Arbitrarily Precise Computation of Gauss Maps and Visibility Sets for Freeform Surfaces [1, 2]

Gershon Elber[3]
Department of Computer Science
Technion, Israel Institute of Technology
Haifa 32000, Israel

Elaine Cohen
Department of Computer Science
University of Utah
Salt Lake City, UT 84112 USA

Abstract

The need to compute visibility and accessibility of surfaces occurs in a broad range of applications from computer aided design and manufacturing to computer graphics and vision. Surface-surface intersection is an essential task in modeling systems that support boolean operations. Recently, Gauss maps and visibility sets were shown [3, 4, 20, 21] to be helpful in robustly solving the above problems.

This paper presents a symbolic based method to both compute and exploit the Gauss map of a freeform surface or a model consists of several, possibly trimmed, freeform surfaces. Unlike other approaches to the computation of the Gauss map, the method presented here can be made arbitrarily precise for piecewise polynomial and rational surfaces.

The Gauss map is then employed to compute the set of views from which a freeform surface is completely locally visible.

[1] This work was supported in part by DARPA (N00014-92-J-4113). All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

[2] This work was supported in part by grant No. 92-00223 from the United States-Israel Binational Science Foundation (BSF), Jerusalem, Israel. However, opinions, conclusions or recommendations arising out of supported research activities are those of the author or the grantee and should not be presented as implying that they are the views of the BSF.

[3] While visiting at the University of Utah.

1 Introduction

The directions from which a surface is completely visible are of fundamental interest for many applications. In graphics, this problem is closely related to the hidden surface removal problem [8, 9]. In manufacturing, a surface can be machined only when it can be reached or "seen" by a tool. In [21], the directions from which a feature can be approached for machining purposes without interference is investigated.

The Gauss map of surface S [2, 17], $\mathcal{G}_s$, and the visibility set of surface S [3, 4, 20], $\mathcal{V}_s$, can be used for both shape recognition and matching since they provide a unique characterization of a surface. Throughout this paper, a subscript will denote the surface for which the map or the set is computed.

We first review some intuitive elementary aspects of visibility. For a point p on a surface S to be visible to an observer's eye located at a point E two tests must be satisfied. First, p must be "forward facing" and not on the "back side" of S, which can be determined by locally examining the surface. Second, p cannot be occluded by another part of S or by a different surface in the scene, that is, another surface lies along the ray from the eye to p but is closer to the eye than p. This last property can only be determined by examining global properties. We will more rigorously define the first requirement, and consider collections of such eye points.

Definition 1 *We define $\mathcal{G}_s$, the Gauss map of a surface S, to be a map from the surface S to the unit sphere, S^2, which takes each point $p \in S$ to its unit normal vector, $\vec{n_p}$. That is, $\vec{n_p} = \mathcal{G}_s(p)$.*

Definition 2 *Given a unit direction vector $\vec{v}$, we say that a point $p \in S$, is locally visible from direction $\vec{v}$ iff $\langle \vec{v}, \vec{n_p} \rangle \geq 0$.*

Denote by $Int(S)$ the interior of S. Then,

Definition 3 *Given a unit direction vector $\vec{v}$, we say that a point $p \in Int(S)$ has local neighborhood visibility from*

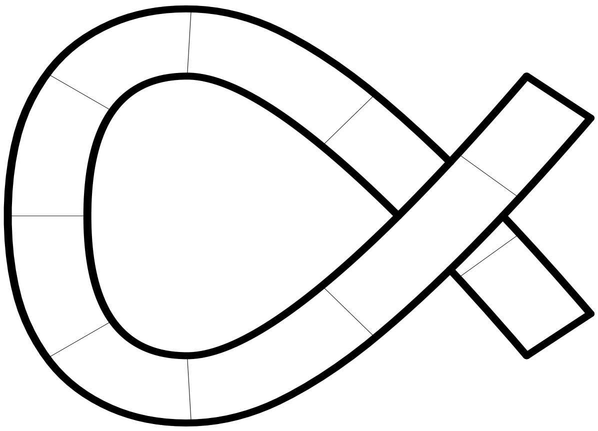

Figure 1: A surface can be locally visible, yet can hide itself globally by looping around itself.

direction $\vec{v}$ iff there exist a small open disk $\mathcal{D} \subset Int(S)$ such that each $q \in \mathcal{D}$ is locally visible from $\vec{v}$.

Definition 4 *The visibility set of surface S, $\mathcal{V}_s$, is the set of points on S^2 that correspond to directions from which every point $p \in Int(S)$ has local neighborhood visibility. That is, $\vec{v} \in S^2$ is also in $\mathcal{V}_s$ if for all $p \in Int(S)$, p has local neighborhood visibility from $\vec{v}$. We say that S is locally visible from each element of $\mathcal{V}_s$*

Definition 4 prescribes the set of directions from which the entire surface is locally visible. Those points are only globally visible if the surface does not loop around itself, as that in Figure 1, or if another surface is not closer to the eye.

A common and simple method to determine a bound on the normal directions of a given surface S is based upon the computation of a bounding cone [16, 19] of all possible normal directions of S. The normal cone is derived from the cross product relation between the two cones representing all the possible partial derivative directions. The normal cone bound of the image of the Gauss map of S is fairly loose and can provide insufficient information in many applications.

In this paper, a method is presented to compute an arbitrarily precise bound on the directions from which a surface is completely locally visible. That is, we present an approach to the computation of $\mathcal{G}_s$ and $\mathcal{V}_s$ that prescribes an arbitrary precise representation of the boundaries of $\mathcal{G}_s$ and $\mathcal{V}_s$, not a conical upper bound. Unlike existing schemes that exercise bounding cones [16, 19] or convex hull bounds using a mesh of the normal surface [15], the method presented herein can be easily extended to provide $\mathcal{G}_s$ and $\mathcal{V}_s$ of a *trimmed* surface or even of a set of several trimmed surfaces with arbitrary precision.

Section 2 discusses the necessary background. In section 3, we develop a method to compute the image of the Gauss map, $\mathcal{G}_s$, of a given surface S. $\mathcal{G}_s$, as a unit vector field [2], is represented and bounded by its boundary curves on the unit sphere, S^2.

One application that demonstrates the importance of the Gauss map is the computation of the visibility set of a surface. We follow the approach proposed in [3, 4] and generalize it to support freeform surfaces. Using the methods presented herein, the image of $\mathcal{G}_s$ is used to compute the visibility set, $\mathcal{V}_s$, of $S(u,v)$ in section 4. $\mathcal{V}_s$ bounds all the directions from which $S(u,v)$ is completely locally visible and is also represented by determining its boundary curves on S^2.

All figures in this paper were created from an implementation of the algorithms using the Alpha_1 solid modeler developed at the University of Utah.

