İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

DESIGN AND ANALYSIS OF 3D HIERARCHICAL MESHES

M.Sc. Thesis by
Serkan ERÖKSÜZ

Department : **Electronics & Communication Engineering**

Programme: **Telecommunication Engineering**

JANUARY 2007

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

DESIGN AND ANALYSIS OF 3D HIERARCHICAL MESHES

M.Sc. Thesis by

Serkan ERÖKSÜZ
504021325

Date of submission : **25 December 2006**

Date of defence examination: **29 January 2007**

Supervisor: **Assoc. Prof. Dr. Işıl CELASUN (İ.T.Ü)**

Members of the Examining Committee: **Prof.Dr. Melih PAZARCI (İ.T.Ü.)**

Prof.Dr. Lale AKARUN (B.Ü.)

JANUARY 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

3B SIRADÜZENSEL TELFİLELERİN TASARIMI VE ANALİZİ

YÜKSEK LİSANS TEZİ

Serkan ERÖKSÜZ
504021325

Tezin Enstitüye Verildiği Tarih : 25 Aralık 2006

Tezin Savunulduğu Tarih : 29 Ocak 2007

Tez Danışmanı : Assoc. Prof. Dr. Işıl CELASUN (İ.T.Ü)

Diğer Jüri Üyeleri: Prof.Dr. Melih PAZARCI (İ.T.Ü.)

Prof.Dr. Lale AKARUN (B.Ü.)

OCAK 2007

PREFACE

The processing, transmission, storage and visualization are gaining importance in the recent years of development. This research work can be termed as link to that chain.

According to ancient Greek philosophy a great teacher is like a candle which consumes itself to light the others for way. Similarly I would like to express my sincere gratitude to my advisor Dr. Işıl CELASUN for her guidance and support, her dedication for research, for her advice, and her perpetual encouragement during this study.

I am also indebted to my friend Rizwan A SIDDIQUI for his endless support and the time he has spared for me and for my work.

A word of thanks is also uttered in the favor of all those friends who are working with me at ITU-CSCRS (Istanbul Technical University Center for Satellite Communication and Remote Sensing) who have cushioned my working hours while I had to toil with my research.

I am also thankful to Derya ŞENOCAK and Cercis Özgür SOLMAZ for their invaluable support. I will never forget the years that we have passed together, the grief we have shared, the laughs we have laughed and the friendship that continues.

A thankful gesture is also extended to my aunts, Şaziye and Aynur ŞİMŞEK, and their families who have always supported me morally and due to whom I have never felt myself lonely or a foreigner in a huge city like Istanbul.

And last but not the least I will always be grateful to my parents Firdevs and Muharrem ERÖKSÜZ, who were my first teachers and its all due to them that I had the courage and the will to carry out my studies. They had always been my greatest supporters. I am also thankful to my loving sister Serap ERÖKSÜZ whose love has been my greatest asset.

January, 2007 Serkan ERÖKSÜZ

TABLE OF CONTENTS

ABBREVIATIONS

VRML	: Virtual Reality Modeling Language
MPEG-4	: Motion Picture Expert Group 4
VTK	: Visualization Toolkit
WWW	: World Wide Web
3D	: Three Dimensional
2D	: Two Dimensional
DT	: Delaunay Triangulation
LOD	: Level of Detail
NV	: Number of vertices
NBV	: Number of Boundary vertices
GA	: Global Alpha
QT	: Quality Test
WA	: Weighted Alpha

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\mathbf{\Sigma}$	: n-ary sumation
$\mathbf{R^2}$	: two dimensional space
$\mathbf{d_E}$	: euclidean distance
$\boldsymbol{\pi}$	: pi
ψ_{t_0}	: space angel
Θ	: dihedral angle
$\boldsymbol{\alpha}$	: shape reconstruction parameter
r_{in}	: radius of the inscribed sphere
R_{out}	: radius of the circumscribed sphere
$\mathbf{l_e}$	: edge length
$L_{\max}$	: length of longest side of the tetrahedron
L_n	: n^{th} edge length
$\cos^{-1}$	: inverse cosine
$D_{\max}$	: maximum distance
$\frac{P}{A}$	: perimeter to area ratio
$\frac{A}{V}$	: area to volume ratio
e	: edge
T	: tetrahedron
t_0	: vertex
S	: delaunay triangulation
RV	: voronoi region
Vor	: voronoi diagram
n_tetra	: number of tetrahedron
$\subset$	: subset of
ε	: member of
$\cup$	: union
kNN	: K- Nearest Neighbor

SUMMARY

The transmission of 3D objects models over the internet and their representation to the end users have drawn considerable interest and attention. Some standards such as VRML (Virtual Reality Modeling Language) and MPEG-4 (Motion Picture Expert Group) have been developed for 3D object representation. One of another choice, for implementing it is hierarchical mesh compression for progressive transmission.

In the hierarchical mesh decimation the Delaunay Triangulation is a building block. The Delaunay Triangulation is considered optimal for the visualization of 2D data, and has many characteristics such as automatic implementation, unique in implementation etc., but has some degeneracy, like formation of needles, Caps and Slivers when are applied for the visualization of 3D data. Additionally the alpha value implementation on the Delaunay triangulation plays a major role for better visualization. The alpha can be classified as the global and the local. The global alpha can be considered as the average edge length whereas the local refers to the different averages of length of edges in the different regions of the data. The visual results can be improved if combination of both is applied on the data.

The hierarchical mesh compression is based upon first classifying the given vertex data into boundary and interior vertices. Both of these vertices are decimated according to different parameters which are proposed in this work. The boundary vertices are decimated according to the nature of region and the density of the vertices. The interior vertices are decimated on the basis of neighbor counts and the volume of tetrahedron formed. The process is performed iteratively on the data to reach from finest to coarsest level. While transmitting, the coarsest is sent first as the base data and then remaining data is added progressively.

While decimation it is important verify whether the data is capable being visualized. To ascertain it we have proposed another technique through which we continuously monitor the quality of tetrahedral mesh being generated at each hierarchy level. Those tests are like radius ratio tests, the inner radius to maximum edge length test, volume to edge lengths ratio etc.

ÖZET

Üç boyutlu (3B) nesnelerin internet üzerinden iletimi ve görselleştirilmeleri son yıllarda ilgi çeken konular arasında yer almaktadır. Teknoloji ve altyapı olanakların gelişmesi sayesinde uygulama alanlarıda artmaktadır. VRML veMPEG-4 bu tür veriler için geliştirilmiş stadartların başında yer almaktadır. Sıradüzensel telfile gösterimi ve sadeleştirme yöntemleri bu standartların içerisinde yer almaktadır.

Sıradüzensel telfile gösterimi ve sadeleştirmesi bilgisayarla grafikte 3B nesnelerin uyarlamalı detay seviye kaplamasında kullanılmaktadır. Sadeleştirme metodu kullanarak her seviyedeki telfilelerin Delaunay topolojisini koruması ve telfilelerin istenen geometrik özelliklere sahip olması sağlanır. Delaunay üçgenlemesi uzaydaki düğümlerin birleştirilmesini sağlamak amaçlı kullanılmıştır. Delaunay Üçgenlemesi 2 Boyutlu uzayda uygulandığı zaman tekil sonuçlar vermesine rağmen 3 Boyutlu uzayda tekil sonuç vermediği gibi aynı zamanda bozuk dörtyüzlülerde oluşturmaktadır. 3B Delaunay Üçgenlemesi yapılırken alfa parametresinden yararlanılmaktadır. 3D Delaunay dörtyüzlemesi için kullanılan alfa parametresi her dörtyüzlüyü çevreleyen kürenin yarıçapını belirlemektedir. Sadece çevreleyen bu küre içinde kalan dörtyüzlü kullanılabilir. Alfa parametresi genel ve yerel olarak 2 farklı şekilde seçilebilir. Genel alfa değerini kullanıldığında şekil üzerindeki bazı bölgelerin doldurulamadığı gözlenmiştir. Bunun yanında düğüm aralıkları genel alfadan küçük olan bölgelerde ise detayların kaybolmasına istenmeyen birleşmelerin oluşmasına sebep olmuştur. Buradaki hatalı ve istenmeyen dörtyüzlülerin topolojide bulunmasını engellemek için nesneleri düğüm yoğunluğuna göre farklı bölgelere ayırarak, her bölge için ayrı alfa parametresi elde edilmesi ve bölgelerin kendi içinde delaunay algoritması uygulanması gerekmektedir.

