Simplifying Polygonal Approximation of 2D Shapes

$\mathcal{P}$iyush $\mathcal{K}$umar
Dept. of Mathematics
I.I.T. Kharagpur

September 10, 1998

Postal Address:
A-309,LLR Hall of Residence, I.I.T. Kharagpur
Kharagpur, West Bengal
INDIA. PIN:721302
Telephone:
+91-3222-55221 to 55223,77390 to 77392 Extn:5616
+91-3222-55224, 77393 DID 5616
FAX: +91-3222-55303
$E-Mail$ Address: *piyush@acm.org*

Abstract

Shape Approximation requires simplifying a polygonalization of a smooth shape S with another polygonalization that has fewer number of vertices. The computed polygonalization should be homeomorphic to S and geometrically close to it. We present an algorithm that can provably simplify the initial polygonalization when the sampling density is high enough with respect to the feature size of the boundary of the shape S. Our algorithm is based on a simple weighted decimation algorithm which calculates it's weight from the circumradius-to-shortest edge ratio(r/l) that is a well known metric in mesh generation for producing good quality triangulations. Our Implementation clearly shows that our output surpasses our theoretical guarantees.

Keywords. Curve Simplification, Surface Simplification, Computational Geometry, Polygonal Approximation

1 Introduction

Smooth 2D shape representation by a piecewise linear approximation that has less number of vertices is a frequently encountered problem in various fields like cartography, computer vision, computer graphics, and web transmission of 2D shapes[9].

Naturally it's desirable that the piecewise linear approximation be perceptually pleasing. Our Algorithm while preserving the overall shape also guarantees theoretically that the piecewise linear approximation is homeomorphic to the 2D shape and that the simplified approximation never self-intersects. It is similar to an automatic adaptive approximation scheme which does not ask the user for tolerance value of any kind as is commonly done by polygonal approximation algorithms[14, 10, 13, 12]. The Algorithm is fast and simple and the error is theoretically bounded in terms of the local feature size.

Generally in a simplification process, a user ought to know how much approximation is required, either in terms of the number of vertices in the output or in terms of some tolerance value (in some distance measure). One of the drawbacks of my algorithm is that it stops at a particular output that approximates the 2D shape and can't simplify further however the approximation at which it stops further simplification is visually satisfying and has its error bounded by theoretical guarantees. It can stop at a denser approximation than what it calls it's base mesh but can't simplify a polygonalization more than it's base mesh which is again defined in terms of the local feature size. This base mesh could have applications in web based transmission of meshes. Progressive Meshes[8] send a single vertex down to the client and keeps on updating it till the user stops or the mesh is totally transferred. In our case a base mesh which approximates the shape well, is sent first and then it is updated with vertices. This shape(the base mesh) gives the client a clear idea of the

final output.

Polygonal Approximation algorithms generally require some kind of tolerance values at the vertices as an input to the algorithm. Many surface simplification methods suggest the use of tolerance values as a function of curvature of the surface [18, 16]. In this paper we suggest the use of tolerance as a function of the local feature size of the surface at the vertices. This leads to strong theoretical guarantees. This suggestion comes on account of the observation that the local feature size varies, depending not just only on the curvature of the surface but also upon the importance of the object perceptually. The algorithm also guarantees a bound in terms of the Hausdorff Metric.

Another very important aspect of the algorithm is it's generalization. It can be directly generalized to closed smooth curves and 2-manifolds in 3D(Although at this point of time we don't have theoretical guarantees for the approximations produced).Thus the approach has applications wherever topology preserving simplification is required.

Curve approximation algorithms[9] don't take into consideration the overall shape that may be inherent in the 2D shape. Simplifying a 2D shape boundary using a curve approximation algorithm can lead to problems like self-intersections and genus changes. Surface simplification algorithms can be used to do shape simplification in 2D but very few of them are as fast (Preprocessing in $\Theta(n)$ for shapes in 2D homeomorphic to a disc in 2D, $\Theta(nlogn)$ for other shapes, and $\Theta(log(n))$ for each vertex dropped) along with having strong theoretical guarantees.

