Optimization of reverse engineering data for regeneration of triangular mesh

By

Ashish Deb, Toronto, May 05, 2003

A project report

presented to Ryerson University

in partial fulfillment of the

requirement for the degree of

Master~~s~~ of Engineering

in the Program of

Mechanical Engineering

Toronto, Ontario, Canada, 2003

Declaration

I hereby declare that I am the sole author of this project report.

I authorize Ryerson University to lend this project report to other institutions or individuals for the purpose of scholarly research.

Ashish Deb

I further authorize Ryerson University to reproduce this project report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Ashish Deb

Borrower's page

Ryerson University requires the signatures of all persons using or photocopying this project report. Please sign below, and give address and date.

Abstract

Surface reconstruction of 3d reverse engineering data through the application of a triangulated mesh is a popular method. This is because triangles are the simplest polygons for computations. The purpose of a triangular mesh is to cover the point set with a surface mesh of triangular facets. During the last decade, reverse engineering and surface reconstruction have received much attention from manufacturing industries. Many algorithms have been developed to create a mesh, but algorithms of mesh are often poorly shaped and thus prone with error.

This report proposes a new simple genetic algorithm, an artificial intelligence method, to optimize triangular mesh generation. This technique reduces the number of data points required to depict an object without sacrificing the details and accuracy.

Acknowledgement

I would like to express my sincere thanks to my Supervisor, Dr. Vincent Chan, for all advice and mentor-ship. He has always been there to offer helpful advice when I have been stuck on a problem and has also encouraged me a lot.

Finally, I would like to thank my wife, for all her support to finish up my Masters of Mechanical Engineering degree.

Ashish Deb
Mechanical Engineering Department
Ryerson University, Toronto
Dated May 05, 2003

Table of Contents:

List of Figures & Table

Chapter 1
Introduction

This chapter provides an overview and the contribution of the project. It highlights the basic concepts of triangulation, surface fitting, adaptive subdivision technique and optimization of reverse engineering data by a new concept: "adaptive subdivision technique by genetic algorithm".

1.1 Introduction

Triangulation is a surface fitting method, which divides the object into triangles. Because of its simplicity, this method is used in many different fields. Triangulation of surface reconstruction for reverse engineering achieves a superior result compared to other mesh generation techniques. But a triangular mesh of a given set of points is not unique. There are many possible ways to make surfaces using triangulation, depending on the desired properties.

This report focuses on the optimization of reverse engineering data through a new triangulation method.

1.2 Goal of this report

The main goal of this research is to build a model of a large surface with the help of an adaptive subdivision technique. In this technique a genetic algorithm is used as an important tool to generate a comparatively smooth surface. The triangulation method for reconstructing surfaces in reverse engineering is still a very active area of research and could provide various levels of smoothness.

The following two main ideas should be considered before developing algorithm: .1). Simplify the mesh fast and then 2). Optimize the triangulation.

1.3 Overview of project

This report is organized in the following way: Chapter 2 provides basic ideas related to reverse engineering, mesh generation and genetic algorithms, triangulation, optimal triangulation and its operators, which are used to build different triangular meshes, general idea of mesh simplification and Delaunay triangulation. It also discusses previous works on non-artificial intelligence and genetic algorithm based artificial intelligence techniques. Chapter 3 contains detailed description of the proposed algorithm, to build a model from reverse engineering data. Chapter 4 contains experimental results and Chapter 5 presents the conclusions and recommendations.

1.4 Contribution of this project:

The principle contributions of this project are:

1. It presents an algorithm (phase 1) for assembling the top and the bottom scans of a model.
2. It introduces a new concept (phase 2) to organize more points at curved regions and fewer points at plain regions before optimization of the triangulation.
3. It introduces a new genetic algorithm (phase 3) for optimizing triangles for surface reconstruction without sacrificing accuracy.

Chapter 2

Background

This chapter provides basic idea of reverse engineering, its complexity and mesh generation, properties of meshes, optimization of mesh by genetic algorithm and its comparison with traditional methods. It presents the detailed information about triangulation, properties of optimal triangulation, internal operators, types and simplification of triangulations. Lastly, it describes the properties and technique for computing Delaunay triangulation.

2.1 Reverse Engineering

Reverse engineering technology creates or clones an existing part by reconstructing its surface geometry in a three-dimensional digital file using a scanning or measurement device [1]. Generally, three steps are considered in reverse engineering:

1. Applying some input device or technique to collect the raw geometry of the object. These data are usually in the form of (x, y, z) points on the object's surface relative to some local coordinate system.
2. Using some techniques to read these data and to convert those into a usable form.
3. Transferring the previous result into some 3D modeling or application software, to perform the desired action on the geometry.

But problems may occur in the following the main steps:

1. Digitizing the necessary geometric data from physical objects.
2. Fitting those data into surface models.

The digitizing process often provides much more data than are needed to accurately recreate the surface or object. In order to use such data in a real time visual simulation, a significant reduction in the data is needed to accurately render the sample surfaces.

2.1.1 Digitizing/Scanning

Scanning is used to gather data on an undefined 3-dimensional surface. During the scanning process, a scanning probe is commanded to contact and move back and forth across an unknown surface. During this process, the system records information about the surface in the form of numerical data. This data may then be used to create a geometric variant of the shape. The terms "digitizing" and "scanning" are often used to describe the same process. Traditionally, scanning is referred to as the process of taking discrete points and forming a surface using a touch-trigger probe. However, with the introduction of new technologies in capturing data such as laser, camera, vision systems, the term digitizing is now used as the generic description for the process of acquiring data from undefined surfaces [2,3].

The purpose of the digitizing system is to digitize a set of multiple points-x, y, z coordinates on the object's surface. 3D digitizers are of two types: contact and non-contact system. Contact digitizers are based on contact between the object's surface and the measuring tool. Some of them are operated manually using a hand-guided probe and some others are run automatically, such as CMM, ultrasonic digitizers. Non-contact digitizers have become popular recently which use structured light for acquiring the point data set.

Once digitized, point data sets have to be converted into a surface. These points do not provide information about adjacent points within the data set. Therefore, algorithms are used for reconstructing surfaces starting from unorganized points.

2.1.2 Fitting data into surface model

Another data-processing problem arises when digitization creates large numbers of unnecessary data points. These data can not be easily manipulated or are necessary [4]. The raw image data should be thinned to make the conversion to a CAD model easier. There are different strategies for completing this step of the process. The ultimate goal is developing a surface mesh with the minimal set without sacrificing surface details.

2.2 Mesh generation

A mesh is the discretization of a geometric domain into small simple shapes. These objects include a). Triangles and quadrilaterals in two dimensions and b). Tetrahedra and hexahedra in three dimensions. These objects can be represented as a set of points or a set of connections between these points. The combination of points and connections forms a mesh.

2.2.1 Quality of mesh

There are numerous criteria to define the quality of a mesh. A few of these criteria applicable to planar/surface meshes are [5]:

1. The variation in the area of the elements should not be large.
2. The aspect ratio of triangle elements should be as close to 1 as possible.
3. The ratio of the largest to the smallest edge/angle of the element should be close to 1.
4. The ratio of area of the largest and smallest element to all immediate neighbors should not be drastically high or low.

2.2.2 Types of mesh

Meshes can be categorized as structured or unstructured. A structured mesh is one where all interior vertices are topologically alike. An unstructured mesh is one where vertices may have arbitrarily varying local neighborhoods. A hybrid mesh is formed by a number of small structured meshes, combined in an overall unstructured pattern.

In general, structured meshes offer simplicity and easy data access, while unstructured meshes offer more convenient mesh adaptability and a better fit to complicated domains. On the other hand, hybrid meshes might be somewhere in between.

Two-dimensional structured meshes usually use quadrilaterals, while unstructured meshes use triangles. In three-dimensional cases the analogous element shapes are hexahedra and tetrahedra. However there is no essential reason for structured and unstructured meshes to use different element shapes.

Structured meshes offer simplicity and efficiency. A structured mesh requires significantly less memory, a factor of three less than an unstructured mesh with the same number of elements [5]. However, it can be difficult or impossible to compute a structured mesh for a complicated geometric domain. Furthermore, a structured mesh may require many more elements than an unstructured mesh for the same problem, because elements cannot change in size as rapidly. The advantages of unstructured meshes are flexibility in fitting complicated domains, rapid grading from small to large elements and relatively easy refinement and de-refinement. Three principle approaches to unstructured mesh generation are used: Delaunay triangulation, constrained Delaunay triangulation and quadtrees.

This report deals with unstructured triangular meshes only. The detail of triangulations and triangular mesh development will be discussed in the next chapter.

2.3 Genetic algorithm (GA)

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They were developed in an attempt to simulate some of the processes in natural evolution. Those algorithms converge quickly to the optimum structure with a minimum effort, having to test only a small fraction of the design space to find out either the near optimum or the optimum solution [6].

Genetic algorithms have many advantages over traditional search methods. Among other considerations, they do not need further additional information than objective function values or fitness information. This information is used to determine the success or failure of a design in a particular environment. Moreover, it is used as a powerful tool in non-conventional optimization problems due to its ability to find out better optimum solution.

2.4 Genetic algorithm vs. traditional methods

The simple genetic algorithm is described by Goldberg [7].“GA operates on a population of solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.”

The most significant differences of genetic algorithm are:

1. GA searches a population of points in parallel, not a single point.
2. GA does not require derivative information or other auxiliary knowledge. It needs only the objective function and corresponding fitness levels influence the directions of search.
3. GA uses probabilistic transition rules, not deterministic ones.

Major elements of Genetic Algorithm are as below and the detail of each term of the proposed algorithm is discussed in Chapter 3.

1. Population representation and initialization.
2. The objective and fitness functions.
3. Selection method.
4. Crossover.

5. Mutation.

6. Termination of the algorithm.

Genetic algorithms have been widely used in various fields such as space search, nonlinear optimization, machine learning, travelling salesman problems etc. Both theoretical and experimental studies by different authors' [8,9] show that the genetic technique is an efficient and robust heuristic for search in complex optimization problem. Therefore, it is possible to develop the best optimal 3D triangulation by using GA technique.

2.5 Triangulation

Triangulation is the division of a surface or plane polygons into a set of triangles with the restriction that each of the triangle's sides are entirely shared by two adjacent triangles. Triangular meshes are formed either from direct triangulation or via other extraction methods can often be poorly formed with regard to vertex distribution. The density of triangles can be varied in different areas on the surface to achieve good approximations of the physical surface areas.

Mathematically: "Given a finite set S of n points in the plane, a triangulation is a maximal set of non-intersecting line segments (called edges) that join points in S. Any triangulation of S partitions the interior of the convex hull of S into triangles. A triangulation of S can be viewed as a planar graph whose vertex set is S and whose edge set is a subset of S*S. The Euclidean relation for planar graphs implies that the number e (S) of edges and the number t (S) of triangles, do not depend on the way of triangulating S"[10]. In particular

$$e(S)= 3n-3-h \qquad (2.1)$$

$$t(S)= 2n-2-h \qquad (2.2)$$

Where, h denotes the number of edges bounding the convex hull of S. The number of different Triangulations of S is, however, an exponential function of n points.

Triangulations are also used in areas such as finite element methods or interpolation of numerical data, where each triangle defines a facet along which a projection of each point can be used in piecewise linear interpolation [11].

A triangular structure corresponding to a given set of points is not unique. Triangulation can be made in many ways. There are many different triangulation methods used for numerous and disparate applications. Different types of triangulations: Delaunay triangulation, Greedy triangulation, and Minimum weight triangulation are discussed in this chapter.

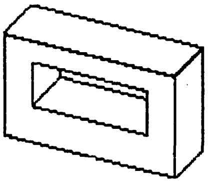


Figure 2.1: Object before triangulation

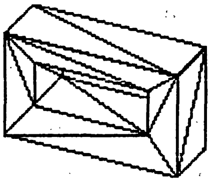


Figure 2.2: Object after triangulation

2.6 Application of triangulation

The necessity of triangulation arises in a wide variety of applications ranging from physics through meteorology to mathematics. The triangulations are used in the following places:

1. Planing the motion of robot
2. Computer graphics to perform hidden surface elimination
3. Rendering images
4. Computing magnetic field
5. Finite element analysis
6. Computer vision
7. Nearest neighbor computation
8. Interpolation of multi-variate functions by triangulating domain
9. Differential equation in chaos theory
10. Surface representation

A triangulation technique for surface representation of an object is the main concern of this report.

2.7 Generation of triangles

The most straightforward way to find an optimal triangulation (might be more than one) for a given problem would be to generate all possible triangulations and then compare them to each other. This is feasible for planar straight-line graph (PLSG) with few vertices. But when the number of vertices grow large, the numbers of different triangulations grow even faster. Let P be a convex polygon with n vertices and t_n is the number of different triangulations (t_I for i=3,4,5,6 are displayed in figure 2.3).

How many possible triangulations can be generated, is given by the following formula [12]:

$$t_n = \frac{(2n-4)!}{(n-1)!(n-2)!} \qquad (2.3)$$

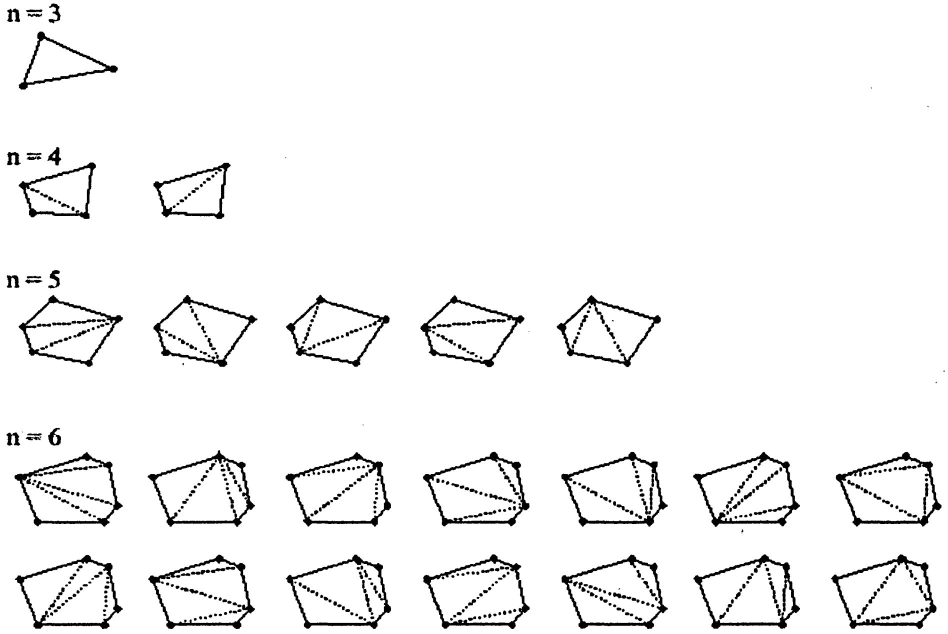


Figure 2.3:Different possibility of triangles with increment vertex

2.8 Optimal triangulation

Input of the mesh generation plays a significant role to find the best triangulation. If the input is a simple polygon, most optimal triangulation problems are solvable by dynamic programming. But if the input is a point set, polygon with holes, or multiple domains, these problems become much harder.

Different problems put different constraints on the shape of triangles. Some of the popular optimization criteria are:

1. Angle.
2. Edge length.
3. Average edge length.
4. Area.
5. Height.
6. Aspect ratio.

Many researchers have worked on the optimization of triangular mesh [13,14,15,16,17,18,19,20]. Two main classes of criteria are considered for optimization of triangulation. The first one is connected with the weight (defined as a sum of lengths) of edges, the second one with the size of angles. The most important criteria of the first class include the following-

1. Minimization of the triangulation weight.
2. Maximization of the minimal triangle height (minimum Euclidean distance from a vertex to the opposite edge). The height of a triangle is the minimum distance from a vertex to the opposite edge. A maximum height triangulation of S maximizes the smallest height of its triangles, over all triangulation of S. Although the maximum height, the maximum angle and the minimum angle criteria all tend to avoid thin and elongated triangles, they do not necessarily define the same optima.
3. Minimization of the maximum edge length.
4. Minimization of the maximal aspect ratio (the ratio of the length of the longest side of the triangle to the height of the triangle, where the height of a triangle is the Euclidean distance of the longest edge to its opposite vertex).

The second class includes the following criteria:

1. Maximization of the minimum angle (Delaunay triangulation).
2. Minimization of the maximum angle.
3. Maximization of the sum of the angles.

2.9 Local operators for simplification/optimization over triangle meshes

There are many surface operators, which are the key ingredient of the majority of surface simplification methods. The followings are the popular methods:

1. Vertex-removal: The vertex-removal operator takes as parameter the vertex to be removed and it preserves the topology of the mesh. The selection of the vertex to be removed is commonly based on curvature estimation at the vertex. Vertices with low curvature values are removed first.

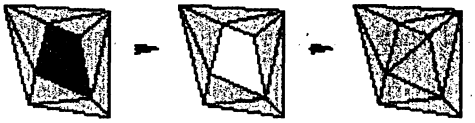


Figure 2.4: Vertex-removal

2. Edge-collapse: The edge-collapse operator takes as parameter the edge to be collapsed, or its equivalent, a pair of vertices sharing an edge. The two vertices are collapsed in one vertex. As a result of this collapse, the triangles sharing edge degenerate in a segment and are removed. The only computed parameter is the new vertex position, which usually is that of one of the two old vertices, or a weighted average. The selection of the edge to be removed is commonly based on an estimation of the error in the Hausdorff distance.

