Clustering-Based Generation Of Hierarchical Surface Models

Bjoern Heckel, Antonio Uva and Bernd Hamann

Center for Image Processing and Integrated Computing (CIPIC)
Department of Computer Science
University of California, Davis

ABSTRACT

We present a highly efficient, automatic method for the generation of hierarchical surface triangulation. Given a set of scattered points in three-dimensional space, without known connectivity information, our method reconstructs a valid triangulated surface model in a two-step procedure. First, we apply clustering to the set of given points and identify point subsets in locally nearly planar regions. Second, we construct a surface triangulation from the output of the clustering step. The output of the clustering step is a set of 2-manifold tiles, which locally approximate the underlying unknown surface. We construct the triangulation of the entire surface by triangulating the individual tiles and triangulating the gaps between the tiles. Since we apply point clustering in a hierarchical fashion we can generate model hierarchies by triangulating various levels resulting from the hierarchical clustering step.

Keywords: Hierarchical Clustering, Surface Reconstruction, Data Reduction, Reverse Engineering, Multiresolution Representation, Triangulation

1 INTRODUCTION

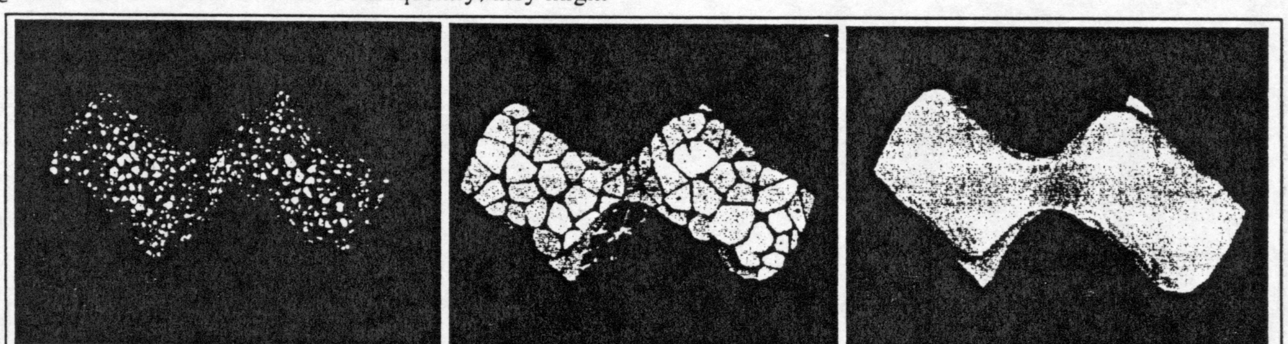
Surface reconstruction is concerned with the extraction of shape information from point sets. Often, these point sets describing complex objects are generated by scanning physical objects, by sampling other digital representations (e.g., contour functions) or by merging data from different sources. Consequently, they might embody incompleteness, noise and redundancy making a general approach for reconstructing surfaces a challenging problem. In many instances, high complexity and high level of detail characterize the described objects. Different levels of representation are needed to allow rapid rendering and interactive exploration and manipulation. Surface reconstruction problems arise in a wide range of scientific and engineering applications including reverse engineering and industrial design, geometric modeling and grid generation, multiresolution rendering.

We introduce an extremely fast surface reconstruction method that is based on cluster analysis. Our approach generates a multiresolution of reconstructed surfaces from arbitrary point sets. The reconstructed model is generated in two stages. First, applying clustering to the point set yields a set of almost flat shapes - so-called tiles - that locally represent the underlying surface. In the second stage, the gaps between the unconnected tiles are filled by inserting triangles producing a valid geometrical and topological multiresolution model.

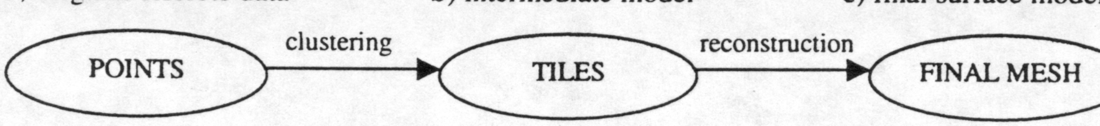
2 RELATED WORK

2.1 Surface Reconstruction

The goal of surface reconstruction methods can be described like this:
Given a set of sample points X assumed to lie on or near an unknown surface U, construct a surface model S approximating U.



