Exact Implementation of Arrangements of Geodesic Arcs on the Sphere with Applications[*]

Efi Fogel[†] Ophir Setter[†] Dan Halperin[†]

Abstract

Recently, the Arrangement_2 package of CGAL, the Computational Geometry Algorithms Library, has been greatly extended to support arrangements of curves embedded on two-dimensional parametric surfaces. The general framework for sweeping a set of curves embedded on a two-dimensional parametric surface was introduced in [3]. In this paper we concentrate on the specific algorithms and implementation details involved in the exact construction and maintenance of arrangements induced by arcs of great circles embedded on the sphere, also known as geodesic arcs, and on the exact computation of Voronoi diagrams on the sphere, the bisectors of which are geodesic arcs. This class of Voronoi diagrams includes the subclass of Voronoi diagrams of points and its generalization, power diagrams, also known as Laguerre Voronoi diagrams. The resulting diagrams are represented as arrangements, and can be passed as input to consecutive operations supported by the Arrangement_2 package and its derivatives. The implementation is complete in the sense that it handles degenerate input, and it produces exact results. An example that uses real world data is included. Additional material is available at http://www.cs.tau.ac.il/~efif/VOS.

1 Introduction

Given a finite collection $\mathcal{C}$ of geometric objects (such as lines, planes, or spheres) the *arrangement* $\mathcal{A}(\mathcal{C})$ is the subdivision of the space where these objects reside into cells as induced by the objects in $\mathcal{C}$. In this paper we concentrate on the particular class of arrangements, where the embedding space is the sphere, and the inducing objects are geodesic arcs. There is an analogy between this class of arrangements and the class of planar arrangements induced by linear curves (i.e., segments, rays, and lines), as properties of linear curves in the plane can be often, (but not always), adapted to geodesic arcs on the sphere. The ability to robustly construct arrangements of geodesic arcs on the sphere, and carry out exact operations on them using only (exact) rational arithmetic is a key property that enables an efficient implementation.

Recently, a software package that computes exact arrangements of general circles on the sphere was introduced [5]. The extended Arrangement_2 package was used to compute arrangements on quadrics [3] and on Dupin cyclides [4], which contain the torus as a special case. The technique to compute Voronoi diagrams on two-dimensional parametric surfaces described in this paper can be applied to these surfaces as well, conditioned on the ability to handle bisectors of sites embedded on these surfaces.

Voronoi diagrams were thoroughly investigated and were used to solve many geometric problems [1, 17]. One of the interesting properties observed about this decomposition of a space is its strong connection to arrangements [6], a property that yields a very general approach for computing Voronoi diagrams.

The concept of computing cells of points that are closer to a certain object than to any other object, among finite number of objects, was extended to various kinds of geometric sites, ambient spaces, and distance functions, e.g., power diagrams of circles in the plane, multiplicatively weighted Voronoi diagrams, additively weighted Voronoi diagrams [1, 2, 17]. One immediate extension is computing Voronoi diagrams on two-dimensional parametric surfaces [12] in general, and on the sphere [15, 16] in particular.

2 Arrangements on Surfaces

A parameterized surface S is defined by a function $f_S : \mathbb{P} \to \mathbb{R}^3$, where the domain $\mathbb{P} = U \times V$ is a rectangular two-dimensional parameter space with bottom, top, left, and right boundaries, and the range f_S is a continuous function. We allow $U = [u_{\min}, u_{\max}]$, $U = [u_{\min}, +\infty)$, $U = (-\infty, u_{\max}]$, or $U = (-\infty, +\infty)$, and similarly for V. A *contraction point* $p \in S$ is a singular point, which is the mapping of a whole boundary of the domain $\mathbb{P}$. For example, if the top boundary is contracted, we have $\forall u \in U, f_S(u, v_{\max}) = p'$ for some fixed point $p' \in \mathbb{R}^3$. An *identification curve* $C \subset S$ is a continuous curve, which is the mapping of opposite closed boundaries of the domain $\mathbb{P}$. For example, if the left and right boundaries are identified, we have

[*]This work has been supported in part by the IST Programme of the EU as Shared-cost RTD (FET Open) Project under Contract No IST-006413 (ACS - Algorithms for Complex Shapes), by the Israel Science Foundation (grant no. 236/06), and by the Hermann Minkowski–Minerva Center for Geometry at Tel Aviv University.