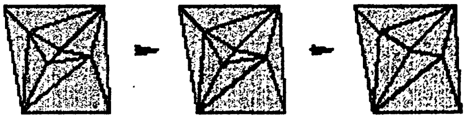


Figure 2.5: Edge-collapse

3. Vertex-clustering: The vertex-clustering operator takes as the parameter the two vertices to be collapsed. When these operators are connected along an edge, this operation is an edge-collapse, with disconnected vertices allowed. If the number of vertices is reduced the triangles sharing the edge degenerate in to a segment and are removed. The computed parameter is the new vertex position, which usually is that of one of the two old vertices or a weighted average. Unlike edge-collapse operators, the pair contraction does not preserve the topology of the mesh but the selection of the vertices to be collapsed is based on geometric proximity.

4. Face-removal: The face-removal takes as parameter the triangle to be removed. This triangle and all of its neighbors sharing vertices with the triangle are removed. The resulting hole is triangulated with the help of a new vertex. Due to the high number of triangles involved, incremental methods using face-removal rapidly arrive to a mesh which can not be further simplified while maintaining its validity.

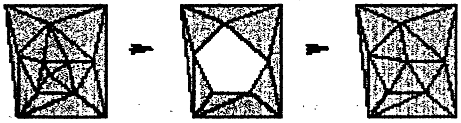


Figure 2.6: Face-removal

5. Vertex-displacement: The vertex-displacement operator takes as parameter the vertex to be moved. The only computed parameters are the new co-ordinates, usually given as an offset vector. It preserves the topology of the mesh and is used to locally improve the fitting of the simplification to the original surface.

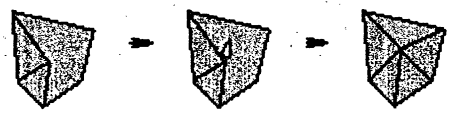


Figure 2.7: Vertex-displacement

6. Edge-flip: The edge-flip operator (also known as edge-swap) takes as parameter two adjacent triangles (or their shared edge). The edge-flip operator preserves the topology of the mesh. The edge-flip is used for two purposes, a). To improve the fitting of the simplification of the original surface, in non-flat regions, and b). To create well-shaped triangles, in flat regions.

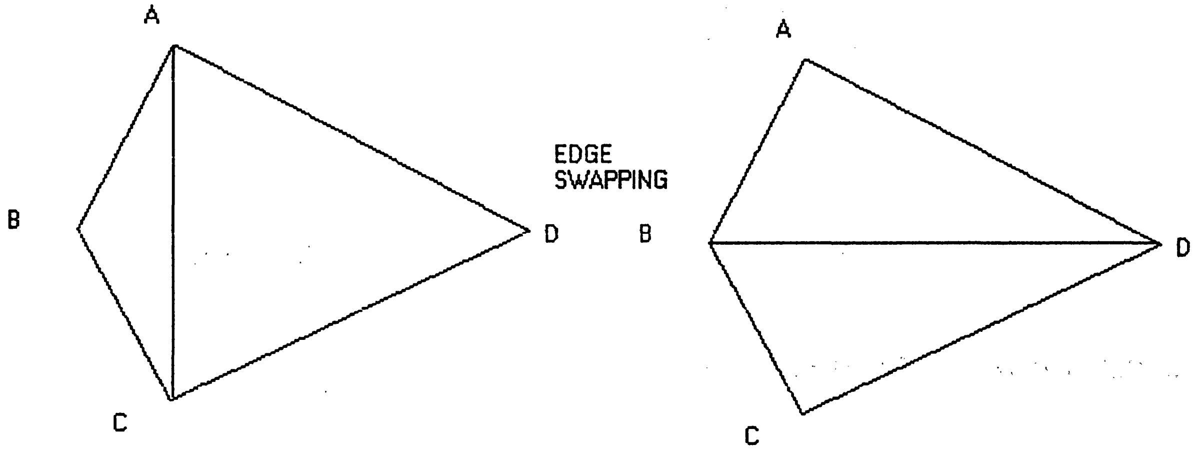


Figure 2.8: Edge-flip

2.10 Triangulation Methods

Triangulation is a surface fitting method where a polygon meshes can be obtained from the data. Different researchers developed different types of triangular meshes, which are discussed below.

2.10.1 Delaunay triangulation

Delaunay triangulation is the creation of a mesh through a set of points in the unique set of triangles such that the circumcircle of any triangle does not contain any other point [21]. Delaunay triangulation avoids long and skinny triangles. Detail is explained in Section 2.13 to 2.17.

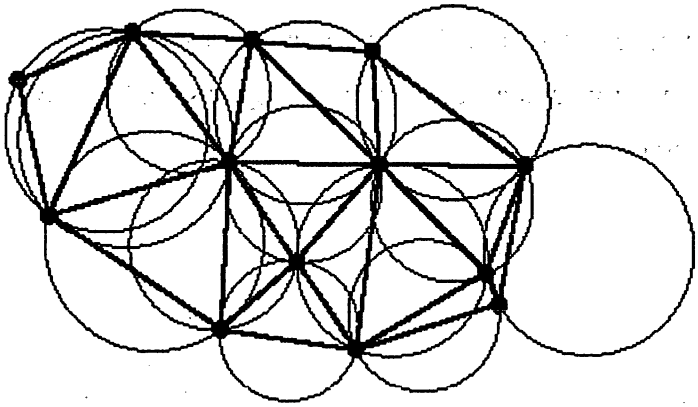


Figure 2.9: Delaunay triangulation with empty circumcircle property

2.10.2 Advancing front method

A popular family of triangle and tetrahedral mesh generation algorithm is the advancing front or moving front method. Starting with an initial triangulation along the faces of the domain generates a mesh. Then the mesh is extended inward, one layer at a time, to generate a full mesh [22]. In three-dimensions, for each triangular facet on the front, an ideal location for a new fourth node is computed. It also crosschecks for any existing nodes on the front that may form well-shaped tetrahedra with the facet. Either the algorithm selects the new fourth node or an existing node to form the new tetrahedron based on which will form the best tetrahedra. It also requires for intersection checks to ensure that tetrahedron do not overlap as opposing fronts advance towards each other. A sizing function can also be defined in this method to control element sizes. Simple advancing front method is explained in step by step process in figure 2.10,2.11 & 2.12.

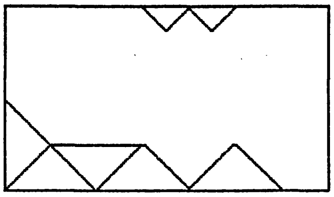


Figure 2.10: Stage-1

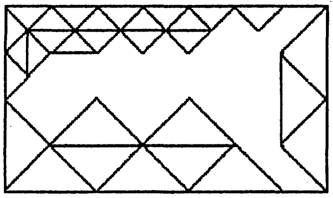


Figure 2.11: Stage-2

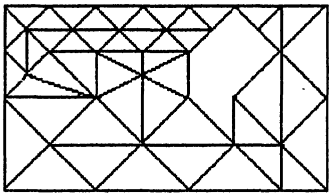


Figure 2.12: Stage-3

Advancing front algorithm [22]

2.10.3 Octree Triangulation

This is a 3D-grid generation method that involves subdividing a hexahedral domain along all three axes. In other words, it generates 8 hexahedra from one. The process can be repeated for each of the newly generated hexahedra. The octree technique does not match a pre-defined surface mesh, as an advancing front or Delaunay mesh might, rather surface facets are formed wherever the internal octree structure intersects the boundary. The resulting mesh will change as the orientation of the cubes in the octree structure is changed. Ensuring element sizes do not change too dramatically, a maximum difference in octree subdivision level between adjacent cubes can be limited to one. Smoothing and clean up operations can also be employed to improve element shapes [23].

2.10.4 Minimum Weight Triangulation

For some applications, it is useful to consider the triangulation that minimizes the total length of all edges in the triangulation (where the length of an edge is the distance between its endpoints measured by the standard Euclidean metric). Such a triangulation is referred to as the minimum weight triangulation [24]. If the edge is the costs, then a minimum cost triangulation will be a triangulation in which the total cost of the edges is minimum. If n is the total number of points and k is the number of points lying on the convex hull of the set of points, then any triangulation has exactly 3n-3-k edges.

2.10.5 Greedy Triangulation

The length of all pairs of points are calculated and sorted first in greedy triangulation algorithm. The algorithm builds the Greedy triangulation one edge at a time by examining each edge in increasing order of length, adding each edge if it does not cross previously added edges. If two or more distances between pairs of points in triangulation are exactly the same, greedy triangulation is not necessarily unique.

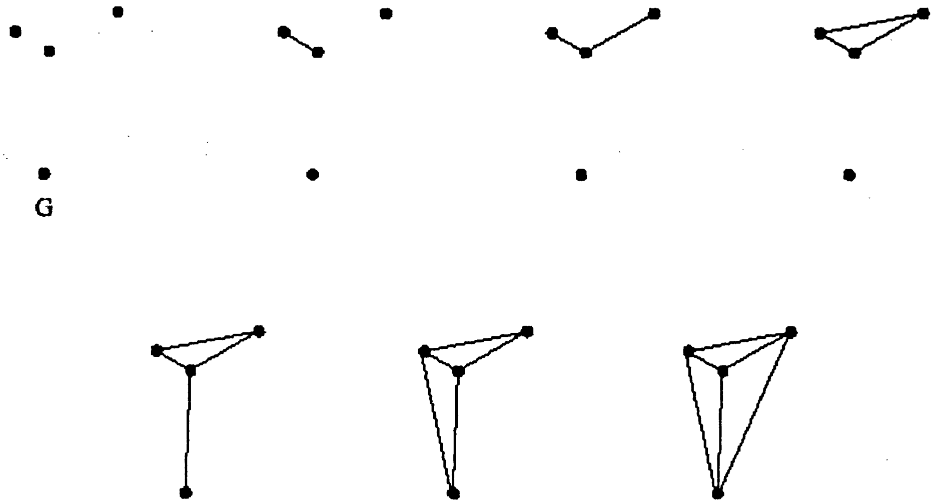


Figure 2.13: Simple greedy triangulation process is started at G

2.10.6 Locally Optimal Triangulation

A locally optimal triangulation (LOT) includes both the greedy triangulation and the minimum weight triangulation. Local optimality is one of the primary properties of the greedy triangulation that is used in analyzing its usefulness, and in developing algorithms to compute it. Relaxing the greedy edge-ordering requirement, but still requiring local optimality may lead to triangulation that are good minimum weight triangulation approximations, but are much simpler to compute.

2.10.7 Quadtree Triangulation

A Quadtree is a recursive partition of a region of plane into axis-aligned squares. One square covers the region that is to be partitioned. Each square may be divided into four child squares, by splitting it with horizontal and vertical line segments through the center of the square. Size of each child (length of the sides of the square) is proportional to half the parent's size. Thus the collection of squares forms a tree, with smaller squares at lower levels of the tree. A leaf square is one that has not been further subdivided into children. Each leaf square in the Quadtree has a set of neighbors, those leaf squares sharing either corner vertices or portions of sides with the square. A neighbor is orthogonal adjacent when it shares a portion of the square's sides, and diagonally adjacent when it shares a corner points only. The algorithm first builds a Quadtree covering the set of input points. Then it triangulates a rectangular region of the Quadtree, including all line segments bounding leaf squares as edges in the triangulation [25,26].

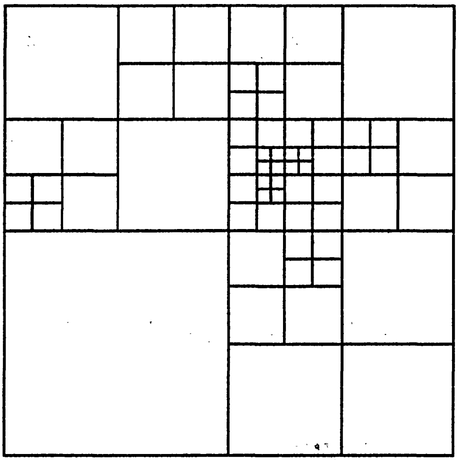


Figure 2.14: A simple Quadtree developing process

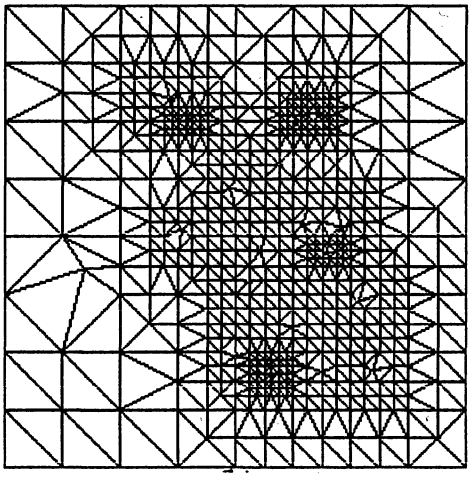


Figure 2.15: A Quadtree based triangulation [25]

2.11 Simplification of triangle mesh

Scanning a physical part and modeling the surface with millions of triangles generates meshes. Mesh decimation techniques have been developed to reduce the number of faces of a polygonal mesh, because of many downstream algorithms cannot handle these much data efficiently. Simplification is the act of transforming a three dimensional model into a simpler version. It reduces the number of polygon/triangles needed to represent a model while trying to retain a good approximation of the original shape and appearance.

As a general classification of algorithm can be in the following categories: Adaptive subdivision, Geometry removal and Sampling [27,28,29,30].

2.11.1 Adaptive Subdivision

An adaptive subdivision algorithm starts with a very simple base model and recursively subdivides it. Then it adds more and more detail to local areas of the model at each step. When the subdivided model approximates the original model to some user-specified degree, the algorithm stops.

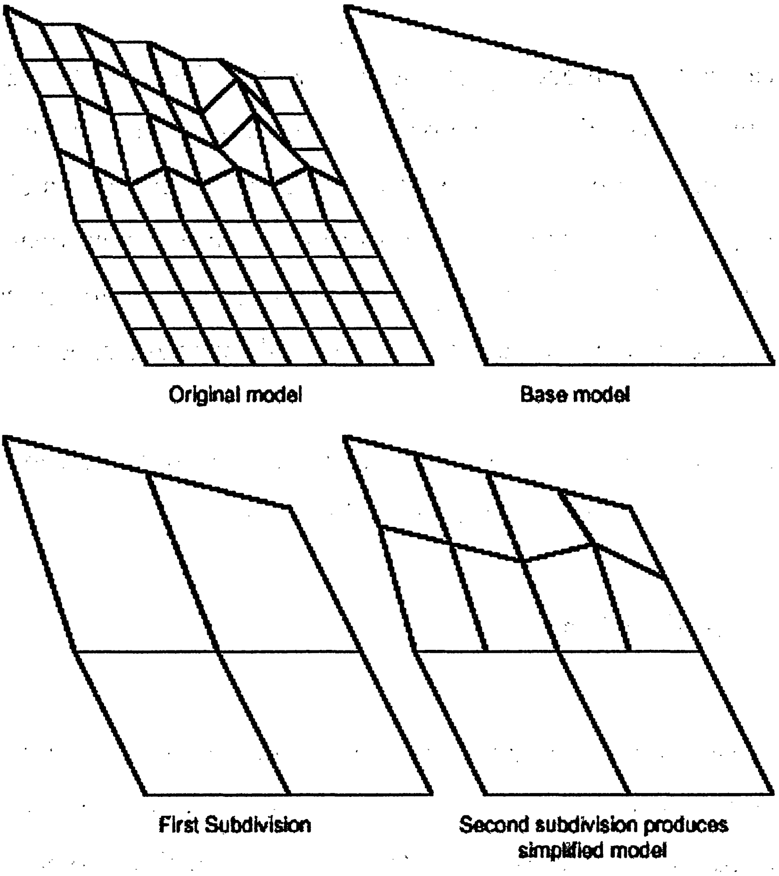


Figure 2.16: Adaptive Submission technique

2.11.2 Geometry Removal

A geometry removal algorithm starts with the original model and simplifies by repeatedly removing faces or vertices from the model. It removes geometry until the model can no longer be simplified and still meet some user-specified degree of approximation to the original. Most of these algorithms do not allow vertex and face removal operations that cause topology violations.