a) original discrete data **b**) intermediate model **c**) final surface model



Address: Department of Computer Science, University of California, Davis, Davis, CA 95616-8562
Email: {heckel, uva, hamann}@cs.ucdavis.edu
Project homepage: http:\\graphics.ucdavis.edu\people\heckel\projects\reconstruction\index.html

Most surface reconstruction methods require additional topological knowledge, such as connectivity information, surface topological type. Common approaches are *parametric reconstruction, function reconstruction*, and *constriction methods*. Only a few methods developed recently by Hoppe et al. [3][4] are capable of creating valid topological and geometrical models from only 3D coordinates of sample points. Another surface reconstruction method that is related to the reconstruction stage of our approach is the concept of *alpha shapes* (α-shapes) proposed by Edelsbrunner and Muecke [1]. Compared with other approaches, our method's main advantage is the overall performance.

2.2 Multiresolution modeling

A multiresolution model is a model defined by a set of levels of detail of an object, which can be used to efficiently access any one of those levels on demand. *Surface simplification* is a particular form of multiresolution modeling in which the goal is to take a polygonal model as input and generate a simplified model (i.e., approximation of the original) as output. A variety of methods have been developed, including *image pyramids* [9], *volume methods* [10], *edge, face and vertex decimation techniques* [11][6][7][13][2], *vertex clustering* [12], *simplification envelopes* [14], and *wavelet methods* [5][8].

3 SURFACE RECONSTRUCTION

3.1 Clustering

The input for our clustering process is a set of scattered points in 3-dimensional space. Our method for creating the cluster hierarchy is based on an incremental and divisive paradigm. Initially, all points are placed in one cluster, which is recursively split. A *cluster* C is a subset of the given data set. The *center of a cluster* $C_{center}=(cx,cy,cz)$ is defined as the geometric mean of the points $p_i=(x_{pi},y_{pi},z_{pi})$ with $i\in\{1..k\}$ associated with the cluster of size k. At each stage of the clustering process, every point is associated with exactly one cluster, that is the cluster with its center being the closest in terms of Euclidean distance. The *internal error* of a cluster is the sum of the distances from the cluster center to the associated points. In each iteration of the clustering process, the cluster C_i with the highest internal error is split into two clusters. Alternatively, the cluster with the highest eigenvalue could be selected for the splitting process, which in our experience yields similar results. The centers of the generated clusters are determined by adding/subtracting an offset vector to/from the center of the original cluster. This offset vector lies either in direction of highest deviation of the cluster data or in the direction of maximum variance. The direction of maximum variance is computed by performing principal component analysis (PCA) on the 3-by-3 covariance-matrix M of the cluster's normalized k-by-3 data matrix Δ:

$$\Delta=\begin{bmatrix} x_{p1}-cx & y_{p1}-cy & z_{p1}-cz \\ \vdots & \vdots & \vdots \\ x_{pk}-cx & y_{pk}-cy & z_{pk}-cz \end{bmatrix}$$

The direction of maximum variance is equivalent to the eigendirection with the largest eigenvalue. Using the direction of maximum variance is generally more accurate, but is only feasible if the number of points is relatively small (e.g., less than 500). Therefore, a threshold variable τ_{PCA} is used to control which splitting mechanism is used. After splitting a cluster, a local reclassification scheme is used to improve the quality of the classification. The points to be reclassified are given by all points that are assigned to the split cluster and its neighbor clusters in the Gabriel graph [15] of the cluster centers. After each reclassification step we update the local neighborhood. The cluster centers are moved to reflect the changes in the point-cluster association. The Gabriel graph is updated locally and another cluster is split subsequently. The clustering process terminates when the smallest eigenvalues of all clusters do not exceed the threshold $\tau_{PCAlimit}$.

3.2 Reconstruction

At this point, we have a set of clusters that partition the original data set. Since we choose a relatively small value for $\tau_{PCAlimit}$, the points associated with a certain cluster are near-coplanar. Thus, the point clusters have an almost flat shape. For each cluster C_i, the cluster center and the two eigendirections with the two largest eigenvalues define a plane P_i that locally minimizes the sum of the plane-to-point distances for the associated points. We project all points p_k associated with cluster C_i into the plane P_i and compute the convex hull H_i for the projected points p_k' in the associated plane. We then map the points p_l' on H_i back to their original locations in three-dimensional space. The result is a flat, convex polygon in 3D-space defining the so-called *tile* T_i of cluster C_i, that consists of a set of points p_l'' lying in H_i. For each cluster we generate a *pseudo-tile* T_i' by replacing the points p_l'' defining the tile boundary by their original points p_l. Pseudo-tiles are not necessarily planar.