[†]School of Computer Science, Tel-Aviv University, 69978, Israel. {efif,ophirset,danha}@post.tau.ac.il

$\forall v \in V, f_S(u_{\min}, v) = f_S(u_{\max}, v)$. A *curve* in the domain is defined as a function $\gamma : I \rightarrow \mathbb{P}$ where (i) I is an open, half-open, or closed interval with endpoints 0 and 1; (ii) γ is continuous and injective, except for closed curves, where $\gamma(0) = \gamma(1)$; (iii) if $0 \notin I$, the curve has no start point, and emanates "from infinity". It holds that $\lim_{t \rightarrow 0+} \|\gamma(t)\| = \infty$ (we have a similar condition if $1 \notin I$), and we assume that these limits exist. A *weakly u-monotone curve* $C \subset S$ is the mapping of a curve γ, such that if $t_1 < t_2$ then $\gamma(t_1)$ is lexicographically smaller than $\gamma(t_2)$.

The `Arrangement_2` package of CGAL, the Computational Geometry Algorithms Library,[1] included in Version 3.3 supports planar arrangements induces by planar curves. Recently, this package has been extended to support arrangements of curves embedded on a two-dimensional parametric surface [3]. The extended package can handle curves that approach a boundary in case it is unbounded, or reach a boundary in case it is bounded. In the bounded case, a boundary can define either a contraction point or an identification curve[2]. The extended package is realized as a prototypical CGAL package, and is planned to be included in the next public release.

The main class of the `Arrangement_2` package represents the embedding of a set of continuous weakly u-monotone curves that are pairwise disjoint in their interiors on a two-dimensional parametric surface. The package offers various operations on arrangements stored in this representation, such as point location, insertion of curves, removal of curves, and overlay computation.

Code reuse is maximized by generalizing the prevalent algorithms and their implementations. The generalized code handles features embedded on a modified surface $\widetilde{S} : f_{\widetilde{S}} = f_S(u, v) \,|\, (u, v) \in \widetilde{\mathbb{P}}$ defined over a modified parameter space $\widetilde{\mathbb{P}}$, where the boundaries are removed. Specific code that handles features that approach or reach the boundaries is added to yield a complete implementation.

The implementation of the various algorithms that construct and manipulate arrangements is generic, as it is independent on the type of curves they handle. All steps of the algorithms are enabled by a minimal set of geometric primitives, such as comparing two points in uv-lexicographic order, computing intersection points, etc. These primitives are gathered in a traits class, which models a *geometry-traits* concept [19]. Different geometry-traits classes are provided in the `Arrangement_2` package to handle various families of curves, e.g., line segments, conic arcs, etc.

The geometry-traits concept is factored into a hierarchy of refined concepts. The refinement hierarchy is defined according to the identified minimal

[1] `http://www.cgal.org`

[2] We do not support surfaces, which contain a contracted identification curve.

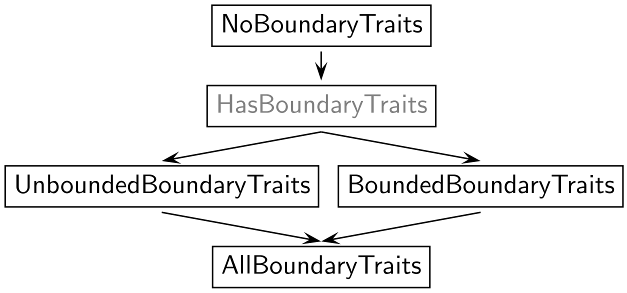


Figure 1: Hierarchy of Geometry Traits Concepts for Arrangement on Surface.

requirements imposed by different algorithms that operate on arrangements, thus alleviating the production of traits classes, and increasing the usability of the algorithms. We refer to the entire hierarchy of refinements defined in Version 3.3 as a single concept called *NoBoundaryTraits* for simplicity. The extended package introduces new concepts, models of which are able to handle unbounded curves or bounded curves, the endpoints of which coincide with contraction points or lie on identification curves; see Figure 1. The "abstract" *HasBoundaryTraits* sub-hierarchy lists additional predicates required to handle both curves that reach or approach the boundaries of the parameter space. It has no models. The refined *BoundedBoundaryTraits* and *UnboundedBoundaryTraits* sub-hierarchies list additional predicates required to handle bounded and unbounded curves respectively. The geometry-traits class that handles arcs of great circles models the *BoundedBoundaryTraits* concept, as the parameter space is bounded in all four directions. Finally, the *AllBoundaryTraits* sub-hierarchy refines all the above. A model of this concept can handle unbounded curves in some directions and bounded curves in others.