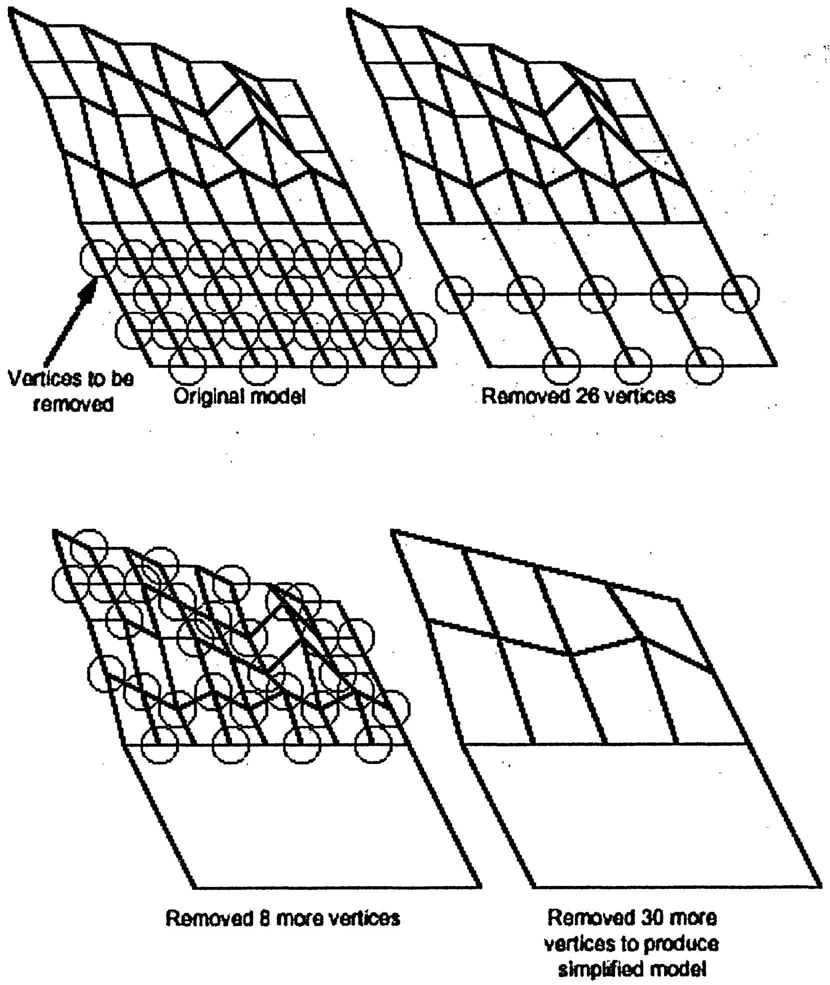


Figure 2.17: Geometry Removal Technique

2.11.3 Sampling

A sampling algorithm initially samples the geometry of the original model by either taking a certain number of random points from its surface or by overlaying the model with a three-dimensional grid and sampling each box of the grid. The algorithm tries to create a simplified model that closely fits the sampled data. The user can control the amount of sampling by specifying the number of points to sample or the dimensions of the overlay grid.

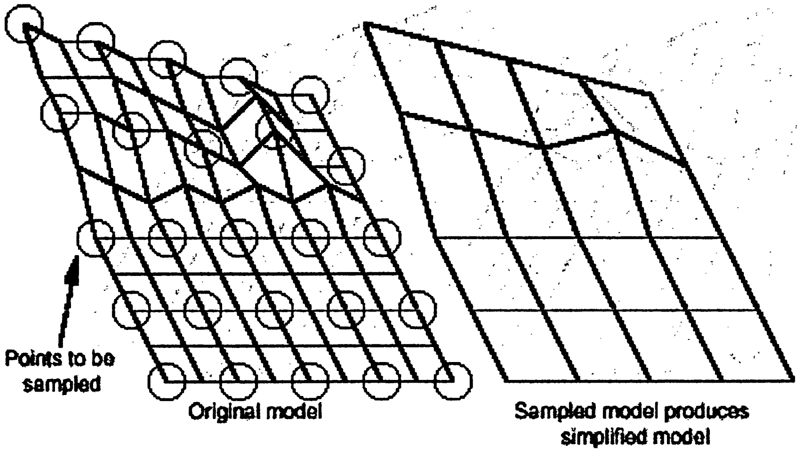


Figure 2.18: Sampling technique

The proposed technique in this report is a combination of adaptive submission and genetic algorithm. First, starting with a coarse representation of the object and then a refinement scheme is applied. Different combinations of adaptive decomposition can be used during triangular mesh generation:

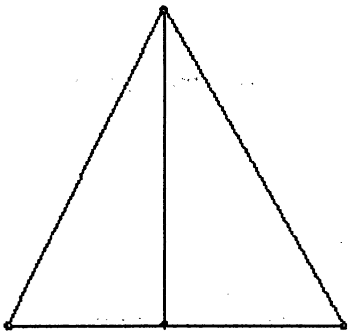


Figure 2.19: One edge split

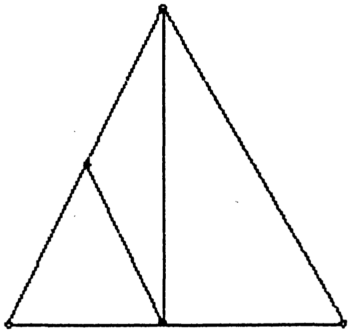


Figure 2.20: Two edges split

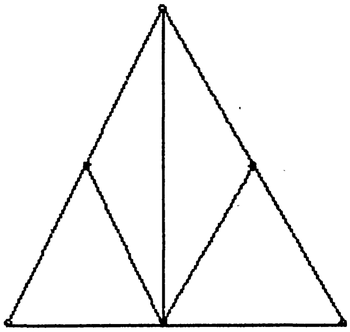


Figure 2.21: Three edges split

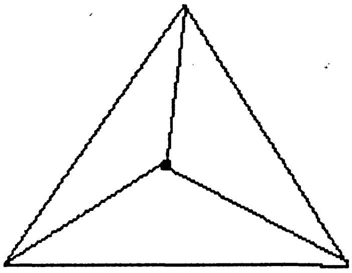


Figure 2.22: One vertex insertion

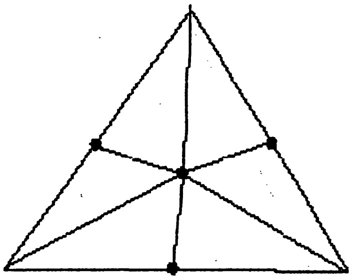


Figure 2.23: More than one vertex insertion

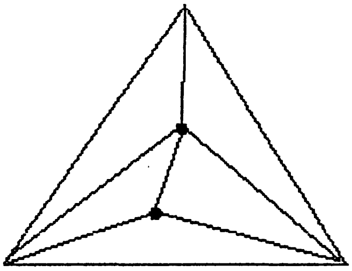


Figure 2.24: Edge-Vertex hybrid insertion

The following refinement scheme is used in this report. Edges are split in two sub-edges and re-triangulation of all the points or inserting new vertices at the centroid and re-triangulating all the vertices. The refinement scheme is a hybrid type, considering the equiangular concept of triangle as a fitness value by simple genetic algorithm.

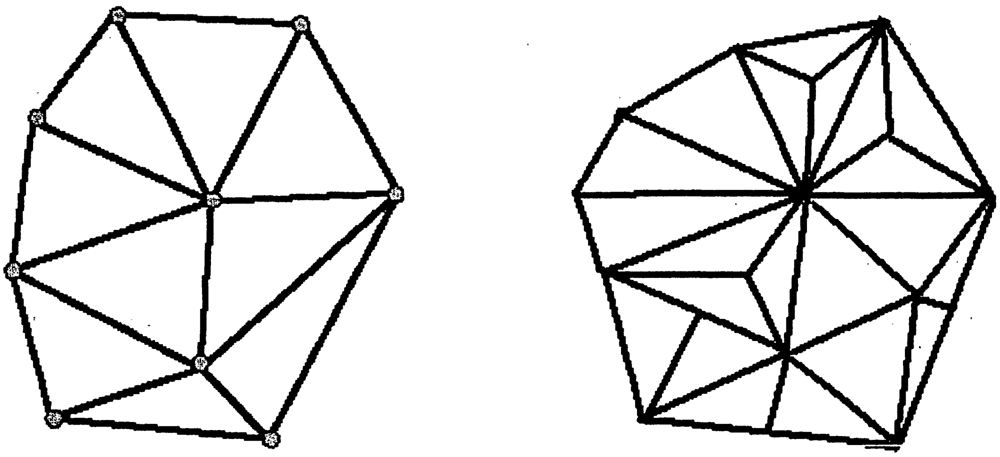


Figure 2.25: Insertion of new vertex for triangulation

2.12 Constraints of triangulation

The following constraints of triangulation are considered for making triangle meshes in this report.

1. The vertices are listed in counter clockwise order when looking at the object from outside.
2. Each triangle must share two vertices with each of its adjacent triangles.
3. No crossing is allowed between all sides of formed triangles.
4. Triangulation must be faithful to its geometry. All edges of newly formed triangles must be inside the blank region.

2.13 Definition of Delaunay Triangulation

The Delaunay triangulation is a geometric structure that has enjoyed a great popularity in mesh generation. There are many ways of generating the Delaunay triangulation, but one of the easiest methods; is the dual of the Voronoi diagram. For building Delaunay

triangulation, a line segment can be drawn between any two sites whose Voronoi regions share an edge. This procedure splits the convex hull of the sites into a triangle.
The Delaunay triangulation D of V (a set of vertices in a set T triangles), introduced by Delaunay in 1934, is defined as follows: "Any circle in the plane is said to be empty if it enclose no vertex of V (Vertices are permitted on the circle). Let u and v be two vertices of V. A circumscribing circle of the edge u, v is in D if and only if there exists an empty circumcircle of u, v. An edge satisfying this property is said to be Delaunay. In other words, the circumcircle of a triangle is the unique circle that passes through all three of its vertices. A triangle is said to be Delaunay if and only if its circumcircle is empty" [31].

A survey of Delaunay triangulations, their properties and algorithm shows that Aurenhammer, Bern and Eppstein, Fortune are considered superior for detailed information [32,33,34]. Delaunay triangulations can be computed either using flipping algorithm or divide-and-conquer, or incremental method or plane-sweep algorithms.

2.14 Properties of Delaunay Triangulation

The Delaunay triangulation has a number of desirable properties.

1. It's a dual to the Voronoi diagram, so computing one automatically gives the other.
2. The empty circle property: drawing a circle through the vertices of any Delaunay triangle, no other sites will be inside that circle.
3. It's a planner graph. By Euler's formula, it has at most 3n-6 edges and at most 2n-5 triangles.
4. It contains fat triangles, in the sense that the minimum angle of any Delaunay triangle is as large as possible.

2.15 Computing of Delaunay triangulation

The Delaunay triangulation criterion in itself, is not an algorithm for generating a mesh. It merely provides the criteria for which to connect a set of existing points in space. Therefore, it is necessary to provide a method for generating node locations within the geometry. A typical approach is to first mesh the boundary of the geometry to provide an initial set of nodes. The boundary nodes are then triangulated according to the Delaunay criterion. Nodes are then inserted incrementally into existing mesh, redefining the triangles or tetrahedra locally as each new node is inserted to maintain the Delaunay criterion. It is the method that is chosen for defining where to locate the interior nodes that distinguishes one Delaunay algorithm from another.

Most of the triangulations are based on Bowyer-Watson's algorithm [35], which is summarized here for 3-D applications as below:

At first, define the convex hull within which all points will lie.

1. Introduce a new point X_{n+1}.
2. Find all tetrahedra whose circum-sphere contains X_{n+1}; these are the tetrahedra that will be deleted.
3. Find all the points belonging to these tetrahedra.
4. Find all the external faces of the void that results when deleting these tetrahedra.
5. Form new tetrahedra by connecting the external faces to the new points X_{n+1}.
6. Add the new elements and the points to their respective lists.
7. Update all data structures.
8. If more points are to be introduced, go to step 2.

The algorithm is described above, assumes a given point distribution.

2.16 Delaunay Triangulation construction algorithm

Like all Delaunay triangulation methods, the Bowyer-Watson algorithm as discussed before assumes that the points to be triangulated are already known and so only addresses half of the mesh generation problem. The points can be pre-generated, say from the vertices of overlapping structured grids about component parts of the boundary, with some filtering and smoothing of the point distribution to get good triangulations. However, the most useful method is to generate the points simultaneously with the triangulation, choosing the points to improve the quality of the triangles. Several researchers worked on Delaunay triangulation [36,37,38]. Four fundamental algorithms are both theoretically important and reasonable for implementation, are discussed here.

1. Flipping algorithm

The flipping algorithm is the simplest algorithm for computing Delaunay triangulation in two dimensions. It starts with arbitrary triangulation of the set of points and produces the Delaunay triangulation by a sequence of local modification.

Let S be of $n \geq 3$ sites in E^2. The simplest algorithm for computing the Delaunay triangulation of S is edge flipping.

- Determine some triangulation of S
- While there are opposite triangle ABC and ACD that are not locally Delaunay, flip the diagonal, that is, replace the two triangles with triangles ABD and BCD. This flip is a Delaunay diagonal flip (same as figure 2.8).

2. The incremental algorithm

This algorithm is more complicated than flipping algorithm. Sites are added one by one, with the Delaunay triangulation updated to include the new site after each addition. The advantages of the incremental algorithm are that it generalizes readily to higher dimension.

3. The random incremental algorithm

This algorithm is comparatively new. It is the incremental algorithm with insertion order chosen at random with all permutations equally likely. The random insertion order guarantees good worst-case performance.

4. The plane-sweep algorithm

The plane-sweep algorithm is based on the plane-sweep paradigm. It provides a contrast because its analysis is based on different principles from the others. This will be discussed in more detailed, as this algorithm is used in Matlab software to build Delaunay triangulation.

5. Divide and conquer algorithm

This algorithm subdivides the area into two partial areas, computes recursively the Delaunay triangulation of the partial areas and merges finally both triangulations.

2.17 Plane-sweep algorithm

Steven Fortune [33] described the plane-sweep algorithm in his paper, "Voronoi diagrams and Delaunay triangulations". The plane sweep algorithm computes a planar Delaunay triangulation using a horizontal line that sweeps upwards across the plane. The algorithm discovers a Delaunay triangle when the sweep-line passes through the topmost point of its circumcircle as in figure 2.26. The Delaunay triangles shown in black have already been discovered. A sweep-line data structure stores an ordered list of sites, the entry for site s corresponds to an interval Is on the sweep-line where each maximal empty circle with topmost point in Is touches site s. The sweep-line moves in discrete steps only when the order changes. This happens when a new site is encountered or when a new Delaunay triangle is discovered (at the topmost point of the circumcircle of three sites that are consecutive on the sweep-line list).

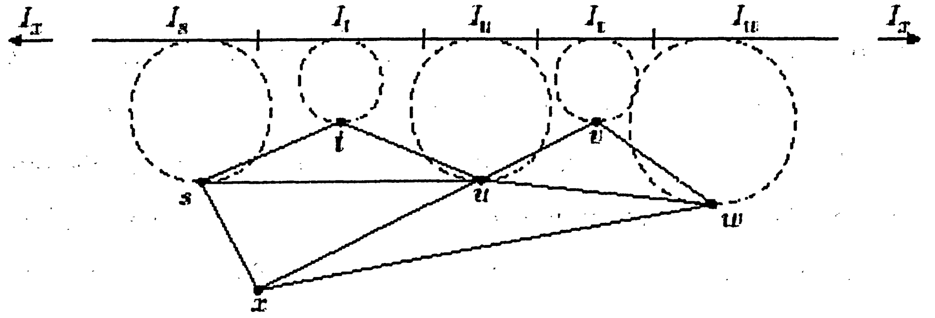


Figure 2.26: The sweep-line list is x, s, t, u, v, w, x. The next Delaunay triangle is tuv [33]

A priority queue is needed to determine the next sweep-line move. The running time of the algorithm is O (n log n) since the sweep-line moves O (n) times once per site and once per triangle and it costs time O (log n) per move to maintain the priority queue and sweep-line data structure.

2.18 Previous works

In recent years, many works related to optimization of surface triangulation have been published. But most of those are based on non-AI techniques. Genetic algorithm as an artificial intelligence based technique has been implemented successfully in different fields. A few researchers used genetic algorithms for developing surface triangulation. Some of the popular approaches which are non-AI and AI based (genetic algorithm) techniques, are discussed here.

- "Simplification of objects rendered by polygonal approximation" by Michael Dehaemer and Michael Zyda [39]: The input model obtained from a three-dimensional scanner is assumed. A scanner samples a model by returning a height value for each point along a regular grid of points. This method takes advantage of the regular layout of its input and thus can not simplify unstructured three-dimensional models. The user specifies an error tolerance in terms of Euclidean distance. The algorithm computes a simplified model that is within this error tolerance of the original.

  If the error tolerance is high, this method greatly simplifies the model. If the error tolerance is low, it marginally simplifies the model.

- "Re-tilling polygonal surfaces" by Greg Turk [40]: He described a new set of vertices, which could be distributed over the surface of a model and connected to one another. This creates a re-tiling surface that is faithful to both the geometry and the topology of the original surface. The main contributions of this paper are a). A robust method of connecting together new vertices over a surface b). A way of using an estimate of surface curvature to distribute more new vertices at regions of higher curvature and c). A method of smoothly interpolating between models that represent the same object at different levels of details.

This method is best suited to models that represent curved surfaces rather than models containing sharp discontinuities.

- "Decimation of triangular meshes" by W. J. Schroeder, J. A. Zarge, and W. E. Lorensen [41]: They described an algorithm called vertex decimation. The goal of the decimation algorithm is to reduce the total number of triangles in mesh, preserving the original topology and a good approximation to the original geometry.

  There are three steps of the algorithm:

  1. Characterize the local vertex geometry and topology
  2. Evaluate the decimation criteria
  3. Triangulate the resulting hole

  It uses a less expensive and less accurate error measure and it deletes multiple vertices per pass. Consequently it is faster but has lower quality.

