8—36

Auto-Surface Reconstruction With Alpha-Shape

Xiaolong Xu[1]
Graduate School of Engineering
Hiroshima University

Koichi Harada[2]
Department of Engineering
Hiroshima University

Abstract

In this paper, we present an algorithm for the reconstruction of piecewise linear surfaces from unorganized sample points with improved α-shape. Alpha-shapes is based on efficient, discrete mathematics which determine exact relationships between points, shapes, and spaces. It generates a family of shapes according to the selected α parameter. The method discussed in this paper might be applied for surface reconstruction, and the process is fully automatic.

1 Introduction

The process of converting a set of sample points in 3D space into a computer graphics model generally involves several steps: the reconstruction of an initial piecewise-linear model, cleanup, simplification, and perhaps fitting with surface patches. In this paper, we put our emphasis on the first step. The input of the process is a set of points in 3D space, without any additional structure or organization while the output is a polygonal mesh. Alpha-shape[1] is a very powerful tool, as α varies, one can obtains different α-shapes from the point set itself to the convex hull. Notice that in general an α-shape is a non-connected, non-regular polytope, so it is not directly suitable for surface reconstruction.

Two improved α-shape methods that have already been presented are "Alpha-solid"[2] and "Anisotropic density-scaled α-shape"[3]. Melkemi and Chen[4] also introduced the conception "A-shape" in 2D and 3D. By constructing another point set A and then A-shape, it can solve the nonuniform problem.

Based on the Marek and Michael's work, we first find the triangle with minimum area and the α value that keeps it on the surface, then the α will be adjusted with the point's density.

2 Improved α- shape Method

In this section, we discuss scaling α locally as a factor of the sampling density of each triangle vertex.

2.1 Interval Calculation

Given the point set as input, we first construct the Delaunay triangulation, called D. For each simplex $\sigma_T \in D$ there is a single interval such that σ_T belongs to the surface iff the α value is contained in this interval. Seeing Fig 2-1 for example, if $\alpha \in [r, R]$, line MN will be kept on the surface as 1-simplex, this range is the interval of the line MN. For points M and N, their interval is [0 , r]. Out of the interval, the simplex will not keep itself on the surface and will change to lower or higher order simplex. The details about the interval see reference [1].

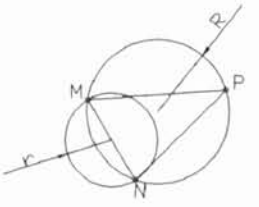

Fig 2-1 Intervals' Calculation in 2D

2.2 Initial Triangle

Alpha shape consists of points, edges, triangles and tetrahedron. Since we wish to obtain a surface, we only need to carefully select a subset of the triangles from the Delaunay triangulation. The initial triangle is our "seed" with the property that satisfy minimum criterion. With this "seed", we can propagate to the whole surface.

2.3 Density Determination

How to calculate the point density in surface reconstruction is important in the method. In general, two rather intuitive formulations of adaptive or variable bandwidth estimators have been considered[5]. The first varies the fixed bandwidth with the estimation point and is often referred to as a balloon estimator. Its form is given by

$$\hat{f}_{\lambda}(x) = \frac{1}{n\lambda_x}\sum_{i=1}^{n} K\left(\frac{x - x_i}{\lambda_x}\right) = \frac{1}{n}\sum_{i=1}^{n} K_{\lambda_x}(x - x_i)$$

[1] Address: 1-7-1 kagamiyama, Higashi-hiroshima, Hiroshima 739-8521, Japan. E-mail: xuxl@hiroshima-u.ac.jp
[2] Address: 1-7-1 kagamiyama, Higashi-hiroshima, Hiroshima 739-8521, Japan. E-mail: harada@mis.hiroshima-u.ac.jp

(2-1)

The balloon estimator was first introduced by Loftsgaarden and Quesenberry[9] in the particular form of the *k*th nearest neighbor estimator. Much have been investigated about the kth nearest neighbor estimator, and it seems clear that it is not an effective density estimator in the univariate case. However, Terrell and Scott[6] show that the kth nearest neighbor estimator improves as dimensionality increases and will perform well in dimensions greater than 4.

The second variable bandwidth procedure is referred to as the sample-point estimator, in which the bandwidth is varied with each data point and not with the estimation point. The function form of the sample-point estimator is given by

$$\hat{f}(x)=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\lambda_i}K(\frac{x-x_i}{\lambda_i})=\frac{1}{n}\sum_{i=1}^{n}K_{\lambda_i}(x-x_i)$$

(2-2)

Different from the two methods mentioned above, we present yet another density determination method: its bandwidth varies with the estimation point, while the estimator is constructed with k-points cluster method. In 3-dimension, we use enclosing ball with the smallest radius as the closeness measure, and select k=4. In this case, each point has a tetrahedral with the smallest enclosing ball. The radius is the point's density.