2 Background

Given a regular C^1 surface $S(u,v)$, let $\mathcal{N}_s(u,v) = n_{S(u,v)}$ be the unit normal surface to $S(u,v)$. We use this new notation to emphasize that n_p, defined in Definition 1, is dependent on the parametrization. Thus,

$$\mathcal{N}_s(u,v) = \frac{\frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}}{\left\| \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v} \right\|}. \tag{1}$$

A surface is considered completely locally visible from direction $\vec{\alpha}$ if the following holds.

Definition 5 *Given a C^1 surface $S(u,v)$, and a vector $\vec{\alpha}$, $S(u,v)$ is completely locally visible from direction $\vec{\alpha}$ iff each point in S is locally visible from $\vec{\alpha}$.*

A surface S is *partially locally visible* from $\vec{\alpha}$ if only some of the points in S are locally visible from $\vec{\alpha}$. As defined, surface normals must point to the "inside" of the surface (object).

Using Definition 5, one can immediately see from Definition 4 that

Corollary 1

$$\mathcal{V}_s = \{\vec{\alpha} \mid \langle \vec{\alpha}, \mathcal{N}_s(u,v) \rangle \geq 0, \forall u, v, \ \|\vec{\alpha}\| = 1\}.$$

$\mathcal{V}_s \subset S^2$ since it contains only unit vectors. In this paper, we compute $\partial\mathcal{V}_s$, the boundary of $\mathcal{V}_s$, for a given surface $S(u,v)$. We assume that the surface is completely locally visible from at least one direction, i.e. $\mathcal{V}_s$ is not empty. This is obviously not always the case. S^2 itself, considered as a single surface, has no direction from which it is completely locally visible. However, a surface can always be subdivided and tested using loose bounding techniques, such as the bounding cones [16, 19], until this criterion holds.

Whenever possible, the unnormalized normal surface will be used instead of $\mathcal{N}_s(u,v)$. The unnormalized normal, referred to as $\hat{\mathcal{N}}_s$, is defined as,

$$\hat{\mathcal{N}}_s(u,v) = \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}, \tag{2}$$

which has a parametrization dependent magnitude. In the ensuing discussion, we symbolically compute the unnormalized normal surface of $\hat{\mathcal{N}}_s$, and refer to it as $\hat{\mathcal{N}}_s^2$. $S(u,v)$ is assumed to be regular, that is $\|\hat{\mathcal{N}}_s(u,v)\| \neq 0$, for all u, v, and sufficiently differentiable so that $\hat{\mathcal{N}}_s(u,v)$ and $\hat{\mathcal{N}}_s^2(u,v)$ are continuous.

Algorithms to symbolically compute the sum, difference, and product of B-spline or Bézier surfaces are well known [6, 7, 10, 18]. This paper will exercise symbolic computation and representation of surfaces, computed as scalar and vector fields that represent properties such as curvature [11], normals [8], offset error [13], or surface slopes and speed bounds [12]. For example, the Bézier or NURBs representations of $\hat{\mathcal{N}}_s(u,v)$ is a normal vector field that can be symbolically computed as the (cross) product of the partial derivatives of $S(u,v)$ (equation (2)).

Let $q \in \mathcal{R}^3$, $\|q\| \neq 0$. Let L be a ray from the origin $\mathcal{O}$ to q, and let q_{s^2} be the point where L intersects S^2.

Definition 6 *q is said to be centrally projected from $\mathcal{O}$ onto S^2 using L, and that q_{s^2} is its projected point. This mapping is called a* central *or* gnomonic *[5] projection.*

In the ensuing discussion and unless stated otherwise, $\mathcal{O}$ is assumed to be the center of the central projection. Let P be an arbitrary plane that does not contain $\mathcal{O}$, and q_p be the intersection point of L with P.

Definition 7 *q is said to be centrally projected onto a plane P using L, and that q_p is its projected point.*

In Definition 7, we notice that the perspective projection, used in computer graphics, is a special case of a central projection onto a plane.

Definition 8 *A closed curve, $C(t) \subset \mathcal{R}^3$, is said to be* centrally convex, *if there exist a plane P that does not contain $\mathcal{O}$ such that the central projection of $C(t)$ onto P is a closed convex curve. That is, the line segment connecting two arbitrary points on the central projection of $C(t)$ onto P is totally within the closed region of P bounded by the central projection of $C(t)$.*

3 Computation of the Image of the Gauss Map

Given an unnormalized normal surface $\hat{\mathcal{N}}_s(u,v)$ our goal is to compute the image of the Gauss map and represent the boundary of this image, $\partial\mathcal{G}_s$, on S^2. One can compute a loose bound on the image of the Gauss map using cones bounding the partials derivatives of S and then compute the bounding cone for the normal field as their cross product [16, 19]. Alternatively, one could compute $\hat{\mathcal{N}}_s(u,v)$ symbolically, centrally project it onto S^2 and compute a bound on the projected image. In [15], the control mesh of $\hat{\mathcal{N}}_s(u,v)$, represented as a Bézier surface, was centrally projected onto S^2 and used as a bound for the directions of the normals of the surface. In this section, we provide an arbitrarily precise computation of $\partial\mathcal{G}_s$ by detecting the exact extreme points of $\hat{\mathcal{N}}_s(u,v)$.

A surface displayed in an orthographic view can assume extreme values only along its boundary or its silhouette locations. Similarly, a surface centrally projected onto a plane can assume extreme values on the plane only along its boundary or its *central silhouette*. In a central projection, the viewing direction is a line through both the origin, $\mathcal{O}$, and the viewed point $S(u,v)$. Therefore, the central projection viewing direction of $S(u,v)$ is collinear with $S(u,v) - \mathcal{O} = S(u,v)$.

Definition 9 *The* central silhouette *set of S, $\mathcal{S}_s$, is the set of parameter values for which $\hat{\mathcal{N}}_s$ is perpendicular to the central projection viewing direction,*

$$\mathcal{S}_s = \left\{(u,v) \middle| \left\langle S(u,v), \hat{\mathcal{N}}_s(u,v)\right\rangle = 0\right\}. \tag{3}$$

To find the extreme values that $\hat{\mathcal{N}}_s$ can assume in a central projection, one needs to compute the boundaries and the central silhouette of $\hat{\mathcal{N}}_s$. At a silhouette point $\hat{\mathcal{N}}_s(u_0,v_0)$ (see Figure 3) the vector $\hat{\mathcal{N}}_s(u_0,v_0)$ is perpendicular to $\hat{\mathcal{N}}_s^2(u_0,v_0)$. In order to compute the silhouette curves of $\hat{\mathcal{N}}_s(u,v)$, one can compute the unnormalized normal vector field of $\hat{\mathcal{N}}_s(u,v)$, $\hat{\mathcal{N}}_s^2(u,v)$, and find the zero set of their inner product,

$$\mathcal{S} = \left\{(u,v) \mid \left\langle \hat{\mathcal{N}}_s(u,v), \hat{\mathcal{N}}_s^2(u,v)\right\rangle = 0\right\}, \tag{4}$$

to yield the set of all central silhouette points.