- "Mesh optimization" by Hoppe Hugues [42]: A general method for automatic reconstruction of accurate, concise, piecewise smooth surfaces from unorganized 3D points are described. The reconstruction method has three major phases 1. Initial surface estimation, 2. Mesh optimization 3. Piecewise smooth surface optimization.

  The algorithm uses an energy function equal to the sum of three components: a distance term, a vertex term and a spring term which represent the trade off between geometric fit and compact representation. The distance term is proportional to the sum of squared distances between the sample points and the mesh being created. It measures the closeness of fit between the original and simplified mesh. The vertex term is proportional to the number of vertices in the simplified mesh and therefore penalizes a mesh containing too many vertices. The spring term simulates the effect of placing springs on the edges of the mesh being created. As the algorithm iterates, a gradual change of the spring term guides the energy function to a local minimum. The initial step randomly samples a specific number of points from the original mesh. It adds the original vertices of the mesh to this list of points.

No guarantee of finding a global minimum optimization mesh. This method consumes more memory than necessary by keeping a copy of the original mesh in memory for comparison to the current mesh at each iteration. This method is slow due to large number of computation but surface smoothness is better.

- "Constructing Hierarchies for Triangle Meshes" by Gieng, Joy, Schussman, Trotts [43]: They proposed an algorithm, which has three types-removing vertices, removing, edges and removing faces. The proposed algorithm is produced a hierarchy of triangle meshes that can blend different levels of detail in a smooth fashion through a set of triangle-collapse operations. They assigned a weight to each triangle of the original mesh, which depends on the absolute curvature, a shape measure, a topological measure, an error measure and the triangulation area. Users weight the four factors later. The triangles with the smallest weight are collapsed first. The criterion used when collapsing triangles is a certain percentage of triangles to be collapsed in a single step.

- "A genetic algorithm minimum weight triangulation of planar points" by Kaihuai Qin and Minglun Gong [44]: The authors presented an algorithm named Genetic Minimum Weight Triangulation, which is based on the rationale of a genetic algorithm. The computing result of genetic minimum weight triangulations are compared with greedy triangulation and proved more optimal than the greedy triangulation. The authors used user-defined adaptive constant, probability of crossover and mutation for different problem for speed up convergence of genetic minimum weight triangulation. But their algorithm is implemented for 2D triangulation only. They did not apply for 3D case. The authors future focus would be to develop better coding of mesh generation for finite element method.

- "Genetic algorithm, Another tool for Quad meshes optimization" by Mike Holder, Jim Richardson [45]: The authors investigated the use of genetic algorithm to perform a finite element analysis mesh smoothing process in this paper. It is the goal of this paper to take a simple quadrilateral mesh and smooth, using a genetic algorithm into a useful model that will provide a correct solution. A distortion metric is used to quantify the goodness of a quadrilateral element and serves as the fitness function for the genetic algorithm. This paper describes application of genetic algorithm to smooth poorly formed simple convex quadrilateral mesh. They implemented their algorithm for simple example case and will focus on taking advantages of the inherent flexibility and robustness of the genetic algorithm as problem solvers for 3D non-convex surface.

- "Geometric surface mesh optimization" by Pascal J. Frey, Houman Borouchaki [46]: They presented a surface mesh optimization method that is suitable for obtaining a geometric finite-element mesh, given an initial arbitrary surface triangulation. The first step of their method involves constructing a G^1 continuous geometric support, associated with the initial surface triangulation, which represents an adequate approximation of the underlying surface geometry. The initial triangulation is then optimized with respect to this geometry as well as the element shape. However, the procedure of mesh optimization is also a very time-consuming iteration scheme and not a real time mesh optimization technique.

- "Genetic algorithm for mesh optimization" by C. C. L. Wang, M. M. F. Yuen [47]: The authors described the problem of optimizing an arbitrary triangular mesh surface with respect to prescribed criteria to obtain a unit surface mesh, where all the edges have a normalized length close to the optimal value. They described the mesh optimization is an edged based optimization method. It involves subdividing the longest edges, collapsing the shortest edges (the edge length is computed with respect to the criteria of optimization) and swapping some edges to reach angle-optimal. Practically, this stage involves edge split, edge collapse and edge swap operations.
  The gap between the approximated mesh surface and the true surface is an important factor. They assumed local spherical surface is as true surface and find the gap between mesh surface and assumed true surface which might give inaccurate result which authors will solve in future research.

- "Multicriteria-optimized triangulations" by I. Kolingerova, A. Ferko [48]: In this paper the authors claimed that they developed a triangulation which satisfied more than one criteria or which contains parts developed according to several criteria. They said by geometrically based deterministic methods, it is not possible to compute a triangle mesh, which is a compromise between extremes, providing small weight as well as good angles. In addition, it is not possible to generate a mesh angularity optimized in some of the points but weight optimized in the rest. "By genetic approach, it can also construct a triangulation which optimizes a combined function of more criteria, e.g. to construct a triangulation which is 60 percent minimum weight triangulation and 40 percent Delaunay triangulation or a triangulation which optimizes both minimum angle and minimum weight as much as possible." The most important argument against their genetic algorithm is time requirement. If several thousand triangles are generated instead of one triangulation, computation time is worst than if a deterministic triangulation algorithm computing only one mesh. Authors claimed their algorithm is better if quality is preferred to speed (for 500 point's data it will take 12*60 times more compare to calculate the same by Delaunay triangulation) which is very expensive.

Speed, quality of triangulation and generality of solution are the main criteria of algorithm, which are developed by different researchers. After development of artificial intelligence, researchers are trying to implement AI technique for generation of accurate triangular mesh. The research on genetic algorithm and their applications are actually a very attractive field. Both theoretical and experimental studies by different authors show that the genetic technique is an efficient and robust heuristic for search in complex optimization problem. Hence the proposed simple genetic algorithm technique for optimization of reverse engineering data will be efficient for regeneration of triangular mesh. There are three steps in the algorithms: screening of reverse engineered data, organizing of data for putting more points in curvature zones and less in plain zone and then to develop a triangulation with the fewest number of points. Finally a simple genetic algorithm is implemented with optimal sacrificing of object accuracy for generation of optimal points and hence, less triangulation consequently.

Chapter 3
Problem Formulation

This chapter describes all the three steps of algorithm. Simple figures and flow charts are added to illustrate the details of algorithm. It introduces all the elements of a genetic algorithm relevant to the proposed algorithm.

3.1 Introduction of first phase algorithm

The scanning points of a leg implant are given data for this project. The leg implant is fixed on plane surface and the scanning machine is moved parallel on top of the leg implant with certain distance to collect the curvature of surface uniformly with a distance of 1/1000 inch in the x and y direction. A 3D scanner, Roland DR. Picza is used to collect the scanning data. Top and bottom scans are executed to fully digitize the leg implant.

The basic idea for developing the object with the help of scanning data (top and bottom data files are in matrix form) is as follows. At first, both the matrixes are rotated individually along x, y-axis to place the object in same platform. Then unwanted x, y, z co-ordinate were filtered from both the matrixes. Lastly, the bottom portion was rotated 180 degrees along x-axis, then rotated along x-y plane to match with the top portion, before combining the files.

Photographic view of leg implant is presented in figure 3.1 .The picture of top and bottom part of leg implant which is developed from Scanner Roland Dr. Picza as bmp. file, presented in figure 3.2 & 3.3, is not clear as figure 3.1.

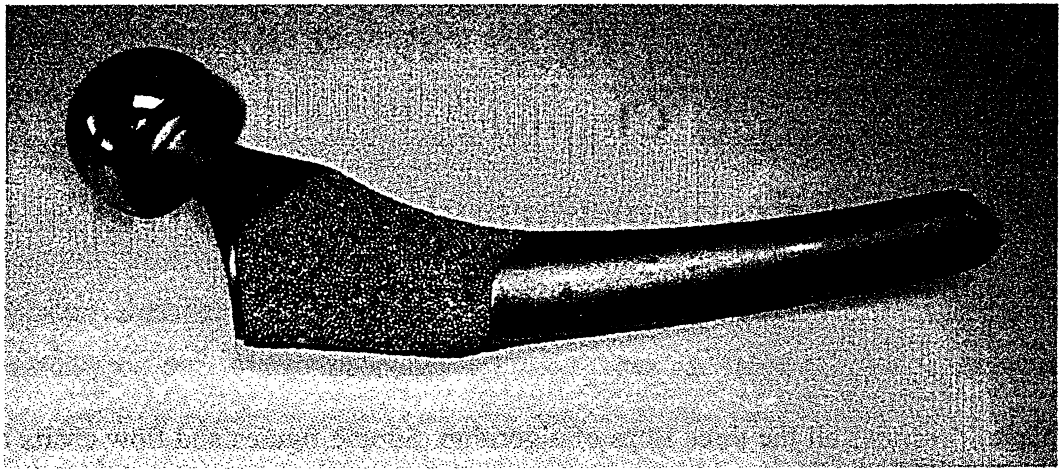


Figure 3.1: Photographic view of leg implant, which is scanned later on

3.1.1 First phase algorithm

For leg implant top:

1. Rotate the matrix along x-axis and y-axis to place in plain level.
2. Filter the scanning data of leg implant top with the following limit: x_max=184;x_min=7;y_max=71;y_min=4.5;z_min=15.5 unit.
3. Shift the data considering x_min and y_min as 0,0 co-ordinate and find out the value of x_max and y_max point.
4. Rotate the matrix along x-y plant to match with the bottom portion of the leg implant.

For leg implant bottom:

1. Rotate the matrix along x-axis and y-axis to place in plain level.
2. Filter the scanning data of leg implant bottom with the following limit: x_max=184;x_min=7;y_max=-72;y_min=1;z_min=-14 unit.
3. Shift the data considering x_min and y_min as 0,0 co-ordinate and find out the value of x_max and y_max point.

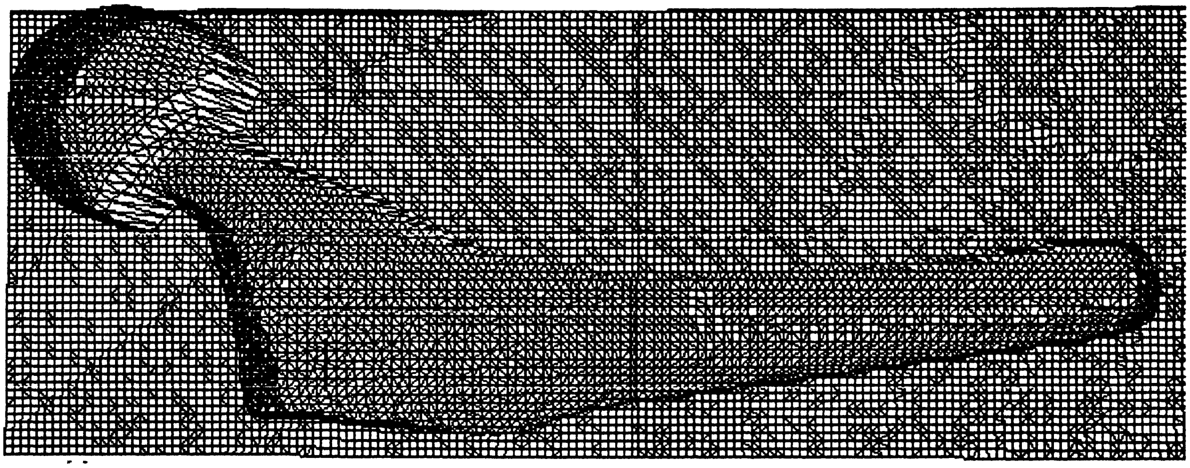


Figure 3.2: Scanning view of leg implant top (bmp. image)

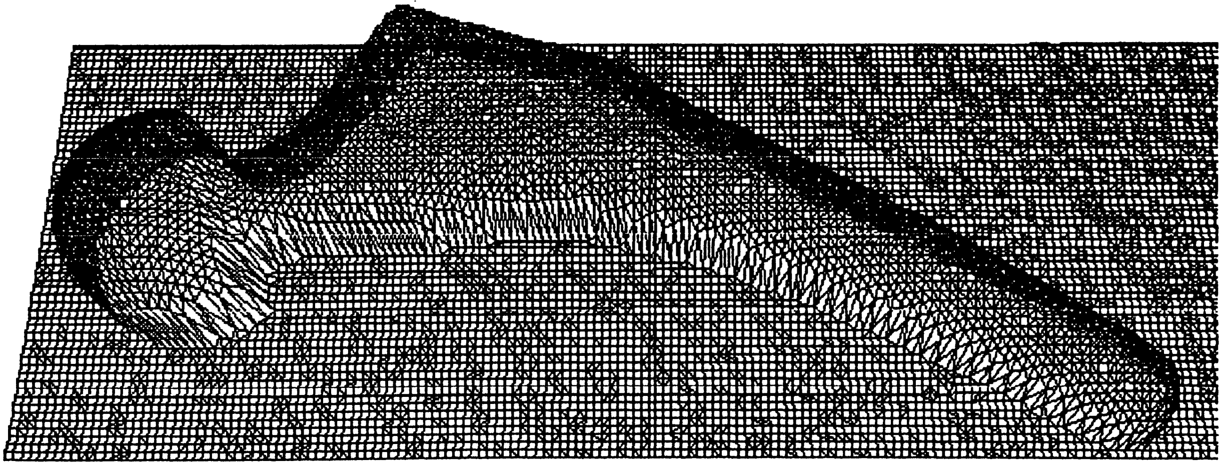


Figure 3.3: Scanning view of leg implant top (bmp. Image)

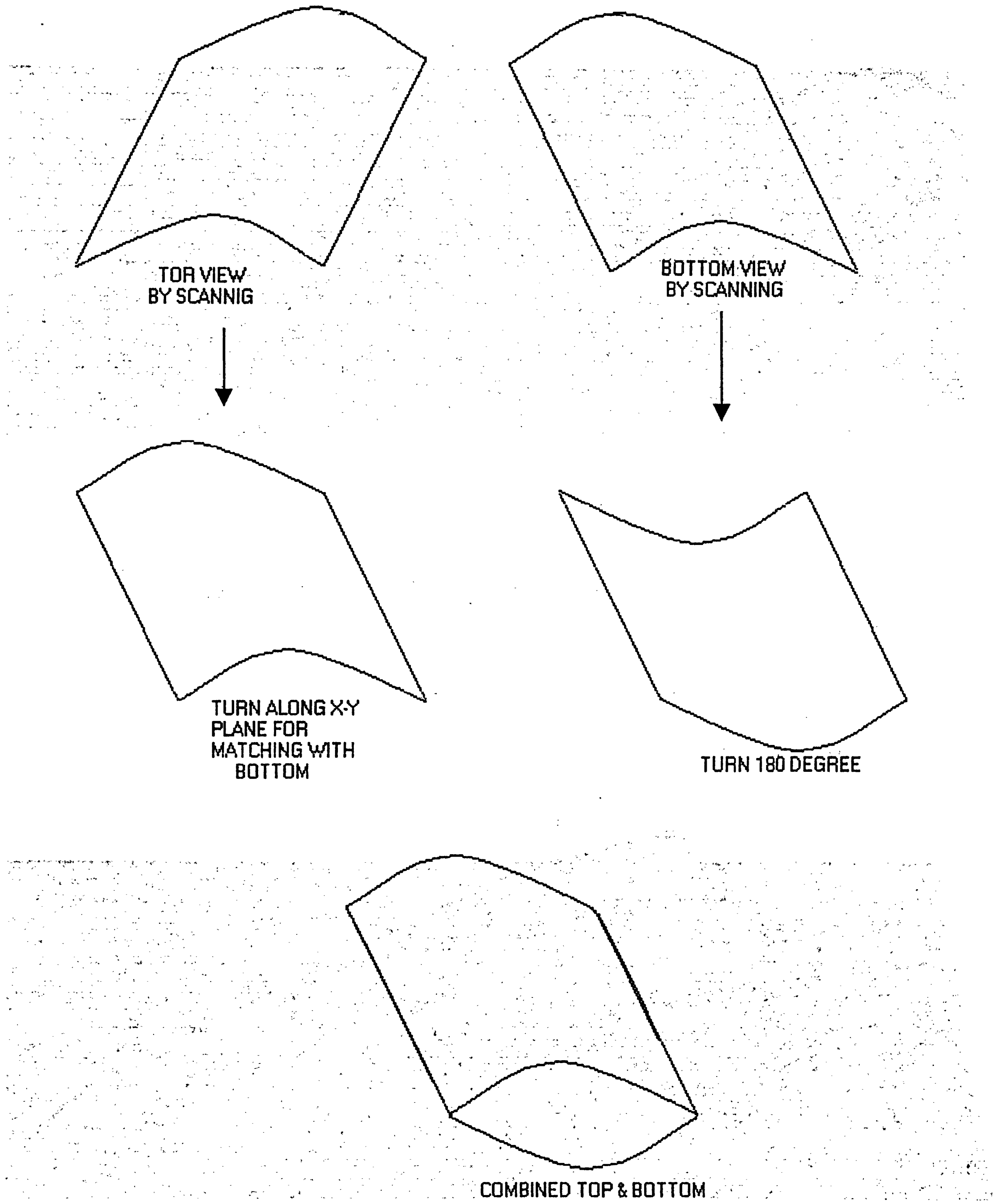


Figure 3.4: basic principle for 1st phase algorithm is illustrated here

4. Rotate the leg implant 180-degree along x-axis.
5. Flip the y-co-ordinate of the matrix to match with the top portion of leg implant.

