# Terrain Modeling using Voronoi Hierarchies

Martin Bertram[1], Shirley E. Konkle[2], Hans Hagen[1], Bernd Hamann[3], and Kenneth I. Joy[3]

[1] University of Kaiserslautern, Department of Computer Science, P.O. Box 3049, D-67653 Kaiserslautern, Germany.
[2] Pixar Animation Studios, 1200 Park Ave., Emeryville, CA 94608, U.S.A.
[3] Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis, CA 95616-8562, U.S.A.

**Abstract.** We present a new algorithm for terrain modeling based on Voronoi diagrams and Sibson's interpolant. Starting with a set of scattered sites in the plane with associated function values defining a height field, our algorithm constructs a top-down hierarchy of smooth approximations. We use the convex hull of the given sites as domain for our hierarchical representation. Sibson's interpolant is used to approximate the underlying height field based on the associated function values of selected subsets of the sites. Therefore, our algorithm constructs a hierarchy of Voronoi diagrams for nested subsets of the given sites. The quality of approximations obtained with our method compares favorably to results obtained from other multiresolution algorithms like wavelet transforms. For every level of resolution, our approximations are $C^1$ continuous, except at the selected sites, where only $C^0$ continuity is satisfied. Considering $n$ sites, the expected time complexity of our algorithm is $O(n \; log \; n)$. In addition to a hierarchy of smooth approximations, our method provides a cluster hierarchy based on convex cells and an importance ranking for sites.

## 1 Introduction

Clustering techniques [7] can be used to generate a data-dependent partitioning of space representing inherent topological and geometric structures of scattered data. Adaptive clustering methods recursively refine a partitioning resulting in a multiresolution representation that is of advantage for applications like progressive transmission, compression, view-dependent rendering, and topology reconstruction. For example, topological structures of two-manifold surfaces can be reconstructed from scattered points in three-dimensional space using adaptive clustering methods [6]. In contrast to mesh-simplification algorithms, adaptive clustering methods do not require a grid structure connecting data points. A cluster hierarchy is built in a "top-down" approach, so that coarse levels of resolution require less computation times than finer levels.

We present a Voronoi-based adaptive clustering method for terrain modeling. Arbitrary samples taken from large-scale terrain models are recursively selected according to their relevance. Continuous approximations of the terrain model are constructed based on the individual sets of selected sites using

Sibson's interpolant [10]. We have implemented this algorithm using a Delaunay triangulation, i.e., the dual of a Voronoi diagram, as underlying data structure. Constructing a Delaunay triangulation is simpler than constructing the corresponding Voronoi diagram, since a large number of special cases (where Voronoi vertices have valences greater than three) can be ignored. A major drawback of Delaunay triangulations is that they are not unique, in general. This becomes evident when the selected sites are sampled from regular, rectilinear grids such that either diagonal of a quadrilateral can be used, resulting in random choices affecting the approximation. The corresponding Voronoi diagram, however, is uniquely defined and can be derived directly from a Delaunay triangulation. Sibson's interpolant is also efficiently computed from a Delaunay triangulation. The advantage of our method when compared to Delaunay-based multiresolution methods [4] is that our approximations are unique and $C^1$ continuous almost everywhere.

## 2 Adaptive Clustering Approach

Adaptive clustering schemes construct a hierarchy of tesselations, each of which is associated with a simplified representation of the data. We assume that a data set is represented at its finest level of resolution by a set $P$ of $n$ points in the plane with associated function values:

$$P \;=\; \{(\mathbf{p}_i,\ f_i) \mid \mathbf{p}_i \in \mathbb{R}^2,\ f_i \in \mathbb{R},\ i = 1, \ldots, n\}.$$

This set can be considered as a sampled version of a continuous function $f : D \to \mathbb{R}$, where $D \subset \mathbb{R}^2$ is a compact domain containing all points $\mathbf{p}_i$. The points $\mathbf{p}_i$ define the associated parameter values for the samples $f_i$. We do not assume any kind of "connectivity" or grid structure for the points $\mathbf{p}_i$. For applications more general than terrain modeling, the points $p_i$ can have $s$ dimensions with $t$-dimensional function values $f_i$, see Figure 1.

