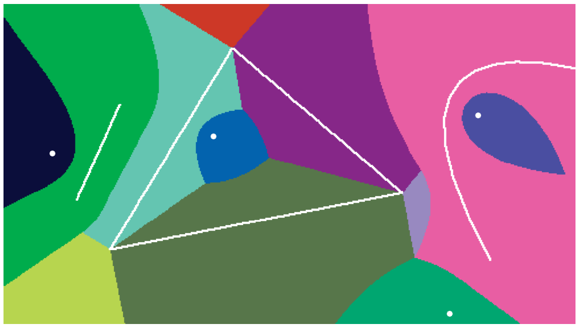
Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware

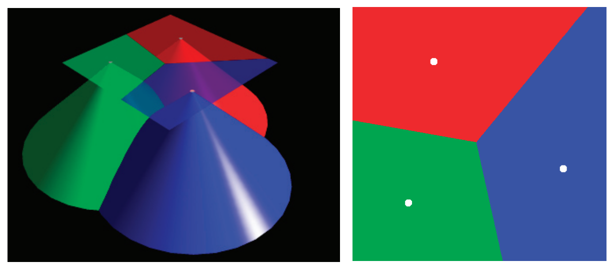
Department of Computer Science | University of North Carolina at Chapel Hill | February 2004



Discrete volumetric approximation of the generalized Voronoi diagram of four points, a line, a triangle, and one cubic Bézier curve computed interactively on a PC. We show Voronoi regions for the edge and vertex features of the triangle. Higher-order curved primitives are handled through tessellation into linear segments.

Highlights

- **Fast, simple algorithm to compute generalized Voronoi diagrams using standard graphics hardware.**
- **Efficient and practical, with little or no precomputation, making it suitable for dynamic geometry.**
- **Applicable to robot motion planning, proximity queries, non-photorealistic rendering, and reconstruction in global illumination.**

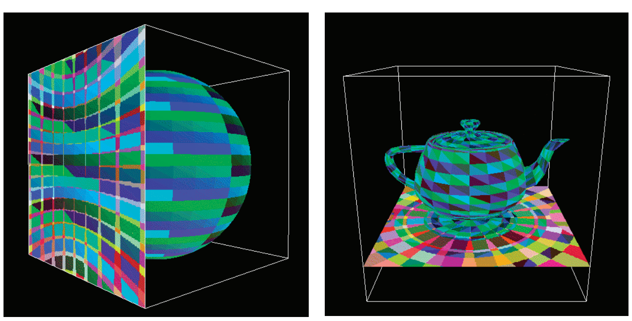


Voronoi diagram of 3 points in 2D to illustrate basic principles. We construct mesh approximations of the distance functions for the sites representing the distance as depth. When viewed in a direction perpendicular to the plane, we obtain the Voronoi diagram.

Background

Given a collection of geometric primitives, a Voronoi diagram is a subdivision of space into cells such that all points in a cell are closer to one primitive than to any other. The geometric primitives are Voronoi sites, the cells are called Voronoi regions, and the Voronoi diagram is the set of all Voronoi regions. Voronoi diagrams have two top-level classifications: ordinary, which refers to diagrams computed over points in any dimension using the Euclidean distance metric, and generalized, which refers to diagrams with higher-order site geometry or with varying distance metrics.

The Voronoi diagram is a fundamental concept that has been independently rediscovered many times in many different fields over the past 450 years. As a result, the concept has widespread applicability.



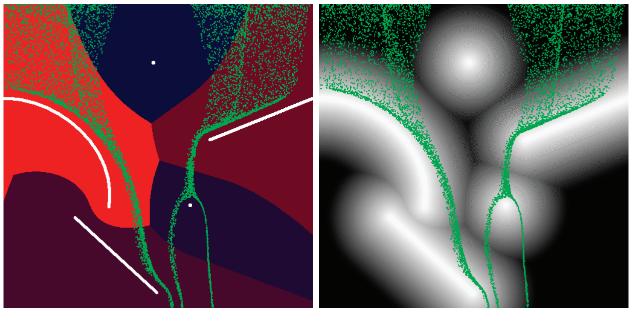
3D Voronoi diagrams of complex polygonal models are computed one 2D slice at a time. Here we show a single slice for a sphere and a teapot model. The colors indicate Voronoi regions for faces, edge, and vertex sites. Only the face site colors are shown on the model.

Generalizations of the distance metric: Standard nearest-site Voronoi regions (left). Farthest regions for the same sites (middle). Weighted regions (right). Weights: line (2); dark point (1); light point (0.5).

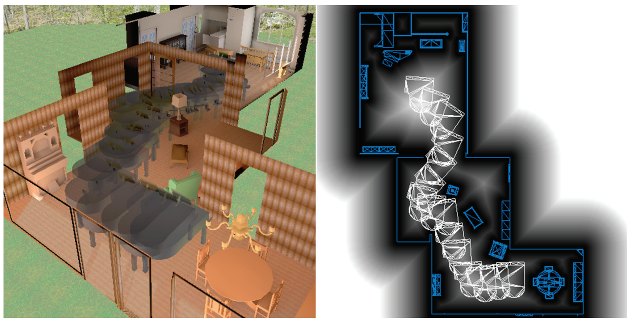
More recently, Voronoi diagrams have been useful in a number of computer graphics-related applications, including collision detection and proximity queries, surface reconstruction, robot motion planning, non-photorealistic rendering, surface simplification, mesh generation, and shape analysis.