For a surface S, $\left\langle \hat{\mathcal{N}}_s, \hat{\mathcal{N}}_s^2\right\rangle$ can be symbolically computed and represented as a piecewise polynomial or rational scalar field [6, 7, 10, 18]. Once a representation is found, subdivision based techniques can be used to find an arbitrarily precise approximation to its zero set [10], as a set of piecewise linear curves.

It is necessary to compute square roots to find the projection of the boundary and central silhouette curves of $\hat{\mathcal{N}}_s$ onto S^2. The square root operation is not representable, in general, as (piecewise) polynomial or rational curves. As a result, an arbitrarily dense set of points can be determined using the central projection, a set that defines a piecewise linear approximation of the boundary and central silhouette curves projected onto S^2.

The smallest locally simply connected set that contains the boundary and the central silhouette curves of $\mathcal{N}_s$ is the exact boundary of $\mathcal{G}_s$. Thus, $\mathcal{G}_s$ can be computed using a two-dimensional boolean union operation on all the regions bounded by the extracted boundary and silhouette curves on S^2.

The set of boundary and silhouette curves of $\hat{\mathcal{N}}_s(u,v)$ may be topologically complex (see Figure 4), making the computation of this two-dimensional boolean union difficult. Fortunately, for the application demonstrated herein, namely the computation of the visibility set [3, 4], $\mathcal{V}_s$, it is unnecessary to compute the exact boundaries of $\mathcal{G}_s$ in order to find the exact boundaries of $\mathcal{V}_s$.

The approach to the computation of $\mathcal{V}_s$ from $\mathcal{G}_s$ generalizes the approach taken in [3, 4] to encompass freeform surface representations.

Lemma 1 *The directional domain from which a surface is completely visible, $\mathcal{V}_s$, is centrally* convex.

Proof: For a vector $\vec{\alpha} \in S^2$ there exists a hemisphere $\mathcal{H} \subset S^2$ from which a surface point with normal $\vec{\alpha}$ is locally visible (see Figure 5),

$$\mathcal{H} = \left\{\vec{\beta} \mid \|\vec{\beta}\| = 1, \left\langle \vec{\beta}, \vec{\alpha}\right\rangle \geq 0\right\}.$$

$\mathcal{H}$ holds all vectors in S^2 that have a nonnegative inner product with $\vec{\alpha}$ (Definition 5). For $S(u,v)$ to be visible from

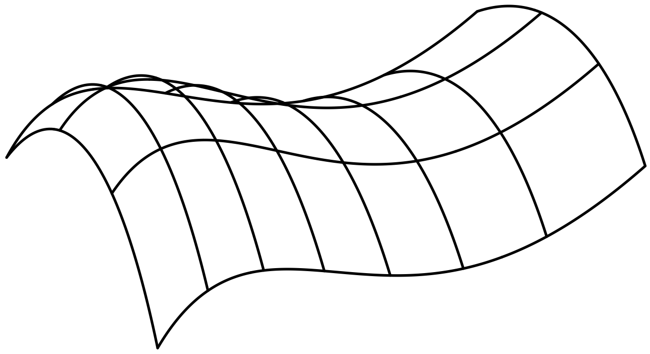

Figure 2: This simple bicubic surface is used throughout this paper as an example.

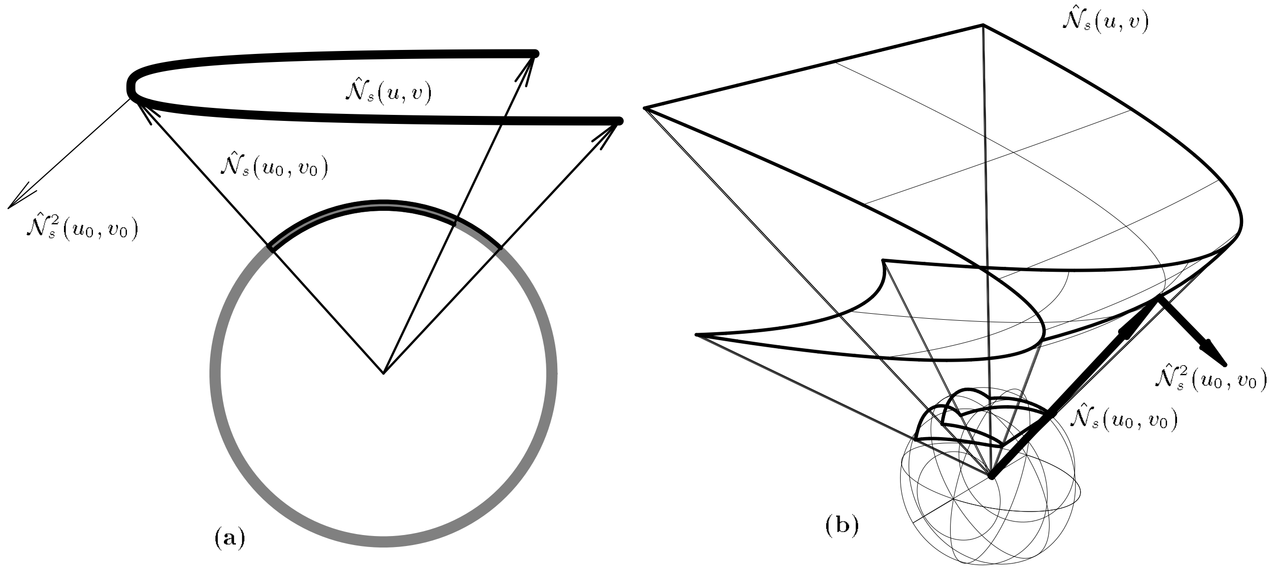

Figure 3: Central view extrema occur along boundaries and central silhouettes. Two-dimensional (a) and three-dimensional (b) cases are considered. The central silhouette set of $\hat{\mathcal{N}}_s(u,v)$ (for the surface $S(u,v)$ in Figure 2, in (b)) is equal to the zero set of $\left\langle \hat{\mathcal{N}}_s(u,v), \hat{\mathcal{N}}_s^2(u,v) \right\rangle$. At (u_0, v_0) we find such a central silhouette point.