Sıradüzensel telfile sadeleştirme yönteminde nesnenin dış şeklini kaybetmemesini sağlamak için sınır ve iç düğümler farklı yöntemler kullanılarak sadeleştirilmektedir. Sınır düğüm sadeleştirme algoritması uzaklık parameteresi kullanılarak dış kabuktaki şekil bozuklukları kontrol altına alınması sağlanmıştır. İç düğüm sadeleştirme algoritmasında düğümler önem kriterine göre sıralanmakta ve sadeleştirmeye en az önemli olan düğümden başlanmaktadır. Sıradüzensel telfile sadeleştirme yöntemi sonucunda en detaylı seviyedeki veriden en kaba seviyedeki veri adım adım elde edilmektedir. İletim ise en kaba seviyeden başlanarak ve ardışıl olarak iyileştirme

seviyelerinin eklenmesi ile gerçekleşir. Telfile sadeleştirme algoritmasında her adım da oluşturulan telfilelerin gösterebilir özelliği kontrol altında olması gerekmektedir. Yüzey telfile nesnelerine iki hacim telfile nesnelerine ise dört farklı test yöntemi uygulanarak gösterebilir özelliği kontrol altına alınır.

1. INTRODUCTION

The last half of twentieth century can be truly recognized as the era of development of communication. If one element has to be chosen amongst all the inventions in the era then computers can be regarded as the most amazing one. The last decade has exacerbated the field of communication with the introduction of laser scanners and satellites it is possible to generate data at tremendous rates. To visualize this data is one of the most important aspects of application. The visualization of 3D data is facilitated by three dimensional polygonal meshes. They are now considered as the fundamental block for the implementation of Virtual Reality Modeling Language (VRML), a standard for storing and interacting with graphic objects and virtual world over the World Wide Web (WWW). Additionally hierarchical representation of 3D meshes has caught the attention because it: 1) provides rendering at various levels of detail; 2) allows progressive/ scalable transmission. Whereas by the term scalability it is meant that terminals of different complexity can extract data of different quality levels from the single bit stream. The term hierarchical representation in computer literature refers to geometric method for fine to coarse 3D mesh simplification. Meshes of tetrahedra have many applications, including interpolation, rendering, compression and numerical methods such as the finite element methods. To ascertain the correct visualization the tetrahedra must be well shaped. In this research work we propose a new hierarchy of 3D Delaunay meshes and we only remove vertices in the fine to coarse design strategy.

The proposed method can be regarded as a tool which will facilitate the progressive transmission on internet. The application area of this technology is in the WWW, telemedicine, architecture and design, military fields, games etc. The method is based upon considering the fundamental issues of design first, encoding for progressive transmission on a channel then taking output from the channel, decoding and in the last as visualization which is clearly depicted in the Figure1.1.

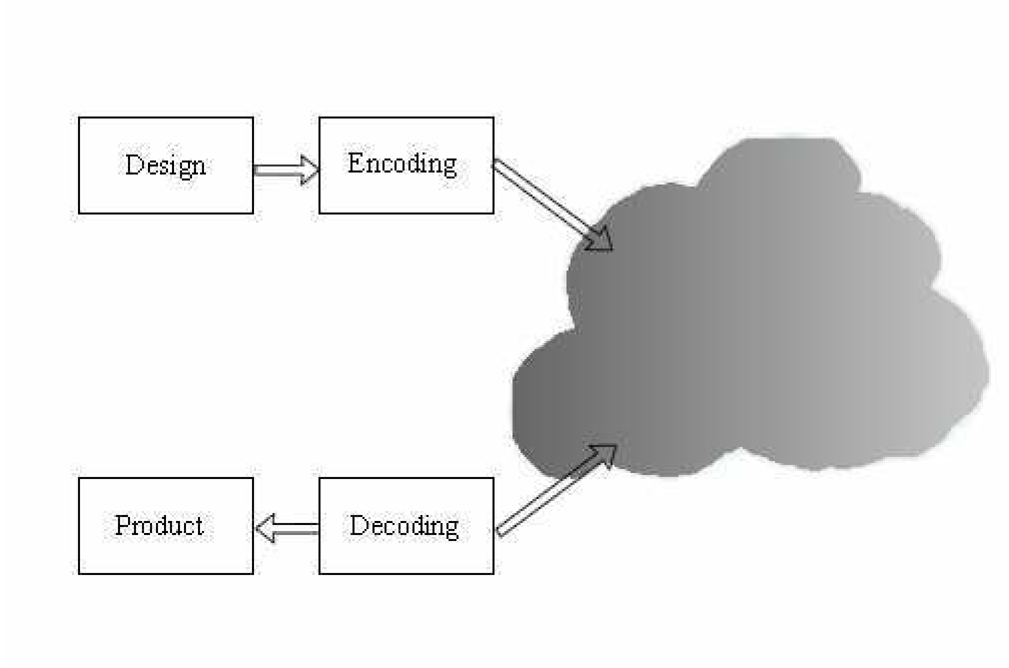


Figure 1.1: Explaining the basis of Coding Scheme.

The proposed method takes the 3D data in the shape of vertices which are processed for 3D Delaunay Triangulation, the data is then divided between boundary vertices and inner vertices by applying a vertex differentiating technique as explained in section

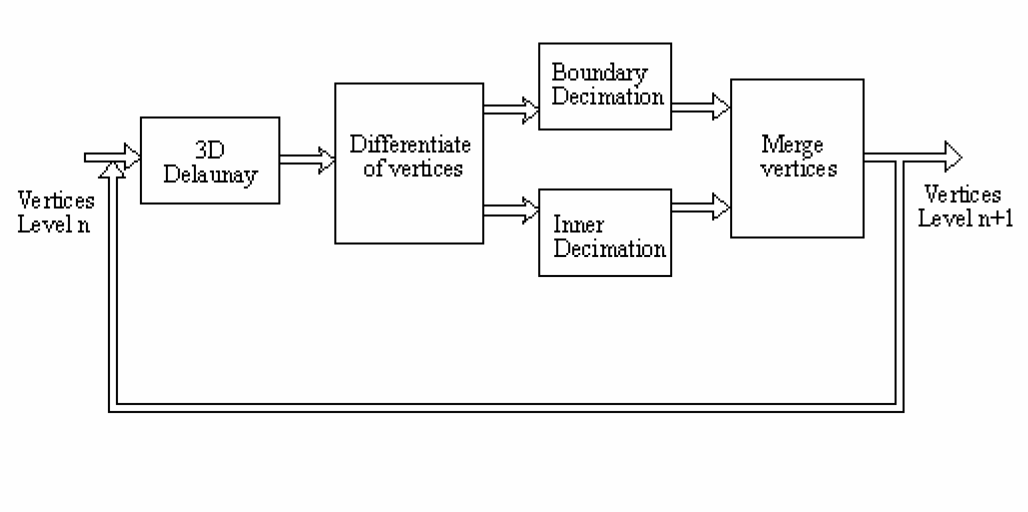


Figure 1.2: Explaining the different processes applied for decimation of data in each level.

Different decimation methods are applied over which is dealt in the Section 3, the vertices are then merged together and processed again for the next level of hierarchy.

At each decimation level the final output from the process discussed above is sent for compression techniques to be applied on it. We have tried the Scalar quantization and Vector quantization also the output of which is then Huffman Coded and at the end Bitstring is also generated.

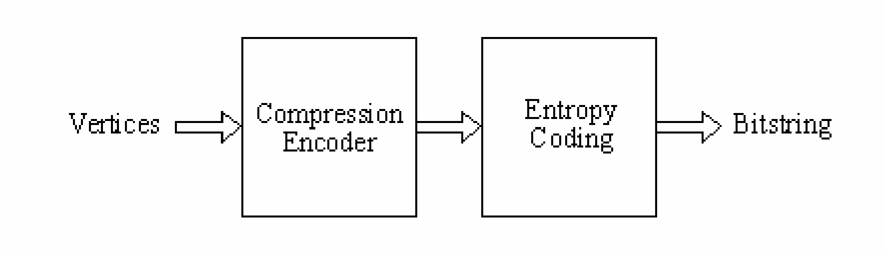


Figure 1.3: Block diagrams of the applied compression scheme.

When looking at the Decoding process of the proposed algorithm, which is depicted in Figure 1.4, it is clear that it is more or less reversal of encoding process. Initially bit string is received at the decoder receiver end which applies the Entropy decoding process over it. The decoded data is then fed to decompression decoder where the reverse of compression algorithm is applied. This reversal process produces the data in vertex format with the information about inner and boundary vertices accompanying a global alpha. The data is then fed to Delaunay triangulation algorithm for visualization which is supplied to the displaying device.