Our algorithm takes its input as a sufficiently dense polygonalization of the 2D shape and outputs a rarer polygonalization whose vertices are a subset of the initial polygonalization. It's output is hierarchical. In other words it numbers the vertices while dropping and this could easily be used to construct a progressive mesh[8].

It has been a tradition in the field of polygonal approximation to show the output of the algorithm designed when the input data points are sampled on a unit circle[10, 14, 13]. In my algorithm's case, it can be theoretically proved that it will simplify the input polygonalization in this case better than any algorithm which outputs a subset of the input vertices.

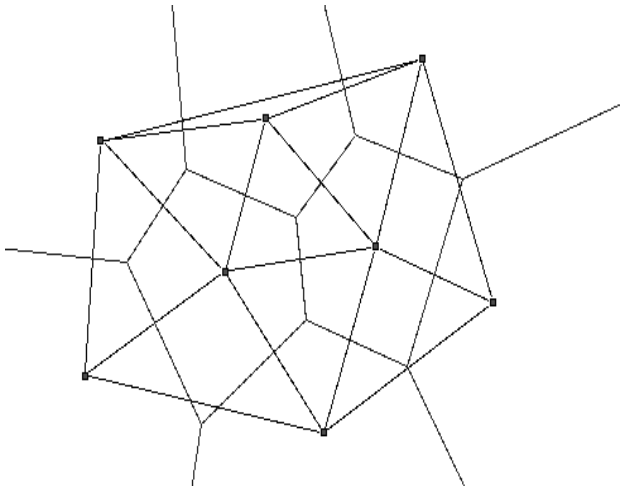

Figure 2.1: Delaunay Tessellation: Dual of the Voronoi Graph

2 Preliminaries

- *Topology*[15]*:*Topology is the theory of shapes which can stretch,compress, and bend, but without tearing or gluing. For example, a square is topologically equivalent(*homeomorphic*) to a circle, since a square can be continuously deformed into a circle. The capital letters of the alphabets "**D**" and "**O**" are similarly topologically equivalent.

- *Surfaces*[15]*:*A surface or d-dimensional manifold, is a shape any small enough neighborhood of which is topologically equivalent to a neighborhood of a point in $\mathbb{R}^d$. A circle and a sphere are examples of 1 and 2 manifolds.The surface of a cube is a surface topologically equivalent to the surface of a sphere. On the other hand if we put an extra wall inside a cube dividing it into two rooms, ,we no longer have a surface, because there are points at which three sheets come together.No small neighborhood of those points is topologically equivalent to a small neighborhood in the plane.

- *Voronoi Region*[5]*:*The Voronoi region of a point $s \in \mathbb{R}^d$ given a set of n points in $\mathbb{R}^d$ is defined as

  $$\textit{Vor(s)} = \{\text{p:}|sp| \leq |s'p|, \forall s' \in S\ \}$$

  where distance between points $x, y \in \mathbb{R}^d$ is $|xy| = ||x - y||$. This region is always convex.

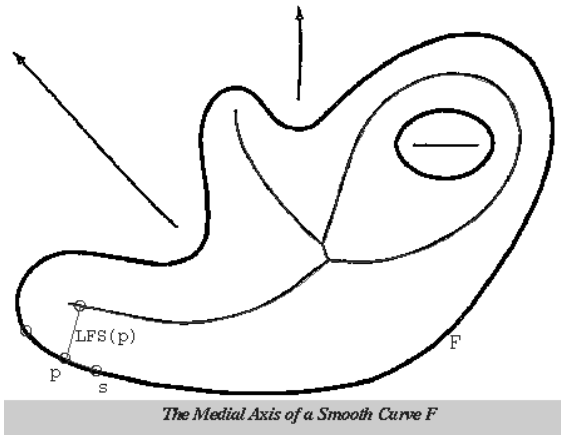

Figure 2.2: Local Feature Size of p with respect to the Medial Axis can also be seen.

