Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 97-055

Multi-criteria geometric
optimization problems in Layered
Manufacturing

by: Jayanth Majhi, Ravi Janardan,
Michiel Smid, and Jorg Schwerdt

Multi-criteria geometric optimization problems in Layered Manufacturing

Jayanth Majhi* Ravi Janardan* Michiel Smid† Jörg Schwerdt†

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. While the algorithms involving “support volume” or “contact area” apply only to convex models, the solutions for “surface finish” and “number of layers” are applicable 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 include construction and searching of certain arrangements on the unit-sphere, 3-dimensional convex hulls, Voronoi diagrams, point location, and hierarchical representations. Additionally, solutions are also provided, for the first time, for the *constrained* versions of two fundamental geometric problems, namely polyhedron width and largest empty disk on the unit-sphere.

1 Introduction

In *Layered Manufacturing* (LM), a physical prototype of a 3D object is built directly from its computer model on a small fabrication device attached to a workstation. The basic principle behind LM is simple: The model is oriented suitably and sliced into layers by horizontal planes. The layers are shipped over a network to the fabrication device which builds them in the vertical direction, each on top of the previous one. LM is the basis of an emerging technology called *Rapid Prototyping and Manufacturing* (RP&M). This technology, which is used extensively in the automotive, aerospace, and medical industries, accelerates dramatically the time it takes to bring a product to the market 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 [12]. Moreover, it opens up the possibility of directly building functional parts composed of multiple materials—something that is not possible via conventional manufacturing methods.

Figure 1 illustrates a well-known LM process called *StereoLithography* [12]. The input is a surface triangulation of the CAD model in a format called STL. The STL model is oriented suitably, sliced

*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A. E-mail: {majhi,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 stereolithography apparatus.

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.

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 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. For this 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. (See Figure 2.)

1.1 Geometric issues in LM

A key process-planning step in LM is choosing an orientation for the model so that it can be built in the vertical direction [12]. An equivalent viewpoint, of course, is to keep the model fixed and choose a (possibly non-vertical) direction in which to build the model, i.e., choose the *build direction*. For convenience, we will adopt this latter viewpoint. Among other things, the build direction affects the quantity of supports used and the surface finish—factors which impact the

speed and accuracy of the process. As a simple example, consider building the object in Figure 2: If it is built in the direction $\mathbf{d}$ indicated in the figure, then the manufactured solid will have a stair-stepped finish and will require supports under facet 14 (along its entire length normal to the paper). However, if the build direction is 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 would be enormous. In current systems, the build direction is often chosen by the human operator, based on experience, so that the number of layers and the amount of supports used 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, 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.

1.2 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 $\mathcal{P}$. (Note that there is no loss of generality in assuming a polyhedral model since the input—the STL representation—is polyhedral, even if the original CAD model is not.) We let $\mathbf{d}$ denote the build direction. The criteria of interest include:

Stair-step error (Criterion $\mathcal{C}_{str}$): Due to the non-zero slice thickness, the manufactured part will have a stair-stepped finish on any facet f that is not parallel to $\mathbf{d}$. The degree of stair-stepping on a facet f depends on the angle, $\theta_f(\mathbf{d})$, between the facet normal and $\mathbf{d}$, and it can be mitigated by a suitable choice of $\mathbf{d}$. In [3], the notion of an *error-triangle* for a facet is introduced as a way of quantifying stair-stepping. 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 both the build time and the cost of raw materials. If $\mathcal{P}$ is convex, then, for a build direction $\mathbf{d}$, the support volume, $\mathcal{C}_{vol}(\mathbf{d})$, is the volume of the region enclosed between $\mathcal{P}$ and the platform, i.e., the volume of the polyhedral “cylinder” which is bounded from below by the platform and from above¹ 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. (Figure 3 illustrates this; for convenience, the figure is drawn in 2D.)

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. If $\mathcal{P}$ is convex, then $\mathcal{C}_{area}(\mathbf{d})$ is the total area of the back facets. 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 certain *front facets*, i.e., facets whose outward normals make an angle of at most $\pi/2$ radians with $\mathbf{d}$.)

¹Throughout, we use the terms such as below/above w.r.t. $\mathbf{d}$.

Figure 3: *Supports for a non-convex polygon.*

Number of layers (Criterion C_{wid}): Since 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 $C_{wid}(\mathbf{d})$.