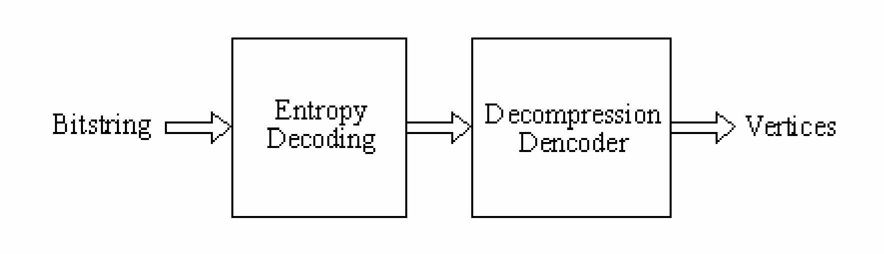


Figure 1.4: Block diagrams of the applied Decoder.

2. THE FUNDAMENTALS

There are various types of data sets which are encountered while carrying out the research for three dimensional data visualization. Some of their important types are illustrated in Figure 2.1. A dataset consists of two pieces which are the structure and the attributes. The structure of the dataset is composed of two parts topology and geometry. The structured data has regular structure in topology and in attributes. The examples are structured points, rectilinear grid, and structured grid. Whereas, the unstructured dataset has an irregular structure which can not be expressed by any mathematical expression, the examples are polygonal data, unstructured points and unstructured grid.

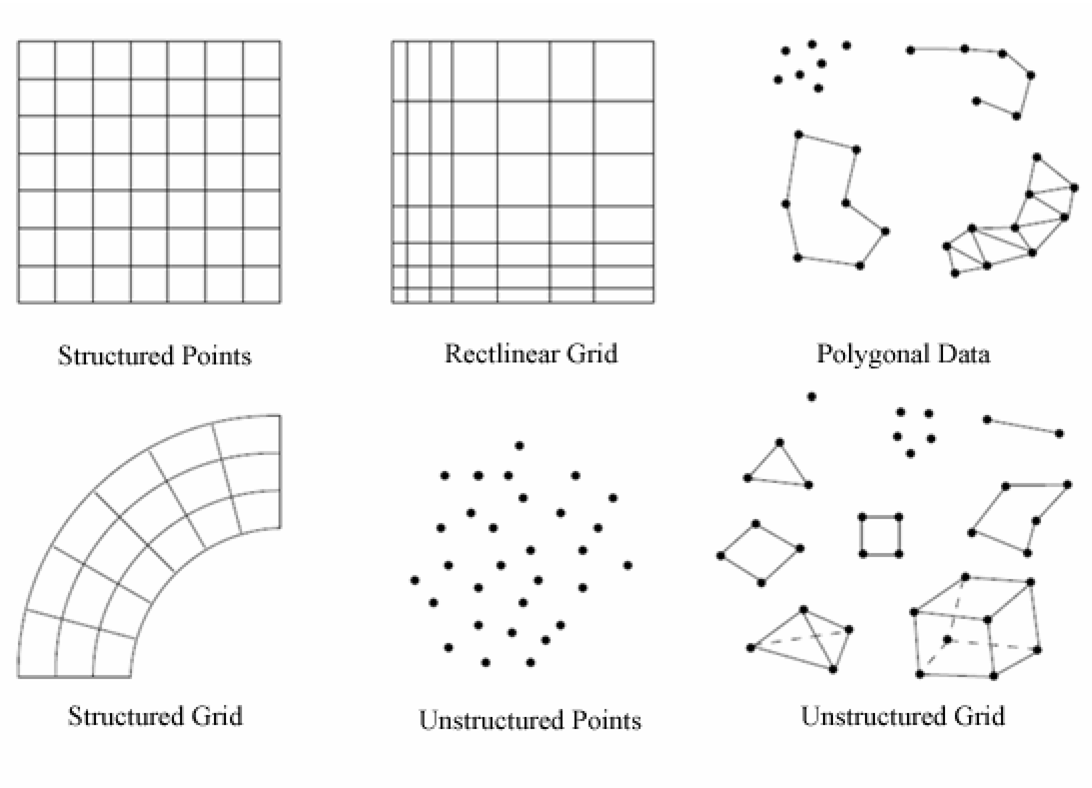


Figure 2.1: Illustration depicting various types of data sets.

Continuous physical systems, such as the airflow around an aircraft, the stress concentration in a dam, the electric field in an integrated circuit, or the concentration of reactants in a chemical reactor, are generally modeled using partial differential equations. To perform simulations of these systems on a computer, these continuum equations need to be discretised, resulting in a finite number of points in space (and time) at which variables such as velocity, density and, electric field are calculated. The usual methods of discretisation are finite differences, finite volumes and finite elements, use neighboring points to calculate derivatives, so there is the concept of a mesh or grid on which the computation is performed.

There are two mesh types, characterized by the connectivity of the points. Structured meshes have a regular connectivity, which means that each point has the same number of neighbors (for some grids a small number of points will have a different number of neighbors). Unstructured meshes have irregular connectivity: each point can have a different number of neighbors.

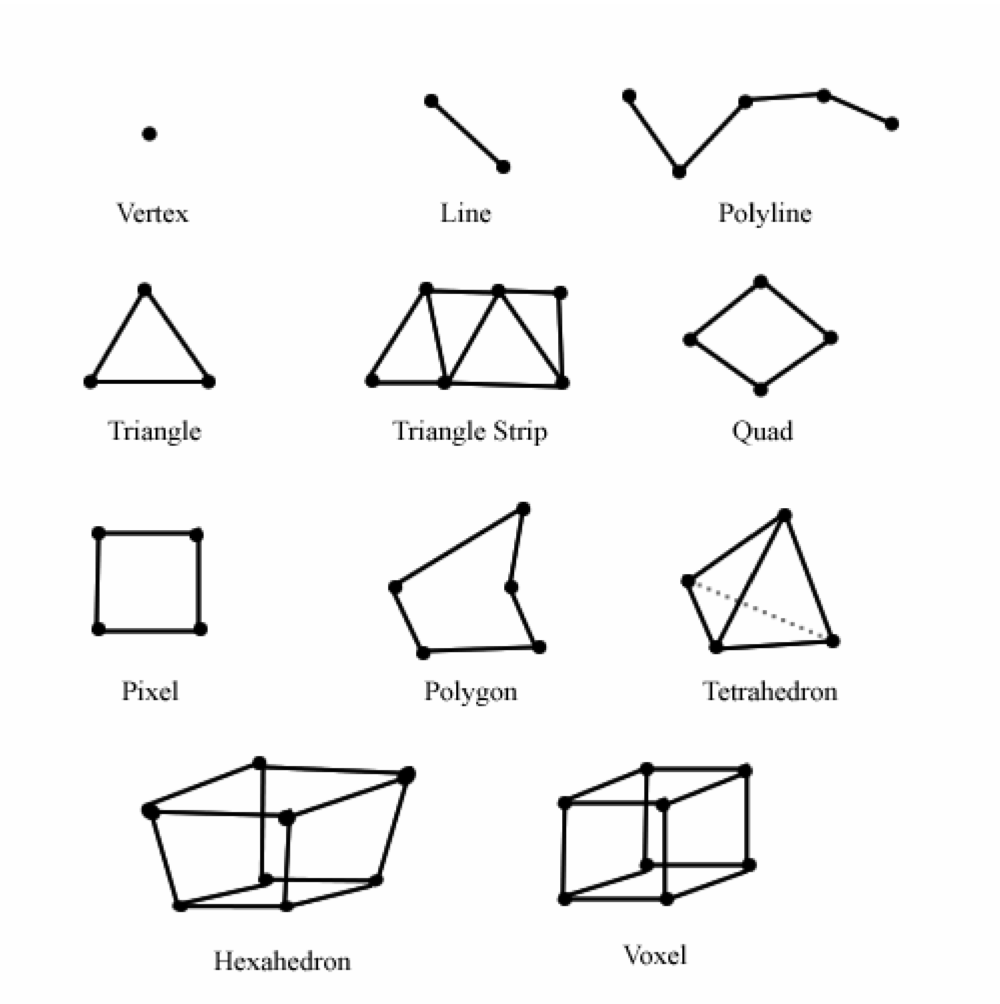


Figure 2.2: Unstructured Grid Data Set

2.1 Unstructured Points and Unstructured Grid

Unstructured points [1] are a simple but important type of data set. Normally they possess no inherent structure, and part of the visualization task is to create it. Combination of these unstructured points via unstructured cells forms the unstructured grid data type. In unstructured grid data both the topology and geometry is unstructured. All cell types can be combined in arbitrary combinations in an unstructured grid. Hence the topology of the cells ranges from 0-D (vertex) to 3-D (tetrahedron). Any dataset can be expressed as an unstructured grid.