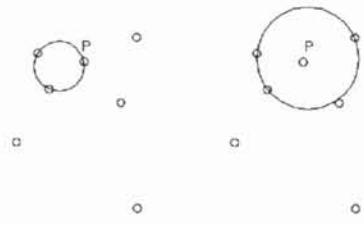

Fig2-2 Point density in 2D(Left: K=3, Right: K=4)

Finding the k-point cluster with the smallest ball, its 2D condition see the "On enclosing k points by a circle" [7]. In that, the author presents randomized algorithms with O(nk) space and O(n log n + nk) expected running time, resp., O(n) space and O(n log n + nk log k) time. In higher dimension, it is more complex and time consuming[8]. Our work is a little different from it. For each point, we have to find its k neighbors with the smallest ball. The value of k is chosen by the analyst to specify the desired degree of smoothing of the data. Small k values result in a small bandwidth, producing a spiky map with little smoothing. Larger k values result in a larger bandwidth and smoother density map. In our implementation, we choose K=4(k=K-1) because small features reconstruction need small K values. So for each estimation point, we find the tetrahedron with the smallest circumsphere by searching the neighbor cells of the Voronoi diagram. In the case K>4, we need to calculate the higher order Voronoi diagram, this will make our algorithm much more complex.

2.4 Scaling Algorithm

In sections 2.2 and 2.3, we have obtained the initial triangle and points' density. The α value that keeps the initial triangle on the surface is our initial α value and is called α_0. This value will change according to the point density, we discuss it in 2-dimension.

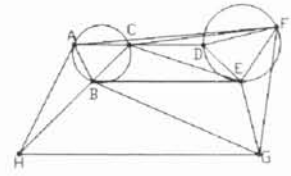

Fig 2-3 Scaling example(2D)

For example in Fig2-3, according to our density calculation, ρ_A(the density of point A)=ρ_B =ρ_C , and ρ_D=ρ_E=ρ_F. We choose the BC as our "seed" line. For BC, we can get it average density by:

$$\rho_{BC}=(\rho_B+\rho_C)/2$$

Then, we get the α value corresponding to each line using the equation:

$$\alpha_{i,j}=\alpha_0\gamma(\rho_{ij}/\rho_{BC}) \qquad (2\text{-}3)$$

where γ is the penalty factor, it is decided by:

$$\gamma=\left(1+\sqrt{\frac{\sum_{i=1}^{n}((\rho_i-\overline{\rho})/\overline{\rho})^2}{n-1}}\right)^{0.2} \qquad (2\text{-}4)$$

ρ_i : local point density

$\overline{\rho}$: mean of local point density

n : number of points

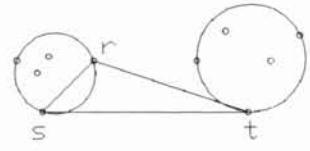

Fig 2-4 Scaling example(3D, K=5)

In 3D case (see Fig 2-4), we have:

$$\alpha_\Delta = \alpha_0 \gamma(\rho_\Delta / \rho_0) \quad (2\text{-}5)$$

where:

$$\rho_\Delta = (\rho_r + \rho_s + \rho_t)/3 \quad (2\text{-}6)$$

ρ_r, ρ_s, ρ_t: local point density of three vertexes of the triangle.

γ: penalty factor, calculated by (2-4).

α_0, ρ_0: Initial triangle and the corresponding density.

If $\rho_r = \rho_s = \rho_t$, it means these three vertices belong to one cluster, and γ=1.0 according to (2-4). In Fig 2-4, we can see that $\rho_r = \rho_s < \rho_t$ because r and s belong to one cluster and t belongs to another one. With the increases of variance of ρ_r, ρ_s and ρ_t, the penalty factor γ also should increase because the variance between the clusters increases. In the following example(see Fig 2-5), we set γ=1.0 as the constant factor. We can see that in each cluster the algorithm works well but between clusters it fails because we ignore the variance of ρ_r, ρ_s and ρ_t.

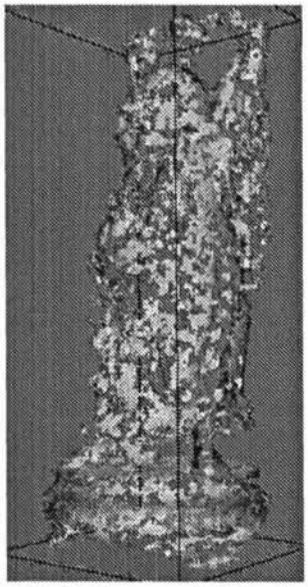

Fig 2-5 Scaling example(3D) γ=1.0

3 Implementation and Results

Our algorithm consists of four steps:

1) Construction of Delaunay triangulation.

2) Computes the α-intervals for the 2-simplex(triangle) in the Delaunay triangulation.

3) Finds the initial triangle and calculates the point's density.

4) Get the triangle set on the surface by comparing the α_{ijk} and the interval, then render the result.

Starting from the dense region, our method can easily capture the small features and construct the shape automatically. The quality and the time of reconstruction rely heavily on the procedure of density estimation. There are still some problems that need further investigation, such as how to assess the quality of the estimate and extend to multivariate adaptive procedure.

We also give the comparison of powercrust method, tightcocone and our algorithm in table 3-1. All computations were carried out on PC, with AMD-K6 3D processor and 192M RAM.