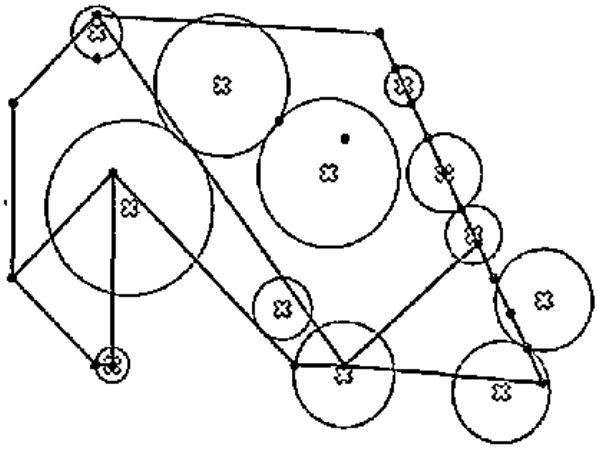

Figure 2.3: An Example diagram illustrating the Local Feature Size function of the centers of the discs.

- *Voronoi Graph*[5]*:*

$$VG(S) = \bigcup_{\forall p \in S} \partial Vor(p)$$

where $\partial Vor(p)$ denotes the boundary of Voronoi Region of point p. The Dual of the Voronoi Graph is a unique decomposition of the convex hull of S named the Delaunay Tessellation.See Figure 2.1.

- *Medial Axis*[1]*:* The medial axis of a (d-1)-dimensional surface F embedded in $\mathbb{R}^d$ is the closure of set of points which have more than one nearest neighbor in F.See Figure 2.2 An excellent intuitive perception is afforded by the famous grassfire analogy by Blum[6]. Let's assume that the shape Z is homogeneously covered with dry grass. Setting fire to the border of S will result in it's medial axis, the locus where the fire fronts meet and quench.

- *Polygonalization*[1]*:* A *Polygonalization* of F from S is a graph that connects every pair of samples adjacent along F, and no others.

- *Local Feature Size*[1]*:* The Local Feature Size of a point p is defined as the Euclidean distance of p to the (nearest neighbor of) *medial axis*. Ruppert introduced this function that is defined on all points. His original definition was for any PSLG Z. He defined the local feature size of a point p as the radius of the smallest ball that touches two features of Z that are not incident on the same point.See Figure 2.3.

- *r-sampled smooth curve*[1]*:* F is said to be r-sampled by a set S of sample points if no point p on F is farther than the local feature size of p from a sample point. This condition has the nice property that less detailed sections of the curve don't have to be sampled as densely.

3 Algorithm Characteristics

This chararacterisation is based upon the survey paper of Heckbert and Garland[9] on polygonal surface simplification algorithms.

3.1 Problem Characteristics

- **Topology and Geometry of Input:**Our input will be a polygon/polyhedra, whose vertices r-sample a smooth curve F.(F is a manifold without boundary) In 2D the input polygon should r-sample F with $r \leq 1$. If the input has $r > 1$ at some places, fictious points could be used to make $r \leq 1$. This can easily be done using the same concept that is used in Constrained Delaunay[2] where we insert mid-point of edges in the Delaunay Triangulation. But in generality we assume that r has a lesser value than 1 for the curve that needs a simplification.

  If the input is not a shape homeomorphic to a unit 2D disc, the r-sampling required is $r \leq 0.40$[1]. This

is required because the linear approximation of the boundary of the shape should be present in the Delaunay Triangulation of the vertices and the Voronoi vertices of the 2D shape. In 3D, although theoretically we can't give a sampling rate to begin with but intuitively we feel that this method can be used for simplification of reconstructed 3D surfaces[19].

- **Domain of Output Vertices:**Vertices of the output are restricted to be a subset of the input vertices.

