O(n^2)$ time.* ∎

4.3 Sequential formulation

Here we wish to find among all directions that minimize $\mathcal{C}_1$, the one that minimizes $\mathcal{C}_2$.

STAIR-STEP ERROR AND WIDTH: Consider the case $\mathcal{C}_1 = \mathcal{C}_{str}$ and $\mathcal{C}_2 = \mathcal{C}_{wid}$. Recall from Section 2.1 that the directions that minimize $\mathcal{C}_{str}$ must coincide with a vertex of *Vor(S)*. There could be $O(n)$ such directions. Among these directions, we need to find a $\mathbf{d}$ minimizing $\mathcal{C}_{wid}(\mathbf{d})$. We do this by computing and preprocessing $CH(\mathcal{P})$ into the hierarchical data structure described in [9], so that given any query direction the extreme vertices in that direction, and hence the width of $\mathcal{P}$ in that direction, can be found in $O(\log n)$ time. Thus, we have:

Theorem 4.4 *For an n-vertex polyhedron $\mathcal{P}$, the sequential formulation for pair $(\mathcal{C}_{str}, \mathcal{C}_{wid})$ can be solved in $O(n \log n)$ time.* ∎

STAIR-STEP ERROR AND SUPPORT VOLUME: Next, let $\mathcal{C}_1 = \mathcal{C}_{str}$ and $\mathcal{C}_2 = \mathcal{C}_{vol}$. We construct the arrangement $\mathcal{A}_{vol}$, and compute and store the formula for the volume of supports, $\mathcal{C}_{vol}$, for each face, R, in $\mathcal{A}_{vol}$ (Section 2.3). Then for each candidate direction, $\mathbf{d}$ we do a point location in $\mathcal{A}_{vol}$ to determine the face R and compute $\mathcal{C}_{vol}(\mathbf{d})$. We then find the overall minimum. All this can be done in $O(n^2)$ time ($O(n \log n)$ time to find the candidate directions, $O(n^2)$ time to compute $\mathcal{A}_{vol}$ and the formulae for $\mathcal{C}_{vol}$ within each face of $\mathcal{A}_{vol}$, and $O(n \log n)$ time for the $O(n)$ point location queries). Clearly, the same approach can be used to solve the problem if $\mathcal{C}_2 = \mathcal{C}_{area}$.

Theorem 4.5 *For an n-vertex convex polyhedron $\mathcal{P}$, the sequential formulation for pairs $(\mathcal{C}_{str}, \mathcal{C}_{vol})$ and $(\mathcal{C}_{str}, \mathcal{C}_{area})$ can be solved in $O(n^2)$ time.* ∎

WIDTH AND SUPPORT VOLUME: We could use the above approach for $(\mathcal{C}_{str}, \mathcal{C}_{vol})$ to also handle the pair $(\mathcal{C}_{wid}, \mathcal{C}_{vol})$. However, this will require $O(n^2 \log n)$ time as there could be $O(n^2)$ candidate directions (since each vertex of $\mathcal{A}_{wid}$ is a candidate direction). However, we can solve this problem in $O(n^2)$ time using a different approach, which does not require point-location. Let $\mathcal{A}$ be the overlay of $\mathcal{A}_{wid}$ and $\mathcal{A}_{vol}$. Note that the minimum width can occur only at vertices of $\mathcal{A}$ that are also vertices of $\mathcal{A}_{wid}$. We visit all the faces of $\mathcal{A}$ in the order given by a depth-first search of the dual graph of $\mathcal{A}$. As we visit each face, R, we update the width formula $\mathcal{C}_{wid}(R)$, for R, calculate $\mathcal{C}_{wid}$ at each vertex of R, and keep track of the minimum value $\mathcal{C}^*_{wid}$, of $\mathcal{C}_{wid}$. Next, we perform a second depth-first search of the dual of $\mathcal{A}$, this time keeping track of the volume formula, $\mathcal{C}_{vol}(R)$, for R. At each vertex, $\mathbf{v}$ of R, if $\mathcal{C}_{wid}(\mathbf{v}) = \mathcal{C}^*_{wid}$, then we calculate $\mathcal{C}_{vol}(\mathbf{v})$. We finally output that $\mathbf{v}$ for which $\mathcal{C}_{wid}(\mathbf{v}) = \mathcal{C}^*_{wid}$ and $\mathcal{C}_{vol}(\mathbf{v})$ is the minimum. All this can be done in $O(n^2)$ time. A similar approach can be used if $\mathcal{C}_2 = \mathcal{C}_{area}$ or $\mathcal{C}_2 = \mathcal{C}_{str}$, with the latter solution holding for any polyhedron. Therefore, we have:

Theorem 4.6 *For an n-vertex polyhedron $\mathcal{P}$, the sequential formulation for pair $(\mathcal{C}_{wid}, \mathcal{C}_{str})$ can be solved in $O(n^2)$ time. If $\mathcal{P}$ is convex, then the problem for pairs $(\mathcal{C}_{wid}, \mathcal{C}_{vol})$ and $(\mathcal{C}_{wid}, \mathcal{C}_{area})$ can be solved in $O(n^2)$ time.* ∎

<u>Support contact-area and width:</u> Let the minimum value of $\mathcal{C}_{area}$ be $\mathcal{C}^*_{area}$. We minimize $\mathcal{C}_{wid}$ within each face $R \in \mathcal{A}_{area}$ such that $\mathcal{C}_{area}(R) = \mathcal{C}^*_{area}$ and find the overall minimum. This problem can be solved in $O(n^2)$ time as discussed in Section 4.1 (Theorem 4.2). Clearly, a similar approach works if $\mathcal{C}_2 = \mathcal{C}_{str}$ or $\mathcal{C}_2 = \mathcal{C}_{vol}$.

Theorem 4.7 *For an n-vertex convex polyhedron $\mathcal{P}$, the sequential formulation for the pairs $(\mathcal{C}_{area}, \mathcal{C}_{wid})$, $(\mathcal{C}_{area}, \mathcal{C}_{str})$ and $(\mathcal{C}_{area}, \mathcal{C}_{vol})$ can be solved in $O(n^2)$ time.* ∎

5 Some experimental results

We have implemented a simplified version of our algorithm for the pairs $(\mathcal{C}_{str}, \mathcal{C}_{wid})$ and $(\mathcal{C}_{wid}, \mathcal{C}_{str})$, under the sequential formulation. We chose the criteria $\mathcal{C}_{str}$ and $\mathcal{C}_{wid}$ since our algorithms for these are applicable to any polyhedron. We have tested our implementation on several real-world STL models that we have obtained from our industrial partner, Stratasys, Inc.. Both implementations are written in C, run on an SGI 100MHz, R4000 processor machine, and do convex hull computations using the `qhull` routine (`www.geom.umn.edu/software/download/qhull.html`). We ran `qhull` with the specification that whenever the cosine of the angle between the normals of two neighboring facets of the convex hull was greater than 0.99, then the facets should be treated as co-planar and merged.

5.1 $(\mathcal{C}_{str}, \mathcal{C}_{wid})$ under sequential formulation

In this version, we have not implemented the hierarchical data structure of [9] to find the extreme vertices. Instead, we do this in time linear in the size of $CH(\mathcal{P})$, which is usually much smaller than that of $\mathcal{P}$ (as seen in Tables 1 and 2). Let S be the set of sites on $\mathbb{S}^2$ corresponding to the facet normals of $\mathcal{P}$ and their negations (Section 2.1). To compute $\mathcal{C}_{str}$, we compute $CH(S)$ and find the facet(s) closest to the origin; the normal from the origin to each such facet yields the center of a largest empty disk. Figure 6 shows the optimal orientation found by our algorithm for one of our test parts—a speedometer component for an automobile, containing 16,720 facets. The value of $\mathcal{C}_{str}$ is $0.968 \times L$ and that of $\mathcal{C}_{wid}$ is 3.33, where L is the layer thickness. By comparison, the width of this part (ignoring the stair-stepping criterion) is 1.89—about half as large as found above. For this part, the algorithm took about 9 seconds, excluding the time for graphical output.

Table 1 gives the results for all our test parts. (The sizes of the polyhedron and the convex hulls in the table are the number of facets; also the $\mathcal{C}_{str}$ values should be multiplied by the layer thickness L.) The test parts, chosen so as to encompass a variety of geometries, included (i) a box cover with