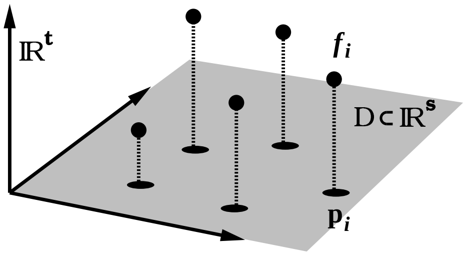


**Fig. 1.** Scattered sites with associated function values.

The output of an adaptive clustering scheme consists of a number of levels $L_j,\ j = 0, 1, \ldots$, defined as

$$L_j \;=\; \{(\tau_k^j,\ \tilde{f}_k^j,\ \varepsilon_k^j) \mid k = 1, \ldots, n_j\},$$

where, for every level with index $j$, the tiles (or regions) $\{\tau_k^j \subseteq D \mid k = 1, \ldots, n_j\}$ form a partitioning of the domain $D$, the functions $\tilde{f}_k^j : \tau_k^j \to \mathbb{R}$ approximate the function values of points in the tiles $\tau_k^j$, *i.e.*,

$$\tilde{f}_k^j(\mathbf{p}_i) \;\approx\; f_i \quad \forall \mathbf{p}_i \in \tau_k^j,$$

and the residuals $\varepsilon_k^j \in \mathbb{R} \geq 0$ estimate the approximation error. In principle, any error norm can be chosen to compute the residuals $\varepsilon_k^j$. We note that the error norm has a high impact on the efficiency and quality of the clustering algorithm, since it defines an optimization criterion for the approximations at every level of resolution. We suggest to use the following norm:

$$\varepsilon_k^j = \left( \frac{1}{n_k^j} \sum_{\mathbf{p}_i \in \tau_k^j} \left| \tilde{f}_k^j(\mathbf{p}_i) - f_i \right|^p \right)^{\frac{1}{p}}, \quad p \in [1, \infty], \tag{1}$$

where $n_k^j = |\{\mathbf{p}_i \in \tau_k^j\}|$ is the number of points located in tile $\tau_k^j$. In the case of $p = \infty$, the residual is simply the maximal error considering all sites in the corresponding tile. This error norm can be easily adapted to higher-dimensional function values $f_i \in \mathbb{R}^t$ by using the Euclidean norm of the individual differences, $\|\tilde{f}_k^j(\mathbf{p}_i) - f_i\|$.

A global error $\varepsilon^j$ with respect to this norm can be computed efficiently for every level of resolution from the residuals $\varepsilon_i^j$ as

$$\begin{aligned} \varepsilon^j &= \left( \frac{1}{n} \sum_{k=1}^{n_j} \sum_{\mathbf{p}_i \in \tau_k^j} \left| \tilde{f}_k^j(\mathbf{p}_i) - f_i \right|^p \right)^{\frac{1}{p}} \\ &= \left( \frac{1}{n} \sum_{k=1}^{n_j} n_k^j \left( \varepsilon_k^j \right)^p \right)^{\frac{1}{p}}. \end{aligned} \tag{2}$$

Starting with a coarse approximation $L_0$, an adaptive clustering algorithm computes finer levels $L_{j+1}$ from $L_j$ until a maximal number of clusters is reached or a prescribed error bound is satisfied. To keep the clustering algorithm simple and efficient, the approximation $L_{j+1}$ should differ from $L_j$ only in cluster regions with large residuals in $L_j$. As the clustering is refined, it should eventually converge to a space partitioning, where every tile contains exactly one data point or where the number of points in every tile is sufficiently low leading to zero residuals.