3.2 Algorithm Characteristics

- **Speed/Quality Tradeoffs:**Algorithms that are optimal(minimal error and size) are generally very slow. It's known that $L_\infty-$optimal polygonal approximation of convex surfaces is NP-Hard.[7] Our Algorithm goes in for a tradeoff between speed and quality.
- **Refinement/Decimation:**Our algorithm is based upon the idea of incremental decimation which takes a fine to coarse approach. It discards details at every step and creates less and less accurate approximations.
- **Generalization:**Very few algorithms except for Optimal Algorithms(Where time complexity is very high) and Algorithms based on tessellation of simple geometric objects(for example the quad-tree based algorithms that depend upon the distribution of inputs.) generalize to surface simplification from curve simplification. We give an algorithm which is generalizable from the polygon simplification to polyhedra simplification although on a restricted class of inputs.

4 The 2D Case

Let us consider the case when F is homeomorphic to a unit 2D disc. Let P^n be a polygon $p_1, p_2, ..., p_n$ such that the set $\{p_1, p_2, ..., p_n\}$ r-samples F. Our aim is to incrementally drop vertices from P^n such that it's distance from F is in some sense bounded and so it retains the shape of F. We ultimately get a coarse polygon P^B which we call our base polygon and which we don't refine further. This base polygon or in general the base mesh P^B is not a very bad approximation of F as we shall see in our theoretical guarantees section. It is always topologically equivalent to F. The number of vertices dropped is controllable by the user but it can't go below the no. of vertices in P^B. Thus a sequence of k successive vertex drops yield a sequence of successively simpler meshes:

$$P^n \overset{Vdrop_1}{\rightarrow} P^{n-1} \overset{Vdrop_2}{\rightarrow} P^{n-2} ... P^{m-1} \overset{Vdrop_{n-m}}{\rightarrow} P^m$$

where m denotes the no. of vertices in P^B. We can retrieve the higher detail meshes from the simplest mesh P^B by using a sequence of vertex split transformations that are dual to the corresponding vertex drop transformations:

$$P^m \overset{Vsplit_0}{\rightarrow} P^{m+1} \overset{Vsplit_1}{\rightarrow} P^{m+2} ... P^{n-1} \overset{Vsplit_{n-m}}{\rightarrow} P^n$$

This idea is now well known in the graphics community as Progressive Meshes[8] PM's support progressive transmission and mesh compression. The continuous sequence of approximations $P^B ... P^n$ can be quickly transversed for interactive level-of-detail control.

5 Algorithm

1. Calculate the set S which is the union of points in P and the vertices of it's voronoi diagram. Delaunay Triangulate S.
2. Let v be a vertex in P. Associate a vertex q with v such that q is the nearest neighbor of v and $\exists$ an edge e between q and v st $e \in P$. Let $nn_P(v) = |vq|$. Similarly associate a vertex q' nearest to v st q' is a vertex of the voronoi diagram of P. Let $nn_V(v) = |vq'|$. Calculate

   $$r_{Approximated} = \frac{nn_P(v)}{nn_V(v)}, \forall v \in P$$

   Store the weights and their corresponding vertices in a Priority Queue. (Preferably a min-heap.)
3. Drop the root of the Priority Queue if $r_{Approximated} < 1$ for the vertex associated with the root node.
4. Update the $r_{Approximated}$ for the neighbors of the root vertex in P. Connect the adjacent vertices of the vertex dropped in P. Retriangulate the hole formed in the Delaunay Triangulation of S. Goto the Previous Step if $r_{Approximated} < 1$ at the root node or size of Priority Queue equals the size of the Output Mesh desired.

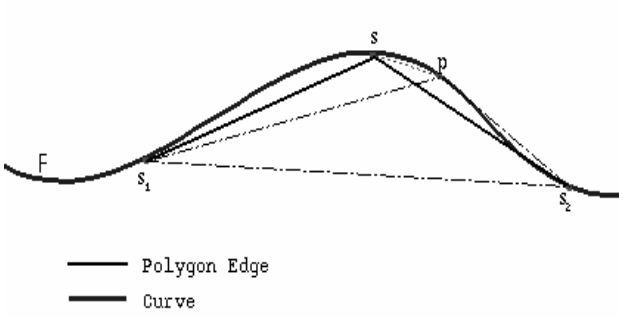

Figure 6.4: Diagram for proof of Lemma 6.5

The above Algorithm takes O(nlogn) in both 2D and 3D for preprocessing(Step 1,2) and takes O(log(n)) for dropping a vertex(Step 3) from P_n that is for the transition $P_n \rightarrow P_{n-1}$.