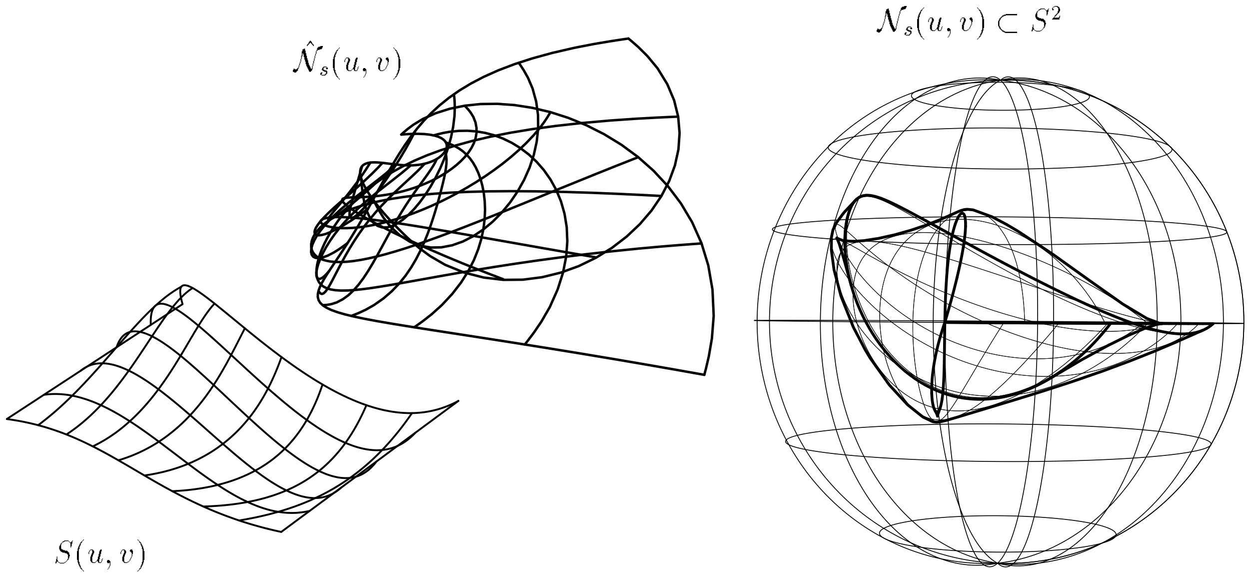

Figure 4: The silhouette set in the central projection of $\mathcal{N}_s(u,v)$ can be extremely complex. On the left, is a bicubic surface, with its complex normal surface, $\hat{\mathcal{N}}_s$, in the middle. The boundaries and central silhouette set of $\hat{\mathcal{N}}_s$ are projected on S^2, on the right.

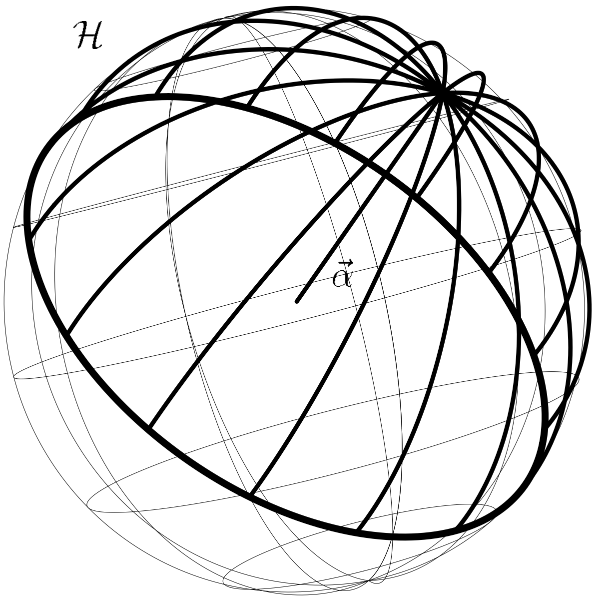

Figure 5: Each vector, $\vec{\alpha}$ is associated with a hemisphere $\mathcal{H} = \left\{\vec{\beta} \;\mid\; \|\vec{\beta}\| = 1, \left\langle \vec{\beta}, \vec{\alpha} \right\rangle \geq 0\right\}$.

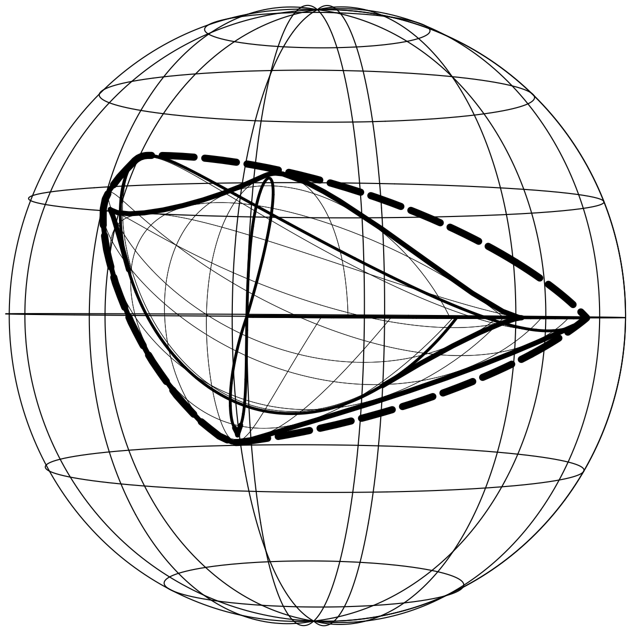

Figure 6: Central convex hull (dashed) of the silhouette and boundary curves of $\mathcal{N}_s$ in Figure 4.

a given direction $\vec{\alpha}$, $\vec{\alpha} \in \mathcal{H}$, $\forall \mathcal{H}$ of all vectors in $\mathcal{N}_s(u,v)$. That is, only a direction $\vec{\alpha}$ contained in (the intersection of) all hemispheres of all unit vectors in $\mathcal{N}_s(u,v)$ is in $\mathcal{V}_s$.

Centrally mapped onto a plane, a hemisphere is a half-plane. Because an intersection of convex regions is convex, the domain of directions from which a surface is completely visible, $\mathcal{V}_s$, is centrally convex as well. ■

Lemma 2 *Let R be a region of a cylinder. Then, $\mathcal{N}_r$ (and $\mathcal{G}_r$) is a great circle segment in S^2 and $\mathcal{V}_r$ is equal to the intersection of the two hemispheres associated with the two end vectors of $\mathcal{G}_r$ (Figure 7).*

Proof: Without loss of generality, one can assume R is a region of a cylinder around the z axis. That is, $R(u,v) = (acos(u), asin(u), v)$. The unit normal of region R is equal to $\mathcal{N}_r(u,v) = (-cos(u), -sin(u), 0)$ and is obviously a great circle segment on S^2 that is also in the xy plane. Call the great circle segment $\mathcal{C}$.