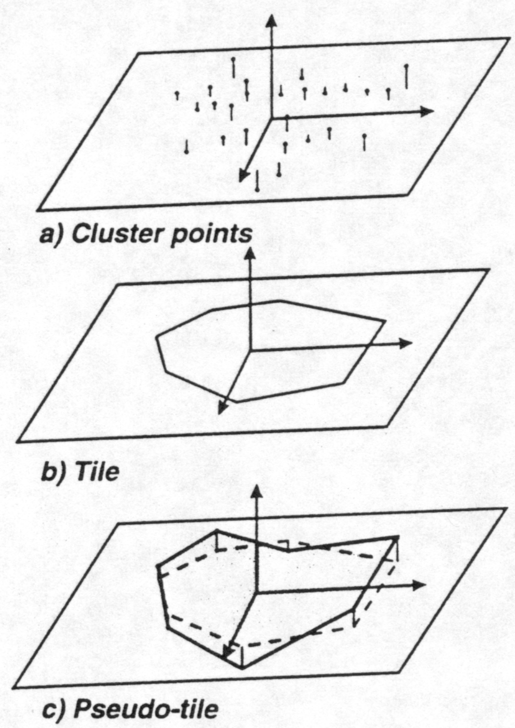


Figure 2: Tile generation.

After the tile generation is completed, each cluster C_i has two corresponding representations by its tile T_i and its pseudo-tile T_i'. The principal orientation of T_i and T_i' is described by the cluster's eigendirection with the smallest eigenvalue.

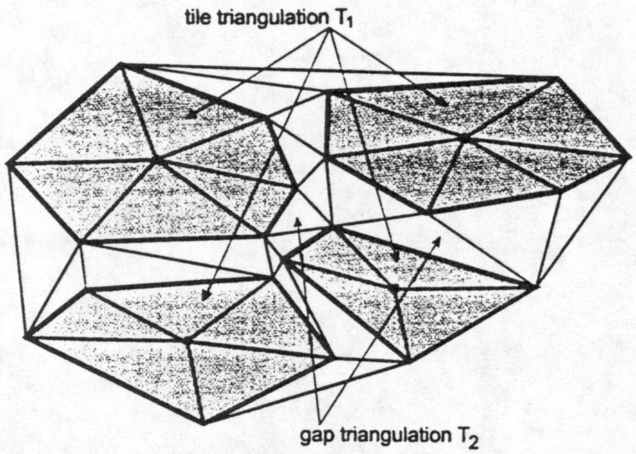


Figure 3: Triangulation of tiles and between tiles.

For the reconstruction process we triangulate pseudo-tiles by connecting the cluster center with the vertices of its boundary polygon. The resulting set of triangles T1 (tile triangulation) is a close approximation of the unknown surface. However, it is not a consistent model due to the lack of connectivity information for the tiles. A connected model is obtained adding a set of triangles T2 (gap triangulation) to T1 that fills the gaps between the tiles. To determine T2, we apply a Delaunay Triangulation (DT) algorithm to the boundary points of the pseudo-tiles. Since the boundary of the Delaunay Triangulation describes the convex hull of the point set, we have to remove certain elements of the mesh (triangles or tetrahedra) which "do not belong" to the desired surface. We remove triangles from the Delaunay Triangulation, if (1) all points lie in one tile or (2) if the points do not fit in a sphere of radius β, the value of β is chosen by locally adjusting the global alpha threshold depending on the tile area. Since we have "almost defined a topology" after the tile generation step, completing the reconstruction is a far simpler problem than applying the α-shape approach directly to the original set of points. Because the gaps between the tiles are relatively small, one can choose a very small α-value without removing important features of the model. Removing the undesired triangles from the Delaunay Triangulation of the boundary points yields the set of triangles $\mathbf{T_2}$ that fill the gap between tiles. By merging $\mathbf{T_1}$ and $\mathbf{T_2}$ we obtain a consistent model.