In previous work [15], 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 $C_{str}(\mathbf{d})$ in time $O(n \log n)$, for any simple polyhedron $\mathcal{P}$, and how to find a $\mathbf{d}$ which minimizes $C_{vol}(\mathbf{d})$ or $C_{area}(\mathbf{d})$ in $O(n^2)$ time for convex $\mathcal{P}$. Finally, we note that finding a $\mathbf{d}$ that minimizes $C_{wid}(\mathbf{d})$ is the well-studied *width* problem, which can be solved in $O(n^2)$ time in the worst case [11].²

1.3 Formulation of multi-criteria optimization problems

We investigate our problem under three formulations of multi-criteria optimization. We describe these below in the context of two generic criteria, C_1 and 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 $C_1(\mathbf{d}) \leq \rho_1$ and $C_2(\mathbf{d}) \leq \rho_2$, where $\rho_1 \geq 0$ and $\rho_2 \geq 0$ are designer-specified real numbers.

For instance, if C_1 and 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.

Weighted formulation: Find a build direction such that $w_1 C_1 + w_2 C_2$ 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 that minimize C_1 , find one which minimizes C_2 .

1.4 Contributions of this paper

Our work makes three main contributions:

²A faster randomized algorithm, which runs in roughly $O(n^{1.5})$ time is known [1] but it is not well suited for our purpose.

(1) A SUITE OF ALGORITHMS: We have developed a set of (over a dozen) algorithms that provably optimize different combinations of the above-mentioned criteria under the three formulations. To our knowledge, ours are the first such results for this important problem. All our algorithms run in $O(n^2)$ time, with the exception of one algorithm which runs in $O(n \log n)$ time and one which runs in $O(n^3)$ time. These solutions employ our earlier algorithms for single-criterion optimization [15] as building blocks. Moreover, the algorithms are modular in the sense that solutions for different combinations of criteria can be derived quickly by combining the appropriate single-criterion algorithms.

(2) IMPLEMENTATION AND TESTING ON REAL DATA: To demonstrate the practicality of some of our solutions, we have implemented two of our solutions and tested them on large, real-world STL files, of up to 17,000 facets, obtained from industry (Figures 8 and 9). These implementations are valid for any polyhedral model, even one with holes. The implementations are somewhat simplified versions of the asymptotically efficient algorithms described in the paper. Nevertheless, they run extremely fast, taking under 15 seconds on the largest model. By contrast, a straightforward, brute-force approach that we also implemented for comparison took nearly 20 minutes on the largest model.

(3) NEW RESULTS FOR CONSTRAINED GEOMETRIC PROBLEMS: In the course of solving the multi-criteria problems, we also provide—for the first time—algorithms for the constrained versions of two fundamental geometric problems, namely, computing the *width* of a polyhedron in 3-space and computing the *largest empty cap* on the unit-sphere, $\mathbb{S}^2$. 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 cap) is restricted to a polygonal region on $\mathbb{S}^2$. Previously, only the unconstrained width problem and the planar version of the largest empty cap (disk) problem had been investigated [11, 17]. We establish new geometric characterizations for these constrained problems that should be of independent interest.

We remark that our results also demonstrate the applicability and power of a wide range of advanced techniques from computational geometry in an applied setting. The geometric techniques that we bring into play include algorithms for constructing and searching various types of arrangements on the unit-sphere, 3-dimensional convex hulls, Voronoi diagrams, point location, and hierarchical representations of polyhedra. Additionally, our solutions also employ methods from continuous optimization such as Lagrange Multipliers.

A limitation of our work is that, at the present time, the solutions involving “support volume” or “contact area” as a criterion are applicable only to convex polyhedra. We are investigating these problems for simple 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 some work done on single-criterion optimization for LM (e.g., [2, 10, 15]), 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 [13, 18] 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.

Figure 4: The error triangle height for a facet f .

In the remainder of this paper, we review briefly the earlier results on single-criterion optimization (Section 2), describe our new results for constrained geometric problems (Section 3), and show how to combine these results to solve the multi-criteria problems (Section 4). We describe the implementation of two of our algorithms in Section 5 and we conclude in Section 6.

2 Single-criterion optimization—a quick review of results

Throughout, we will represent directions in space by points on the unit sphere, $\mathbf{S}^2$, and use the terms “direction” and “point” interchangeably.

2.1 Minimizing stair-step error

Given a simple polyhedron $\mathcal{P}$, we wish to find a direction $\mathbf{d}$ which minimizes the maximum error-triangle height, $C_{str}(\mathbf{d})$. 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 4.) We seek a direction $\mathbf{d}$ such that $C_{str}(\mathbf{d}) = \max_f L \cos \theta_f(\mathbf{d})$ is minimized, i.e., $\min_f \theta_f(\mathbf{d})$ is maximized.

