Hierarchical Data Representations Based on Planar Voronoi Diagrams

Shirley Schussman, Martin Bertram, Bernd Hamann, and Kenneth I. Joy

Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis, CA 95616-8562, USA
e-mail: {schussms,bertram,hamann,joy}@cs.ucdavis.edu

Abstract

Multiresolution representations of high-dimensional scattered data is an outstanding problem in the field of scientific visualization. This paper introduces a data hierarchy of Voronoi diagrams as a versatile solution. We implemented two programs to demonstrate this, the first of which uses a constant function to approximate the data within each Voronoi tile and the second program uses the Sibson interpolant within each tile [5].

1 Introduction

This paper presents a new solution for constructing multiresolution data representations: data hierarchies based on Voronoi diagrams. This approach is motivated by the need to interactively render very large data sets that consist of scattered or arbitrarily gridded data. A hierarchy of Voronoi diagrams is a natural solution for a number of reasons. First, Voronoi diagrams provide a "natural mesh" for scattered data, data without explicit point connectivity. Second, point insertion and deletion operations on a Voronoi diagram are constant-time operations [3]. In addition, Voronoi tiles can be sorted in depth in linear time for volume visualization and Voronoi diagrams can be extended to n dimensions. Although the Voronoi diagram's dual the Delaunay triangulation has the same properties, the Voronoi diagram provides a more intuitive tesselation.

2 Related Work

The most common methods for visualizing massive data sets are based on mesh reduction. Many algorithms reduce the meshes well for surface data, but do not extend to higher-dimensional data sets [4]. Some algorithms accelerate either isosurface extraction and decimation or volume rendering, but do not extend to other visualization techniques [2, 6]. Subdivision for Delaunay triangulations [1] and tetrahedral meshes [7] has been explored, but with severe restrictions.

3 Voronoi Hierarchies

A Voronoi hierarchy consists of a set of Voronoi diagrams that represent different resolutions, and different approximations of a given data set. As a result, there is a lot of room for variation in an implementation. Implementations define a method for selecting points to define a level in the hierarchy, to select an interpolant within each Voronoi tile, and to choose an error metric to determine the overall accuracy of each level in the Voronoi hierarchy.

3.1 Refinement of a Voronoi Diagram

Given a Voronoi diagram V defined by k original data points, the remaining original data points $p_0, \ldots, p_m$, and a corresponding set of function values $f_0, \ldots, f_m$, a method for refining V to better represent all n original data points is outlined.

Refinement is based on an error ε_{t_j} that is calculated for each tile t_j in the Voronoi diagram, where each tile overlaps n_{t_j} original data points. Since a constant function $f(p_i)$ is used to represent the region of space in the tile t_j, an error is associated with each point in that region. The l_1 error norm is used to calculate the error ε_{t_j} for all points within the tile.

$$\varepsilon_{t_j} = \frac{1}{n_{t_j}} \sum_{p_i \in t_j} \| f(p_i) - f_i \| \tag{1}$$

This result is compared with the global average error ε_{avg} for all $n = k + m$ original data points.

$$\varepsilon_{avg} = \frac{1}{n} \sum_{j=1}^{k} \sum_{p_i \in t_j} \| f(p_i) - f_i \| \tag{2}$$

If the tile error t_j exceeds the average tile error ε_{avg}, then a point p_{worst}, associated with ε_{worst}, is inserted into it.

$$\varepsilon_{worst} = \max_{p_i \in t_j} \| f(p_i) - f_i \| \tag{3}$$

As a result, only the areas in the Voronoi diagram with high error are refined.

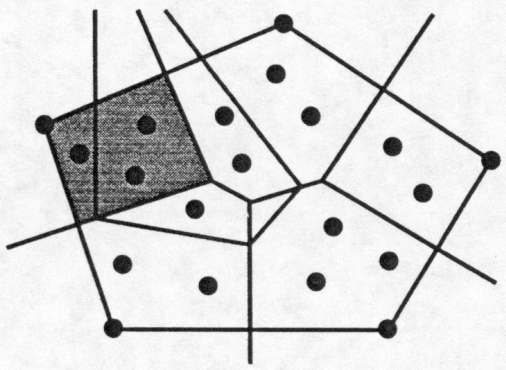

Figure 1: Example of vertex insertion. Black lines: current Voronoi diagram; blue lines: convex hull of Voronoi vertices; grey region: Voronoi tile with maximal error; red point: vertex of maximal error p_{worst}; red lines: Voronoi tile to be inserted.

3.2 Constant Function Based Implementation

A variation of the general algorithm was used for the constant function based implementation. Before points are inserted into a Voronoi diagram, a predetermined threshold for point insertion is calculated using a percentage of the current Voronoi diagram's average error, ε_{avg}. All tiles with an average error that exceeds this threshold are inserted. Typically, a threshold of $1.1 * \varepsilon_{avg}$ was used, so that only tiles with average errors that are worse than the global error are refined.