3 Handling Arcs of Great Circles on the Sphere

We use the following parameterization of the unit sphere: $\mathbb{P} = [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$ and $f_S(u, v) = (\cos u \cos v, \sin u \cos v, \sin v)$. This parameterization induces two contraction points $p_s = (0, 0, -1)$ and $p_n = (0, 0, 1)$, referred to as the south and north poles respectively, and an identification curve that coincides with the opposite Prime (Greenwich) Meridian.

The geometry-traits class for geodesic arcs on the sphere is parameterized with a geometric kernel [10] that encapsulates the number type used to represent coordinates of geometric objects and to carry out algebraic operations on those objects. The implementation handles all degeneracies, and is exact as long as the underlying number type supports the arithmetic operations $+$, $-$, $*$, and $/$ in unlimited precision over the rationals, such as the one provided by GMP[3]. A point in our arrangement is defined to be an unnormalized vector that emanates from the origin,

[3] `http://www.swox.com/gmp/`

extended with an enumeration that indicates whether the vector (i) pierces the south pole, (ii) pierces the north pole, (iii) intersects the identification arc, or (iv) is in any other direction. An arc of a great circle is represented by its two endpoints, by the normal of the plane that contains the arc, and some Boolean flags that cache information. The orientation of the plane and the source and target endpoints determine which one of the two great arcs is considered. The flags are used to expedite the performance.

All the required geometric operations listed in the traits concept are implemented using only rational arithmetic. Degeneracies, such as overlapping arcs that occur during intersection computation, are properly handled. The end result is a robust yet efficient implementation.

4 Applications

Armed with the geometry-traits for geodesic arcs on the sphere, we can use all the arrangement machinery to solve a variety of problems involving such arrangements. In particular, we compute Minkowski sums of convex polyhedra [7], by overlaying their respective Gaussian maps, which are arrangements of geodesics on the sphere. We also compute various Voronoi diagrams on the sphere through the computation of the lower envelope of the site-distance functions over the sphere. This section describes the latter application.

We define lower envelopes of functions on the sphere in a way similar to the standard definition of lower envelopes of bivariate functions in space [8]:

Definition 1 *Given a set of bivariate functions* $F = \{f_1, \ldots, f_n\}$, *where* $f_i : \mathbb{S}^2 \rightarrow \mathbb{R}$, *their* lower envelope $\Psi(u, v)$ *is defined to be their pointwise minimum* $\Psi(u, v) = \min_{1 \leq i \leq n} f_i(u, v)$.

The *minimization diagram* $\mathcal{M}(F)$ of the set F is the two-dimensional map obtained by central projection of the lower envelope onto $\mathbb{S}^2$.

Definition 2 *Given two points* $p_i, p_j \in \mathbb{S}^2$, *the* distance *between them* $\rho(p_i, p_j)$ *is defined to be the length of a geodesic arc that connects* p_i *and* p_j.

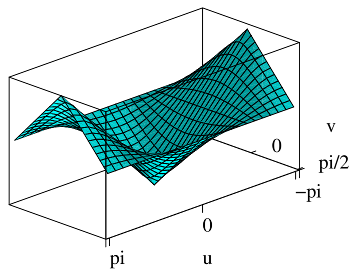
Definition 3 *Given a set of* n *points* $P = \{p_1, \ldots, p_n\}$, $p_i \in \mathbb{S}^2$, *we define* $R(P, p_i) = \{x \in \mathbb{S}^2 \mid \rho(x, p_i) < \rho(x, p_j), j \neq i\}$. $R(P, p_i)$ *is the region of all points that are closer to* p_i *then to any other point in* P.

The *Voronoi diagram* of P over $\mathbb{S}^2$ is defined to be the regions $R(P, p_1), R(P, p_2), \ldots, R(P, p_n)$ and their boundaries.