In the special case when the algorithm has an input a simple polygon in 2D the preprocessing can be done faster (Linear time) using the algorithm in [17] to compute the Medial axis in Linear time.

6 Theoretical Guarantees

Lemma 6.2 follows from the triangle inequality[2].

Lemma 6.1 (Amenta,Bern,Eppstein[1]) Let F be an r-sampled smooth curve. If $r \leq .40$ then the Delaunay triangulation of the set of vertices of the polygon and it's voronoi vertices contains all the edges of the polygon.

Lemma 6.2 (Ruppert[2]) $f(q) \leq f(p) + |pq|$

Lemma 6.3 (Dey,Kumar[3]) $l(e) \leq \frac{2r}{1-r} f(p) \quad \forall e \in P \; if r < 1$

Lemma 6.4 (Amenta,Bern,Eppstein[1]) *The distance from a point p on an r-sampled smooth curve F to some point on the polygonal reconstruction of the samples is at most* $\frac{r^2}{2}$LFS(p)

Lemma 6.5 *The change in r is most if the edges incident on a dropped point s make an angle* π *with each other.*

Proof: Let p be a point in the portion of F over which $s_1 s_2$ are adjacent after removing s such that

$$d_{ps} \leq r f(p) \tag{1}$$

See Figure 6.4. Since s has been removed, the LHS of (1) changes to $d = min(d_{ps_1}, d_{ps_2})$. Let r' be the new value of r because of removal of s. If $r' > r$, it must be determined by certain p. If that be the case:

$$d_{min(ps_1, ps_2)} \leq d_{s_1 s_2} if r' \leq 1$$

By elementry geometry. From the above equation we get:

$$r' = \frac{d_{min(ps_1, ps_2)}}{f(p)} \leq \frac{d_{s_1 s_2}}{f(p)}$$

and $d_{s_1 s_2}$ is maximum when $\theta = \pi$ which clearly implies that r can change most if $\theta = \pi$. □

Theorem 6.6 *By dropping* $n/3$ *number of points on an r-sampled smooth curve using the above algorithm the new value of r follows the following inequality:*

$$\boxed{r' \leq r\sqrt{\frac{1+r}{1-r}}}$$

Proof: Let p be the point being dropped. By Lemma 6.5, the change in r is maximum if the incident edges of p make an angle π with each other. We consider this worst case. Using Lemma 6.4 we can say that the maximum distance F has from the two incident edges at p say e_1, e_2 such that $e_1 > e_2$ can be $\frac{r^2}{2} f(p)$. Let the maximum distance be denoted by D. Let the distance after the drop be D' and the sampling after the drop be r'. Let the other endpoint of e_1 be q.Then

$$\begin{aligned}
D &\leq \frac{r^2}{2} f(p) \\
&\leq \frac{r^2}{2}\left[|pq| + f(q)\right] \\
&\leq \frac{r^2}{2}\left[\frac{2r}{1-r} f(q) + f(q)\right] \\
&\leq \frac{r^2}{2}\left[\frac{1+r}{1-r}\right] f(q) \\
D' &\leq \frac{r'^2}{2} f(q) \\
D' &\leq D \\
\frac{r'^2}{2} f(q) &\leq \frac{r^2}{2}\left[\frac{1+r}{1-r}\right] f(q) \\
r' &\leq r\sqrt{\frac{1+r}{1-r}}
\end{aligned}$$

The drop of a point p from an r-sampled smooth curve can affect at most it's neighbors as far as the calculation

of r is concerned at each vertex. The Algorithm picks up the minimum r each time. This clearly implies that till the algorithm drops at least $\frac{n}{3}$ number of points from the polygon the transition from r to r' sampling doesn't take place. □