Consider the set $S = \{\mathbf{n}_f \cap \mathbf{S}^2, -\mathbf{n}_f \cap \mathbf{S}^2 | f \text{ is a facet of } \mathcal{P}\}$ of points on $\mathbf{S}^2$. The size of S is $O(n)$ and we call each of its points a *site*. Our problem is to find a point $\mathbf{d}$ on $\mathbf{S}^2$ such that the minimum angle between it and the sites is maximized. Let us define a *cap* on $\mathbf{S}^2$, with *pole* $\mathbf{d}$ and *radius* θ , as the set of all points on $\mathbf{S}^2$ that are at a distance of at most θ from $\mathbf{d}$, as measured along the surface of $\mathbf{S}^2$. Clearly, our problem is to find the largest radius cap which does not contain any site in its interior, i.e., a *largest empty cap*; the pole of this cap is the build direction which minimizes the stair-step error.

Let $CH(S)$ be the convex hull of S . Then, as argued in [15], the plane bounding the largest empty cap contains a facet of $CH(S)$. Specifically, the normal from the origin to the plane containing the facet of $CH(S)$ that is closest to the origin is the desired optimum build direction. Thus it can be found in $O(n \log n)$ time. An alternate way of solving the problem is to construct the *Voronoi diagram* [17], $Vor(S)$, of S on $\mathbf{S}^2$. It can be shown that the pole of the largest empty cap must coincide with a vertex of $Vor(S)$. (The argument is similar to the one for the largest empty disk in the plane [17].)

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 cap with

its pole 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 $\mathbf{d} = xi + yj + zk$ 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 build direction $\mathbf{d}$ iff it is a back facet w.r.t. $\mathbf{d}$. Thus, the set of directions for which f needs support can be represented on $\mathbf{S}^2$ by an open hemisphere, h_f , whose pole is the point $-\mathbf{n}_f$. 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 $\mathcal{C}_{area}(R)$. If we assign to each h_f a weight equal to f 's area, then $\mathcal{C}_{area}(R)$ is simply the total weight of the h_f 's that cover R . Thus our problem is to find a face R of $\mathcal{A}_{area}$ which is covered by hemispheres of minimum total weight. In [15], we solve this problem by mapping it to the plane, via *central projection* [17], and then applying *topological sweep* [9]. The running time is $O(n^2)$. (In fact, the algorithm computes $\mathcal{C}_{area}(R)$ for each face R .)

2.3 Minimizing volume of supports

Assume $\mathcal{P}$ is convex. For a build direction $\mathbf{d}$, consider the vertex of $\mathcal{P}$ which is farthest away in direction $-\mathbf{d}$ (ties broken arbitrarily). We call this vertex the *extreme vertex* of $\mathcal{P}$, w.r.t. $\mathbf{d}$, and denote it by $v_{ext}(\mathbf{d})$. When $\mathcal{P}$ is built along $\mathbf{d}$, then $v_{ext}(\mathbf{d})$ is in contact with the platform.

For a given $\mathbf{d}$, the combinatorial structure of the supports is determined by the back facets and the extreme vertex of $\mathcal{P}$. Let $\mathcal{A}_{vol}$ be an arrangement on $\mathbf{S}^2$ 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 structure of the supports (but not their volume) is invariant within R .

In [15] we find a support volume-minimizing direction as follows: We compute $\mathcal{A}_{vol}$ as the overlay of two arrangements $\mathcal{A}_{area}$ and $\mathcal{A}_{ext}$. Faces of $\mathcal{A}_{area}$ represent directions for which the back facets are the same (Section 2.2). Faces of $\mathcal{A}_{ext}$ represent directions for which the extreme vertex is the same. $\mathcal{A}_{area}$ is an arrangement of great circles and can be computed in $O(n^2)$ time [6], while $\mathcal{A}_{ext}$ is an arrangement of great arcs and can also be computed in $O(n^2)$ time [4]. Their overlay can be computed in $O(n^2)$ additional time [15].

Next, we pick a face, R , of $\mathcal{A}_{vol}$ and generate a formula for the support volume for any direction $\mathbf{d} = xi + yj + zk$ in R , which is of the form $Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz + G$. Finding the optimal direction $\mathbf{d} \in R$ involves solving the following constrained minimization problem:

$$\text{Minimize } Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz$$

$$\begin{aligned} \text{Subject to } & x^2 + y^2 + z^2 = 1 \\ & a_i x + b_i y + c_i z \geq 0 \text{ for each great arc } r_i \text{ bounding } R, \end{aligned}$$

where $A-F$ are constants depending only on the back facets and the extreme vertex, while a_i-c_i are constants depending on r_i .