Let $\mathcal{H}$ be a hemisphere associated with a vector in $\mathcal{C}$. Let $\vec{\alpha_1}$ and $\vec{\alpha_2}$ be the two vectors of the end points of the arc of $\mathcal{C}$ and let $\mathcal{H}_1$ and $\mathcal{H}_2$ be the two hemispheres associated with $\vec{\alpha_1}$ and $\vec{\alpha_2}$, respectively. $\partial\mathcal{H}$, the great circle boundary of $\mathcal{H}$, is in a plane that always contains the z axis. Therefore, by moving the vector $\vec{\alpha}$ along $\mathcal{C}$ in the xy plane, the great circle $\partial\mathcal{H}$ rotates around the z axis. This transformation is continuous and monotone, when R is regular. Since $\vec{\alpha_1}$ and $\vec{\alpha_2}$ provide the two extreme locations and $\mathcal{H}_1$ and $\mathcal{H}_2$ form the hemispheres for the two extrema locations, the intersection of all hemispheres associated with a vector in $\mathcal{G}_r$ is equal to $\mathcal{H}_1 \bigcap \mathcal{H}_2$. ■

Corollary 2 *Suppose the Gauss map of S is two isolated points on S^2. Then, $\mathcal{V}_s$, is the same as if $\mathcal{G}_s$ was the great circle segment connecting the two vectors on S^2.*

Given $\mathcal{G}_s$, for any two vectors $\vec{\alpha_1}, \vec{\alpha_2} \in \mathcal{G}_s$, one can include a great circle segment between them, selecting the smaller one ($\vec{\alpha_1}$ and $\vec{\alpha_1}$ have at least two such segments, one that is greater then or equal to 180 degrees and one that is less than or equal to that), without affecting $\mathcal{V}_s$. Much like a planar convex hull, in which for each two points in the convex hull, the line segment between the two points is also in the convex hull, for each two points in a central convex hull on S^2, the great circle segment between them is also contained in the central convex hull:

Definition 10 *The* central convex hull*, $\mathcal{CH} \subset S^2$, of some set $\mathcal{S} \subset S^2$, is the smallest subset of S^2 containing $\mathcal{S}$ bounded by a centrally convex curve (Definition 8).*

The edges of $\mathcal{CH}$ are great circle segments, seen as straight lines from the origin. For any two points in $\mathcal{CH}$, the great circle segment on S^2 between them is completely in $\mathcal{CH}$. Specifically, for any two points in $\mathcal{G}_s$, one can add the great circle segment between them having no affect on the visibility set. Applying Corollary 2 allows one to employ the central convex hull of $\mathcal{G}_s$, $\overline{\mathcal{G}}_s$, instead of the exact and possibly topologically complex boundary of $\mathcal{G}_s$ (see Figures 4 and 6), while obtaining the *same tight bound* on the visibility set, $\mathcal{V}_s$.

The convex hull must be computed centrally. That is, edges of the convex hull are great circle segments on S^2 (see Figures 6 and 8). Optimal planar convex hull algorithms have been known for sometime [1, 14]. We cannot use planar convex hull algorithms directly, since we are interested in the central convex hull. A central projection from S^2 onto a plane (Definition 7) that maps great circles to straight lines is first employed. Then, we apply the planar convex hull algorithm to the two dimensional planar set only to centrally project the edges of the convex hull back onto S^2 as great circles.

$\mathcal{G}_s$ and $\mathcal{V}_s$ are required to strictly fit into a hemisphere so that the mapping to a plane is homeomorphic, a requirement equivalent to the constraint for the surface to be visible in the local from at least one direction. Satisfying this requirement always allows one to uniquely project great circle

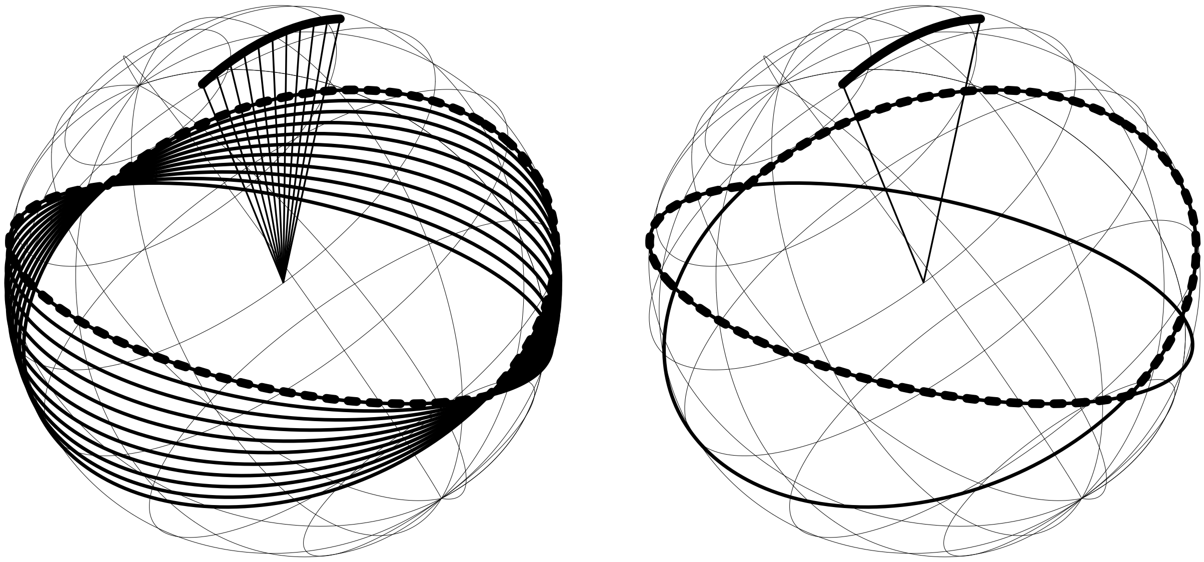

Figure 7: Let $\mathcal{G}_s$ be a great circle segment. Then, the boundary of $\mathcal{V}_s$ (dotted) is the boundary of the intersection of the two hemispheres associated with the end vectors of $\mathcal{G}_s$.

segments as line segments on the $Z = 1$ plane. It is necessary for neither $\mathcal{G}_s$ nor $\mathcal{V}_s$ to be in the $Z > 0$ hemisphere. Instead, a rotation, $\mathcal{R}$, could map $\mathcal{G}_s$ or $\mathcal{V}_s$ into the upper hemisphere. Further, if $\mathcal{P}$ is the plane through the origin that dichotomizes S^2 into two hemispheres, one strictly containing $\mathcal{G}_s$ (or $\mathcal{V}_s$), then $\mathcal{R}$ is the mapping that transforms the normal of $\mathcal{P}$ to the z axis. One can find $\mathcal{P}$ by using loose bounds such as bounding cones [16, 19] or by computing the three dimensional convex hull of $\mathcal{G}_s$ (or $\mathcal{V}_s$). Alternatively, an $O(n)$ algorithm to rotate points on a unit sphere as far away from the equator as possible, is described in [3].