## 3 Constructing Voronoi Hierarchies

In the following, we describe our adaptive clustering approach for multiresolution representation of scattered data: a hierarchy of Voronoi diagrams [1,9] constructed from nested subsets of the original point set.

The *Voronoi diagram* of a set of points $\mathbf{p}_i,\ i = 1, \ldots, n$ in the plane is a space partitioning consisting of $n$ tiles $\tau_i$. Every tile $\tau_i$ is defined as a subset of $\mathbb{R}^2$ containing all points that are closer to $\mathbf{p}_i$ than to any $\mathbf{p}_j,\ j \neq i$, with respect to the Euclidean norm.

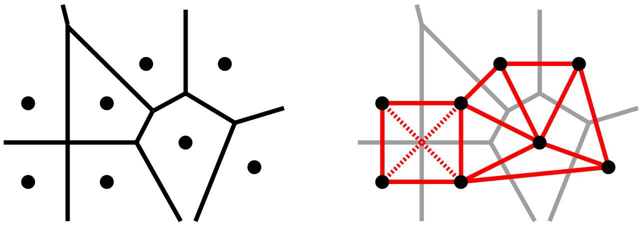


**Fig. 2.** Planar Voronoi diagram and its dual, the (not uniquely defined) Delaunay triangulation.

A Voronoi diagram can be derived from its dual, the *Delaunay triangulation* [2,5,3,4], see Figure 2. The circumscribed circle of every triangle in a Delaunay triangulation does not contain any other data points. If more than three points are located on such a circle, the Delaunay triangulation is not unique. The Voronoi vertices are located at the centers of circumscribed circles of Delaunay triangles, which can be exploited for constructing a Voronoi diagram. The Voronoi diagram is unique, in contrast to the Delaunay triangulation.

Our method constructs a Delaunay triangulation incrementally by successive insertion of selected points. For every point inserted, all triangles whose circumscribed circles contain the new point are erased. The points belonging to erased triangles are then connected to the new point, defining new triangles that satisfy the Delaunay property, see Figure 3.

When inserting a point located inside a prescribed Voronoi tile, the corresponding tile center is incident to one or more Delaunay triangles to be removed. Since all triangles to be removed define a connected region, point insertion is a local operation of expected constant time complexity (and of $O(n)$ complexity in the extremely rare worst case).

For applications in $s$-dimensional spaces ($s > 2$), the Delaunay triangulation consists of $s$-simplices whose circumscribed $s$-dimensional hyperspheres contain no other point. Our algorithm remains valid for applications using data defined on higher-dimensional domains.

Our adaptive clustering algorithm uses *Sibson's interpolant* [10] for constructing the functions $\tilde{f}_k^j$. Sibson's interpolant is based on blending function values $f_i$ associated with the points $\mathbf{p}_i$ defining the Voronoi diagram. The blending weights for Sibson's interpolant at a point $\mathbf{p} \in \mathbb{R}^2$ are computed by inserting $\mathbf{p}$ temporarily into the Voronoi diagram and by computing the areas $a_i$ that are "cut away" from Voronoi tiles $\tau_i$, see Figure 4. The value of

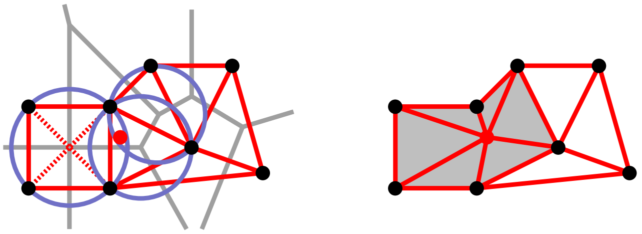


**Fig. 3.** Construction of Delaunay triangulation by point insertion. Every triangle whose circumscribed circle contains the inserted point is erased. The points belonging to removed triangles are connected to the new point.