The Challenge

Despite the long history of Voronoi diagrams, efficient algorithms for computing then did not appear until the 1970s, with the birth of the field of computational geometry. There are two general classes of algorithms: exact and approximate. *Exact algorithms* directly compute an analytically exact representation of the Voronoi boundaries. This requires representing and manipulating high-degree curves and surfaces and their intersections. These algorithms are often complex and difficult to implement, and they suffer from robustness and accuracy problems. In an attempt to overcome these difficulties, *approximate algorithms* are often used. These algorithms simplify the problem by either discretizing the sites or by discretizing the space containing the sites, using point sampling. We form an approximate boundary representation by computing the Voronoi diagram over the point-sampled sites, or we create a volumetric representation where point samples in the space containing the sites are classified as belonging to a particular Voronoi region. We compute this volumetric representation.



Motion planning of falling particles. Sites are avoided by using the Voronoi diagram's distance buffer (right) to create a potential field. This same principle is used in the rigid-body planner.

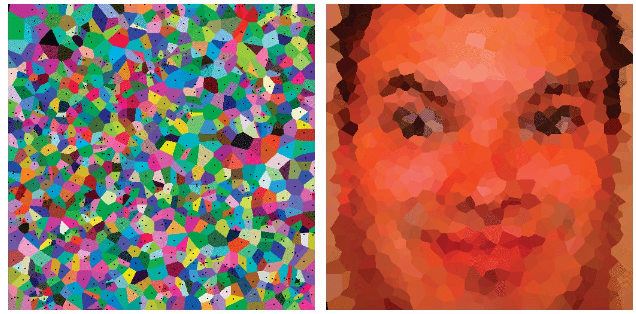


The potential field from the Voronoi diagram of the house floorplan to plan the motion of a piano through a complex scene at interactive rates with no precomputation.

Previous approximate algorithms do provide practical solutions, but are either difficult to error-bound, restricted to static geometry, or are still relatively slow. In this work, we tried to address some of the shortcomings of previous approaches. Our goal was to find an approximate algorithm with the following characteristics: simple to understand and implement; easily generalized in terms of site geometry, distance metrics, and dimension; and efficient and practical with little or no precomputation, making it suitable for dynamic geometry. And, finally, all sources of error must be fully enumerated.

The Approach

We present a new approach for computing generalized 2D and 3D Voronoi diagrams, using interpolation-based polygon rasterization hardware. We compute a discrete Voronoi diagram by rendering a 3D distance mesh for each Voronoi site. The polygonal mesh is a bounded-error approximation of a (possibly) non-linear function of the distance between a site and a 2D planar grid of sample points. For each sample point, we compute the closest site and the distance to that site using polygon scan-conversion and the Z-buffer depth comparison. We construct distance meshes for points, line segments, polygons, polyhedra, curves, and curved surfaces in 2D and 3D. We generalize to weighted and farthest-site Voronoi diagrams, and present efficient techniques for computing the Voronoi boundaries, Voronoi neighbors, and the Delaunay triangulation of points. We also show how to adaptively refine the solution through a simple windowing operation. The algorithm has been implemented on SGI workstations and PCs using OpenGL, and has been applied to complex datasets. We demonstrate the application of our algorithm to fast motion planning in static and dynamic environments, fast proximity queries,



The Voronoi diagram of randomly distributed animated points is used to create real-time dynamic mosaic effects by coloring the Voronoi regions by a texture value at the point site's location.

selection in complex user interfaces, reconstruction in global illumination, and creation of dynamic mosaic tilings. We are also working on a complete generalized Voronoi diagram computation library called HAVOC (Hardware-Accelerated VOronoi Computation), and an accompanying proximity query library called PIVOT (Proximity Information from VOronoi Techniques).

Project Leaders

Dinesh Manocha, professor
Ming Lin, professor

Graduate Research Assistants

Kenneth E. Hoff III

Past Research Assistants

Timothy Culver, John Keyser

Research Sponsors

Intel Corp.; National Science Foundation; National Institutes of Health National Center for Research Resources on Interactive Graphics for Molecular Studies and Microscopy; Office of Naval Research; U.S. Army Research Office; U.S. Department of Energy (ASCI Program)

Selected Publications

Hoff III, K. E., A. Zaferakis, M. Lin, and D. Manocha. "Fast and Simple 2D Geometric Proximity Queries Using Graphics Hardware," *Proc. Symposium on Interactive 3D Graphics*, March 2001.

Hoff III, K. E, T. Culver, J. Keyser, M. Lin, and D. Manocha. "Interactive Motion Planning Using Hardware-Accelerated Computation of Generalized Voronoi Diagrams," *Proc. IEEE International Conference on Robotics and Automation*, 2000, 2931–2937.

Pisula, C., K. E. Hoff III, M. Lin, and D. Manocha. "Randomized Path Planning for a Rigid Body Based on Hardware Accelerated Voronoi Sampling," *Proc. Fourth International Workshop on Algorithmic Foundations of Robotics*, 2000, SA31–SA44.

Key Words

Voronoi diagrams; graphics hardware; polygon rasterization; interpolation; motion planning; proximity query; medial axis; OpenGL; framebuffer techniques; distance computation

www.cs.unc.edu/~geom/voronoi