Therefore, an algorithm to compute the central convex hull of $\mathcal{G}_s$ for surface $S(u,v)$ follows,

Algorithm 1

`Input:`
  $S(u,v)$`, input surface;`

`Output:`
  $\overline{\mathcal{G}}_s$`, central convex hull Gauss map of` $S(u,v)$`;`

`Algorithm:`
  $\hat{\mathcal{N}}_s(u,v)$ $\Leftarrow$ `unnormalized normal surface of` $S(u,v)$`;`
  $\mathcal{B}$ $\Leftarrow$ `Boundaries of` $\hat{\mathcal{N}}_s(u,v)$`, projected onto` S^2`;`
  $\mathcal{S}$ $\Leftarrow$ `Central silhouettes of` $\hat{\mathcal{N}}_s(u,v)$`, as the zero set of` $\left\langle \hat{\mathcal{N}}_s, \hat{\mathcal{N}}_s^2 \right\rangle$`, projected onto` S^2`;`
  $\mathcal{R}$ $\Leftarrow$ `Rotation to map` $\mathcal{B}$ `and` $\mathcal{S}$ `to the` $Z > 0$ `hemisphere;`
  $\overline{\mathcal{G}}_s^P$ $\Leftarrow$ `Central convex hull of` $\mathcal{R}(\mathcal{B} \cup \mathcal{S})$`;`
  $\overline{\mathcal{G}}_s$ $\Leftarrow$ $\mathcal{R}^{-1}\left(\overline{\mathcal{G}}_s^P\right)$`;`

In section 4, $\overline{\mathcal{G}}_s$ is used to compute the visibility set of $S(u,v)$, $\mathcal{V}_s$.

4 Computing the Visibility Set

$\overline{\mathcal{G}}_s$ is now used to compute a tight bound on $\mathcal{V}_s$.

Lemma 3 *The visibility set derived from $\overline{\mathcal{G}}_s$ is the same as the visibility set derived from the boundary of $\overline{\mathcal{G}}_s$, $\partial\overline{\mathcal{G}}_s$.*

Proof: Recall that $\overline{\mathcal{G}}_s$ is centrally convex. For each interior vector $\vec{\alpha} \in \overline{\mathcal{G}}_s$, there exists a great circle segment $\mathcal{C} \subset \overline{\mathcal{G}}_s$, such that $\vec{\alpha} \in \mathcal{C}$ and the two end vectors of $\mathcal{C}$ are in $\partial\overline{\mathcal{G}}_s$. Since the visibility set of $\mathcal{C}$ can be determined from its two end vectors in $\partial\overline{\mathcal{G}}_s$ by Lemma 2, we conclude that it is enough to examine only (the vectors in) $\partial\overline{\mathcal{G}}_s$ in order to determine visibility set from $\overline{\mathcal{G}}_s$ ■

The approximated boundary of $\overline{\mathcal{G}}_s$ is a set of great circle segments. By Lemma 2, it follows immediately that it is enough to examine the end vectors of these great circle segments in order to determine the visibility of $\overline{\mathcal{G}}_s$ (and $\mathcal{G}_s$). These end vectors are the vertices of $\partial\overline{\mathcal{G}}_s$.

Lemma 3 suggests an efficient way to compute $\mathcal{V}_s$ from $\overline{\mathcal{G}}_s$. Given the set of vertices of $\partial\overline{\mathcal{G}}_s$, $\mathcal{F}$, we associate a hemisphere $\mathcal{H}_i$ (Figure 5) with each vertex $f_i \in \mathcal{F}$. The intersections of all these hemispheres is $\mathcal{V}_s$:

$$\mathcal{V}_s = \bigcap_{f_i \in \mathcal{F}} \mathcal{H}_i \tag{5}$$

As in section 3, one can use the central projection and project these hemisphere boundaries (great circles) onto the plane $Z = 1$ from the origin. These great circles are mapped to lines in the plane ($\mathcal{H}_i^P$, the projection of $\mathcal{H}_i$, is now a half plane defined by a line). Furthermore, since $\overline{\mathcal{G}}_s$ is centrally convex, the intersection problem is reduced to intersecting each line in the plane only with its two neighbors, the previous line and the next line. Therefore, the complexity of this stage is linear (see Algorithm 1 for $\mathcal{R}$):

Algorithm 2

`Input:`

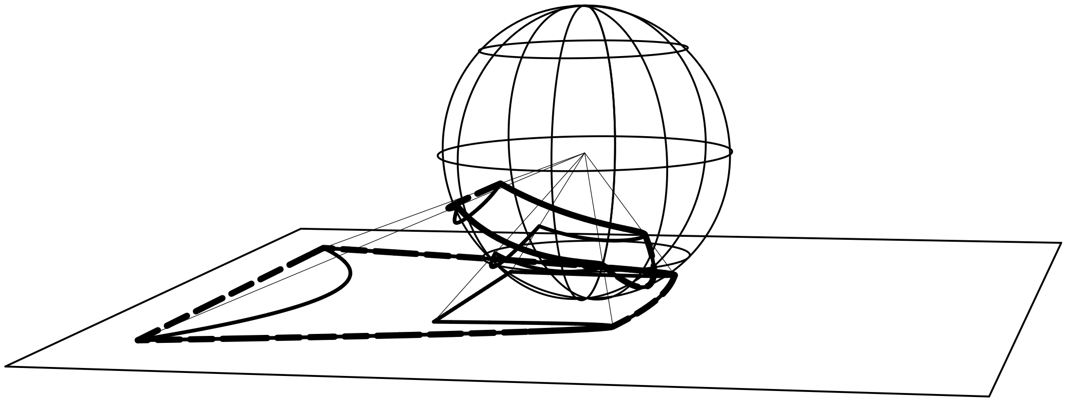

Figure 8: Convex hull (dashed) is computed for the projected great circles as lines in the plane $Z = 1$, using central projection (of surface in figure 2).

```
  𝒢̄_s, central convex hull of the image of 𝒢_s;

Output:
  𝒱_s, tightly bounded visibility set of S;

Algorithm:
  ℋ_i ⇐ hemispheres associated with vertices
                                   of ℛ(𝒢̄_s);
  ℋ_i^P ⇐ ∂ℋ_i centrally projected on plane
                                       Z = 1;
  𝒱_s^P ⇐ intersection set of ∂ℋ_i half planes;
  𝒱_s ⇐ ℛ^{-1}(𝒱_s^P) mapped back to S^2;
```

Figure 9 shows an example of this projection, the resulting $\mathcal{V}_s$ in the plane $Z = 1$, and mapped back onto S^2.

5 Extensions

Bounding cones based methods [16, 19] usually examine the entire parametric domain of the surface. Adopting these methods to support trimmed surfaces can be difficult. In contrast, the method presented herein can be easily extended to support trimmed surfaces. The boundary of the Gauss map will be computed for the *trimmed* boundary curves of $\hat{\mathcal{N}}_s(u, v)$, and its *trimmed* central silhouette curves. No other part of the algorithms developed herein need to be changed. Figure 10(a) shows a trimmed surface constructed from the surface in Figure 2. In Figure 10(b), the extracted silhouette and boundary curves of its Gauss map, its convex hull, and the visibility set (in dotted lines), are all shown mapped onto S^2. Figure 11(a) shows a trimmed region out of S^2. Obviously, the Gauss map of this surface is identical to the surface itself. In Figure 11(b) the Gauss map with its convex hull and the resulting visibility set are shown.