In [15], we solve this problem in $O(|R|)$ time, where $|R|$ is the number of edges on the boundary of R , using the method of Lagrangian Multipliers [14]. We also need to set up the volume formula for each face of $\mathcal{A}_{vol}$. Since doing this from scratch for each face is too expensive (taking $O(n^3)$ total time), we do it by visiting the faces in a depth-first order and updating the formula incrementally

from one face to the next. Specifically, if we visit a face R' from a face R , then either the extreme vertex changes or the set of back facets changes (in the sense that a non-back facet becomes a back facet or a back facet ceases to be one). As shown in [15], in each case we can update efficiently an appropriate term in the volume formula. The overall algorithm runs in $O(n^2)$ time.

2.4 Minimizing the number of layers

Assume $\mathcal{P}$ is a simple polyhedron. We review briefly Houle and Toussaint's *width* algorithm [11], which computes a direction $\mathbf{d}$ that minimizes $C_{wid}(\mathbf{d})$.

Let $CH(\mathcal{P})$ be the convex hull of $\mathcal{P}$. Vertices u and v of $CH(\mathcal{P})$ are called an *antipodal vertex-vertex pair* (or *VV pair*) if there exist parallel planes, one containing u and the other containing v , that enclose $CH(\mathcal{P})$. We say that the parallel planes *support* $CH(\mathcal{P})$. Similarly, one can define *antipodal vertex-edge (VE)*, *vertex-facet (VF)*, and *edge-edge (EE) pairs*. (Edge-facet and facet-facet pairs are subsumed by vertex-facet pairs, and hence not considered.) In [11] it is shown that the width-minimizing direction for $\mathcal{P}$ is perpendicular to the parallel planes associated with a *VF* or an *EE* pair. (The other cases do not sufficiently constrain the parallel planes and one can always find a direction in which to rotate them so as to reduce the width further.)

The *VF* and *EE* antipodal pairs can be determined by using the idea described in [5]. The algorithm in [5] divides $CH(\mathcal{P})$ into an *upper hull* and a *lower hull*, based on whether the outward normals of the facets point upward or downward (w.r.t. the positive z -axis). The upper and lower hulls are mapped to two subdivisions, $\mathcal{A}_u$ and $\mathcal{A}_l$, on $\mathbf{S}^2$. 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 subdivision are in 1-1 correspondence with the vertices of $CH(\mathcal{P})$. Given $\mathcal{A}_u$ and $\mathcal{A}_l$, the *VF* antipodal pairs are obtained by locating each vertex of $\mathcal{A}_u$ in a face of $\mathcal{A}_l$, and vice versa. The *EE* antipodal 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 *features* of $CH(\mathcal{P})$: two vertices, two edges, a vertex and a facet, or a vertex and an edge. Then, for any $\mathbf{d} \in E$, the same two features determine $C_{wid}(\mathbf{d})$. 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 $\mathbf{d} \in E$, $C_{wid}(\mathbf{d}) = (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{d}$ (assuming $\mathbf{u}_1$ is above $\mathbf{u}_2$ w.r.t. $\mathbf{d}$). Thus, $C_{wid}(\mathbf{d})$ is a linear function of the form $A_E x + B_E y + C_E z$, where $A_E - C_E$ depend only on E .

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 $\mathbf{S}^2$. These results will be used as subroutines in our multi-criteria algorithms in Section 4. We begin with two utility lemmas.

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

Proof 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 [6], 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. ■

Lemma 3.2 *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. 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.*

Proof $\mathcal{A}'$ has three types of vertices: (i) vertices of $\mathcal{A}$ inside/on R , (ii) vertices of R , and (iii) intersections between arcs of $\mathcal{A}$ and arcs of R . The number of vertices of type (i) is $O(n^2)$ and the number of vertices of type (ii) is $O(|R|)$. The number of vertices of type (iii) is $O(n)$ since each great arc of $\mathcal{A}$ can intersect R at most twice. Therefore, $\mathcal{A}'$ has size $O(n^2 + |R|)$. It can be computed as follows:

For any great arc a of $\mathcal{A}$, let G_a be the corresponding great circle. We start with the polygon R , and add the great circles one at a time. Let G_a be the current great circle. We compute the intersection (if any) of G_a and R in $O(\log |R|)$ time using binary search on R (type (iii) 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 (i) vertices). We retain only those intersections that are also in a . By the zone theorem [6], applied to G_a and the great circles, this takes $O(n)$ time per G_a . Thus, the overlay can be computed in $O(n^2 + n \log |R| + |R|)$ time. ■

3.1 The constrained width problem

In the *constrained width* problem, we wish to minimize $C_{wid}(\mathbf{d})$, where $\mathbf{d}$ is restricted to lie within a region, R , on $\mathbb{S}^2$. We assume that R is a convex polygon, whose boundary is composed of great arcs (see Figure 5). 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 3.3 *Let $\mathcal{P}$ be a simple polyhedron and let R be a convex polygon on $\mathbb{S}^2$. A width-minimizing direction, $\mathbf{d}$, for $\mathcal{P}$ 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 $CH(\mathcal{P})$.