Combine top and bottom leg implant:

1. Select 4 boundary points from both the top and bottom of leg implant and try to match as close as possible.

2. The Delaunay triangulation technique (based on Plane sweep algorithm by Fortune [33]) is used to create the triangular mesh of the given object. The bigger size triangles (based on total lengths of three edges) are filtered out to avoid unwanted triangles. Then the filtered points are collected for making appropriate triangular meshes.

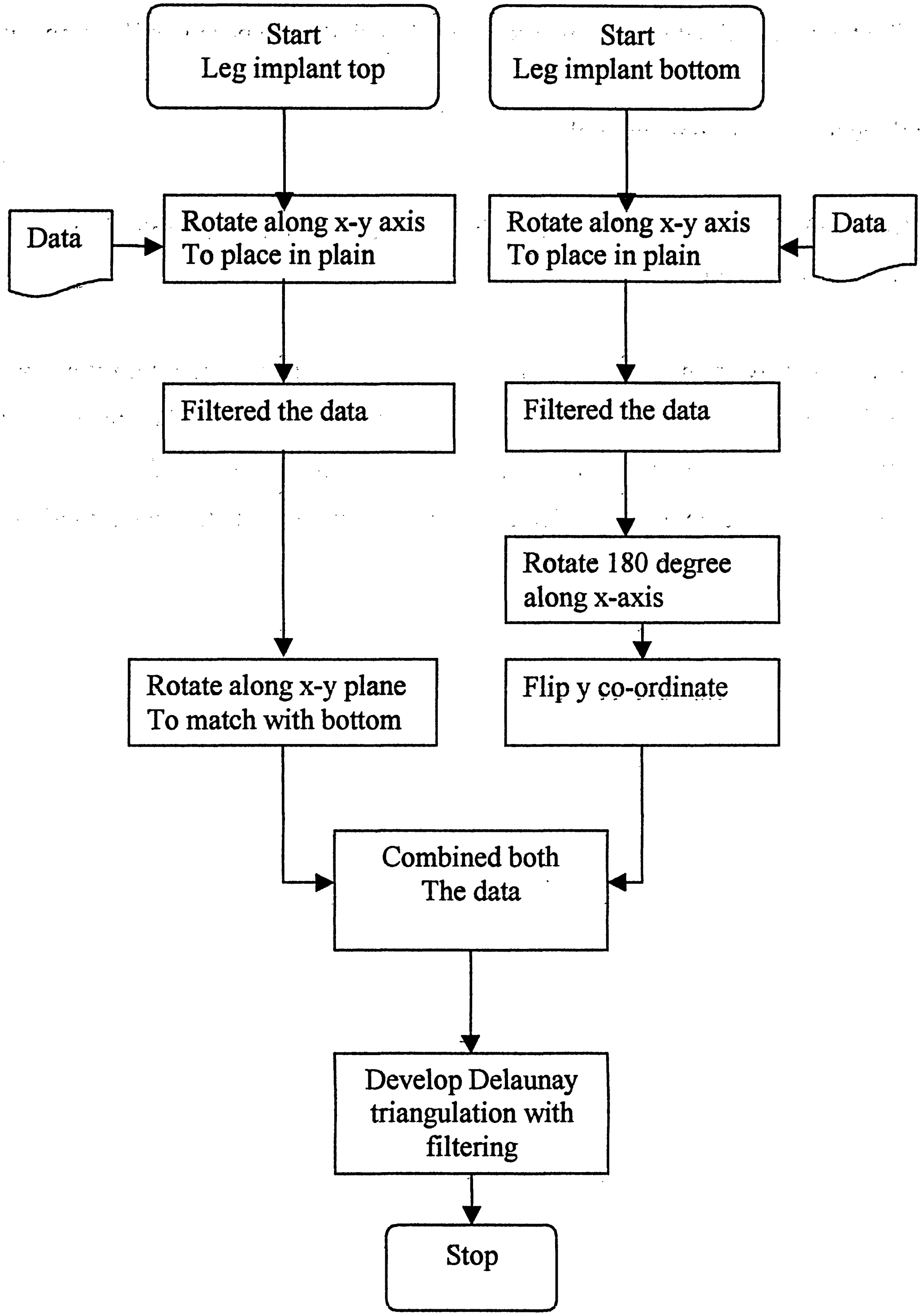


Figure 3.5: Flow chart for 1st phase algorithm

3.2 Introduction of second phase algorithm:

It is normal practiced in mesh generation to use fewer points in plain region and more points in curve region during mesh simplification process. Organizing more points at curve region before proceeding for optimized triangular mesh by genetic algorithm technique would be more attractive than conventional method.

3.2.1 Approximation Error:

The approximate error is found by calculating the difference between the Euclidean R^3 and the approximating 3-D triangular mesh. Each triangle will contain unknown number of points (x, y, z) of total number of points, after applying Delaunay triangulation with selected scanning points. The average difference of Euclidean distance can be calculated by the formula (3.1).

$$D_i = \left| mean(Z_{Inside}) - mean(Z_T) \right| \tag{3.1}$$

Where Z_{inside} defines the z co-ordinates of all the measured points inside each triangle and Z_T defines the z co-ordinates of the three vertices of each triangle. D is equal to the difference between the average of z co-ordinates and the average of z co-ordinates of three vertices in a triangle. E_i is defined, as (T-total numbers of triangles) is the sum of the squares of the Euclidean differences over all the scanning points inside T_i.

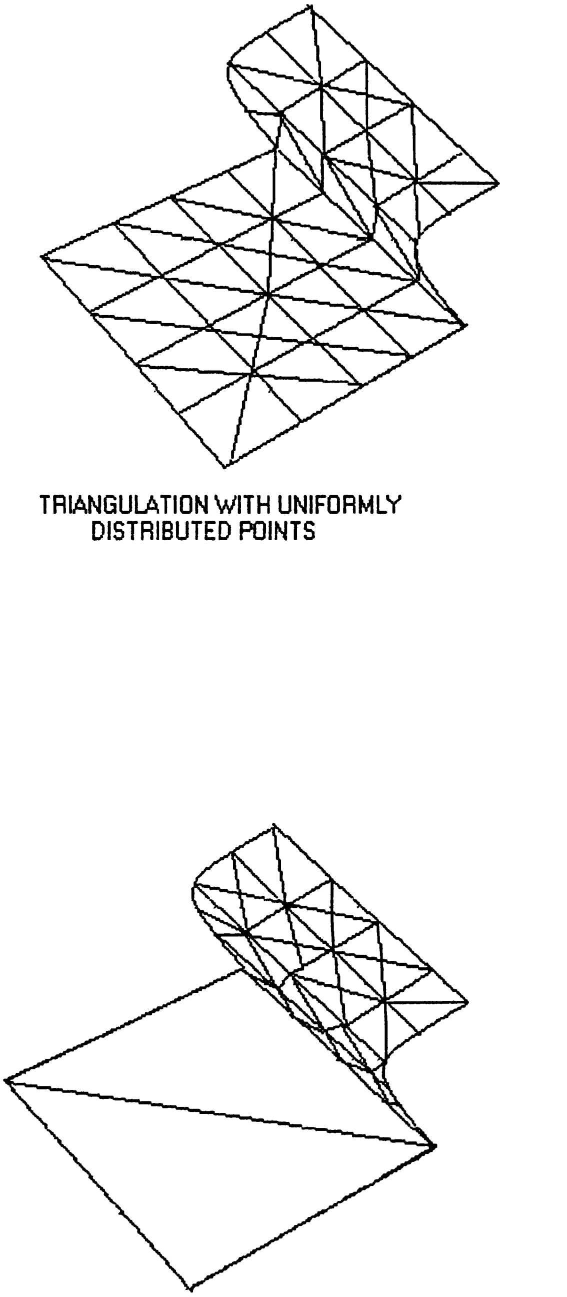
$$E = \sum_T D_i^2 \tag{3.2}$$

The total error is then defined by

$$E = \sum_{i=1:T} E_i \tag{3.3}$$

Constructing Delaunay triangular mesh with a limited number of uniformly distributed sample points can develop an approximate surface. But organizing the same amount of data in triangular mesh, original surface can be approximated better than previous one by using approximation error technique. To reduce the approximation error, the number of points in low curvature regions has to be balanced with those points in high curvature regions.

The portions of high curvature are more difficult to approximate with linear interpolations, thereby causing larger errors than lower curvature regions. The vertex of a triangle with smaller error is randomly deleted while another point is placed inside triangle with larger error; therefore the total of points is unchanged. As a result, organized points will be developed whose smaller number of points are in plain region but higher numbers of points in high curvature region.



TRIANGULATION WITH UNIFORMLY DISTRIBUTED POINTS

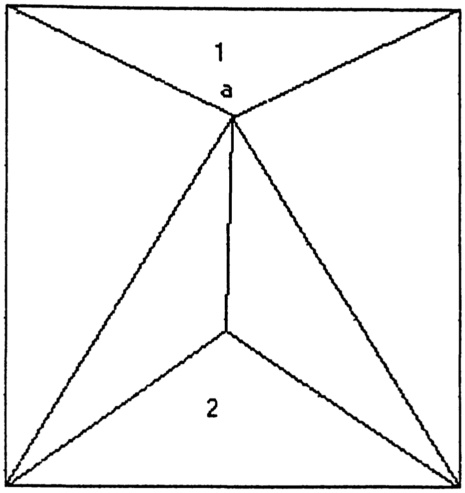
TRIAGULATION AFTER ORGANIZING POINTS CURVATURE ZONE & PLAIN ZONE

Figure 3.6: Organizing points in curvature & plain zone

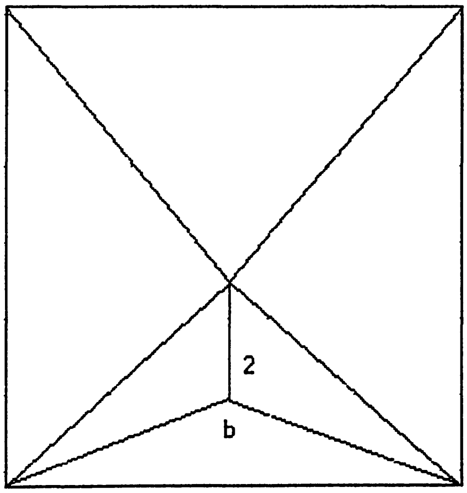
The selection of points for deletion and reproduction is done according to the evaluation of error in the triangulation in this phase of the algorithm.

The second phase algorithm is as below:

1. Collect points from given data, which should be uniformly distributed.
2. Considering the points to generate Delaunay triangulations.
3. Calculate error E_i for each triangle $T_{i.}$
4. Select T_i whose error is the smallest. Then delete any vertices of each selected triangle.
5. Find T_i whose error E_i is the largest. Then create a new point laying at centroid of that triangle and search for nearest neighbor of centroid.
6. Replace the deleted point with the point created at step 5 at near the centroid of a triangle. Thus the number of new points is equal to the number of old points and more points are selected from curvature zone in this way.
7. Continue the process until the total error becomes stable (make sure the deleted zones are plain zones).



Step 1: The error of triangle 1 is minimum where as the error of triangle 2 is maximum



Step 2: The vertex a is replaced by new point b which is real centroid of triangle 2 as well as vertex of new generated triangles. More points are organized in new location.

Figure 3.7: Organizing of points

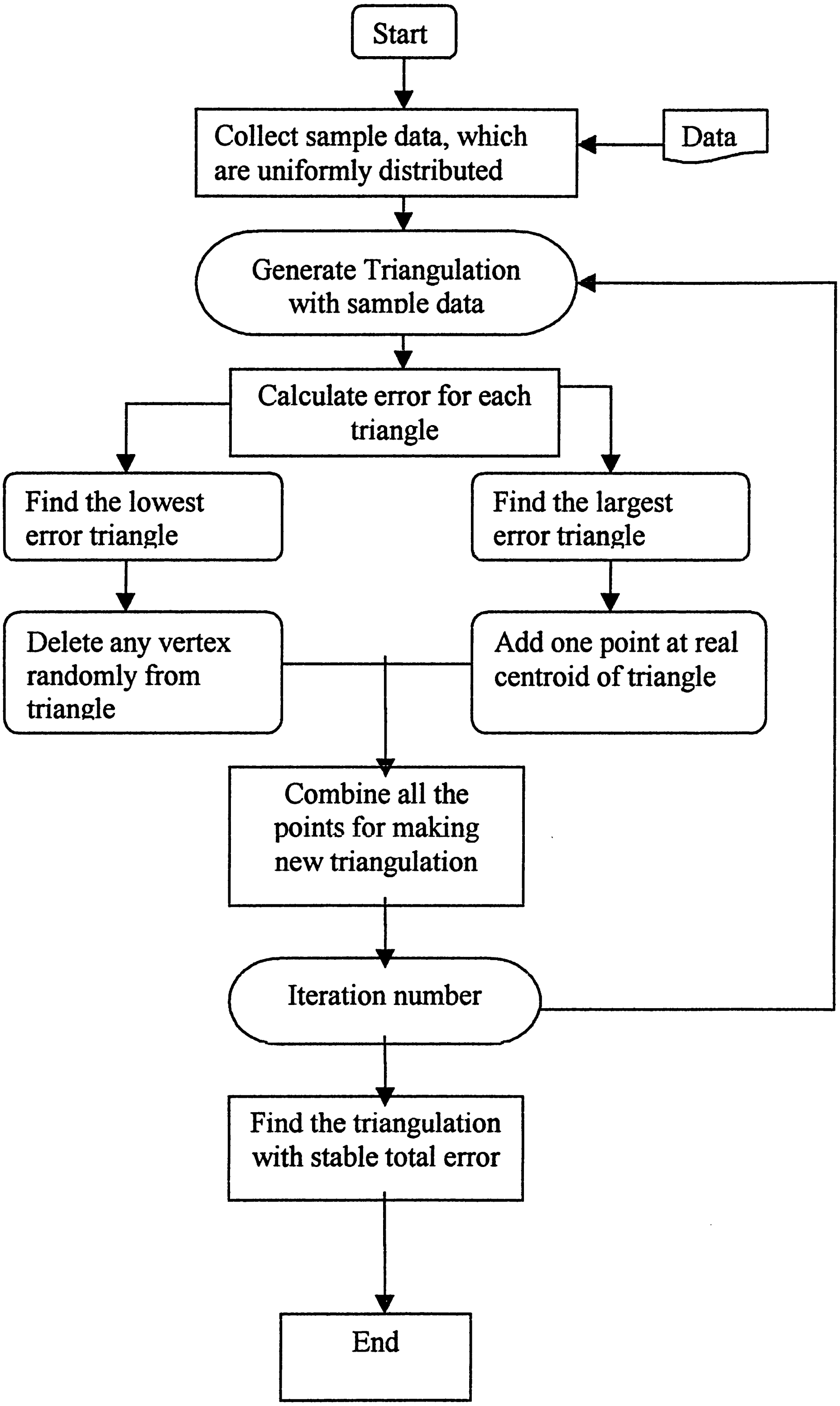

Start
Collect sample data, which are uniformly distributed
Data
Generate Triangulation with samnle data
Calculate error for each triangle
Find the lowest error triangle
Find the largest error triangle
Delete any vertex randomly from triangle
Add one point at real centroid of triangle
Combine all the points for making new triangulation
Iteration number
Find the triangulation with stable total error
End

Figure 3.8: Flow chart for 2nd phase algorithm

3.3 Introduction of third phase algorithm

Genetic algorithms search the solution space of a function through the use of simulated evolution, (i.e., the survival of the fittest strategy). In general, in order to find out the optimum solution of a problem, a genetic algorithm starts from a set of chromosomes (assumed solutions) and evolves different but better sets of chromosomes (sets of solutions) over a sequence of iterations. In each generation (iteration) the fitness function-measuring criterion (objectives function) determines the suitability of each chromosome and based on these values, some of them are selected for reproduction. The number of copies reproduced by an individual parent is expected to be directly proportional to its fitness value, thereby embodying the natural selection procedure, to some extent. The procedure thus selects more fit solution, less fit solutions are eliminated. The basic algorithm can summarized as follows-

1. Select parents.
2. Apply genetic operators in order to create the next generation.
3. Evaluate objective function and constraints.
4. If the optimum has not been replaced, repeat steps 1-4 until the optimum is found or the process will be stopped by end criteria.

This approach is used to build a triangular mesh for developing the given model. The fundamental issues of genetic algorithms in point triangulations as well as the relation with mesh optimization characteristic are described below.

3.3.1 Population initialization

A random number generator creates the initial population. The initial population is a balancing act, too many may result in a loss of efficiency and too few may result in a loss of accuracy.

The population is initialized by uniformly distributing n points and making a triangular mesh using those points. The total number of triangles will be the population. The number of generation is user-defined which will depends on the number of triangles generated in each step or numbers of points developed in each step.

3.3.2 Selection scheme

This operation is used to decide which individuals will act as cross over and mutation operation in order to produce new search points. Normally this process selects individual solution for reproducing new solutions that will be inserted into the population. A user-defined fraction of the highest-ranking solutions are chosen to reproduce new solutions through crossover and mutation. The new solutions usually replace the lowest ranking solution in the population and the total population size remains the same.

All the triangles are considered as the initial population in this work. The population will increase with generation, depending on the number of children for each of the parents.

