Polyhedral Approximation of 3D Objects from Scattered Boundary Points

Thomas Schreiber
University of Kaiserslautern

Extended Abstract

Building a CAD model with a CAD system from scratch is even for an experienced user a time consuming process. If a physical model exist, e.g. in car body design, it should be used to support the shaping process. With modern 3D scanning technology the acquisition of large point sets from the surface of the model becomes possible. Now, the challenge is to derive a CAD model from the measured points automatically.

This talk concerns the first step in this reverse engineering process: Given a set of scattered 3d points, find a polyhedral approximation of the 3d object the points come from. The most natural way is to create a simplical polyhedron. In the past various methods for the generation of a simplical surface have been developed [1],[2],[4],[5],[8]. For an overview see [9] and for an introduction to reverse engineering see [7].

Statement of the Problem

To analyze the problem, let us have a closer look to the analog 2d case. Given a set P of scattered points $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ in the plane, find a simple polygon connecting all the points. A polygon is simple, if it has no self-intersections, no holes and bounds a single connected region. To construct a simple polygon through all points is equivalent to find a Hamilton cycle. Since a simple polygon of n vertices has n edges, the task is to choose n edges from $\binom{n}{2}$ possible. Thus, a brute force algorithm will have a time complexity of $\binom{\binom{n}{2}}{n} = O(n^n)$ which is clearly not feasible. In order to reduce the search space most known algorithms chooses the Delaunay triangulation instead of the complete graph. The Delaunay triangulation of n vertices has at most $3n - 6$ edges. Note, that a naive search is still not feasible, since $\binom{3n-6}{n} = O(3^n)$. However, not every Delaunay triangulation contains a Hamilton cycle [3]. Thus, assumptions has to made to ensure that the edges to be reconstructed are edges in the Delaunay triangulation. Let us assume for the following that this is the case.

Reconstruction of a Simple Polygon

Since the dual of any triangulation of a simple polygon is a tree [6] the dual of the Delaunay triangulation of a simple polygon (assuming existence) forms also a tree. Using the circumcenters of the Delaunay triangles as duals to the Delaunay triangles the dual tree becomes a subgraph of the Voronoi diagram $VD(P)$. O'Rourke et al. [5] observed that the shortest tree connecting those Voronoi points that correspond to Delaunay triangles which form a triangulation of a simple polygon of all points yields a natural solution. They called this tree the *minimal spanning Voronoi tree of P* ($MSVT(P)$) and gave an $O(n^3)$ algorithm for the construction of an approximation.

In this talk we want to describe a new algorithm (of course a heuristic) for the construction of the $MSVT(P)$. The algorithm is based on an other observation that large parts of the $MSVT(P)$ are contained in the ordinary MST of the Voronoi points restricted to the Voronoi edges.

The method consists of the following steps:

(1) Construct the Voronoi diagram of P.

(2) Compute the ordinary minimal spanning tree of the Voronoi points V restricted to the Voronoi edges E. We will call this tree $MST(V, E)$. Since every Voronoi point belongs to the $MST(V, E)$ the dual region is the Delaunay triangulation of P.

(3) Prune the $MST(V, E)$ according to the following rule: Cut of branches, if edges are longer than a given threshold or if the corresponding dual region is not simple. This step results in a set of branches of the $MST(V, E)$ or in a set of dual regions each consisting of a set of Delaunay triangles.

(4) Classify each region to be inside or outside of the desired polygon according to the following classification rule: If there exist a point $\boldsymbol{p}_k \in P$ which lie exactly on two regions from which one is known to be inside or outside than classify the other contrarily to that. If no such point exist than join the two regions which are connected by the shortest edge of the $MST(V, E)$ and form together a simple polygon. Notice, the region outside the convex hull of P is a priori known to be outside.

(5) Collect all regions classified as inside to construct the desired simple polygon.

Step (1) and (2) can be done with complexity $O(n \log n)$. Step (3) can be done in $O(n)$ time by a single walk through the $MST(V, E)$. Thus the result depends on the traversing order but we could not notice any big influence. Let r be the number of generated regions by step (3) than step (4) needs $O(rn)$ time and step (5) takes $O(r)$ time. Overall the method has the time complexity of $O(n(\log n + r))$.

The method explained above can be directly extended into 3d. Because in 3d the number of tetrahedra is $O(n^2)$ the time complexity raises to $O(rn^2)$.

References

[1] J.-D. Boissonnat: Geometric Structures for Three-Dimensional Shape Representation, ACM Trans. on Graphics, Vol. 3, No. 4, 266–286 (1984).

[2] T. DeRose, H. Hoppe, T. Duchamp, J. A. McDonald, W. Stuetzle: Fitting of Surfaces to Scattered Data, SPIE Vol. 1830, 212–220 (1992).

[3] M. B. Dillencourt: A non-Hamiltonian, nondegenerate Delaunay triangulation, Information Processing Letters, Vol. 25, No. 3, 149–151 (1987).

[4] H. Edelsbrunner, E. P. Mücke: Three-dimensional Alpha Shapes, ACM Trans. on Graphics, Vol. 13, No. 1, 43–72 (1994).

[5] J. O'Rourke, H. Booth, R. Washington: Connect-the-Dots: A New Heuristic, Computer Vision, Graphics, and Image Processing 39, 258–266 (1987).

[6] J. O'Rourke: Computational Geometry in C, Cambridge University Press (1993).

[7] T. Varady, R. R. Martin, J. Cox: Reverse Engineering of Geometric Models — An Introduction to appear in CAD.

[8] R. C. Veltkamp: 3D Computational Morphology, Eurographics'93, Vol. 19, No. 3, C-115–127 (1993).

[9] R. C. Veltkamp: Closed Object Boundaries from Scattered Points, LNCS 885, Springer (1994).