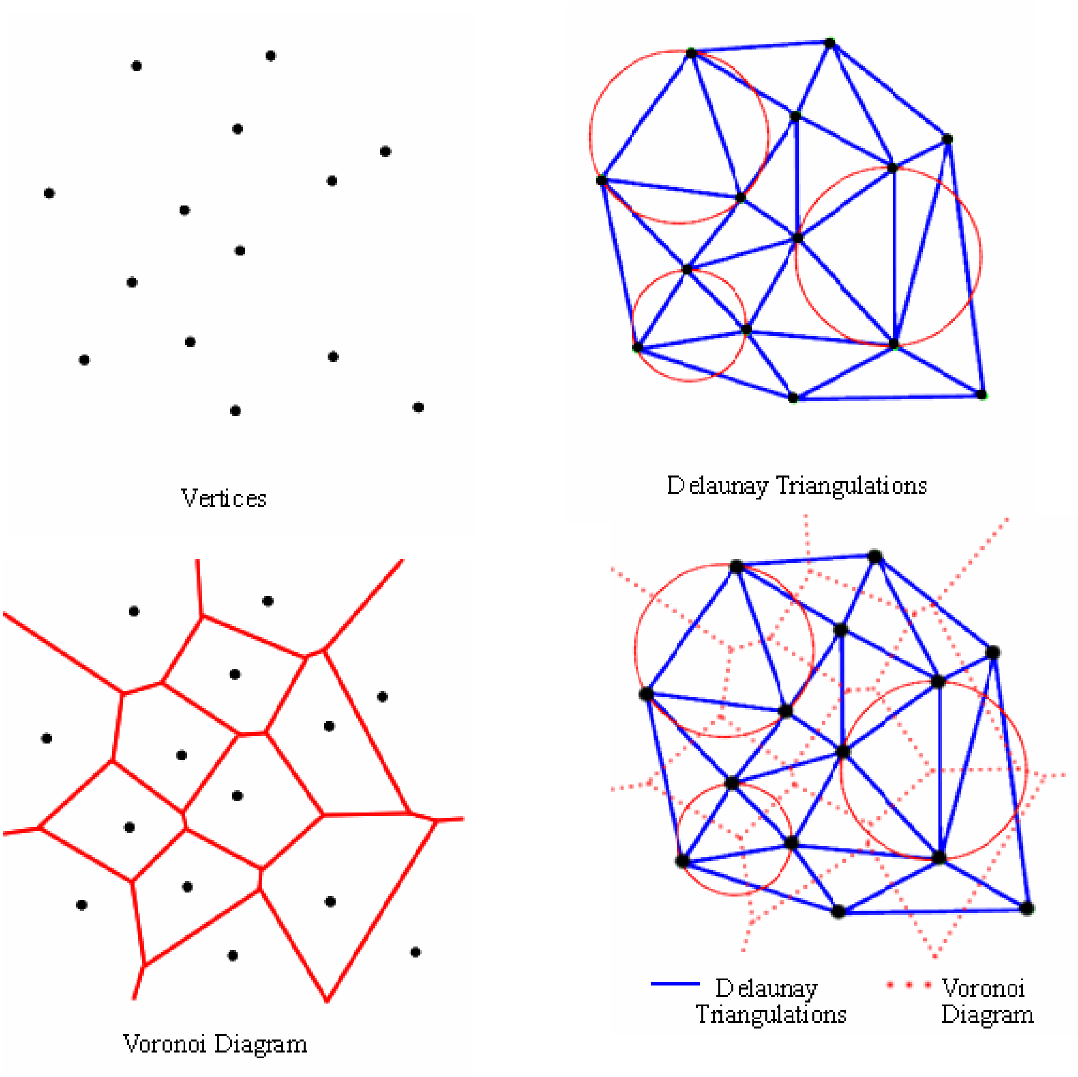


Figure2.3: (a) Showing the unstructured points (b) Showing Delaunay Triangulation (c) its dual the Voronoi or Dirichlet tessellation (d) showing both simultaneously.

2.2 Delaunay Triangulation

To visualize the unstructured points the Delaunay Triangulation [16] is the fastest and the cheapest if the load on computer is considered. The Delaunay triangulation of a set of points has a well developed theory (here triangulation includes `tetrahedralisation' in 3D). The techniques used to generate the triangulation can obviously be used to generate unstructured meshes.

A *Delaunay triangulation* Σ of V (points) is a triangulation of V such that the circum-circle of any triangle belonging to Σ does not contain points of V in its interior. The Delaunay triangulation of a set V of points is *unique* provided that no four or more points of V are co-circular. The Delaunay triangulation is proven to be the dual of another algorithm known as Voronoi pattern which is also drawn on the data. Let V be a finite set of points in the plane: Given a point p ε V, the *Voronoi region* of p in V, denoted as RV(p), is the locus of the point of R^2 defined as follows:

$$RV(p) = \{q \mid \varepsilon\ R^2 \mid d_E(p,q) \leq d_E\ (w,q),$$

for every w | ε | V-{p}}, where d_E denotes the Euclidean distance. the collection of the Voronoi regions of the points of V defines the *Voronoi diagram* of V, denoted Vor (V).

Under the assumption that any four or more points of V are not cocircular, the *Delaunay triangulation* S(V) and the *Voronoi diagram* Vor(V) are *dual* as plane graphs:

- Every point p of V corresponds to a Voronoi region RV(p)
- Every triangle of S(V) correspond to a vertex in Vor(V)
- Every edge e=(p,q) in S(V) corresponds to an edge shared by the two Voronoi regions RV(p) and RV(q)

If any four or more points of V are cocircular, then the dual of the Voronoi diagram is a cell complex (but not a simplical complex), called a *Delaunay diagram*, which is clearly illustrated in Figure 2.3.

2.2.1 Properties of Delaunay Triangulation

Delaunay Triangulation has some very important properties:

2.2.1.1 Empty Circle Property

If points p1, p2, p3 constitutes a triangle then there should be no point inside that triangle. This property in fact enunciates the automatic implementation of Delaunay triangulation. Let *e* be an internal edge of a triangulation, *e* satisfies the *local empty circle* property if the circum-circle of any of the two triangles sharing edge does not contain the vertex of the other triangle in its interior.

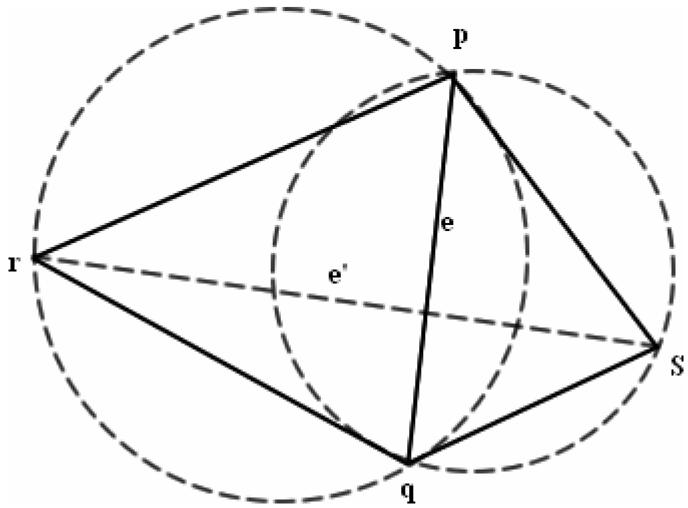


Figure 2.4: Edge e satisfies the empty-circle property**.**

2.2.1.2 Maximum and Minimum Angle Property

Let *e* be an internal edge of a triangulation then according to the *Max-min angle* property *e* satisfies the *max-min angle* property only if either the quadrilateral Q formed by the two triangles sharing edge *e* is not strictly convex (strictly convex quadrilateral= each internal angle < 180°) or *e* is the diagonal of Q which maximizes the minimum of the six internal angles associated with each of the two possible triangulations of Q.