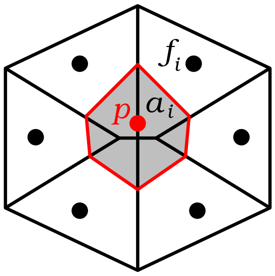


**Fig. 4.** Computing Sibson's interpolant at point $p$ by inserting $p$ into a Voronoi diagram and using the areas cut away from every tile as blending weights.

Sibson's interpolant at $\mathbf{p}$ is defined as

$$f(\mathbf{p}) \;=\; \frac{\sum_i \, a_i f_i}{\sum_i a_i}.$$

Sibson's interpolant is $C^1$ continuous everywhere, except at the points $\mathbf{p}_i$. To avoid infinite areas $a_i$, we clip the Voronoi diagram against the boundary of the compact domain $D$. A natural choice for the domain $D$ is the convex hull of the points $\mathbf{p}_i$.

In the following, we provide the individual steps of our clustering algorithm:

(i) Construct the Voronoi diagram for the minimal point set defining the convex hull of all points $\mathbf{p}_i$, $i = 1, \ldots, n$. The tiles of this Voronoi diagram define the cluster regions $\tau_k^0$, $k = 1, \ldots, n_0$, of level $L_0$.

(ii) From the functions $\tilde{f}_k^j$, defined by Sibson's interpolant and from error norm (1) ($p = 2$), compute all residuals $\varepsilon_k^0$. To avoid square root computations, $(\varepsilon_k^0)^2$ is stored.

(iii) Refinement: $L_j \rightarrow L_{j+1}$. Let $m$ denote the index of a maximal residual in $L_j$, *i.e.*, $\varepsilon_m^j \geq \varepsilon_k^j \; \forall k = 1, \ldots, n_j$. Among all $\mathbf{p}_i \in \tau_m^j$, identify a data point $\mathbf{p}_{max}$ with maximal error given by $max_{\mathbf{p}_i \in \tau_m^j} \{|\tilde{f}_m^j(\mathbf{p}_i) - f_i|\}$. Insert $\mathbf{p}_{max}$ into the Voronoi diagram, resulting in a new tile denoted as $\tau_{n_{j+1}}^{j+1}$, where $n_{j+1} = n_j + 1$.

(iv) Update $\varepsilon^{j+1}_{n_{j+1}}$ and all residuals associated with tiles that have been modified, *i.e.*, all tiles that are adjacent to the new tile $\tau^{j+1}_{n_{j+1}}$ with center $\mathbf{p}_{max}$. (All other tiles remain unchanged, *i.e.*, $\tau_i^{j+1} = \tau_i^j$, $\tilde{f}_i^{j+1} = \tilde{f}_i^j$, and $\varepsilon_i^{j+1} = \varepsilon_i^j$.)

(v) Compute the global approximation error $\varepsilon^j$ using the error norm given by equation (2). Terminate the process when a prescribed global error bound is satisfied or when a maximal number of points has been inserted. Otherwise, increment $j$ and continue with step (iii).

We briefly analyze the complexity of our algorithm. As stated above, point insertion into a Delaunay triangulation is a local operation, provided that the Voronoi tile containing the inserted point is known. Analogously, evaluating Sibson's interpolant at a point inside a certain Voronoi tile is a local, expected constant-time operation. If we had an oracle providing the order of insertion and the indices of the Voronoi tiles containing every inserted point, our algorithm would perform in expected linear time (and in $O(n^2)$ time in the worst case, for example when all points are nearly co-circular).

The overall computational cost of our method is determined by the cost for computing residuals, selecting points for insertion, and keeping track of the Voronoi tiles containing these points. When starting with a sufficiently even distribution of original points, we can assume that every point is relocated into a different tile on average $O(log\ n)$ times. For computing the residuals, Sibson's interpolant is evaluated at every original point also expectedly $O(log\ n)$ times. After locally updating the residuals, a tile with greatest residual can be determined in expected constant time by a comparison-free sorting algorithm like hashing. Thus, the overall expected time complexity of our method is $O(n\ log\ n)$.