The value used for the constant function in a given tile is the function value of the point that defines that tile. As a result, all piecewise constant function values appearing in the Voronoi diagram are original data values.

3.3 Sibson's Interpolant Based Implementation

The implementation based on the Sibson's interpolant varies the general algorithm in a different manner. Rather than identifying a set of tiles with high error, a single tile with the worst area-weighted average is inserted. The error ϵ_{t_j} is redefined for each tile t_j with area a_{t_j} to be:

$$\epsilon_{t_j} = \frac{a_{t_j}}{n_{t_j}} \sum_{p_i \in t_j} \|f(p_i) - f_i\| \tag{4}$$

This approach calculates one point at a time to insert, and then updates the Voronoi diagram. Unlike the general algorithm, updating the Voronoi diagram doesn't mean that a new level in the hierarchy has been calculated. Instead, points are inserted and the Voronoi diagram is updated until either an error bound or a certain number of points is reached. The resulting set of points is used to define the level in the hierarchy.

A better interpolant was used within each tile, the Sibson interpolant [5]. The Sibson interpolant is based on blending function values f_j associated with the points that define a Voronoi diagram. The resulting interpolation defines a smooth function that is C^1-continuous everywhere except at the points which define the Voronoi diagram. The function $f(x, y)$ is evaluated by inserting a point (x, y) temporary into the Voronoi diagram and by estimating the areas a_j cut away from Voronoi tiles t_j. The value of Sibson's interpolant at (x, y) is defined by

$$f(x, y) = \frac{\sum_j a_j f_j}{\sum_j a_j}.$$

4 Results

The image of the Cats Eye Nebula (Figure 2), courtesy of the NASA's Hubble telescope, show results from the two implementations. Even with 580 points, both algorithms pick up the characteristics of the nebula, which results from the manner in which points are inserted. Because p_{worst} in a tile t_j corresponds to either the brightest or darkest point in the tile t_j, bright and dark regions are represented first.

5 Conclusions and Future Work

This paper introduced a method for hierarchical, gridless representation of planar data, which is straightforward and can be to generalized to higher dimensions. We plan to extend the algorithm to volumetric data, and eventually to time-varying data. In addition, efficient ray-casting and isosurface extraction methods for Voronoi diagram hierarchies will be developed.

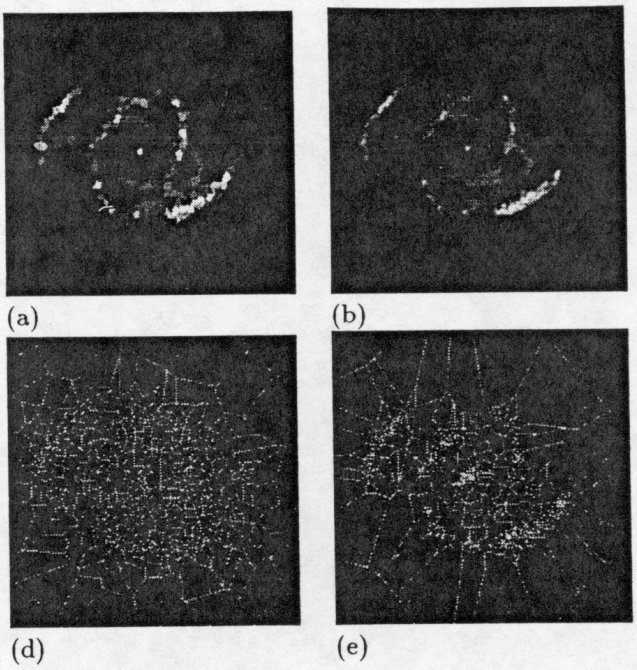

Figure 2: (a) An image of the Cats Eye Nebula approximated with 580 points using the constant function; (b) Sibson's interpolant used to estimate same image with 580 points; (c) and (d) show corresponding Voronoi meshes for (a) and (b), respectively

References

[1] L. De Floriani and E. Puppo. Constrained Delaunay triangulation for multiresolution surface description. In *Proceedings Ninth IEEE International Conference on Pattern Recognition*, pages 566–569, Los Alamitos, California, 1988. CS Press.

[2] David Laur and Pat Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. *Computer Graphics*, 25(4):285–288, July 1991.

[3] Arne Maus. Delaunay triangulation and the convex hull of n points in expected linear time. *BIT*, 24(2):151–163, 1984.

[4] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, July 1992.

[5] R. Sibson. Locally equiangular triangulation. *The Computer Journal*, 21:243–245, 1978.

[6] Jane Wilhelms and Allen Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 11(3):201–227, July 1992.

[7] Yong Zhou, Baoquan Chen, and Arie Kaufman. Multiresolution tetrahedral framework for visualizing regular volume data. In *IEEE Visualization '97*, October 1997.