Figure 5: The region R on $\mathbb{S}^2$.

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; w.l.o.g. 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 [11], one of the two rotations will yield a width less than $C_{wid}(\mathbf{d})$ —a contradiction. A similar discussion applies for a VV pair in case (a) or case (b). ■

Remark 3.1 This argument does not apply to parts (ii) and (iii) of the lemma because the width-reducing rotation may yield a direction outside R .

Lemma 3.4 *The candidate directions for the constrained minimum width problem are:*

- (i) *vertices of R ,*
- (ii) *vertices of $\mathcal{A}_{wid}$ lying in the interior of R or in the interior of an edge of R ,*
- (iii) *intersections of the interior of an edge of $\mathcal{A}_{wid}$ with the interior of an edge of R .*

Proof Follows from Lemma 3.3 and the properties of $\mathcal{A}_{wid}$ in Section 2.4. ■

Theorem 3.1 *A width-minimizing direction for a simple 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 points of types (i)–(iii) in Lemma 3.4 are, respectively, $O(|R|)$, $O(n^2)$, and $O(n)$ (since a great arc of $\mathcal{A}_{wid}$ can intersect a great arc of R at most twice). These points can be identified in $O(n^2 + n \log |R| + |R|)$ time (Lemma 3.2), which implies the same time bound for the constrained width problem. ■

Remark 3.2 This method can be extended to regions R composed of small arcs. The directions given by Lemma 3.3 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 [11]. 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:

$$\text{Minimize } f(x, y, z) = A_E x + B_E y + C_E z$$

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

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

Theorem 3.2 *A width-minimizing direction for a simple 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 cap problem

Let S be a set of n sites on $\mathbb{S}^2$ and let R be a convex polygon on $\mathbb{S}^2$ bounded by great and/or small arcs. In the *constrained largest empty cap* problem, we wish to find a largest cap, D , on $\mathbb{S}^2$ which contains no site in its interior and whose pole lies in R . In the plane, the unconstrained problem can be solved in $O(n \log n)$ time using the Voronoi diagram of the sites [17], and this extends to the unconstrained problem on $\mathbb{S}^2$ as well. The following lemma characterizes the constrained largest empty cap on $\mathbb{S}^2$. Let $\text{Vor}(S)$ denote the Voronoi diagram of the sites on $\mathbb{S}^2$.

Figure 6: Finding the point on the small circle, C , which is furthest away from p .

Lemma 3.5 *The pole, $\mathbf{d}$, of the largest empty cap, D , on $\mathbf{S}^2$, 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 $\text{Vor}(S)$ lying in the interior or on the boundary of R ,*
- (ii) *it is the intersection of an edge of $\text{Vor}(S)$ 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, $F \in \text{Vor}(S)$, and is the farthest point in the interior of $r \cap F$ from the site 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 $\text{Vor}(S)$ and in the interior of R , or (b) in the interior of a face, F , of $\text{Vor}(S)$ 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 $\text{Vor}(S)$ 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 R 's interior, 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 $\mathbf{d}$. This implies that $\mathbf{d}'$ is the pole of an empty cap 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 cap 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 cap than D . ■

The following lemma is useful in identifying directions $\mathbf{d}$ that satisfy condition (iv) in Lemma 3.5.

Lemma 3.6 *Let r be an arc of a small circle and let p be a site on $\mathbf{S}^2$. Then the distance between a point $t \in r$ and p is maximum when t is either one of the endpoints of r or is the intersection between r and the great circle defined by the plane passing through p , the origin, and the center of the small circle containing r .*

Proof Let C be the small circle containing r . Let o_C be the center and r_C the radius of C . Let G be the great circle defined by the intersection of $\mathbb{S}^2$ with the plane containing o_C , p , and the origin. W.l.o.g., we may assume that o_C is on the positive z -axis, that G is in the xz -plane, and $p = (p_x, 0, p_z)$, where $p_x \geq 0$. (See Figure 6.) 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 $\overline{o_C q}$. 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 $\mathbb{S}^2$.