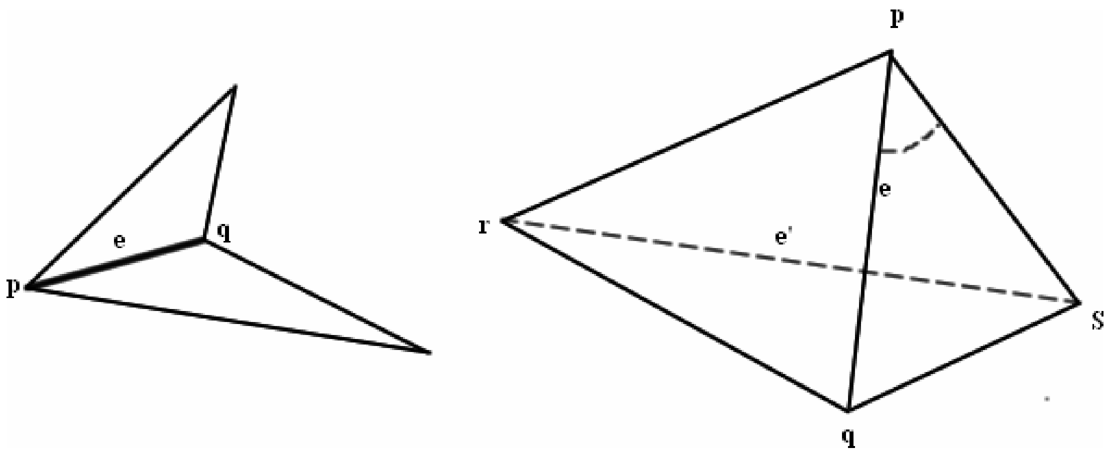


Figure 2.5: pq is the diagonal that maximizes the min. of the six internal angles; the two minimum size angles corresponding to the possible triangulations are shown

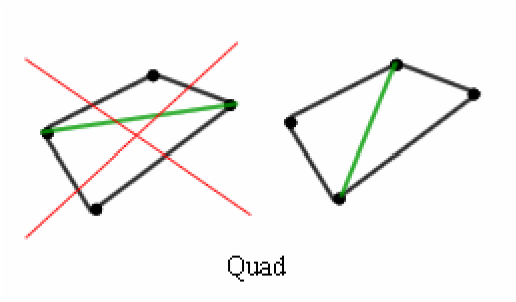


Figure 2.6: Explaining the minimum angle property

2.2.1.3 Local Optimality Property

Any edge in a triangulation which satisfies either the local empty circle or the max-min angle properties is called *locally optimal.* The local empty-circle and the max-min angle properties are locally equivalent. This issue was previously addressed and explained in Figure 2.4.

If we consider four co-circular points, then the two properties give both Subdivisions; in this case angles *cab* = *adb* = q. By moving *point d* outside the circle, the size of angle *cdb* decreases, and, thus, diagonal *ac* satisfies both empty-circle and max-main angle properties. Resultantly a triangulation of a set V of points is a Delaunay triangulation only if all of its internal edges are *locally optimal*.

Conversely, if an edge *e* connecting two points of V is locally optimal, *e* does not necessarily belong to a Delaunay triangulation of V.

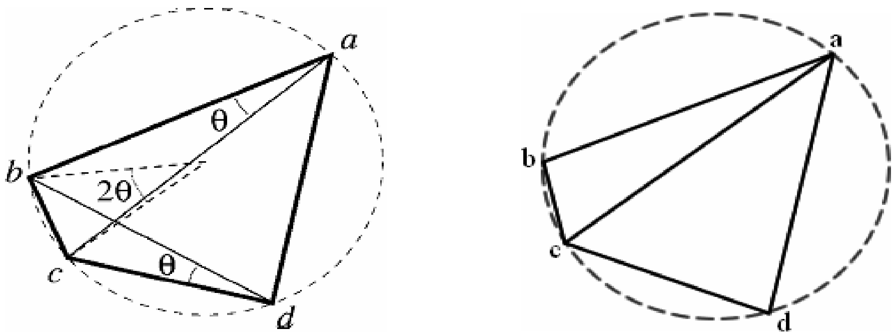


Figure 2.7: Empty-circle and max-min angle properties

2.2.2 Watson's Algorithm for Delaunay Triangulation

The Delaunay triangulation can be manipulated by using different algorithms [1]. In this project Delaunay triangulation is realized by the vtkDelaunay 3D class of Visualization Toolkit (VTK).

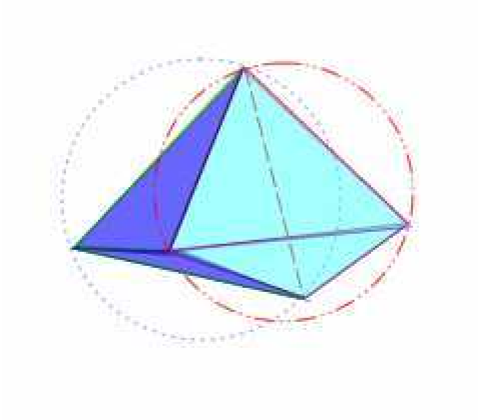


Figure 2.8: 3D Delaunay Triangulation of Volumetric data

VTK uses Bowyer and Watson's algorithm which is described for two dimensions but it can be easily realized to three dimensional. The algorithm can be described as under:

- A Delaunay triangulation is computed by incrementally adding a single point to an existing Delaunay triangulation
- Let $V_i \subset V$ and let Σ (V_i) be a Delaunay triangulation of *Vi*. Let p ε V-Vi

- We want to compute $\Sigma\,(V_{i+1})$ from $\Sigma\,(V_i)$ where $V_{i+1} = V_i \cup \{p\}$

The algorithm although optimal for 2D data but can be extended to 3D data easily. And the resultant 3D examples can be explained by the Figure 2.8.

2.2.3 Problems of 3D Delaunay

The optimality of the Delaunay triangulation 2-D is a proven matter. In 2-D Delaunay triangulation minimum interior angle of a triangle is greater or equal than any other possible triangulation. But in the 3-D Delaunay triangulation is not optimal. In 3-D it is only optimal with respect to smallest containing spheres of tetrahedra. To generate meshes of well shaped tetrahedrons in 3-D is more difficult than in 2-D.

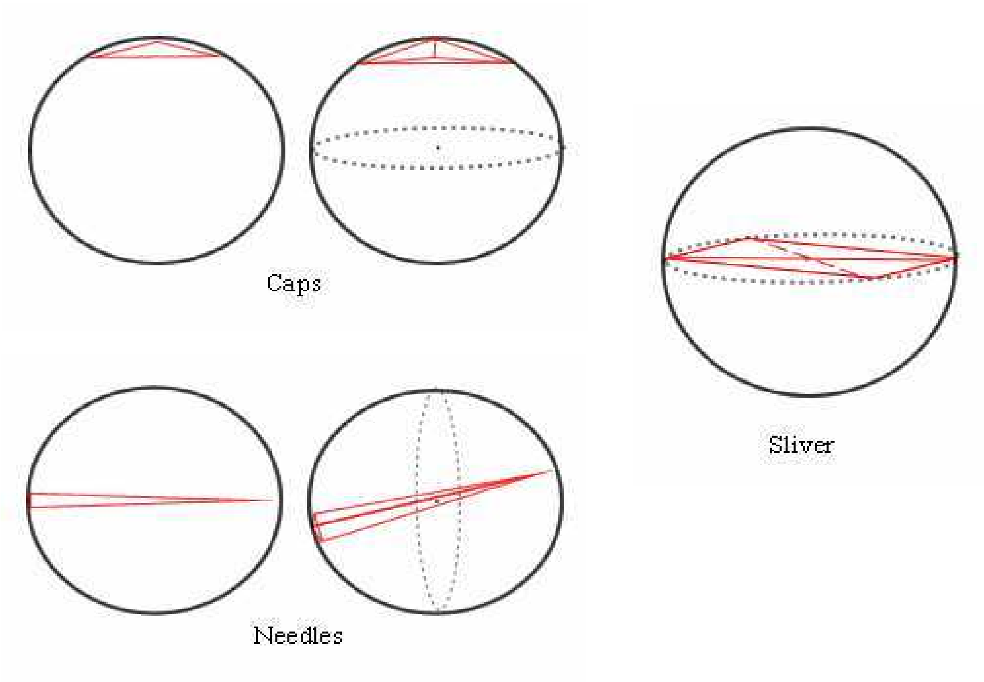


Figure 2.9: Showing different degeneracies in Delaunay Triangulation like Needles, Caps and Sliver.

The basic problem in 3D Delaunay arise when new points are injected and the result maybe a formation of tetrahedrons with poor angles. The main amongst them are:

- Needles and Wedges: Needles and wedges have edges of greatly disparately length.
- Caps have a large solid angle.
- Slivers have neither large edges nor large solid angles, but can have good circumradius-to-shortest edge ratios, so it will not be possible to compute a valid centre for the circumcirle of a triangle or the centre of sphere for a tetrahedron.

If two or more tetrahedrons or triangles are equivalent there will occur a computational confusion in the algorithm and this will cause numerical problems when circumcenters are calculated. For example, points lying at the vertices of a square, rectangle or hexagon can be triangulated more than one possible way.

2.3 Alpha Shapes

The concept of alpha shapes [4] formalizes the intuitive notion of "shape" for spatial point set data, which occurs frequently in the computational sciences. An alpha shape is a concrete geometric object that is uniquely defined for a particular point set. It thus stands in sharp contrast to many common concepts in computer graphics, such as isosurfaces, which are approximated by definition and the exact form depends on the algorithm used to construct them.