4 APPLICATION

We have applied our method to a variety of data sets. Three resolution levels of the Mt St. Helens data set, which consists of 151,728 points, are shown in figure 4. The multiresolution hierarchy for up to 1391 tiles has been generated in about 2 minutes on a SGI O^2 workstation with a 180 MHz R5000 processor. In Figure 5, the result of the application of our method to four other data sets is shown. The corresponding performance data is shown in Table 1. Additional performance information, error measure and pictures can be found on the project homepage (http://graphics.ucdavis.edu/people/heckel/projects/reconstruction /index.html).

Data Set	*# of points*	*reconstruction time in [sec]*	*# of tiles*
Mt St. Helens	151,728	132.8	1391
Car	20,621	16.3	517
3Holes	4,000	4.1	402
Rabbit	35,929	34.7	727
Peak function	20,000	18.1	693

Table 1: Performance data for five data sets.

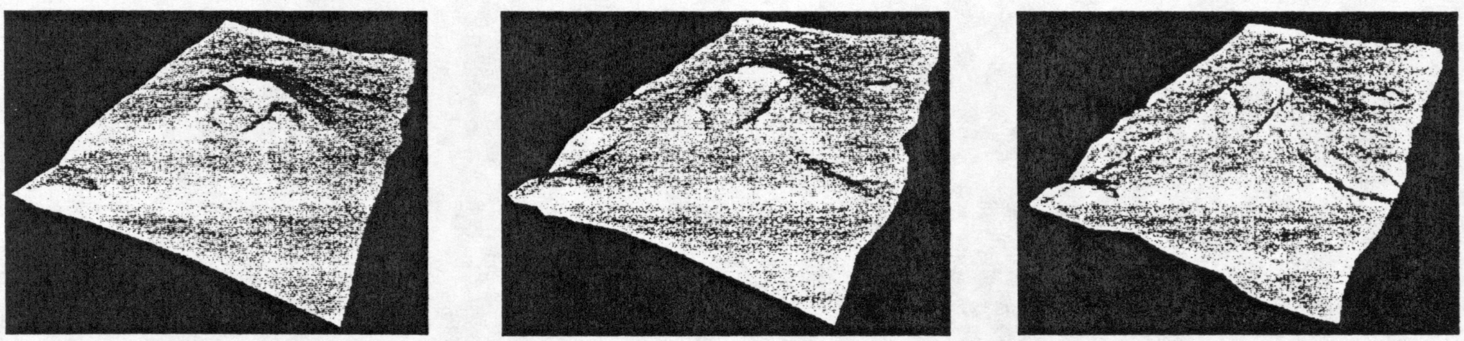


Figure 4: Three resolution levels of Mt St. Helens data set (154, 382 and 1391 tiles).

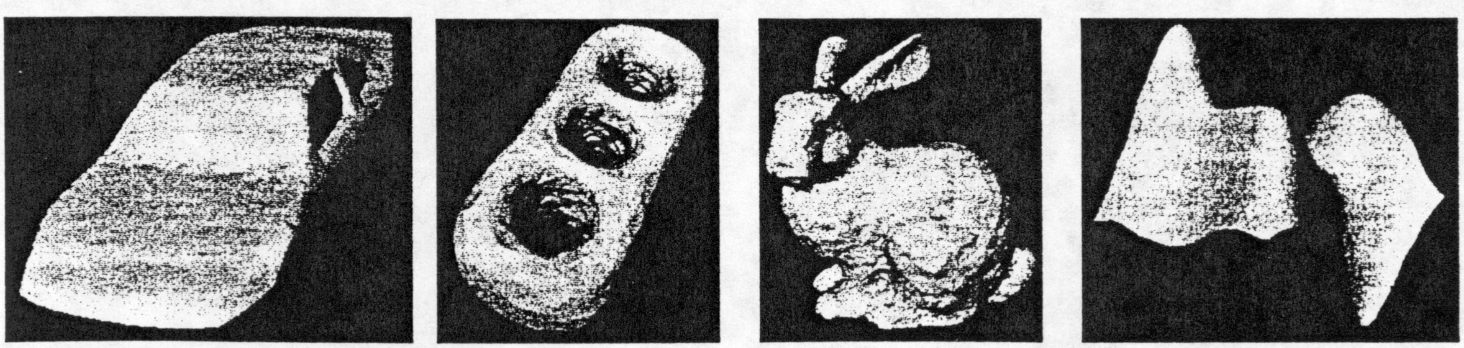


Figure 5: Four reconstructed models.