Suppose that the endpoints of r make angles θ_1 and θ_2 , respectively, with $\overline{o_C q}$. 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 where $\alpha = \pi$. ■

Theorem 3.3 *For n sites on $\mathbb{S}^2$, a largest empty cap whose pole 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 $\text{Vor}(S)$ 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 3.5, and these can be identified in $O(n \log |R|)$ time. (There are only $O(n)$ candidate points of type (ii) since each great arc of $\text{Vor}(S)$ can intersect R at most twice. To identify 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 $\text{Vor}(S)$ 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 $\text{Vor}(S)$. We do this by picking any edge of R and testing this for intersection with each edge of $\text{Vor}(S)$. Each subsequent intersection is computed by walking along the face of $\text{Vor}(S)$ containing the edge involved in the previous intersection. This takes $O(n + |R|)$ time and 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 3.6. 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 [15] on only those faces of $\mathcal{A}_{\text{vol}}$ (or portions thereof) that are also in R . Towards this end, we compute the overlay of R and $\mathcal{A}_{\text{vol}}$ in $O(n^2 + n \log |R| + |R|)$ time (Lemma 3.2). The arrangement of interest to us is the part of $\mathcal{A}_{\text{vol}}$ which is inside R . This has size $O(n^2 + |R|)$. Thus, the support volume minimization algorithm runs in $O(n^2 + |R|)$ time [15]. Thus we have:

Theorem 3.4 *A support volume minimizing direction for a convex n -vertex polyhedron, where the directions are constrained to lie inside a 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. 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 that our problem is to find a direction $\mathbf{d}$ such that $\mathcal{C}_1(\mathbf{d}) \leq \rho_1$ and $\mathcal{C}_2(\mathbf{d}) \leq \rho_2$, where ρ_1 and ρ_2 are designer-specified real numbers.

SUPPORT CONTACT-AREA AND WIDTH: Suppose that $\mathcal{C}_1 = \mathcal{C}_{area}$ and $\mathcal{C}_2$ is $\mathcal{C}_{wid}$. We construct $\mathcal{A}_{area}$ and compute $\mathcal{C}_{area}(R)$ for each face R , in $O(n^2)$ total 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. 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). $\mathcal{A}$ 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 3.3 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' . We visit all the faces of $\mathcal{A}$ in the order given by a depth-first search (dfs) 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 width formula $\mathcal{C}_{wid}(R')$ for R' (Section 2.4). We can update these quantities incrementally, in $O(1)$ time, using the corresponding quantities from the previous face visited in the dfs. 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 4.1 *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. ■*

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 f , i.e., $\theta_f(\mathbf{d}) \geq \cos^{-1}(\rho_1/L)$ for all f . Let $\cos^{-1}(\rho_1/L) = \theta$. For each 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*,

Figure 7: The band corresponding to a facet f .

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 7). Thus, the desired set of directions $\mathbf{d}$ for which $C_{str}(\mathbf{d}) \leq \rho_1$ is the intersection of such bands for all facets $f \in \mathcal{P}$. This gives two antipodal convex polygons, R and R' , on $\mathbf{S}^2$, whose boundaries are composed of small arcs. Each polygon has size $O(n)$.

We now solve the constrained minimum width problem for R and R' using Theorem 3.1 and check if the resulting width meets the threshold ρ_2 . Since $|R| = |R'| = O(n)$, the total time is $O(n^2)$ in both cases.

A similar approach works if C_2 is C_{vol} except that we solve the constrained minimum support volume problem in R and R' (Theorem 3.4).

Theorem 4.2 *For an n -vertex simple polyhedron $\mathcal{P}$, the threshold formulation for pair (C_{str}, C_{wid}) can be solved in $O(n^2)$ time. If $\mathcal{P}$ is convex, then the problem for pair (C_{str}, C_{vol}) can be solved in $O(n^2)$ time.* ■

WIDTH AND SUPPORT VOLUME: Let $C_1 = C_{wid}$ and $C_2 = C_{vol}$. The idea is to identify a (possibly disconnected) region R on $\mathbf{S}^2$ representing directions $\mathbf{d}$ such that $C_{wid}(\mathbf{d}) \leq \rho_1$. Within each component of R we find the minimum support volume, using Theorem 3.4, and check the threshold ρ_2 .