3.3.3 Crossover

Crossover is defined as swapping of genetic material from the parents, as is applied to the offspring. In order for traits from both the parents to be evident in the children, the swapping of genetic information must be somewhat balanced. In other words, no one parent should dominate the provision of genetic material. Different types of crossovers are possible, the most common being single. In single crossover operation two chromosomes are cut at the same time randomly chosen location and the cut off parts are joined to new offspring chromosomes with the same length as the parent chromosomes. In multi-point crossover the procedure is done at several random points on the parent chromosomes at once.

Crossover technique is used in different approach in this report. Every triangle has three edges. If the mid points of the three edges are connected, then will create three new set of (each set consists of two triangles) triangles. If the centroid of the triangle is connected, it will create one more set of triangles with three triangles inside. These four combinations of triangles will be the children after crossover operation. All the process is described in the figure 3.9.

3.3.4 Real midpoint and centroid

There are four combinations of points within a triangles-three are midpoints and one is centroid. If x1, y1, z1; x2, y2, z2; x3, y3, z3 are the three vertices of a triangle, then we can find three midpoints of edges and one centroid with the following formulas:

$$x_{mid} = \left[(x_1 + x_2)/2, (x_2 + x_3)/2, (x_3 + x_1)/2\right] \tag{3.4}$$
$$y_{mid} = \left[(y_1 + y_2)/2, (y_2 + y_3)/2, (y_3 + y_1)/2\right] \tag{3.5}$$
$$z_{mid} = \left[(z_1 + z_2)/2, (z_2 + z_3)/2, (z_3 + z_1)/2\right] \tag{3.6}$$

And

$$x_{centroid} = \left[(x_1 + x_2 + x_3)/3\right] \tag{3.7}$$
$$y_{centroid} = \left[(y_1 + y_2 + y_3)/3\right] \tag{3.8}$$
$$z_{centroid} = \left[(z_1 + z_2 + z_3)/3\right] \tag{3.9}$$

However the points which are calculated by using the above formula might not be the real scanning point i.e. scanning points could be near the midpoint and centroid of triangle. To find the real point, the neighborhood is searched by calculating the minimum difference of the point with the nearest scanning point. Then lines are drawn among all the vertices and midpoints or centroid as necessary to make new generation of triangles.

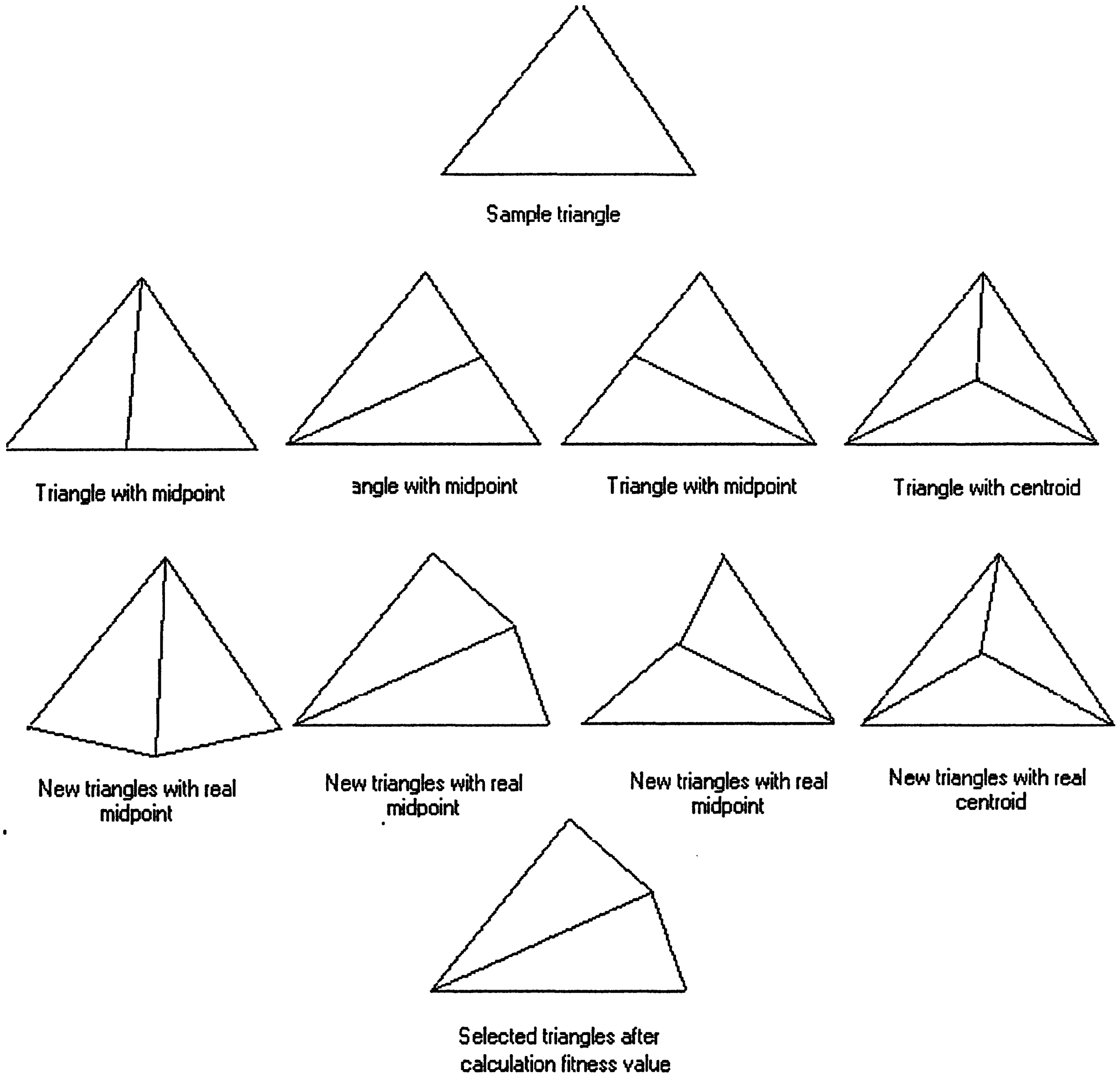

Sample triangle
Triangle with midpoint
angle with midpoint
Triangle with midpoint
Triangle with centroid
New triangles with real midpoint
New triangles with real midpoint
New triangles with real midpoint
New triangles with real centroid
Selected triangles after calculation fitness value

Figure 3.9: Simple steps for selecting different triangulation

Step-1-Select any triangle.

Step-2-Find different combinations of triangles of first three triangles using midpoint of edges and for the last one use centroid of the triangle.

Step-3- Find the nearest point of respected mid points and centroid and reconnects all the vertices to make new triangles.

Step-4- Select the best triangles using the fitness value.

3.3.5 Fitness Function

The fitness function determines which of the individuals will survive and continue on to the next generation. Generally, in genetic algorithms these functions evaluate the solutions (combination of parameters) and return a single score for each solution, the fitness score. There are two ways of evaluating of fitness function; minimization of cost function and maximization of profit function.

Quality of the mesh (described in chapter 2) plays major role for selecting the fitness function for a genetic algorithm. The characteristics of optimization of triangle can play significant role for deciding fitness function in genetic algorithm. The judgement of a good triangulation is based on the shape of newly formed triangular facets. It is desirable to form triangle as equiangular as possible. The equiangularity of a triangle can be evaluated by F, which is defined as:

$$F = \sum_{i=1}^{n-2} \left[\sum_{j=1}^{3} (\alpha_i - \pi / 3)^2 \right] \tag{3.10}$$

Where, α_j is an interior angle of the triangle. For an equiangular triangle as shown in the figure below, the value of F is zero but when a triangle becomes thinner or longer F becomes larger.

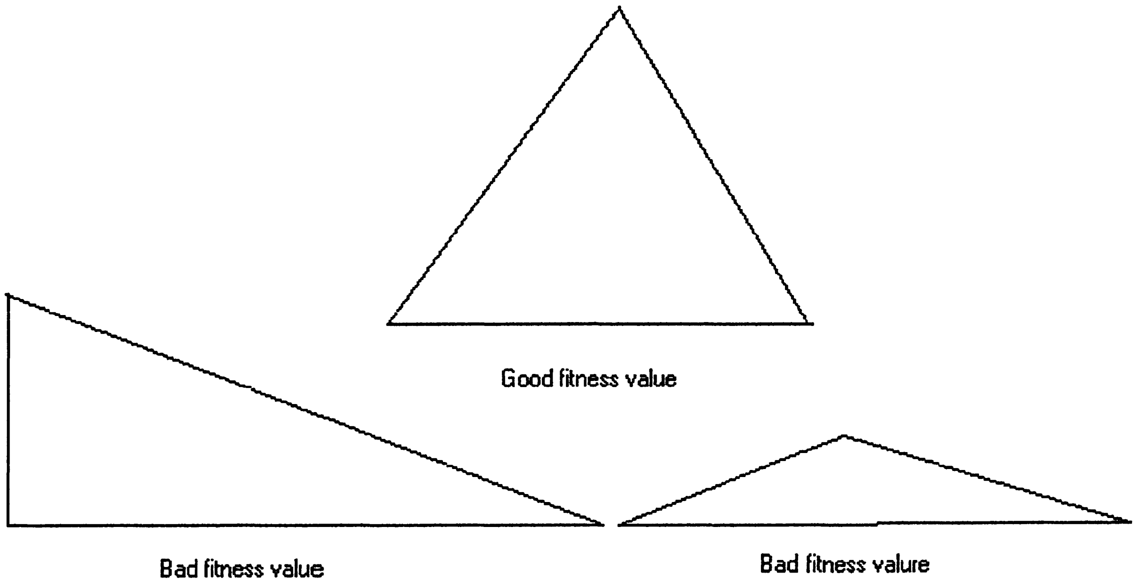


Figure 3.10: Different types of triangles

If the coordinates of the triangles vertices are given by $(x_{i,} y_I)$ where i=1,2,3, then the area is given by the formula:

$$Area = 1/2 \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad (3.11)$$

Therefore, the actual area is obtained by taking the absolute value of the above equation.

If ABC is a triangle with edges a, b, c as bellow, the formula 6.12 establish relation between edges and angles.

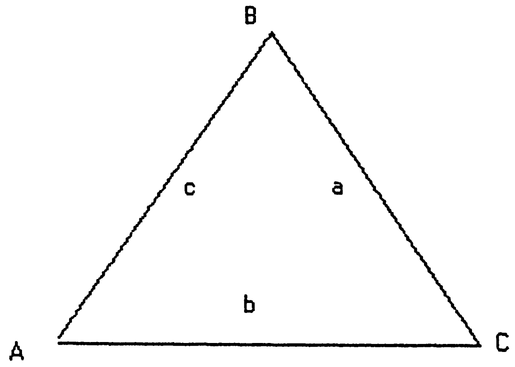


Figure 3.11: ABC is a triangle with edges a, b &c.

$$a \times b \times \sin(C)/2 = b \times c \times \sin(A)/2 = c \times a \times \sin(B)/2 \qquad (3.12)$$

3.3.6 Termination criteria

The genetic algorithm moves from generation to generation selecting and reproducing parents until a termination criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. The second criterion, which was used more often, was to stop the run after the genetic algorithm converged within a user-specified tolerance, and there was no further or only very little improvement over several generations (iterations). The genetic algorithm can be terminated when no further improvement is made for user-specified number of times generation.

As the proposed algorithm considers all the triangles are its initial population and will create at least double (maximum triple) triangles after each generation, special care is taken to select number of triangles for developing optimized triangulation.

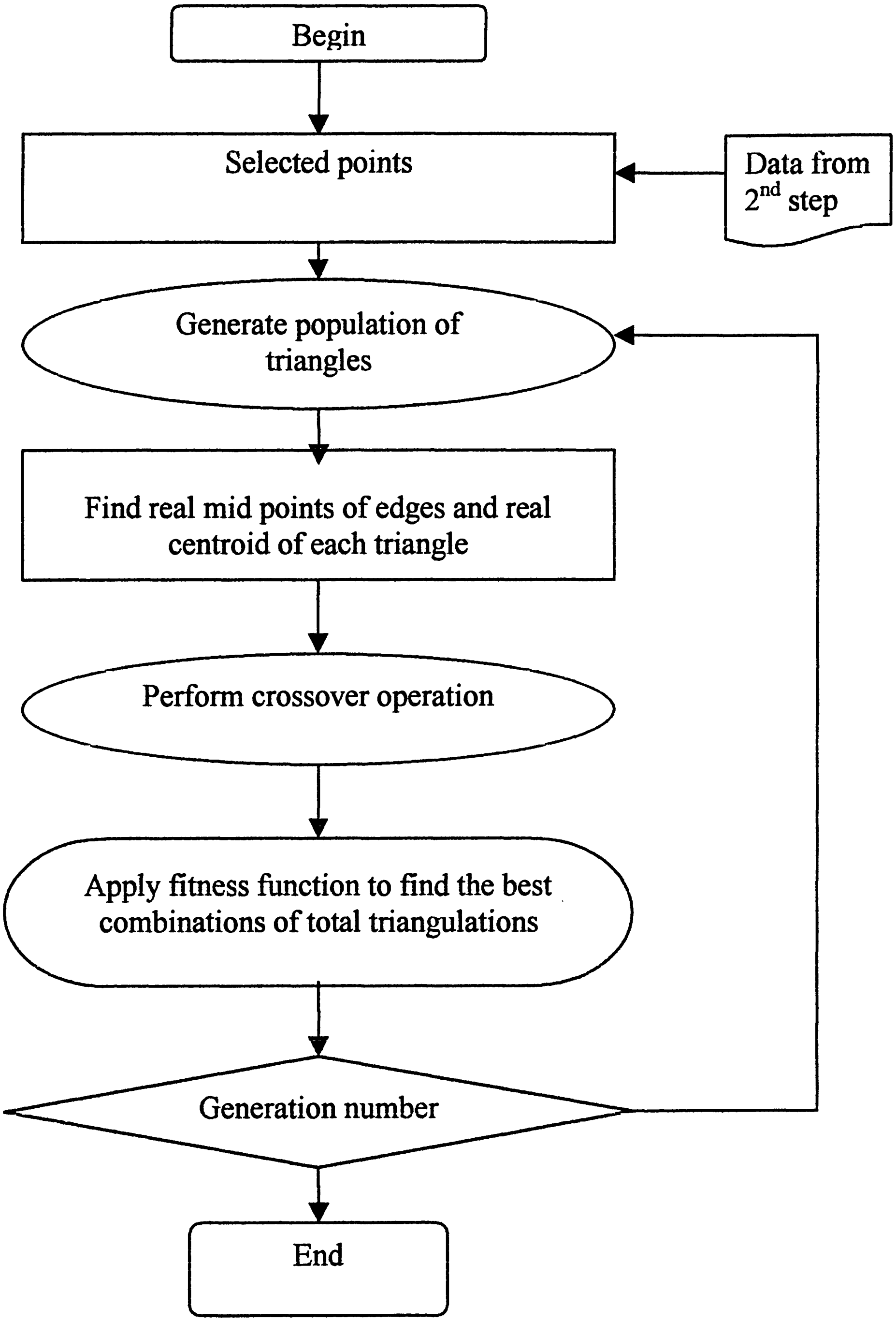

Begin
Selected points
Data from 2nd step
Generate population of triangles
Find real mid points of edges and real centroid of each triangle
Perform crossover operation
Apply fitness function to find the best combinations of total triangulations
Generation number
End

Figure 3.12: Flowchart for 3rd phase algorithm

Chapter 4
Experimental Results

This chapter concentrates on the outcome of proposed algorithm. It describes the criteria of the proposed algorithm after implementing on a given model. It compares the global error between Delaunay based triangulation and the proposed genetic algorithm based triangulation. The triangulation output is written in 4*3 matrix (first three rows are vertices of triangle and last row is the normal of triangle) format.

4.1 Experimental Results

The proposed genetic algorithm is tested by the reverse engineered scanned data of sample model named "Leg implant". The initial total number of scanning points are (9338+9338)=18676 (figure 3.2 & 3.3). These data reduce to 4082 by implementing 1st phase algorithm. Then Constrained Delaunay triangulation technique is used to triangulate the data. It is seen that the object (figure 4.1) is distorted at the boundary zone on the top and bottom portion of leg implant due to complexity of the model.