Table 3-1

Model / Method	Bunny			Head			Knot		
	Pts.	Tri.	T(m)	Pts.	Tri.	T(m)	Pts.	Tri.	T(m)
*PowerCrust	35539	78513	67	12772	92503	8	10000	92780	5
TCocone	35539	71071	6	12772	25534	2	10000	20000	2
Our method	35539	106069	9	12772	29537	5	10000	20726	4

* For the three examples, we take different multiplier in the Powercrust option.

Head: 10

Bunny: 100000

Knot: 100000

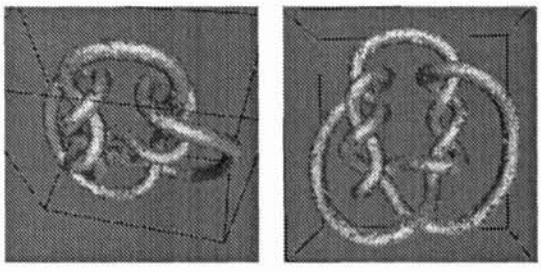

Fig 3-1 Knot Points: 10000 Triangles: 20726)

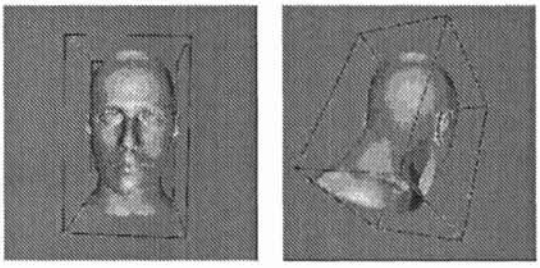

Fig 3-2 Head (Points: 12772 Triangles: 29537)

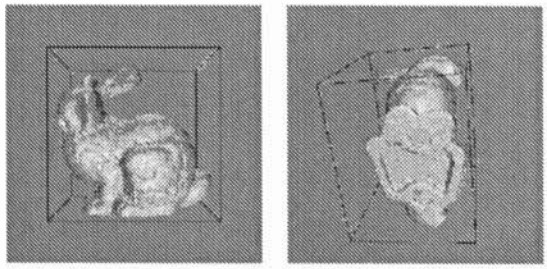

Fig 3-3 Bunny (Points: 35539 Triangles: 106069)

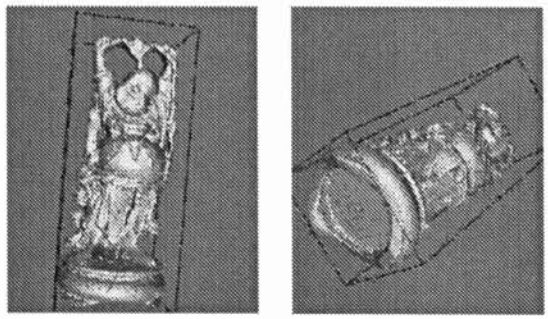

Fig 3-4 Buddha(Points: 32328 Triangles: 74187)

4 Acknowledgments

We thank Ernst Mücke and Herbert Edelsbrunner for the Delaunay triangulation software and the α-shape package. We thank Hughes Hoppe(Microsoft) for the head knot, and the Stanford 3D Scanning Repository for the bunny and buddha. We also thank Tamal Krishna Dey and Nina Amenta for the tightcocone and powercrust which we used in this paper for comparison. We also would like to thank the Geometry Center at the University of Minnesota for *Geomview*, which we used for viewing and rendering models.

References

[1] H.Edelsbrunner and E.P.Mücke. Three dimensional alpha shapes. *ACM Trans.Graph.*,13(1):43-72,January 1994.

[2] F. Bernardini, C. Bajaj, J. Chen and D. Schikore. Automatic Reconstruction of 3D CAD Models form Digital Scans. *Int. J. on Comp. Geom. and Appl.*, 9(4&5):327-370, August-October 1999

[3] Marek Teichmann and Michael Capps. Surface Reconstruction with Anisotropic Density-Scaled Alpha Shapes. *In Proceedings of IEEE Visualization*, pages 67--72, 1998.

[4] M. Melkemi. A-shapes of a finite point set. *In 13th ACM Sympsium on Computational Geometry*, Nice France,367-369,1997.

[5] Jones, M.C. Variable Kernel Density Estimates and Variable Kernel Density Estimates. *Australian Journal of Statistics*, 32, 361-371, 1990.

[6] Terrell, G.R. and Scott, D.W.. Variable Kernel Density Estimation. *The Annals of Statistics*, 20, 1236-1265, 1992.

[7] J.Matousek. On enclosing k points by a circle. *Information Processing Letters*, Vol.53,217-221,1995.

[8] David Eppstein, Jeff Erickson. Iterated nearest neighbors and finding minimal polytopes. *Discrete and Computational Geometry* Vol.11, 321-350,1994.

[9] Loftsgaarden,D.O. and Quesenberry, C.P.. A Nonparametric Estimate of a Multivariate Density Function. Annals of Mathematical Statistics, 36, 1049-1051, 1965.