Edelsbrunner and Seidel [6] observed the connection between Voronoi diagrams in $\mathbb{R}^d$ and lower envelopes of the corresponding distance functions to the sites in $\mathbb{R}^{d+1}$. This also holds for our spherical case. From the above definitions it is clear that if $f_i : \mathbb{S}^2 \rightarrow \mathbb{R}$ is set to be $f_i(x) = \rho(x, p_i)$, for $i = 1, \ldots, n$, then the minimization diagram of $\{f_1, \ldots, f_n\}$ over $\mathbb{S}^2$ is exactly the Voronoi diagram of P over $\mathbb{S}^2$.

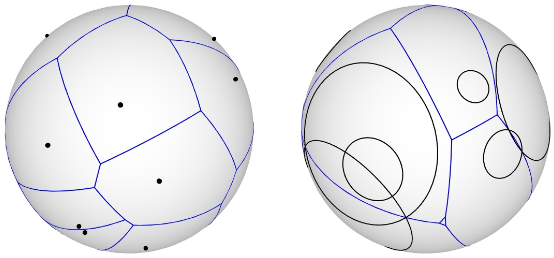
A new framework based on the envelope algorithm of CGAL [13] was developed to compute different types of Voronoi diagrams. The implementation is exact and can handle degenerate input. The framework provides a reduced and convenient interface between the construction of the diagrams and the construction of envelopes, which in turn are computed using the `Envelope_3` package [14]. Obtaining a new type of Voronoi diagrams only amounts to the provision of a traits class that handles the type of bisector curves of the new diagram type [9]. This traits class models the *EnvelopeVoronoiTraits* concept that refines one of the traits concepts mentioned in Section 2. Essentially, every type of Voronoi diagram, the bisectors of which can be handled by an arrangement traits class, can be implemented using this framework. The bisector curves between point sites on the sphere are great circles [16, 17], handled by the newly developed traits class described in Section 3; see Figure 2(a).

We implicitly construct envelopes of distance functions defined over the sphere to compute Voronoi diagrams. The image to the right illustrates the distance function from $(0, 0) \in [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$ on the sphere in the parameter space. The great circle bisector of two point sites on the sphere is the intersection of the sphere and the bisector plane of the points in $\mathbb{R}^3$ (imposed by the Euclidean metric).



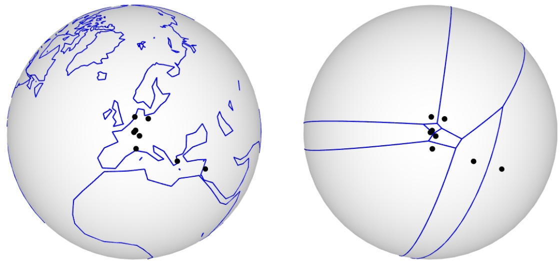
The envelope code together with the traits class for geodesic arcs on the sphere enable the computation of Voronoi diagrams on the sphere, the bisectors of which are great circles or piecewise curves composed of geodesic arcs. Another type of Voronoi diagrams whose bisectors are great circles is the power diagram of circles on the sphere [18], which generalizes the Voronoi diagram of points; see Figure 2(b). Power diagrams on the sphere have several applications similar to the applications of power diagrams in the plane. For example, determining whether a point is included in the union of circles on the sphere, and finding the boundary of the union of circles on the sphere [11, 18].

Given two circles on the sphere c_1 and c_2, let p_1 and p_2 be the planes containing c_1 and c_2 respectively. The bisector of c_1 and c_2 is the intersection of the sphere and the plane that contains the intersection line of p_1 and p_2 and the origin. If p_1 and p_2 are parallel planes, then the bisector is the intersection of the sphere and the plane that contains the origin and is parallel to both p_1 and p_2.



(a) (b)

Figure 2: Voronoi diagrams on the sphere. Sites are drawn in black and Voronoi edges are drawn in blue. (a) A Voronoi diagram of 14 random points. (b) A power diagram of 10 random circles.



(a) (b)

Figure 3: Arrangements on the sphere.