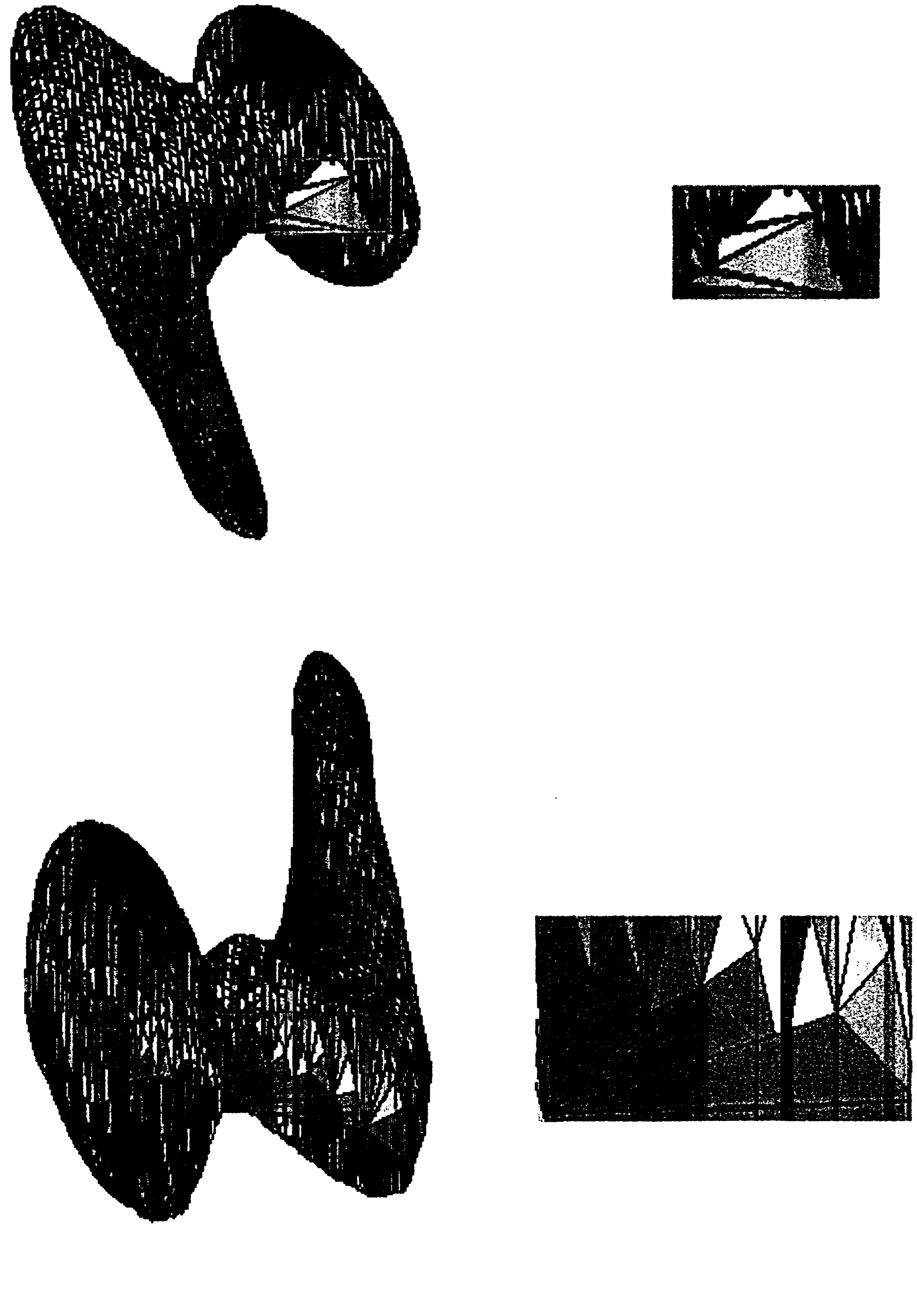


Figure 4.1: Constrained Delaunay Triangulation develops Leg implant (surface 3d view). Zoomed view of small portion at the boundary of top and bottom portion. Distortion is occurred which is hard to control due to complexity of model

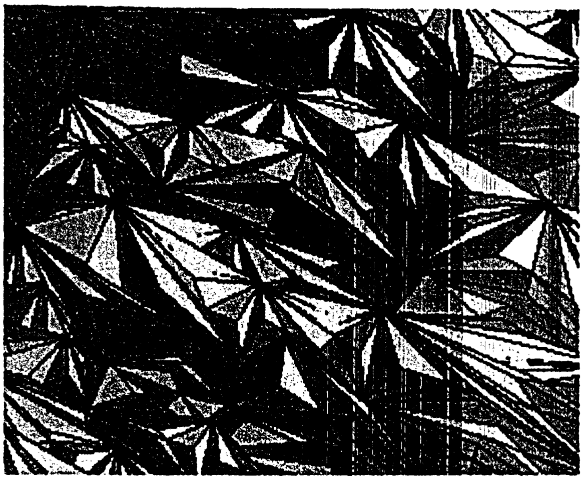


Figure 4.2 Close up view of triangles inside object mesh (surface patch)

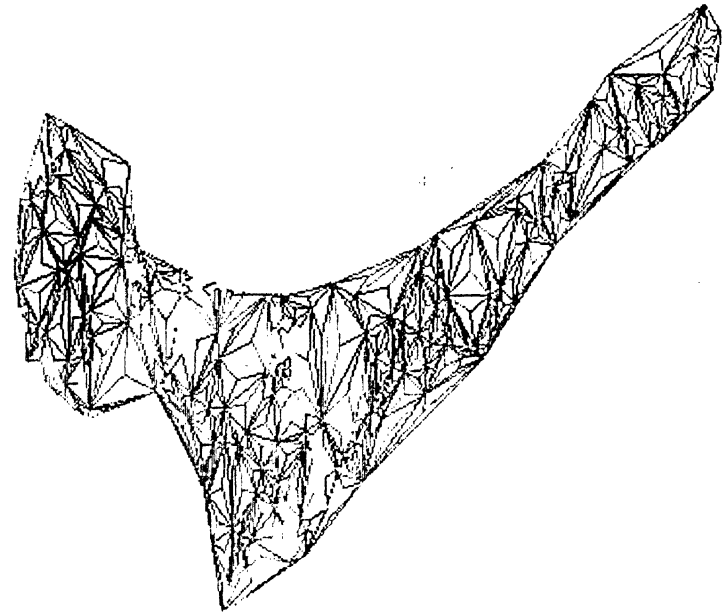


Figure 4.3: 2d view triangular mesh of leg implant (541points)

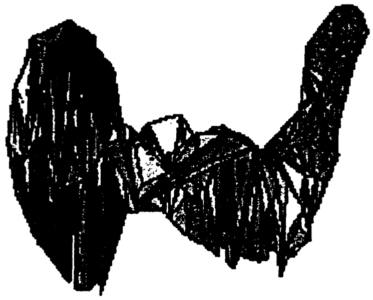


Figure 4.4: 3d view of leg implant (541 points)

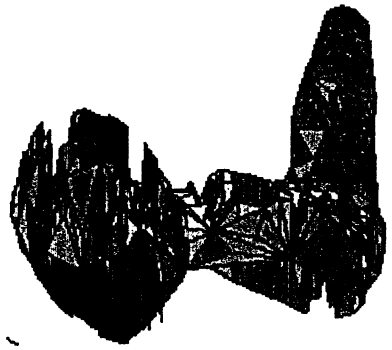


Figure 4.5: 3d view of leg implant (1409 points)

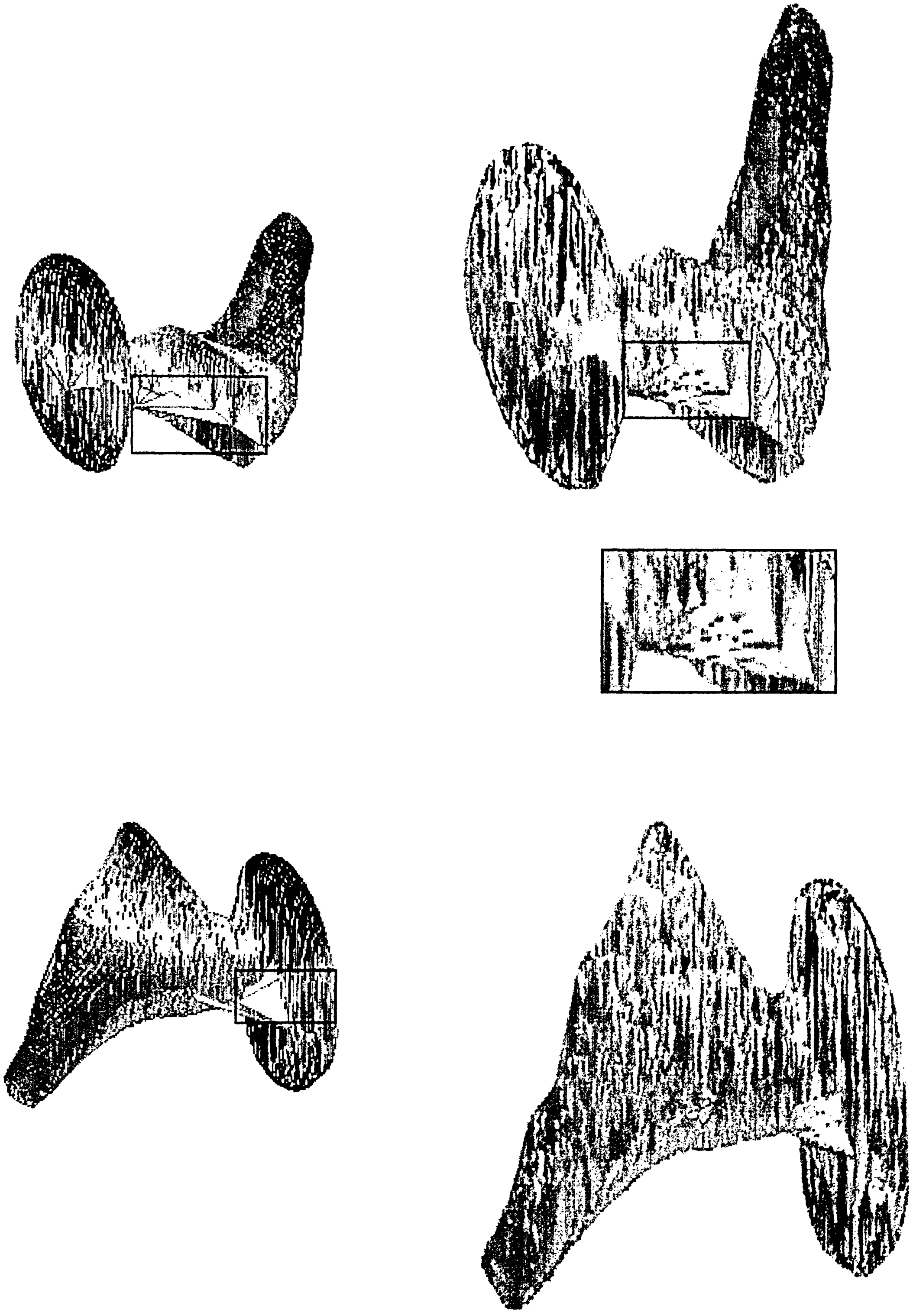


Figure 4.6: Comparison picture of leg implant

Left Side: Leg implant is developed by Constrained Delaunay triangulation

Right Side: Leg implant is developed with 3631 points

The figure 4.2 is a close-up view of the triangulation, developed by the proposed algorithm. The mid point of an edge and centroid of a triangle is distributed according to fitness value. The triangles are generated at each generation by cross over operation, described in previous chapter.

The figure 4.4, 4.5 are formed through different generation (1st, 2nd) of triangles by proposed genetic algorithm. Lastly the figure 4.6 shows the comparison of picture of Delaunay triangulation (4082 points.) and proposed algorithm (3631 points.). It shows distortion at boundary region is improved by new algorithm with less data.

4.2 Face and Vertex normal vectors

Expected output in this project is optimized data, with each group consisting of a 4*3 matrix. First three rows are the vertices of each triangle and last row is the normal of the triangle.

Each face in a mesh has a perpendicular normal vector. The vector's direction is determined by the order in which the vertices are defined. Vertices are listed in counter clockwise order as mentioned in Chapter 2. The normal for any triangle at surface in triangular mesh can be determined by the cross product of two vectors. Let P1, P2, P3 are the three vectors are generated from the three points vertices of the triangle. Using elementary vector subtraction, two edges of the triangle (E1 and E2) can be found by using the following equations:

$$A=P3-P2 \text{ and } B=P2-P1 \qquad (4.1)$$

$$A \times B = n|A|.|B|.\sin\theta \qquad (4.2)$$

Where n is unit vector perpendicular to the plane of A and B.

The output of optimization of points is given in 4*3 matrix form, three vertices of each triangle are the first three rows and the normal of the triangle is the last row.

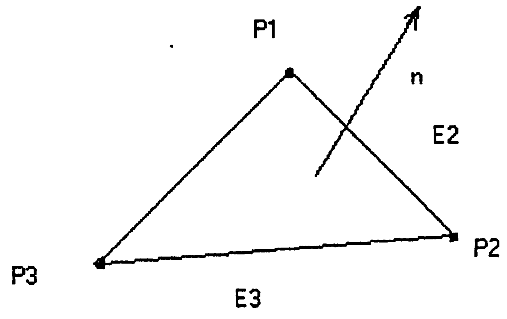


Figure 4.7: Geometry of given triangle and its surface normal

4.3 Global Error Calculation

The error computation is an essential part of the algorithm. The error of a surface point is defined as the vertical distance between the point and the intersection of the vertical line with the triangulation. For simplicity, the difference between the mean of the z co-ordinate of the three vertices of a triangle and the maximum z co-ordinate of the three vertices is measured. Delaunay triangulation developed by the Fortune algorithm, in a Matlab environment is used as a standard triangulation technique for comparing the proposed triangulation. The global error is calculated considering all the points of Delaunay triangulation. The final errors for the different triangulations, based on proposed algorithm are calculated. Experimental results indicate that more points are considered for developing the triangle, less error will result (figure 4.9).

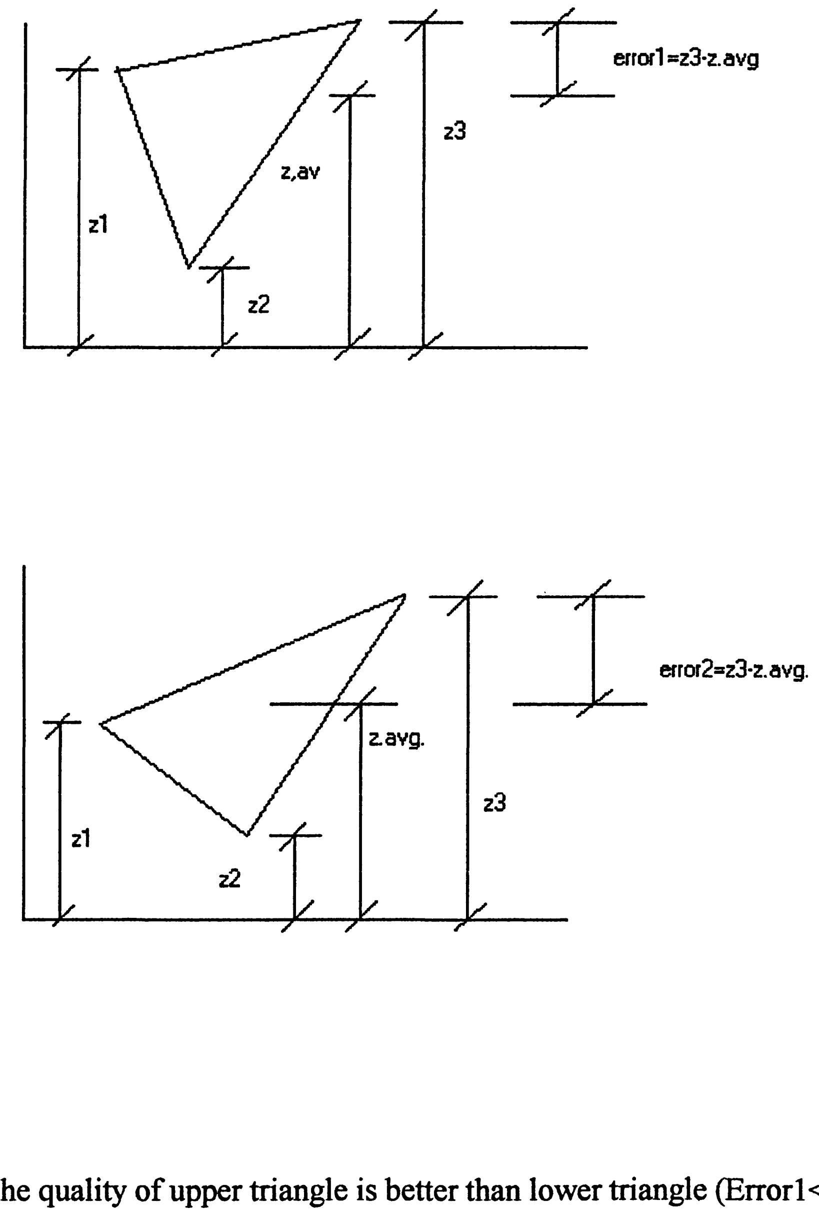


Figure 4.8: The quality of upper triangle is better than lower triangle (Error1<Error2)

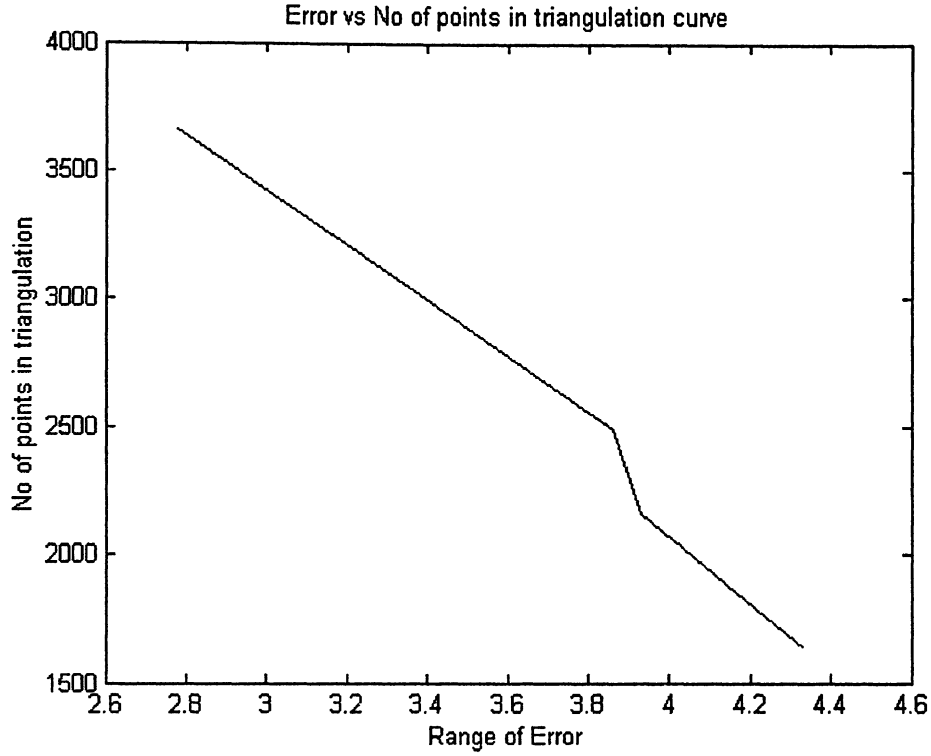


Figure 4.9: Error vs. No of points in triangulation curve

The proposed algorithm is written in Matlab code and applied to the leg implant object with different combination of points. It is seen that the global error is decreased whenever the numbers of points are increased to develop triangulation.