5 SUMMARY AND FUTURE WORK

The algorithm we have presented allows the generation of a hierarchy of surface models from discrete point sets without known connectivity information. While we have demonstrated the power of our approach only for surface models, we are quite confident that the same clustering paradigm, when applied to more general two- or three-manifold, or even time-varying data, would significantly speed up the process of computing multiple-level-of-detail representations.

We plan to extend our approach to the clustering of more general scattered data sets describing scalar and vectors fields, defined over either two-dimensional or three-dimensional domains. Faster algorithms for the generation of data hierarchies for scientific visualization will become more important as our ability to generate ever larger data sets increases: Computing a data hierarchy prior to the application of a visualization algorithm should not require minutes or hours but seconds instead. We believe that our clustering methodology provides one viable answer to this problem. To allow the analysis of massive data sets we have also implemented a scalable parallelization of our approach.

Furthermore, we plan to extend our algorithm to ensure that all tetrahedra are removed, such that the reconstructed model is a true 2-manifold representation. Regarding the triangle elimination phase, we will develop means to guarantee that no "holes" are inserted into the model. Currently, we are also working on alternative gap elimination schemes, that are performed directly during the clustering stage.

6 ACKNOWLEDGEMENTS

This work was supported by the IBM Almaden Research Center, the National Science Foundation under contract ASC 9624034 (CAREER Award) to the University of California, Davis. In addition, this project was funded by 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 under the NRA2-36832 (TLL) program and Lawrence Livermore National Laboratory under contract W-7405-ENG-48 (B335358) to the University of California, Davis. We would like to thank the members of the Visualization Thrust at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis.

7 REFERENCES

[1] H. Edelsbrunner, Ernst P. Muecke, "Three-Dimensional Alpha Shapes," in: ACM Transactions on Graphics, Volume 13, Number 1, January 1994.

[2] Gieng, T. S., Hamann, B., Joy, K. I., Schussman, G. L. and Trotts, I. J., "Smooth hierarchical surface triangulations," in: Yagel, R. and Hagen, H., eds., *Visualization* '97, IEEE Computer Society Press, Los Alamitos, CA, pp. 379--386, 1997.

[3] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W., "Surface reconstruction from unorganized points," in Computer Graphics (SIGGRAPH '92 Proceedings), pp. 71-78, 1992.

[4] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J. and Stuetzle, W., "Piecewise smooth surface reconstruction," in Computer Graphics (SIGGRAPH '94 Proceedings), pp. 295-302,1994.

[5] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W., "Multiresolution analysis of arbitrary meshes," in Computer Graphics (SIGGRAPH '95 Proceedings), pp. 173-182, 1995.

[6] Hoppe, H., "Progressive meshes," in: Proceedings of SIGGRAPH '96, pp. 99-108, 1996.

[7] Popovic, J., Hoppe, H., "Progressive simplicial complexes," in Computer Graphics (SIGGRAPH '97 Proceedings), pp. 217-224, 1997.

[8] Zorin, D., Schroeder, P. and Sweldens, W., "Interactive multiresolution mesh editing," in Computer Graphics (SIGGRAPH '97 Proceedings), pp. 259-268, 1997.

[9] A. Rosenfeld, "Multiresolution Image Processing and Analysis," Springer-Verlag, Berlin, 1984.

[10] T. He, L. Hong, A. Kaufman, A. Varshney and S. Wang, "Voxel-Based Object Simplification," in: Proceedings of IEEE Visualization '95, IEEE Comput. Soc. Press, 1995.

[11] W. Schroeder and J. Zarge and W. E. Lorensen, "Decimation of triangle meshes," in: Proceedings of SIGGRAPH '92, pp. 65-70, 1992.

[12] J. Rossignac and P. Borrel, "Multi-resolution {3D} approximations for rendering complex scenes," in: B. Falcidieno and T. Kunii, eds., Modeling in Computer Graphics: Methods and Applications, Springer-Verlag, Berlin, 1993.

[13] M. Garland and P. S. Heckbert, "Surface Simplification Using Quadric Error Metrics," in: Proceedings of SIGGRAPH '97, pp. 209-216, 1997.

[14] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks and W. Wright, "Simplification Envelopes," in: Proceedings of SIGGRAPH '96, 1996.

[15] A. Okabe, B. Boots, K. Sugihara, "Spatial Tessellations," John Wiley & Sons, Chichester, 1992.