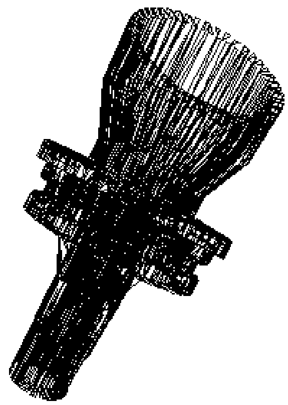

Figure 6: *Optimal orientation found by our algorithm for the pair* $(\mathcal{C}_{str}, \mathcal{C}_{wid})$*, under the sequential formulation. The coordinate axes have been chosen so that the* x*-axis points out of the paper, the* y*-axis points rightwards, and the* z*-axis points upwards. The part has been oriented so that the build direction is upwards, along the* z *axis. The part shown is a speedometer component for an automobile, with about 16,720 facets (of which only a third are displayed for clarity). Notice that the computed orientation is quite different from the natural orientation that one might expect, in which the long axis of the part is vertical. Indeed, in industry, this part is built along its long axis, usually with the narrow end down.*

$(\mathcal{C}_{str}, \mathcal{C}_{wid})$						
Part name	$\|\mathcal{P}\|$	$\|CH(\mathcal{P})\|$	$\|CH(S)\|$	*time (sec.)*	$\mathcal{C}_{str}$	$\mathcal{C}_{wid}$
`cover-5.stl`	906	13	1088	0.58	0.885	11.21
`cylinder.stl`	3600	37	110	1.14	0.706	2.81
`basex.stl`	5487	16	1448	1.97	0.822	5.22
`stlbin.stl`	9530	179	24494	10.17	0.958	3.14
`speedo.stl`	16720	67	17180	8.76	0.968	3.33

Table 1: *Results for the pair* $(\mathcal{C}_{str}, \mathcal{C}_{wid})$*, under the sequential formulation*

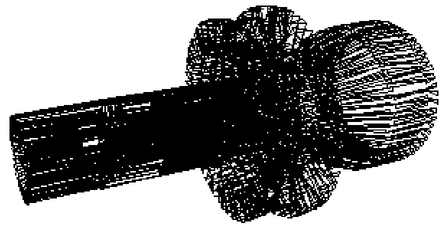

Figure 7: *Optimal orientation found by our algorithm for the pair* $(\mathcal{C}_{wid}, \mathcal{C}_{str})$*, under the sequential formulation. The part has been oriented so that the build direction is vertical, along the* z *axis.*

---

several sloping internal surfaces and one hole (`cover-5.stl`), (ii) a cylinder (`cylinder.stl`), (iii) a rectangular base for a part with numerous ridges (`basex.stl`), (iv) an electrical plug-like part, composed of a hemisphere and a cylinder with four holes and four protrusions (`stlbin.stl`), and (v) the speedometer component shown in Figure 6 (`speedo.stl`). One observation worth making is that for parts whose facet normals were distributed fairly uniformly over $\mathbb{S}^2$, the value of $\mathcal{C}_{str}$ approached the maximum of $1.00 \times L$. This is to be expected since the largest empty disk then becomes quite small.

5.2 $(\mathcal{C}_{wid}, \mathcal{C}_{str})$ under sequential formulation

We first compute the VF pairs by finding for each facet of $CH(\mathcal{P})$, the vertex farthest away from the plane through the facet. Next we compute $\mathcal{A}_u$ and $\mathcal{A}_l$ directly on $\mathbb{S}^2$ and find all EE pairs by computing edge intersections between $\mathcal{A}_u$ and $\mathcal{A}_l$. For each width-minimizing direction thus found, we compute $\mathcal{C}_{str}$ by checking each point in S, where S is the set of sites on $\mathbb{S}^2$ corresponding to the facet normals of $\mathcal{P}$ and their negations. Figure 7 shows the optimal orientation found by our algorithm for the speedometer part. The value of $\mathcal{C}_{wid}$ is 1.89 and that of $\mathcal{C}_{str}$ is $0.998 \times L$, while the minimum stair-step error (ignoring the width criterion) is $0.968 \times L$. For this part, the algorithm took about 7 seconds, excluding the time for graphical output. Table 2 gives the results for the same set of test parts as in Table 1. In each case, the width-minimizing direction is realized by a VF-pair of $CH(\mathcal{P})$ and the corresponding facet of $CH(\mathcal{P})$ is parallel (or nearly parallel) to a facet of $\mathcal{P}$, thereby yielding a value for $\mathcal{C}_{str}$ that is equal to (or close to) $1.00 \times L$.

6 Conclusion

We have presented efficient algorithms for several multicriteria optimization problems in Layered Manufacturing, under different formulations. These algorithms employ solutions to the single-criterion optimization problems as building blocks. We note that although we have described our results in the context of two criteria, they can be extended in a straightforward way to handle three or more criteria. In general, the running times are asymptotically the same as for two criteria, since the algorithms involve overlaying three or more arrangements of total size $O(n^2)$.