Alpha shapes are generalizations of the convex hull. Given a finite point set S, and a real parameter alpha, the alpha shape of S is a polytope which is neither necessarily convex nor necessarily connected. The set of all real numbers alpha leads to a family of shapes capturing the intuitive notion of "crude" versus "fine" shape of a point set. For sufficiently large alpha, the alpha shape is identical to the convex hull of S. As alpha decreases, the shape shrinks and gradually develops cavities. These cavities may join to form tunnels and voids. For sufficiently small alpha, the alpha shape is empty. As can be observed in the Figure 2.10 that the convex hull of the data is the shape that is formed with value of Alpha as infinite and with Alpha Zero it just becomes the point data since no triangulation can be performed.

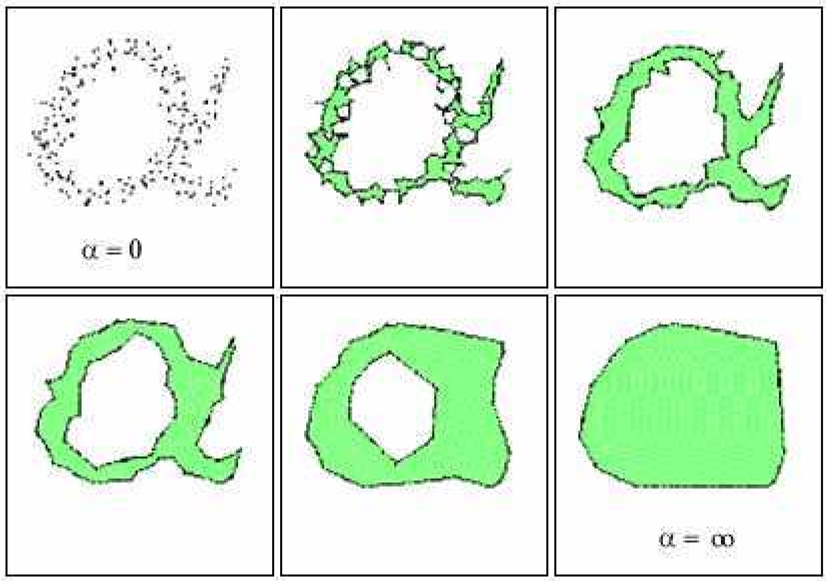


Figure 2.10: Showing the effect of Alpha values while triangulation.

A related combinatorial concept is the alpha complex. Given a point set S, it consists of vertices, edges, triangles, and tetrahedra in space. For each alpha, the alpha complex is a subcomplex of the 3-dimensional Delaunay simplicial complex.

They are specially used for shape reconstruction of unstructured data points and determined according to a parameter called alpha (α). This alpha parameter represents the level of detail of the shape. There are two types of alpha shapes

1. Global (unweighted) Alpha
2. Local (weighted) Alpha

In this project we have addressed both of the Alpha shapes for better visualization and facilitating the data for better transmission.

2.3.1 Global (Unweighted) Alpha

In the original (unweighted) definition, a piece of the polytope disappears when alpha becomes small enough so that a ball with radius alpha, or several such balls, can occupy its space without enclosing any of the points of S. Let a R^3 filled with Styrofoam and the points of S made of solid rock. Now imagine an eraser in the form of a ball with radius alpha. It is omnipresent in the sense that it carves out Styrofoam at all positions where the Styrofoam particle does not contain any of the sprinkled rocks, that is, points of S. The resulting object is called the alpha hull. For good reasons we straighten the surface by substituting straight edges for circular arcs and triangles for spherical caps. The resulting object is the alpha shape of S. It is a polytope in a fairly general sense: it can be concave and even disconnected, it can contain two-dimensional patches of triangles and one-dimensional strings of edges, and its components can be as small as single points.

2.3.1.1 Determination of Global (unweighted) Alpha

In implementing algorithm of VTK 3D Delaunay the triangulation is closely related to alpha shapes. For a simplex in the Delaunay triangulation there is a single interval of alpha values for which it will be alpha-exposed. Each shape in space has a different value of alpha so alpha should be calculated from the given unstructured data set. In order to calculate an alpha value for each shape, let E denote the class of edges and l_e denote the length of each edge. Alpha is determined by using the average edge length of the convex hull.

$$\alpha = \frac{1}{E}\sum_{i=1}^{E} l_{e_i}$$

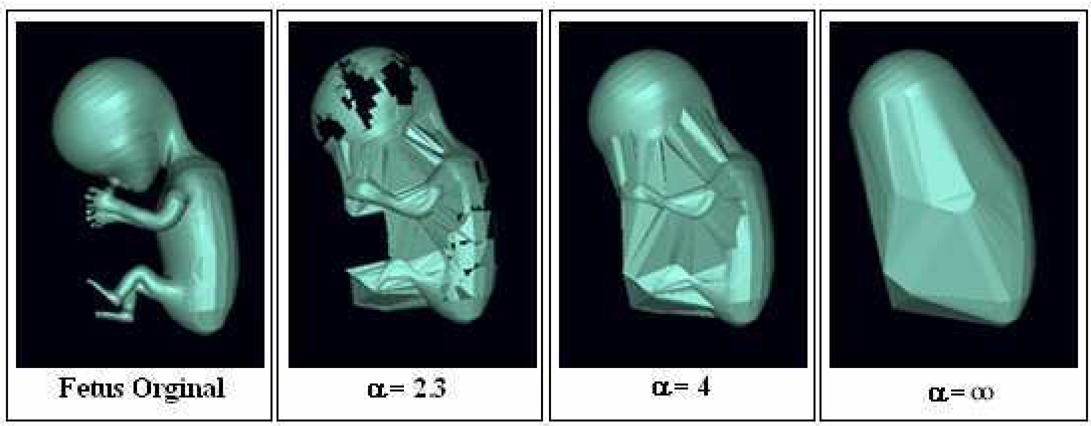


Figure2.11: The figure showing convex hull and then alpha 0.08 and 0.14 on the cat data which is inherently a surface data.

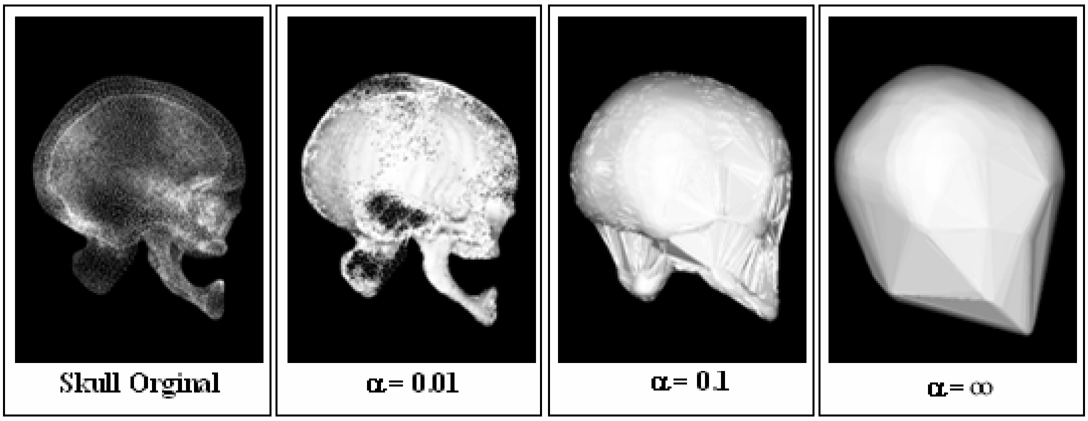


Figure 2.12: The figure showing effects of Alpha value application on volumetric data of skull starting from zero resultantly showing just the point data till it becomes a convex hull when the value of Alpha reaches infinity.

We have implemented the visualization while calculating different alpha values on fetus data, which is a surface data, and the results are evident the in the Figure2.11. Similarly the different Alpha values are applied on volumetric data of skull and the results are shown in Figure 2.12

2.3.2 Local (weighted) Alpha

Since the global unweighted alpha [3] is implemented by calculating the average length of the edges. But the implementation of global alpha only results in some parts being unfilled.

For calculating the weighted alpha we have employed grow and learn [21] on the data. According to which the data is clustered with an error constrained applied while processing. It is similar to nearest neighbor implementation but can be further elaborated in the manner which is given below:

- First entries from the data are considered as the cluster centre.
- To induce the new entry, the error is calculated between the new point and all the cluster centers.
- If the error is within range then the point is allotted in that cluster.
- If the larger than the threshold then it is labeled and placed between the two clusters according to its location.

Similarly whole of the data is traversed and clusterized. If the error criterion is very low then we will end up having large number of clusters with small size. Conversely if error threshold is kept high then we will have fewer numbers of clusters with large size.