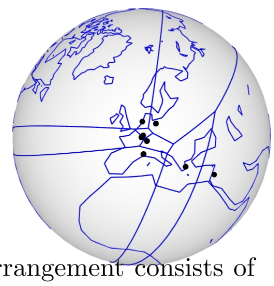


Figure 3(a) shows an arrangement on the sphere induced by (i) the continents and some of the islands on earth, and (ii) the institutions that participate in the ACS project,[4] which appear as isolated vertices. The sphere is oriented such that Nancy is at the center. The arrangement consists of 1053 vertices, 1081 edges, and 117 faces. The data was taken from gnuplot[5] and from google maps[6]. Figure 3(b) shows an arrangement that represents the Voronoi diagram of the eight cities, the institutions above are located at, namely Athens, Berlin, Groningen, Nancy, Saarbrücken, Sophia-Antipolis, Tel Aviv, and Zurich. The figure above shows the *overlay* of the two arrangements shown in Figure 3. Recall that arrangement points are represented as an unnormalized vector; see Section 3. The coordinates of such points are converted into machine floating-point only for rendering purposes.

References

[1] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[2] F. Aurenhammer and R. Klein. Voronoi diagrams. In J. Sack and G. Urrutia, editors, *Handb. Comput. Geom.*, chapter 5, pages 201–290. Elsevier, 2000.

[3] E. Berberich, E. Fogel, D. Halperin, K. Melhorn, and R. Wein. Sweeping and maintaining two-dimensional arrangements on surfaces: A first step. In *Proc. 15th Annu. Eur. Symp. Alg.*, pages 645–656, 2007.

[4] E. Berberich and M. Kerber. Exact arrangements on tori and Dupin cyclides, 2008. Manuscript.

[5] F. Cazals and S. Loriot. Computing the exact arrangement of circles on a sphere, with applications in structural biology. Technical Report 6049, Inria Sophia-Antipolis, 2006.

[6] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Disc. Comput. Geom.*, 1:25–44, 1986.

[7] E. Fogel and D. Halperin. Exact and efficient construction of Minkowski sums of convex polyhedra with applications. *Computer-Aided Design*, 39(11):929–940, 2007.

[8] D. Halperin. Arrangements. In J. E. Goodman and J. O'Rourke, editors, *Handb. Disc. Comput. Geom.*, chapter 24, pages 529–562. Chapman & Hall/CRC, 2nd edition, 2004.

[9] D. Halperin, O. Setter, and M. Sharir. Exact and efficient construction of general two-dimensional Voronoi diagrams via divide and conquer of envelopes in space, 2008. Manuscript.

[10] S. Hert, M. Hoffmann, L. Kettner, S. Pion, and M. Seel. An adaptable and extensible geometry kernel. In *Proc. Workshop Alg. Eng.*, volume 2141 of *LNCS*, pages 79–90. Springer, 2001.

[11] H. Imai, M. Iri, and K. Murota. Voronoi diagram in the Laguerre geometry and its applications. *SIAM J. on Computing*, 14(1):93–105, 1985.

[12] R. Kunze, F. Wolter, and T. Rausch. Geodesic Voronoi diagrams on parametric surfaces. In *Computer Graphics Int. Conf.*, page 230, Washington, DC, USA, 1997. IEEE Computer Society.

[13] M. Meyerovitch. Robust, generic and efficient construction of envelopes of surfaces in three-dimensional space. In *Proc. 14th Annu. Eur. Symp. Alg.*, pages 792–803, 2006.

[14] M. Meyerovitch, R. Wein, and B. Zukerman. 3D envelopes. In Cgal Editorial Board, editor, Cgal *User and Reference Manual*. 3.3 edition, 2007.

[15] R. E. Miles. Random points, sets and tessellations on the surface of a sphere. *The Indian J. of Statistics*, 33:145–174, 1971.

[16] H.-S. Na, C.-N. Lee, and O. Cheong. Voronoi diagrams on the sphere. *Comput. Geom. Theory Appl.*, 23(2):183–194, 2002.

[17] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, NYC, 2nd edition, 2000.

[4]Algorithms for Complex Shapes: `http://acs.cs.rug.nl`
[5]`http://www.gnuplot.info/`
[6]`http://maps.google.com/`

[18] K. Sugihara. Laguerre Voronoi diagram on the sphere. *J. for Geom. Graphics*, 6(1):69–81, 2002.

[19] R. Wein, E. Fogel, B. Zukerman, and D. Halperin. 2D arrangements. In CGAL Editorial Board, editor, CGAL *User and Reference Manual*. 3.3 edition, 2007.