We form R as follows: Let E be any element (vertex, edge interior, or face interior) of $\mathcal{A}_{wid}$. From Section 2.4, we know that for any $\mathbf{d} \in E$, $C_{wid}(\mathbf{d})$ can be written as $A_E x + B_E y + C_E z$. We want the subset of E for which $A_E x + B_E y + C_E z \leq \rho_1$. Let h_E be the halfspace $A_E x + B_E y + C_E z \leq \rho_1$. Thus we want the subset E' of E contained in $h_E \cap \mathbf{S}^2$. For each E , we can compute E' in $O(|E|)$ time. E' is either empty, a point, an edge, or a portion of a face of $\mathcal{A}_{wid}$, in which case it consists of great arcs and one small arc, or it consists of a face of $\mathcal{A}_{wid}$ with a hole (composed of a small circle). The total time to compute all the E' 's is $O(n^2)$. The desired region R is the union of the E' 's and it has size $O(n^2)$. Using the ideas in Theorems 3.2 and 3.4 (which carry through even though we now have a set of convex polygons and convex polygons with holes), we can compute a direction within R that minimizes the support volume in $O(n^2 + n \sum_R \log |E'| + \sum_R |E'|) = O(n^3)$ time. (Since R can have $O(n^2)$ components E , each of size $O(1)$, the term $n \sum_R \log |E|$ is $O(n^3)$.)

Theorem 4.3 *For an n -vertex convex polyhedron, the threshold formulation for the pair (C_{wid}, C_{vol}) can be solved in $O(n^3)$ time.* ■

4.2 Weighted formulation

Recall that here we wish to find a build direction such that $w_1C_1 + w_2C_2$ is minimum, where w_1 and w_2 are designer-specified real numbers.

STAIR-STEP ERROR AND WIDTH: Suppose that $C_1 = C_{str}$ and $C_2 = C_{wid}$. 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 $C_{str} = A_Rx + B_Ry + C_Rz$ and the formula for the width is of the form $C_{wid} = A'_Rx + B'_Ry + C'_Rz$. Therefore, within R , we need to minimize $w_1(A_Rx + B_Ry + C_Rz) + w_2(A'_Rx + B'_Ry + C'_Rz)$, 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 C_{area} is a constant within a face of $\mathcal{A}_{area}$, while C_{vol} is a quadratic function within a face of $\mathcal{A}_{vol}$ (Section 2.3).

Theorem 4.4 *For an n -vertex simple polyhedron $\mathcal{P}$, the weighted formulation for pair (C_{str}, 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 C_1 , the one that minimizes C_2 .

STAIR-STEP ERROR AND WIDTH: Let us consider the case $C_1 = C_{str}$ and $C_2 = C_{wid}$. Recall from Section 2.1 that the directions that minimize C_{str} must coincide with a vertex of $Vor(S)$. There could be $O(n)$ such directions and they can be found in $O(n \log n)$ time. Among these directions, we need to find a $\mathbf{d}$ minimizing $C_{wid}(\mathbf{d})$. We do this by computing and preprocessing $CH(\mathcal{P})$ into the hierarchical data structure described in [8], 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 4.5 *For an n -vertex simple polyhedron $\mathcal{P}$, the sequential formulation for pair (C_{str}, C_{wid}) can be solved in $O(n \log n)$ time.* ■

STAIR-STEP ERROR AND SUPPORT VOLUME: Next, suppose that $C_1 = C_{str}$ and $C_2 = C_{vol}$. We construct the arrangement $\mathcal{A}_{vol}$, and compute and store the formula for the volume of supports, C_{vol} , for each face, R , in $\mathcal{A}_{vol}$ (Section 2.3). Then for each candidate direction, $\mathbf{d}$, minimizing C_{str} , we do a point location in $\mathcal{A}_{vol}$ to determine the face R and compute $C_{vol}(\mathbf{d})$. 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 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 $C_2 = C_{area}$.

Theorem 4.6 *For an n -vertex convex polyhedron $\mathcal{P}$, the sequential formulation for pairs (C_{str}, C_{vol}) and (C_{str}, C_{area}) can be solved in $O(n^2)$ time.* ■

WIDTH AND SUPPORT VOLUME: We could use the above approach for (C_{str}, C_{vol}) to also handle the pair (C_{wid}, 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. 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 dfs 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 dfs 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 simple polyhedron. Therefore, we have:

Theorem 4.7 *For an n -vertex simple 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. ■*

SUPPORT CONTACT-AREA AND WIDTH: 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 4.1). Clearly, a similar approach works if $\mathcal{C}_2 = \mathcal{C}_{str}$ or $\mathcal{C}_2 = \mathcal{C}_{vol}$.

Theorem 4.8 *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 Implementation

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 criteria are applicable to simple polyhedra. We have tested our implementation on real-world STL models obtained from industry. Both implementations are written in C, run on an SGI Irix 5 machine, and find convex hulls using qhull (www.geom.umn.edu/software/download/qhull.html).