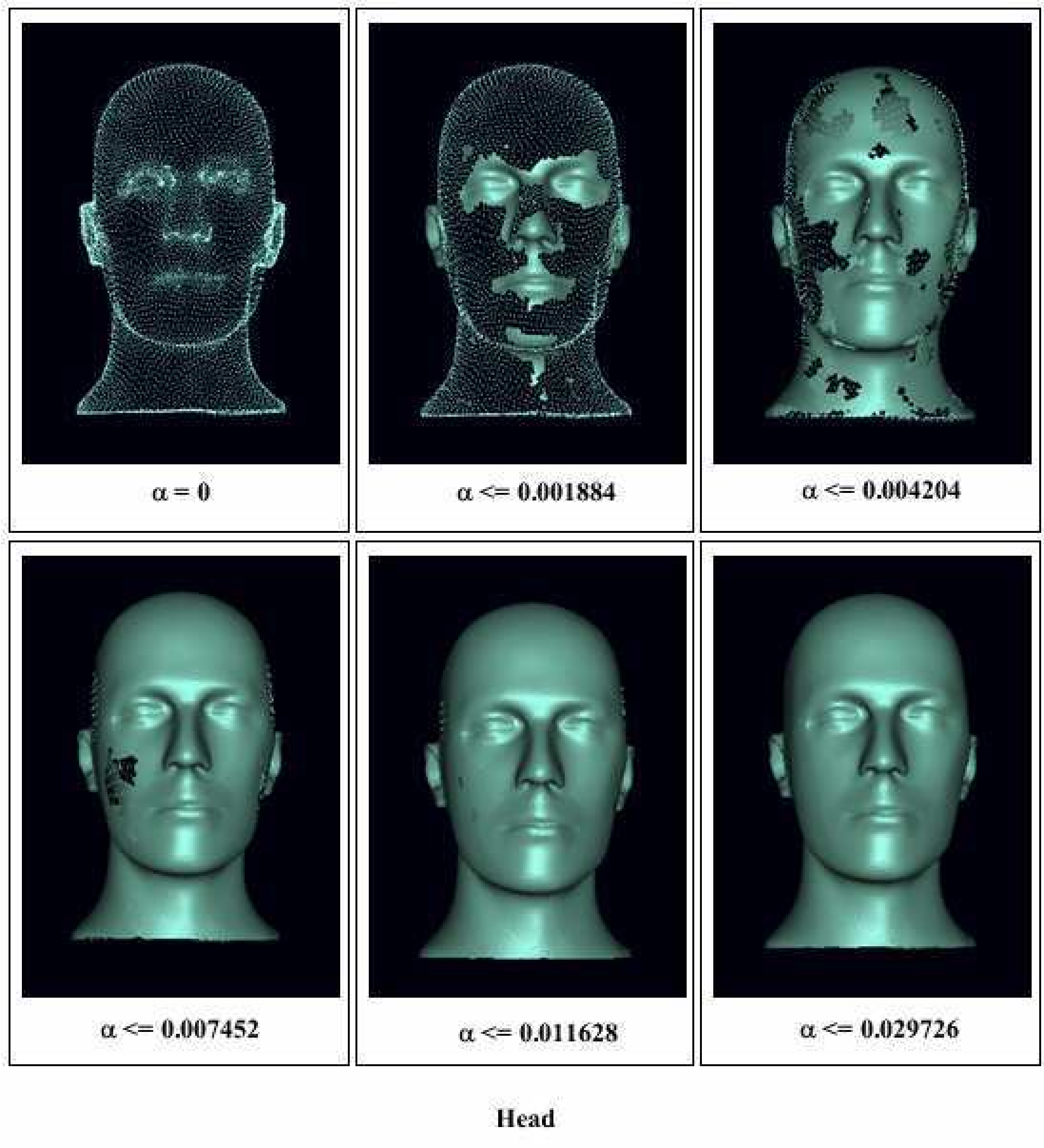


Figure 2.13: Visualization result after implementation of local (weighted) Alpha.

It is due to the fact that Delaunay triangulation is not optimal in 3D some vertices whose distance is more than the alpha value are not included so a hole is left behind.

To address this issue we have been able to devise a unique algorithm that is composed of different filters which are being applied over the data to produce the desired results for visualization. The Figure 2.14 shows the block diagram of the algorithm being applied.

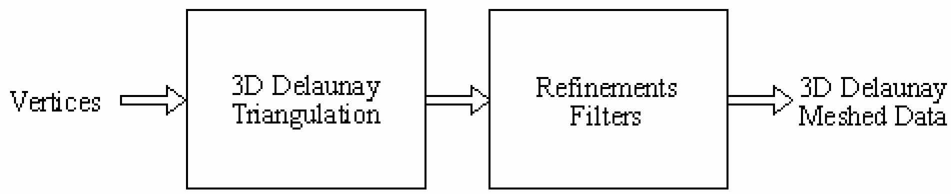


Figure 2.14: The Block Diagram of Local Alpha Implementation

Initially whole of the data, which is in unstructured point data form, is induced and in the first phase 3D Delaunay Triangulation is applied with alpha equals to infinity the resultant is the convex hull visualization of the data. Then there are five filters in line which are applied on the Delaunayed data to reconstruct its surface and volume.

- Circle Radius Filter
- Min/Max Edge Filter
- Density Filter
- Calculating Neighbor Count
- Gaps Control Filter.

The logic of these filters is applied globally on the data initially. For further refinement in visualization similar types of filters are applied locally on the data.

2.3.2.1 Circle Radius Filter

The radius of all the circles forming due to the Delaunay triangulation and tetrahedralization are calculated. An average is taken of all those radii. A filter is then introduced in the algorithm which employs a restriction that if radius of the sphere formed by the tetrahedron is less than the threshold, it let the tetrahedron to be included but if not then it divides the tetrahedron further and analyze the four triangles which constitutes that tetrahedron. In that analysis the triangle are assessed in the manner that their longest edge is taken as the radius and if that is less than the threshold then those triangles are kept within the data and are discarded otherwise. This also eliminates those tetrahedrons such slivers for better visualization. The effects of the filter can very easily be noticed from the Figure 2.15 in which before the implementation of the filter, some points were triangulated which actually distorts the original shape of the data. But after the filter employment we have improved visual results.

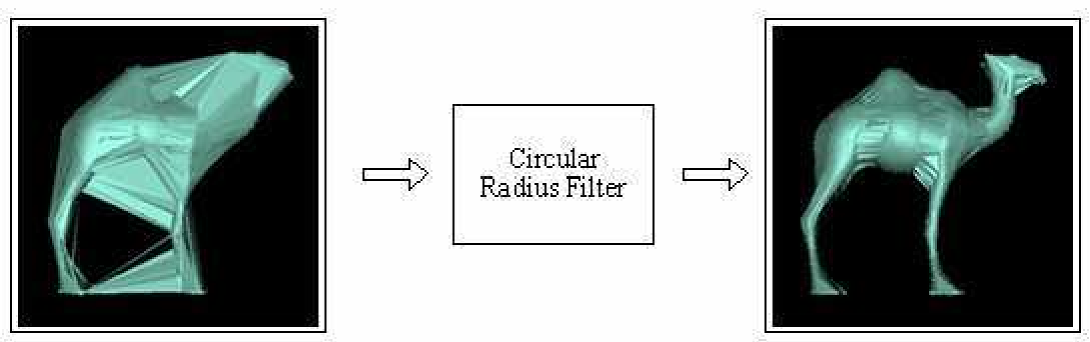


Figure 2.15: Effect of Circular radius filter on the visualization of the data.

But if the applied data is volumetric then the algorithm only deals with tetrahedron and do not further divide them into triangles.

2.3.2.2 Min/Max Edge Ratio Filter

This filter also calculates the min/max edge ratio of all the triangles and tetrahedrons and provides a threshold accordingly. The Filter acts in manner that if the min/max ratio is less than the threshold then it discards that triangle. Similarly when being applied over tetrahedrons it assesses the tetrahedrons first and if the tetrahedrons have lesser min/max edge ratio then it further divides the tetrahedrons into triangle and analyzes them separately. While conducting this analysis it also discards all those triangles which have their value less than the min/max edge ratio and keep those who are lying within the range. The effects of the filter can very easily be noticed from the Figure 2.16 in which before the implementation of the filter, some points were triangulated which actually distorts the original shape of the data. But after the filter employment we have improved filter results.