$(\mathcal{C}_{wid}, \mathcal{C}_{str})$					
Part name	$\vert\mathcal{P}\vert$	$\vert CH(\mathcal{P})\vert$	*time (sec.)*	$\mathcal{C}_{wid}$	$\mathcal{C}_{str}$
`cover-5.stl`	906	13	2.61	4.78	1.000
`cylinder.stl`	3600	37	3.30	1.98	0.997
`basex.stl`	5487	16	4.19	1.18	1.000
`stlbin.stl`	9530	179	5.77	2.48	1.000
`speedo.stl`	16720	67	7.27	1.89	0.998

Table 2: *Results for the pair $(\mathcal{C}_{wid}, \mathcal{C}_{str})$, under the sequential formulation*

Acknowledgement

We thank the software development group at Stratasys, Inc. (Larry Roscoe, Don Holzwarth, Jon Holt, Jeff Kotula, and Tom Studanski) for providing us with test data and for stimulating discussions.

References

[1] Agarwal, P.K. and Sharir, M. (1996). Efficient randomized algorithms for some geometric optimization problems, *Discrete & Computational Geometry*, **16**, 317–337.

[2] Allen, S. and Dutta, D. (1995). Determination and evaluation of support structures in layered manufacturing, *Journal of Design and Manufacturing*, **5**, 153–162.

[3] Allen, S. and Dutta, D. (1998). Wall thickness control in layered manufacturing for surfaces with closed slices, *Computational Geometry: Theory and Applications*, **10**, 223–238.

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

[5] Bose, P., van Kreveld, M., and Toussaint, G. (1993). Filling polyhedral molds, *Proceedings of the 3rd Workshop Algorithms Data Structures*, *Lecture Notes in Computer Science*, Springer-Verlag, **709**, 210–221.

[6] Chazelle, B., Edelsbrunner, H., Guibas, L., and Sharir, M. (1993). Diameter, width, closest line pair and parametric searching, *Discrete & Computational Geometry* **10**, 183–196.

[7] Cheng, W., Fu, J.Y.H., Nee, A.Y.C., Wong, Y.S., Loh, H.T., and Miyazawa, T. (1995). Multi-objective optimization of part-building orientation in stereolithography, *Rapid Prototyping Journal*, **1**:4, 12–23.

[8] Chazelle, B., Guibas, L., and Lee, D.T., (1985). The power of geometric duality, *BIT*, **25**, 76–90.

[9] Dobkin, D. and Kirkpatrick, D. (1985). A linear algorithm for determining the separation of convex polyhedra, *Journal of Algorithms*, **6**, 381–392.

[10] Edelsbrunner, H. and Guibas, L. (1989). Topologically sweeping an arrangement, *Journal of Computer and System Sciences*, **38**, 165–194.

[11] Frank, D. and Fadel, G. (1994). Preferred direction of build for rapid prototyping processes, *Proceedings of the 5th International Conference on Rapid Prototyping*, 191–200.

[12] Houle, M.E. and Toussaint, G. (1988). Computing the width of a set, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**:5, 761–765.

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

[14] Kim, J., Lee, K., and Park, J. (1994). Determination of optimal orientation in stereolithographic rapid prototyping, *Technical Report*, Dept. of Mechanical Design and Production Engineering, Seoul National University.

[15] Kulkarni, P. and Dutta, D. (1996). An accurate slicing procedure for layered manufacturing, *Computer-Aided Design*, **28**, 683–697.

[16] Majhi, J., Janardan, R., Smid, M., and Gupta, P. (1999). On some geometric optimization problems in layered manufacturing, *Computational Geometry: Theory and Applications*, **12**, 219–239.

[17] Majhi, J., Janardan, R., Schwerdt, J., Smid, M., and Gupta, P. (1999). Minimizing support structures and trapped area in two-dimensional layered manufacturing, *Computational Geometry: Theory and Applications*, **12**, 241–267.

[18] Preparata, F. and Shamos, M. (1988). *Computational Geometry–An Introduction.* Springer-Verlag.

[19] Schwerdt, J., Smid, M., and Janardan, R., and Johnson, E., and Majhi, J. (2000). Protecting critical facets in layered manufacturing, *Computational Geometry: Theory and Applications*, **16**, 187–210.

[20] Sreeram, P. and Dutta, D (1994). Determination of optimal orientation based on variable slicing thickness in layered manufacturing, *Technical Report UMEAM–94–14*, University of Michigan.