Given a model $\mathcal{M}$ consisting of several, possibly trimmed, surfaces one can now answer the query, whether or not there exists a direction from which all surfaces are visible. By computing $\mathcal{V}_s$ for each surface, such a direction exists iff $\bigcap_{s_i \in \mathcal{M}} \mathcal{V}_{s_i} \neq \phi$. Furthermore, any vector from the centrally convex set of $\bigcap_{s_i \in \mathcal{M}} \mathcal{V}_{s_i}$ can be exploited as a direction from which all surfaces of $\mathcal{M}$ are visible in the local.

6 Conclusion

An algorithm to provide a tight bound on the visibility directions of a given surface $S(u, v)$ is described, combined with an algorithm to compute bounds on the Gauss map of S. This bound is tight since it provides the (centrally convex) visibility set $\mathcal{V}_s$ of $S(u, v)$. The algorithm can be easily extended to provide a tight visibility bound for a trimmed surface, and/or a set of surfaces.

The symbolic computation has a fixed complexity, given the order and continuity of the surface. The complexity of the convex hull computation of the image of the Gauss map is $n log(n)$, where n is the number of vertices, which is the optimal time. The visibility set is computed in linear time from the Gauss map. The silhouette extraction (zero set of $\left\langle \hat{\mathcal{N}}_s(u, v), \hat{\mathcal{N}}_s^2(u, v) \right\rangle$) is the only numeric computation in this algorithm. Subdivision based techniques [10] were used to compute its zero set.

7 Acknowledgment

The authors are grateful to Rich Riesenfeld for his valuable remarks on the various drafts of this paper.

REFERENCES

[1] K. R. Andreson. A Reevaluation of an Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. Information Processing letters 7 (1978) 53-55.

[2] M. P. DoCarmo. Differential Geometry of Curves and Surfaces. Prentice-Hall 1976.

[3] L. L. Chen and T. C. Woo. Computational Geometry on the Sphere With applications to Automated Machining. Technical Report No. 89-30, Department of Industrial and Operations Engineering, University of Michigan, August 1989.

[4] L. L. Chen, Shuo-Yan Chou, and T. C. Woo. Seperating and Intersecting Spherical Polygons: Computing Machinability on Three-, Four- and Five-Axis Numerically Controlled Machines. ACM Transaction on Graphics, Vol. 12, No. 4, pp 305-326, October 1993.

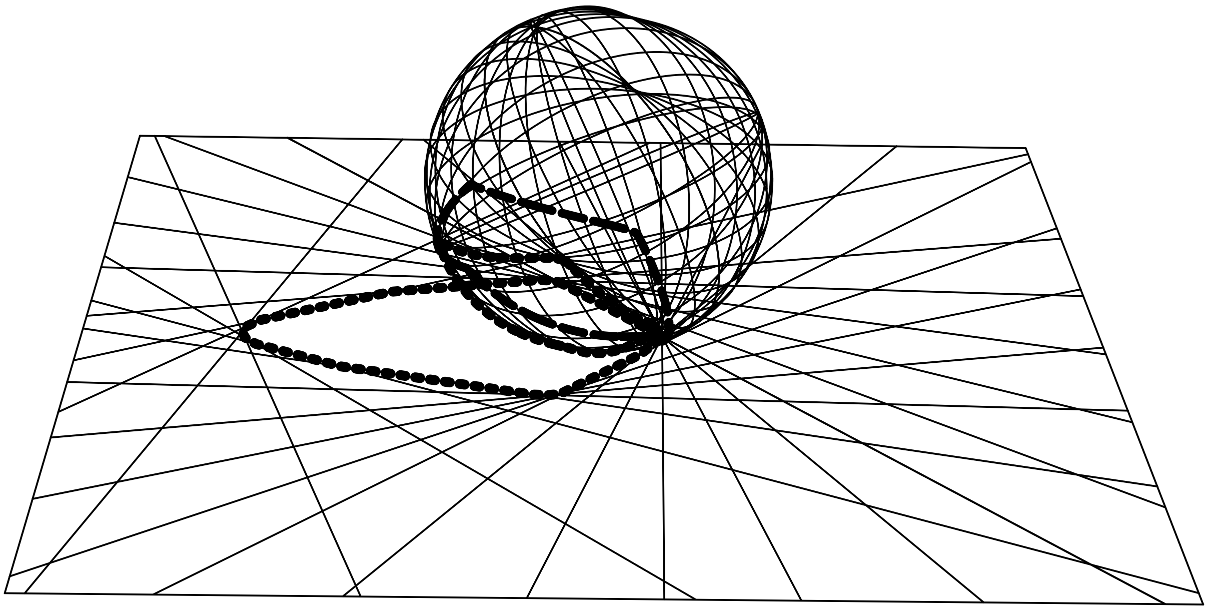

Figure 9: $\mathcal{V}_s$ (dotted) is computed as intersection of hemisphere boundaries (great circles) of $\overline{\mathcal{G}}_s$ vertices (dashed) projected as lines onto the plane $Z = 1$ and mapped back (of surface in Figure 2).

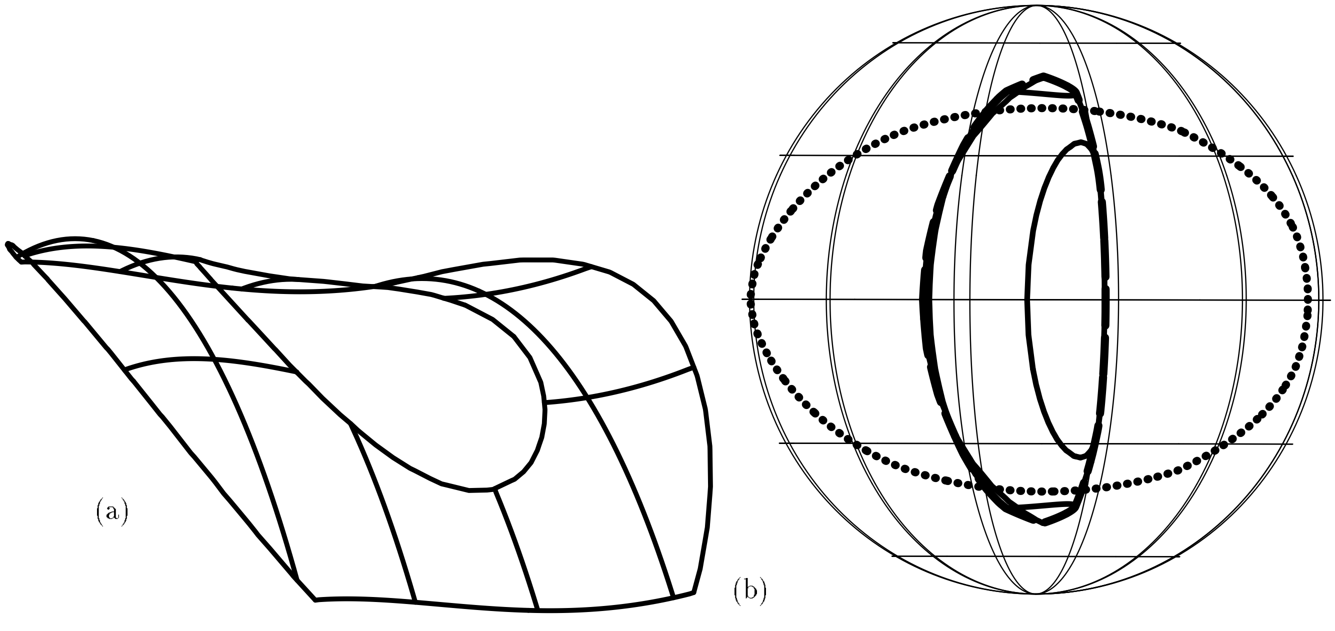

Figure 10: A trimmed region from the surface in Figure 2 is shown in (a). In (b), shown are its $\overline{\mathcal{G}}_s$ (dashed), the silhouette and boundary curves of its $\mathcal{N}_s(u,v)$, and its $\mathcal{V}_s$ (dotted).