Corollary 6.7 The Hausdroff measure between the n-vertex polygon P from F and $\frac{2n}{3}$-vertex polygon P' from F follows the following inequality.

$$\frac{\mathcal{H}(P,F)}{\mathcal{H}(P',F)} \leq \frac{1+r}{1-r}$$

Where $\mathcal{H}$(A,B) is defined as

$$\mathcal{H}(A,B) = Max\{\sup_{a\in A} d(a,B), \sup_{b\in B} d(b,A)\}$$
$$Where\ d(a,B) = \inf_{y\in B} ||a-y||$$

Proof: By Lemma 6.4 $\mathcal{H}(P,F) = \frac{r^2}{2} f(p_k)$ Where p_k is a vertex of P such that $f(p_k) \geq f(p)\ \forall p \in \{\ v | v$ is a vertex of P $\}$. This clearly implies

$$\mathcal{H}(P,F) = \frac{r^2}{2} f(p_k)$$

Similarly

$$\mathcal{H}(P',F) = \frac{r'^2}{2} f(p_{k'})$$

Where $f(p_{k'}) \leq f(p_k)$. Using Theorem 6.6 we get

$$\frac{\mathcal{H}(P,F)}{\mathcal{H}(P',F)} \leq \frac{r'^2}{r^2} \Rightarrow \frac{\mathcal{H}(P,F)}{\mathcal{H}(P',F)} \leq \frac{1+r}{1-r} \quad (2)$$

□

7 Implementation

We implemented our Algorithm on an Indy Workstation (OS:Irix6.2) in C++ using libraries of C++ extensively (STL/Tools.h++/LEDA). The example output given here has a special relevance because we were actually designing a hand recognition system and the problem of simplification arose. After the segmentation of the image inputted, we could extract the outer edge of the hand. Now we intended to store this polygon and geometrically match it with a few other polygons(After the classification step yields these as a high probability solution). To reduce the matching time and storage requirements simplification of the polygons was required. Accuracy and Fastness both were equally important in this setup. This is how this algorithm was born.

Figure 8.5: Closed Input Polygon: 3000 vertices

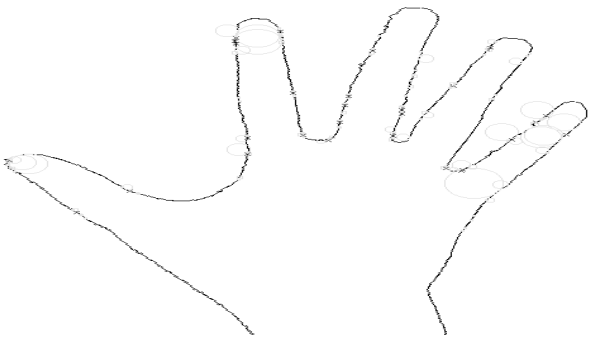

Figure 8.6: Output: 1200 vertices(To be replaced in the camera ready paper.)

8 Results

The three figures of the hand polygon(closed at the end points) clearly show that the algorithm works. The r-value for Fig: 8.7 $>> 1 \approx 5$.
Even at this value of r, it is very clear from the close up of the hand that the curved sections are well preserved as far as the shape is concerned. More Experimental results

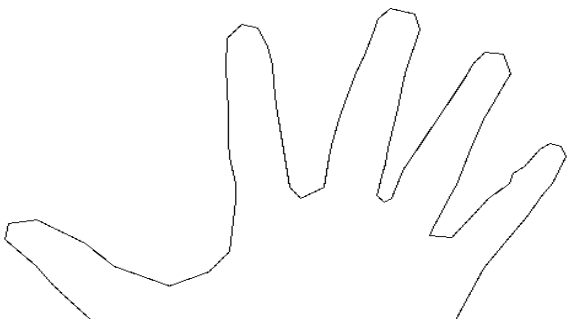

Figure 8.7: Cut out: 100 vertices

9 Conclusions

Our Implementation clearly suggests that the simple idea of using the Local Feature Size is a good choice for both the accuracy of the approximation and the perceptual pleasantness of the simplified shapes, although we cannot just go on dropping vertices one by one till the whole shape is reduced to a single point.