The main objective of this research is accuracy. The proposed algorithm is implemented by Matlab programming. Therefore the computational time is slower than expected. If the algorithm can be written in C++ with optimized structure algorithm, it will give a faster result.

In this work, the combination of two powerful tools in triangulation field, subdivision and genetic algorithm is presented. The use of this combined approach improves the object accuracy by triangulating compare to constrained Delaunay triangulation method.

Chapter 5
Conclusions

This chapter presents a summary of the work outlined in this report. Conclusions drawn from the work are summarized and the contributions to the optimization of reverse engineering data are discussed. Finally, recommendations for future work in this area are outlined.

5.1 Summary and Conclusions

This report has presented a new method for optimizing a triangular mesh from reverse engineering scanning data. A genetic algorithm based optimization method is proposed and implemented.

There is a sharp difference between the photographic view (picture 3.1) and scanned top and bottom view of model from bmp. format (figure 3.2 & 3.3). The sharp collar near the spherical part and abrasive part on the middle of the leg implant is not seen clearly in scanned view model. The report is concerned in the optimization of reverse scanning data. Therefore, the 3d picture, which is developed by the Delaunay triangulation as well as by proposed triangulation techniques in this report, is not as good as in the photographic picture.

During the assembling of the top and bottom leg in the first algorithm, a trial and error method is used to avoid redundant z co-ordinate. This might be the only way to reduce z co-ordinates for unknown data points from 3D scanner Roland DR. Picza. Moreover, during collection of the scanning points of the leg implant, when the leg implant turned upside down, it's position not only changes in x-y plane, but also the z axis direction due to complexity of the model object. As a result it creates distortion during assembling the top and bottom parts (figure 4.1).

During the organization of points at curved regions, the algorithm is not as effective as expected. The algorithm is slow due to huge calculations and the topology of the sample model, which is curved over most of the surface. On the other hand, more points should be placed at the sharp corners as boundary points to get an actual picture of the model object, which is not possible to install, by the proposed algorithm.

The possibility of generated triangulations of given points is faster with the increment of points (2 to 3 times the number of triangles at each step of generation). The algorithm developed in the third phase is computationally expensive even though a simple genetic algorithm is used. Careful steps have been taken for the selection of scanning points at the first step of the algorithm as well as termination criteria. A user can have different triangulation results by adjusting the fitness function.

The following characteristics are noticed when comparing the developed triangulation to Constrained Delaunay triangulation:

1. This method requires subdivision connectivity as a primary level, which is the inherent weakness of the subdivision technique.
2. Genetic algorithm can be used as a very good refinement operator in the adaptive subdivision method.
3. It is possible to model a complex structure if boundary points are selected carefully.
4. The greatest advantage of this method is that it's allowing various degrees of smoothness (level-of-details).
5. Still a very active area of research.

5.2 Recommendation

The success of this project is dependent on how good the top and bottom parts of the leg implant are assembled in the first phase of the algorithm. 3D rotated scanner might help to develop a better picture of a complex object like leg implant. The sharpness criterion and organizing points at curved region algorithm deserves further attention. Second phase of algorithm can be improved by considering sharp corners as the boundary points as well as with a smart algorithm. During crossover operation in third phase algorithm, centroid based triangles probably dominate the mid point based algorithm due to fitness values of newly generated triangles. This problem might be overcome by introducing a mutation for centroid based triangles and crossover for midpoints based triangles for the generation of new triangles. Therefore, additional studies on the subject of optimization criteria in 3d triangulation are expected.

Appendices

Terminology

Topology: refers to the connected polygonal mesh's structure.

Local topology: is the geometric structure surrounding a particular face or vertex of a polygonal model. For example, a vertex and all its adjacent faces define a local topology. When a simplification algorithm preserves local topology, it does not change the local geometric structure and the genus, or number of holes of the object.

Global topology: is the geometric structure of the whole polygonal model. When a simplification algorithm preserves global topology, it preserves local topology and does not create self-intersections in the object. A self-intersection occurs in the model when two non-adjacent faces intersect each other.

Geometric domain: Some simplification algorithms require the input representation to have particular geometric properties, such as convex surface or orthogonal faces. Other methods require two-manifold, non-self intersecting surfaces. This criterion includes pure geometric properties and the incidence properties defined by the topology of vertices, edges and faces.

Surface fitting: The process of creating an engineering surface to represent the shape and trends of a point cloud.

2D manifold: If the neighborhood of every feature consists of a connected ring of polygons forming a single surface. In a triangulated mesh displaying manifold topology, exactly two triangles share every edge and every triangle shares an edge with exactly three neighboring triangles. A 2D manifold with boundary permits boundary edges, which belong to only one triangle.

Surface modeling: studies the representation of two-dimensional surfaces. Surfaces models give detailed information on the geometry of a curved surface, but do not always give sufficient information for determining all geometric properties of the object potentially bounded by the surface.

Solid modeling: deals with complete and valid geometric representation of 3D objects whose interior is considered to be homogeneous and isotropic. By complete mean that representations must be adequate for answering geometric questions about the object.

3D modeling: are devoted to representation of geometric and non-geometric information on 3D objects. A 3D is an object model including both geometric information and non-geometric information, such as visual information.

Surface mesh generation: is a mesh, generated on a planar or a curved surface region. A surface mesh consists of planar or curved surface elements, which are either triangular or quadrilateral.

Genus: is the number of holes in the mesh surface. For example, a sphere and a cube have a genus of zero, while a doughnut and a coffee cup have a genus of zero.

Point cloud: a group of points captured during the scanning process.

Planar graph: is a graph whose vertices and edges can be embedded in R^2 such that their edges do not intersect.

STL FILE: consists of a list of triangular facet data. A unit normal and three vertices uniquely identify each face. Apart from these, triangular facets in an STL file must also obey the following two rules: The orientation of the facet is specified in two ways, which must be consistent. First, the direction of the normal is outward. Second, the vertices are listed in counterclockwise order when looking at the object from the outside.

Maximum curvature: Consider the mesh of data points resulting from digitizing the surface of a cube and picture the normal vectors associated with each such data point. The data points that lie on the topside of the cube will have normal vectors that are all parallel to each other (or nearly parallel). These are not be the case for the normal vectors of data points that lie on opposite sides of an edge. Therefore angle formed between the normal of nearby data points give an indication of how sharply the surface is curving.

Aspect Ratio: of a triangle as the ratios of the longest over the shortest side of the tightest rectangle that encloses the triangle. The tightest triangle is the triangle with the smallest area. Long and skinny triangles have poor aspect ratios, those are much greater than one while regularly shaped triangles have aspect ratios close to one. Having triangles with aspect ratios close to one is a desirable property since long and skinny triangles are more prone to rendering and numerical errors.

Level of detail: is a single representation of a model. A greatly simplified model is a low level of detail. A marginally simplified model is a high level of detail. The original model is the highest level of detail. The levels of details of a model are the models and all its simplified versions grouped together. When rendering an object, graphics system substitutes different levels of detail for the original model according to how far away the viewer is.

Local curvature: of a region of a model is an approximate measures how fast the geometry changes at that area. A region of low curvature remains relatively stable, implying that the area is almost planar. A region of high curvature abruptly changes direction, meaning that the area contains a bump or spike.

Planar Straight Line Graph: is a collection of points and segments. Segments are edges whose endpoints are points in the PSLG, and whose presence in any mesh generation from the PSLG is enforced.

Convex hull: is the smallest convex polygon that contains a set of points in the plane-shaped as a rubber band stretched around a bunch of pins.

Tessellation: is a general subdivision of a spatial domain, where there is no implied maximum dimension.

Tetrahedrization: is a 3D tessellation, which is made up of tetrahedra. To tetrahedralize something is to construct a Tetrahedrization.

Error metric: is the difference between the input model and the simplified model is known as approximation error. The metric used for such a quantification is called error metric. When the error metric is expensive to compute, cheap error estimations are used instead.

Hausdorff Distance: is named after Felix Hausdorff (1868-1942). Hausdorff distance is the maximum distance of a set to the nearest point in the other set. Hausdorff distance from set A to set B is a maximum function, defined as

$$h(A,B) = \max_{a\in A}\left\{\min_{b\in B}\{d(a,b)\}\right\}$$

Where a and b are points of sets A and B respectively, and d (a, b) is any metric these points. For simplicity, d (a, b) is the Euclidean distance between a and b.

References

1. S. Motavalli, " Review of Reverse Engineering Approaches", Computers ind. Engng Vol. 35, Nos 1-2, pp. 25-28.
2. H. Hoppe, " Surface reconstruction from unorganized points" Ph.D. Thesis, University of Washington.
3. V. Chan, I. Archibald, G. Vickers, C. Bradley, " Development of Tools for processing laser scanner data for reverse engineering" University of Victoria, Department of Mechanical engineering, Victoria.
4. A. Fischer and S. Park, " Reverse Engineering: Multilevel-Of-Details Models for Design and Manufacturing", Int. J. Adv. Manuf. Technol (1999) 15:566-572.
5. S. Kumar, " Surface Triangulation: a Survey" Department Computer Science, University of North Carolina, Chapel Hill, July 03,1996.
6. V. Chan, " Lecture Notes for Artificial Intelligence for Mechanical Engineers" Ryerson University, fall term 2002.
7. D. E. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning", Addition Wesley Publishing Company, January 1989.
8. I. Kolingerova, "Genetic approach to triangulations" Department of Computer Science, University of West Bohemia, Plzen, Czech Republic, 2000.
9. Y. H. Chen and Y. Z. Wang, " Genetic algorithms for optimized re-triangulation in the context of reverse engineering", Computer-Aided Design 31(1999) 261-271.
10. http://www.ics.uci.edu/~eppstein/junkyard/triangulation.html
11. K Ho Le, "Finite element mesh generation method: a review and classification".
12. J. Jansson, " Planar Minimum-Weight Triangulations" M. Sc. Thesis, Computer Science and Technology, November 1995.
13. M. Bern and D. Eppstein, "Mesh generation and optimal triangulation" Computing in Euclidean Geometry (Ding-Zhu Du and Frank Hwang, editors), Lecture Notes Series on Computing, World Scientific, Singapore 1995.
14. P. Hinker, C. Hansen, " Geometric optimization" Advanced Computing Laboratory, Los Alamos National Laboratory, Los Alamos NM 87545.
15. D. Luebke, " A developer's survey of polygonal simplification algorithms" IEEE Computer Graphics and Application, May/June 2001.

16. L. P. Chew, " Guaranteed-quality triangular meshes" Technical report, Cornell University, 1989, No. TR. -89-983.

17. A. Singhai, " Optimal Triangulation and Mesh generation" Term paper CS474, spring 1994, DCS, UIUC, Urbana, IL61801.

18. R. L. Drysdale, Scott McElfresh, Jack Scott Snoeyink, " On exclusion regions for optimal triangulation" February 09, 2000.

19. H. Hoppe, T. DeRose, J. McDonald and W. Stuetzle, " Surface reconstruction from unorganized points" Computer Graphics, 26(2), p. p 71-78,July 1992.

20. C. Andujar, " Geometry simplification" February 1999, Universitat Politecnica de Catalunya, Dept. LSI, Diagonal 647, E-08028, Spain.

21. J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and Delaunay triangular" School of Computer Science, Carnegie Mellon University, Pennsylvania 15213.

22. http://www.andrew.cmu.edu/user/sowen/survey/tetsurv.html

23. N. Amenta, M. Bern, D. Eppstein, " Optimal point placement for mesh smoothing".

24. D. Eppstein, "Approximating the Minimum Weight Triangulation" Department of Information and Computer Science, University of California, Irvine, CA 92717, Tech. Report 91-55, July 1991.

25. S. J. Owen, " A survey of Unstructured mesh generation technology" Department of Civil and Environmental Engineering, Carngie Mellon University, Pittsburgh, PA.

26. Y H. Choi, H Y. Shin, Y I Yoon. , J W. Lee " Triangulation of scattered data in 3D space", Computer-Aided Design 1988:20(5): 239-248.

27. C. Erikson, "Polygonal Simplification: An overview" TR96-016, 1996, Department of Computer Science, CB#3175, Sitterson Hall, UNC-Chapel Hill, Chapel Hill, NC 27599-3175.

28. P. Cignoni, C. Montani and T. Scopigno, " A comparison of mesh simplification algorithms" Computers and Graphics, Vol. 22, N0.1, 1998.

29. P. Heckbert and M.Garland, " Survey of polygonal surface simplification algorithms', Siggraph 97 Course Notes, ACM press, New york.

30. M. J. Dehaemer, Jr. and M. J. Zyda, " Simplification of objects rendered by polygonal approximations" Computer and Graphics, Volume 15, No.2, PP 175-184,1991.

31. J. R. Shewchuk, " Lecture notes on Delaunay Mesh Generation" September 1999, Department of Electrical Engineering and computer science, University of California at Berkeley, CA.

32. F. Aurenhammer, "Voronoi diagram-A survey of a fundamental geometric data structure" ACM computing surveys, Vol.23, 1991 page 345-405.

33. S. Fortune, "Voronoi diagram and Delaunay Triangulation" Computing in Euclidean Geometry Lecture Notes Series on Computing, World Scientific, Singapore 1995.

34. J. R. Shewchuk, "Delaunay refinement mesh generation" Ph.D. thesis, May 1997,Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

35. S. Rebay., "Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm" Journal Of Computational Physics, Academic Press, Vol. 106, pp.125-138, 1993.

36. J. Ruppert, " A Delaunay refinement algorithm for quality 2-dimensional mesh generation" Journal of algorithms 18(3): 548-585, May 1995.

37. Y. Zheng, Roland W. Lewis, David T. Gethin, " Three-dimensional unstructured mesh generation: Part 1, Fundamental aspects of triangulation and point creation" Comput. Methods appl. Mech. Engrg. 134(1996) 249-268.

38. P. Su, " A comparison of Sequential Delaunay Triangulation algorithms" Imperative Internet Technologies, Pittsburgh, PA 15213 and Robert L. Scot Drysdale, Department of Computer Science, 6211 Sudikoff Laboratory, Dartmouth College, Hanover NH 03755-3510, USA.

39. M. J. Dehaemer, Jr. and Michael J. Zyda, " Simplification of objects rendered by polygonal approximations" In computer & graphics, Vol. 15, No, 2, 1991, p. p 175-184.

40. G.Turk, " Re-tilling polygonal surfaces" Computer Graphics, 26(2): 55-64, July 1992.

41. W. J. Schroeder., A. Zarge and W. E. Lorensen, "Decimation of triangular meshes", Computer Graphics, 26(2): 65-70, July 1992.

42. H. Hoppe, T. Derose, T. Duchmp, " Mesh optimization" Computer Graphics Proceeding, annual conference series, 1993.

43. T. S. Gieng, B. Hamann, K. I. Joy, G. L. Schussman, and J. Trotts, "Constructiong hierarchies for triangle meshes", IEEE Trans. On Visualization and Computer Graphics, 4(2): 145-161, April-June 1998.

44. K. Qin, M. Gong, B. Wu and Z. Tang, " The genetic minimum weight triangulation of planar points", Journal of Computer-Aided Design & Computer Graphics, Vol. 8, No. Suppl, pp. 141-147, September, 1996.

45. M. Holder and J. Richardson, " Genetic Algorithm, Another Tool for Quad Mesh Optimization", Civil Engineering Dept., University of Albama, Box 870205, Tuscaloosa AL 35407, 1998.

46. P. J. Frey, H. Borouchaki, "Geometric surface mesh optimization", Computing and Visualization in Science @ Spriger-Verlag,1998,1,113-121

47. C. C. L. Wang, M. M. F. Yuen , "A genetic algorithm for mesh optimization", The international journal of Advanced Manufacturing Technology, 2001,18:739-744.

48. I. Kolingerova, and A. Ferko, "Multicriteria-optimized triangulations" Springer-Verlag, August 2,2001.