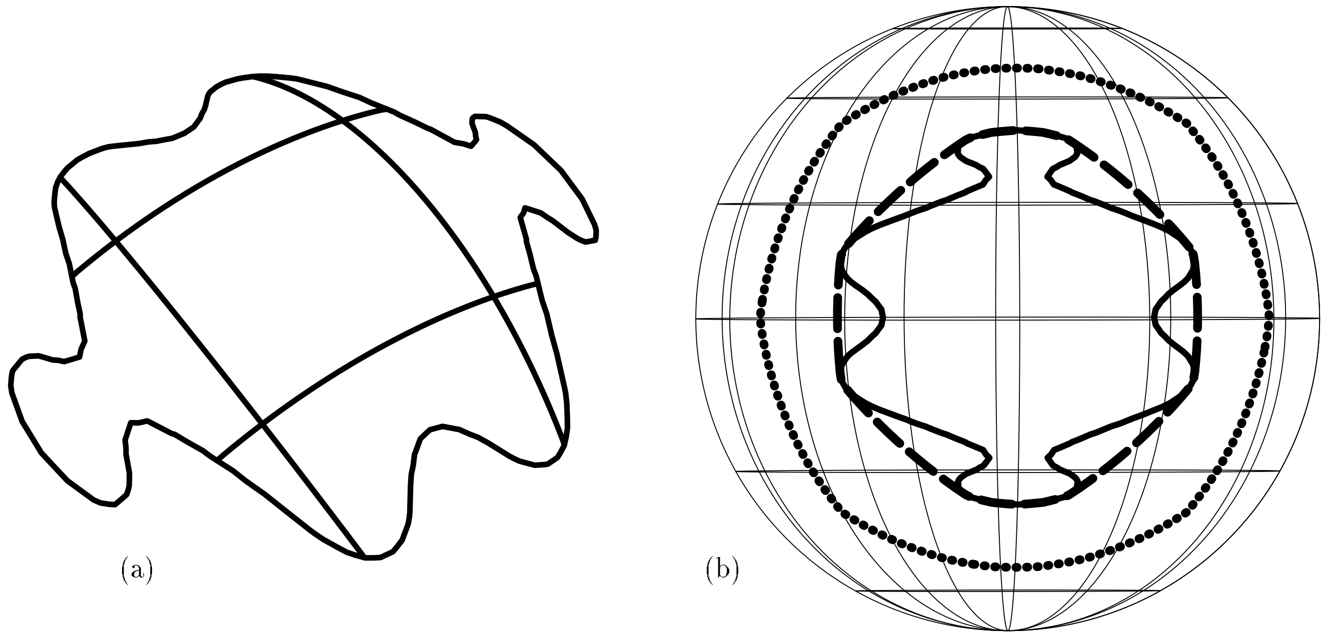

Figure 11: In (a), is a trimmed region from S^2. In (b), is its $\mathcal{G}_s$, $\overline{\mathcal{G}}_s$ (dashed), and $\mathcal{V}_s$ (dotted).

[5] S. Iyanaga, Y. Kawada and K. O. May. Encyclopedic Dictionary of Mathematics. The MIT Press Cambridge, Massachusetts, and London, England.

[6] G. Farin. Curves and Surfaces for Computer Aided Geometric Design Academic Press, Inc. Third Edition 1993.

[7] R. T. Farouki and V. T. Rajan. Algorithms For Polynomials In Bernstein Form. Computer Aided Geometric Design 5, pp 1-26, 1988.

[8] G. Elber and E. Cohen. Hidden Curve Removal for Untrimmed and Trimmed NURB Surfaces. Technical Report No. 89-019, Computer Science, University of Utah.

[9] G. Elber and E. Cohen. Hidden Curve Removal for Free Form Surfaces. SIGGRAPH 90, pp 95-104.

[10] G. Elber. Free Form Surface Analysis using a Hybrid of Symbolic and Numeric Computation. Ph.D. thesis, University of Utah, Computer Science Department, 1992.

[11] G. Elber and E. Cohen. Second Order Surface Analysis Using Hybrid Symbolic and Numeric Operators. ACM Transaction on Graphics, Vol. 12, No. 2, pp 160-178, April 1993.

[12] G. Elber and E. Cohen. Hybrid Symbolic and Numeric Operators as Tools for Analysis of Freeform Surfaces. Modeling in Computer Graphics, B. Falcidieno and T. L. Kunii (Eds.), Working Conference on Geometric Modeling in Computer Graphics (IFIP TC 5/WG 5.10), Genova 1993.

[13] G. Elber and E. Cohen. Error Bounded Variable Distance Offset Operator for Free Form Curves and Surfaces. International Journal of Computational Geometry and Applications, Vol. 1., No. 1, pp 67-78, March 1991.

[14] R. L. Graham. An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. Information Processing letters 1 (1972) 132-133.

[15] M. E. Hohmeyer. A Surface Intersection Algorithm Based on Loop Detection. Symposium in Solid Modeling Foundations and CAD/CAM Applications, Austin, Texas, June 5-7, 1991.

[16] D. S. Kim. Cones on Bezier Curves and Surfaces. Ph.D. dissertation, University of Michigan, 1990.

[17] R. S. Millman and G. D. Parker. Elements of Differential Geometry. Prentice-Hall Inc. 1977.

[18] K. Morken. Some Identities for Products and Degree Raising of Splines. Constructive Approximation 7, pp 195-209, 1991.

[19] T. W. Sederberg and Ray J. Meyers. Loop Detection in Surface Patch Intersections. Computer Aided Geometric Design 5, pp 161-171, 1988.

[20] S. H. Suh and J. K. Kang. Process Planning for Multi-Axis NC Machining of Free Surfaces. To Appear in IJPR.

[21] Y. J. Tseng and S. Joshi. Determining Feasible Tool-approach Directions for Machining Bezier Curves and Surfaces. Computer Aided Design, Vol. 23, No. 5, pp 367-378, June 1991.