5.1 $(\mathcal{C}_{str}, \mathcal{C}_{wid})$ under sequential formulation (THEOREM 4.5)

In this version, we have not implemented the hierarchical data structure of [8] 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 $\mathcal{P}$. Figure 8 shows the optimal orientation found by our algorithm for our largest test part—a speedometer component for an automobile containing roughly 17,000 facets. (For this part, the convex hull had only about 600 facets.) The value of $\mathcal{C}_{str}$ is $0.96 \times L$ 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 15 seconds, excluding the time for graphical output.

Figure 8: Optimal orientation found by our algorithm for the pair (C_{str}, C_{wid}) , under the sequential formulation. The part has been oriented so that the build direction is vertical. The part shown is a speedometer component for an automobile, with about 17,000 facets (of which only a third are displayed for clarity).

Figure 9: Optimal orientation found by our algorithm for the pair (C_{wid}, C_{str}) , under the sequential formulation. The part has been oriented so that the build direction is vertical.

5.2 (C_{wid}, C_{str}) under sequential formulation (THEOREM 4.7)

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 $\mathbf{S}^2$ and find all EE pairs by computing edge intersections between $\mathcal{A}_u$ and $\mathcal{A}_l$. Figure 9 shows the optimal orientation found by our algorithm for the speedometer part. The value of C_{wid} is 1.89 and that of C_{str} is $1.00 \times L$, while the minimum stair-step error (ignoring the width criterion) is $0.96 \times L$. For this part, the algorithm took about 12 seconds, excluding the time for graphical output.

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. Some of our algorithms are practical, as demonstrated by our implementations. 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 $O(n^2)$.

References

- [1] P. K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. *Discrete & Computational Geometry*, vol. 16, pages 317–337, 1996.
- [2] S. Allen and D. Dutta. Determination and evaluation of support structures in layered manufacturing. *Journal of Design and Manufacturing*, 5:153–162, 1995.
- [3] 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 Conference*, 1995.
- [4] P. Bose, M. van Kreveld, and G. Toussaint. Filling polyhedral molds. In *Proc. 3rd Workshop Algorithms Data Structures*, volume 709 of *Lecture Notes in Computer Science*, pages 210–221. Springer-Verlag, 1993.
- [5] B. Chazelle, H. Edelsbrunner, L. Guibas and M. Sharir. Diameter, width, closest line pair and parametric searching. *Discrete & Computational Geometry*, vol. 10, pages 183–196, 1993.
- [6] B. Chazelle, L. J. Guibas and D. T. Lee. The power of geometric duality. *BIT*, vol. 25, pages 76–90, 1985.
- [7] W. Cheng, J.Y.H. Fu, A.Y.C. Nee, Y.S. Wong, H.T. Loh and T. Miyazawa. Multi-objective optimization of part-building orientation in stereolithography. *Rapid Prototyping Journal*, vol. 1, no. 4, pages 12–23, 1995.
- [8] D. P. Dobkin and D. G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, vol. 6, pages 381–392, 1985.

- [9] H. Edelsbrunner and L. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, vol 38, pages 165–194, 1989.
- [10] D. Frank and G. Fadel. Preferred direction of build for rapid prototyping processes. In *Proceedings of the 5th International Conference on Rapid Prototyping*, pages 191–200, 1994.
- [11] M.E. Houle and G.T. Toussaint. Computing the width of a set. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pages 761–765, 1988.
- [12] P. Jacobs. *Rapid Prototyping & Manufacturing: Fundamentals of StereoLithography*. McGraw-Hill, 1992.
- [13] J. Kim, K. Lee and J. Park, Determination of optimal orientation in stereolithographic rapid prototyping. *Tech. Report*, Dept. of Mechanical Design and Production Engineering, Seoul National University, July 1994.
- [14] D.G. Luenberger. *Introduction to linear and non-linear programming*. Addison-Wesley, 1973.
- [15] J. Majhi, R. Janardan, M. Smid and P. Gupta. On some geometric optimization problems in layered manufacturing. Preliminary version in *Proc. 5th Workshop on Algorithms and Data Structures*, vol. 1272 of Lecture Notes in Computer Science, pages 136–149, Springer-Verlag, 1997.
- [16] M. McKenna and R. Seidel. Finding the optimal shadows of a convex polytope. In *Proceedings of the 1st Annual ACM Symposium on Computational Geometry*, pages 24–28, 1985.
- [17] F.P. Preparata and M.I. Shamos. *Computational Geometry—An Introduction*. Springer-Verlag, 1988.
- [18] P. Sreeram and D. Dutta. Determination of optimal orientation based on variable slicing thickness in layered manufacturing. *Tech. Report UMEAM-94-14*, University of Michigan, 1994.