For comparison, an algorithm constructing Delaunay triangulations and convex hulls from the scratch (without providing a hierarchy) in expected linear time is described by Maus [8]. A divide-and-conquer method provides a worst-case solution with $O(n\ log\ n)$ time complexity.

## 4 Numerical Results

We have applied our Voronoi-based clustering approach to approximate the "Crater Lake" terrain data set, courtesy of U.S. Geological Survey. This data set consists of 159272 samples at full resolution. Approximation results for multiple levels of resolution are shown in Figure 5 and summarized in Table 1.

The quality of approximations obtained with our method compares favorably to results obtained from other multiresolution algorithms like wavelet transforms. A standard compression method, for example, uses a wavelet transform followed by quantization and arithmetic coding of the resulting coefficients. Using the Haar-wavelet transform for compression of the Crater-Lake data set (re-sampled on a regular grid at approximately the same resolution) results in approximation errors (for $p = 2$) of 0.89 percent for a 1:10

**Table 1.** Approximation errors in percent of amplitude for Crater Lake. Figure 5 shows the different levels of resolution.

| No. Voronoi Tiles | Error ($p = \infty$) [%] | Error ($p = 2$) [%] |
|---|---|---|
| 100 | 31.6 | 3.13 |
| 200 | 17.1 | 1.96 |
| 300 | 16.3 | 1.55 |
| 400 | 13.9 | 1.33 |
| 500 | 11.9 | 1.21 |
| 1000 | 10.8 | 0.80 |

compression and 4.01 percent for a 1:100 compression [1]. For a Voronoi-based compression locations of the samples need to be encoded as well.

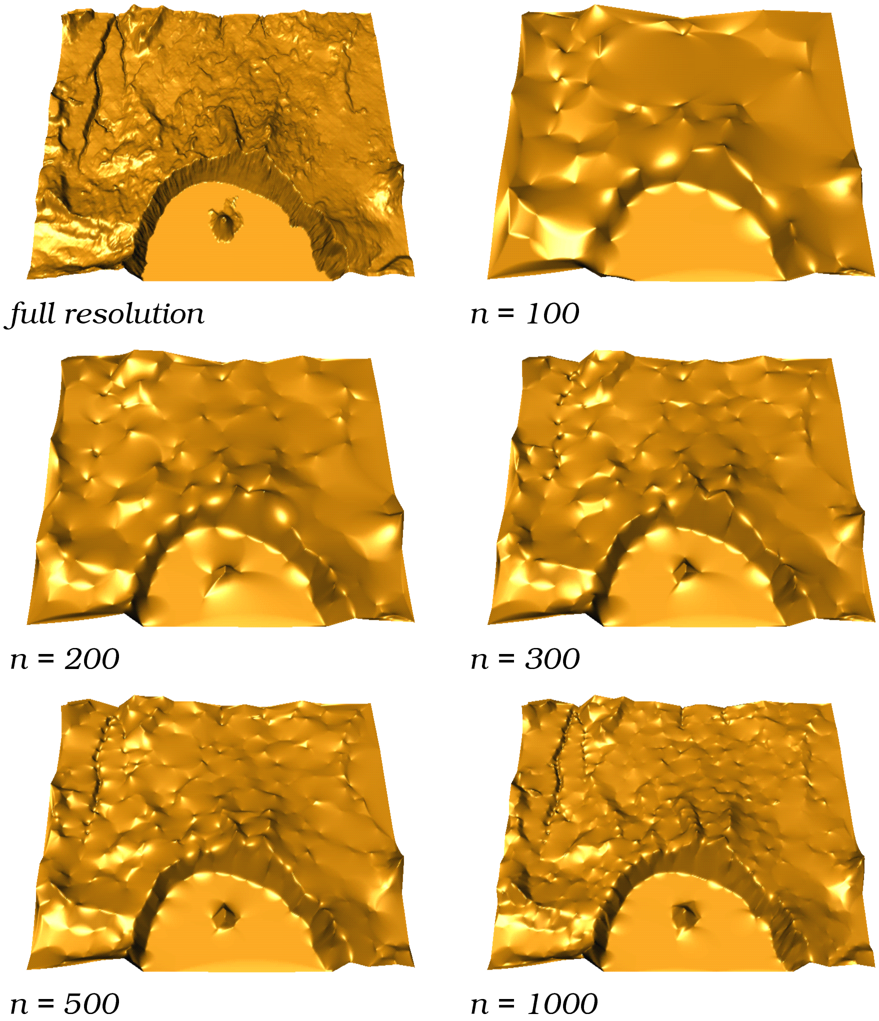
In addition to a hierarchy of smooth approximations, our method provides a cluster hierarchy consisting of convex cells and an importance ranking for sites. Future work will be directed at the explicit representation of discontinuities and sharp features.