ACKNOWLEDGMENTS

I would like to thank V.Vetrivel, A. Bhartiya, N.N.Sharma and T.K.Dey for their helpful discussions. I would also like to thank the MPI-Saarbrucken,Germany for making their LEDA library available for our work and helping us with it.

References

[1] Nina Amenta, Marshall Bern and David Eppstein. The Crust and the β-Skeleton: Combinatorial Curve Reconstruction.*Graphics Models and Image Processing* **60** March (1998) 120-135

[2] Ruppert J., A new and simple algorithm for quality two-dimensional mesh generation. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms* (1993) 725-729

[3] Tamal K. Dey and Piyush Kumar. A simple provable Algorithm for Curve Reconstruction. *To Appear in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms* Jan (1999)

[4] D.Attali. Computing and Simplifying 2D and 3D Continuous Skeletons. *Computer Vision and Image Understanding* **67** (1997) 261-273

[5] H. Edelsbrunner. Algorithms in Combinatorial Geometry. Springer-Verlag, Heidelberg, 1987.

[6] H. Blum. A transformation for extracting descriptions of shape. *Models of the perception of Speech and Visual Form(W.WalHennDunn,ed.)* MIT Press (1967)

[7] H.Bronnimann and M.T.Goodrich. Almost optimal set covers in VC-dimension. In *Proc. 10th Annual ACM Symp. on Computational Geometry* (1994) 293-302

[8] H.Hoppe. Progressive Meshes. *Proc. of ACM-SIGGRAPH* (1996) 99-108

[9] Paul S.Heckbert and Michael Garland Survey of Polygonal Surface Simplification Algorithms.*Multiresolution Surface Modelling Course, SIGGRAPH* (1997)

[10] Laurence Boxer, Chun-Shi Chang, Russ Miller and Andrew Rau-Chaplin Polygonal approximation by boundary reduction.*Pattern Recognition Letters* **14(2)** Feb (1993) 111-119

[11] Leu, J.-G and L.Chen Polygonal approximation of 2D shapes through boundary merging.*Pattern Recognition Letters* **7(4)** April (1988) 231-238

[12] Kento Miyaoku and Koichi Harada. Note:Approximating Polygonal Curves in Two and Three Dimensions. *Graphics Models and Image Processing* **60** May (1998) 222-225

[13] David Eu and G.T.Toussiant On Approximating Polygonal Curves in Two and Three Dimensions. *Graphics Models and Image Processing* **56** May (1994) 231-246

[14] Chrles M.Williams. Note:An Efficient Algorithm for the Piecewise Linear Approximation of Planar Curves. *Computer Graphics and Image Processing* **8** (1978) 286-293

[15] John Conway, Peter Doyle, Jane Gilman, Bill Thurston Geometry and the Imagination in Minneapolis.
http://math.ucsd.edu/ doyle/docs/mpls/handouts/handouts.html June (1991)

[16] Greg Turk Re-tiling polygonal surfaces. *SIGGRAPH 92 Proc.(Computer Graphics)* **26(2)** July (1992) 55-64

[17] Francis Chin, Jack Snoeyink and Cao An Wang Finding the Medial Axis of a Simple Polygon in Linear Time. *Discrete Computational Geometry* http://www.cs.ubc.ca/spider/snoeyink/papers/papers.html (1997)

[18] J.Cohen, A.Varshney, D.Manocha, G.Turk, H.Weber, P.Agarwal, F.Brooks, W.Wright Simplification Envelopes.*SIGGRAPH 96 Proc.(Computer Graphics)* July (1996) 119-128

[19] Nina Amenta and Marshall BernSurface Reconstruction by Voronoi Filtering. In*Proc. 14th Annual ACM Symp. on Computational Geometry* (1998) 39-48