## 5 Acknowledgements

We thank Mark Duchaineau, Daniel Laney, Eric LaMar, and Nelson Max for their helpful ideas and contributions to many discussions. This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the Office of Naval Research under contract N00014-97-1-0222; the Army Research Office under contract ARO 36598-MA-RIP; the NASA Ames Research Center through an NRA award under contract NAG2-1216; the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159; the Lawrence Berkeley National Laboratory; the Los Alamos National Laboratory; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628. We also acknowledge the support of ALSTOM Schilling Robotics and SGI. We thank the members of the Visualization and Graphics Research Group at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis.

## References

1. M. Bertram, *Multiresolution Modeling for Scientific Visualization*, Ph.D. Thesis, Department of Computer Science, University of California, Davis, California, 2000. `http://daddi.informatik.uni-kl.de/~bertram`



*full resolution*

$n$ = 100

$n$ = 200

$n$ = 300

$n$ = 500

$n$ = 1000

**Fig. 5.** Crater-Lake terrain data set at different levels of resolution (using $p = 2$). The full-resolution data set consists of 159272 points, courtesy of U.S. Geological Survey.

2. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer, Berlin, Germany, 1997.
3. L. De Floriani, B. Falcidieno, and C. Pienovi, *A Delaunay-based method for surface approximation*, Proceedings of Eurographics '83, Amsterdam, Netherlands, 1983, pp. 333–350, 401.
4. L. De Floriani and E. Puppo, *Constrained Delaunay triangulation for multiresolution surface description*, Proceedings Ninth IEEE International Conference on Pattern Recognition, IEEE, 1988, pp. 566–569.

5. G. Farin, *Surfaces over Dirichlet tesselations*, Computer Aided Geometric Design, Vol. 7, No. 1–4, 1990, pp 281–292.
6. B. Heckel, A.E. Uva, B. Hamann, and K.I. Joy, *Surface reconstruction using adaptive clustering methods*, Proceedings of the Dagstuhl Seminar on Scientific Visualization, Dagstuhl, Germany, May 21–26, 2000 (to appear).
7. B.F.J. Manly, *Multivariate Statistical Methods, A Primer*, second edition, Chapman & Hall, New York, 1994.
8. A. Maus, *Delaunay triangulation and convex hull of n points in expected linear time*, BIT, Vol. 24, No. 2, pp. 151–163, 1984.
9. S.E. Schussman, M. Bertram, B. Hamann and K.I. Joy, *Hierarchical data representations based on planar Voronoi diagrams*, W.C. de Leeuw and R. van Liere (eds.), Data Visualization 2000, Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization (VisSym), Springer Vienna/New York, 2000, pp. 63–72.
10. R. Sibson, *Locally equiangular triangulation.* The Computer Journal, Vol. 21, No. 2, 1992, pp. 65–70.