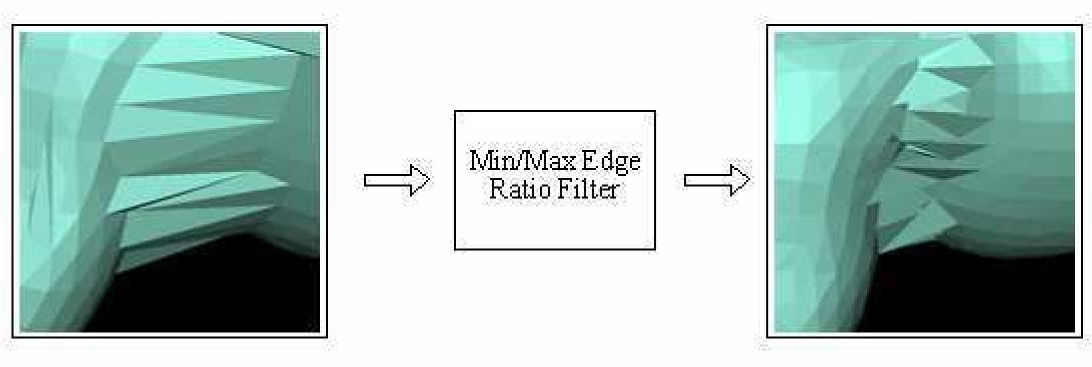


Figure 2.16: Illustration of Min/Max Edge Ratio Filter

But if the applied data is volumetric then the algorithm only deals with tetrahedron and do not further divide them into triangles.

2.3.2.3 Density Threshold Filter

This filter calculates the density of points in per unit area and also the neighboring points of each tetrahedron. Then, it calculates the average density and the neighbor of the tetrahedrons and triangles. It employs a threshold which is based upon the density of the neighboring points. If the triangle has density less than the threshold then it is discarded but is kept otherwise. For tetrahedron like the above discussed filters it initially analyze them as an entity but if one is not eligible to be filtered then it further divide the tetrahedron into triangle and calculate the density individually. While carrying out the analysis those triangles which have density less than the threshold are discarded. The effects of the filter can very easily be noticed from the Figure 2.17 in which before the implementation of the filter, some points were triangulated which actually distorts the original shape of the data. But after the filter employment we have improved visual results.

Here an important point is to be noted that since the density also is dependent upon the size of the tetrahedron also, so this feature is also kept in view.

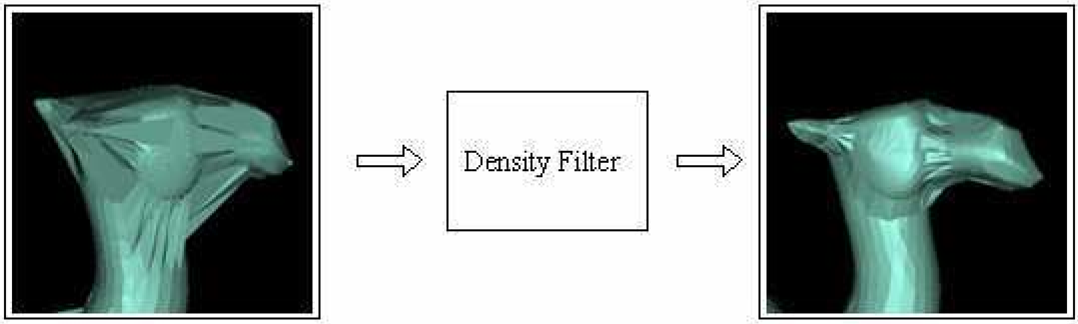


Figure 2.17: Implementation of Density Filter on the data.

But if the applied data is volumetric then the algorithm only deals with tetrahedron and do not further divide them into triangles.

2.3.2.4 Removal of Lone Triangle

This filter is activated in the case when it notice a triangle with least of the neighbors or whose only one point has neighbor and the rest of the two do not have any. In that case it removes that triangle. The effects of the filter can very easily be noticed from the Figure 2.18 in which before the implementation of the filter, some points were triangulated which actually distorts the original shape of the data. And there were some lone triangles connecting those points which are not to be connected. But after the filter employment we have improved filter results.

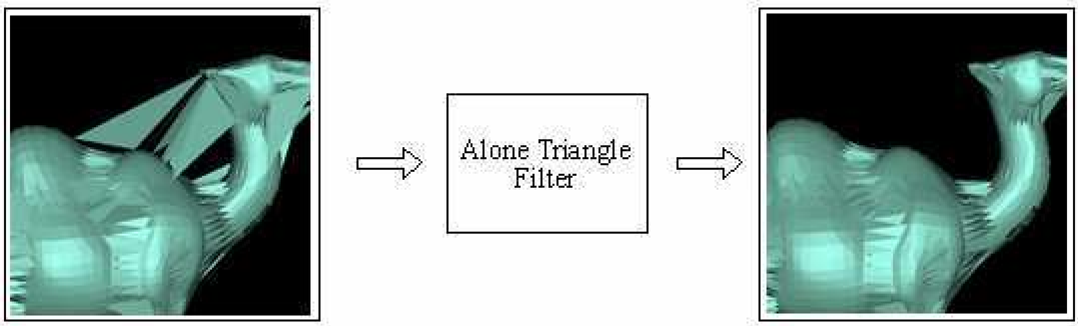


Figure 2.18: Alone Triangle Filter is implemented on the data.

2.3.2.5 Gaps Control

While applying all the filers over the data it is necessary to continuously check the over all topology and care must be taken that in case of deletion of a triangle hole is not produced. To solve that problem we have applied Gaps control method which is continuously assessing the number of neighbor, which if less than two suggests that the hole has appeared so the deletion of that triangle is not executed. Figure 2.19 is clearly showing the appearance of those holes while triangulation and then illustrates that after the implementation of Gaps control filter those hole were refilled and the triangles were not deleted at that particular place. But if the data is of an open type then we have the list of all those triangles which are forming at the boundary of the data so that exception is handled logically.

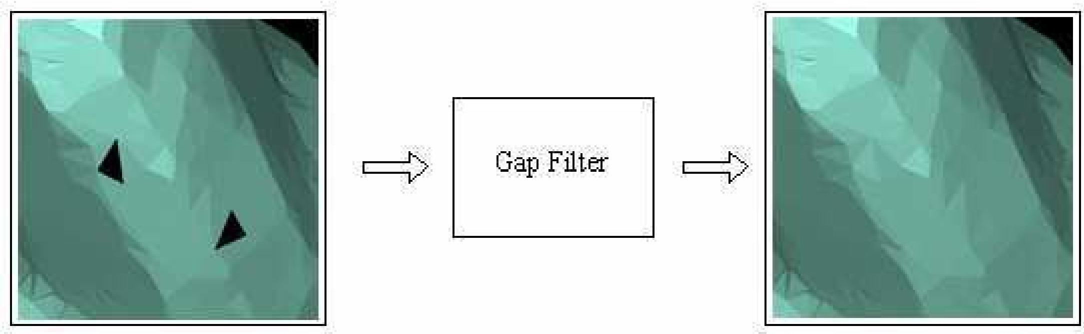


Figure 2.19: Illustration of Gaps Control Filter Implementation on the data.

After the implementation of these four global filters we have two more filters in our algorithm which are based upon local calculations which are:

- Neighbor Circle Radius
- Neighbor Min/Max Edge Ratio.

2.3.2.6 Neighbor Circle Radius Ratio

This local filter is applied on every tetrahedron and triangle to assess its over all standing and affirmative with its neighbors. If the circle radius of a triangle or tetrahedron is within the range in the average circle radius of its neighbors then it is kept within the data otherwise discarded. In similar to its global counterpart it also assesses the tetrahedrons initially as the single entity but break them further down, if the radius is greater than the average of its neighbors, into triangles and then discard that triangle which has the value greater than the average. Figure 2.20 shows the effect of localized neighbor circle radius ratio filter when applied over the data. The improvement in the visualization can be noticed very easily.

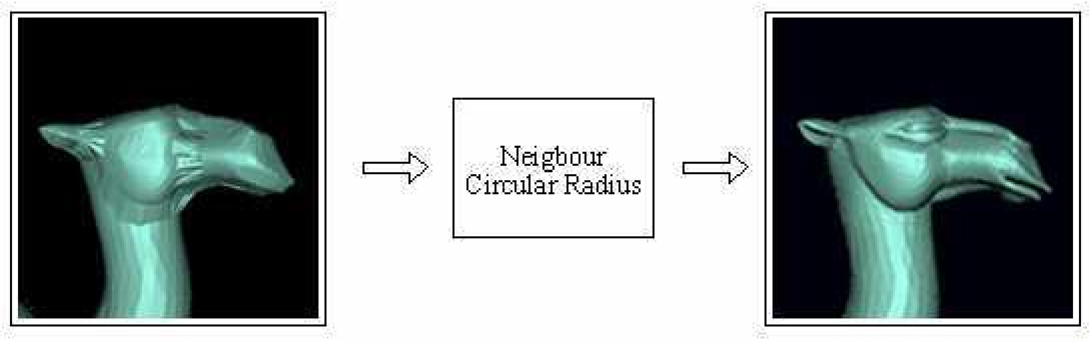


Figure 2.20: Effects of Neighbor Circle Radius Ratio Filter on the visualization of the data.

2.3.2.7 Neighbor Min/Max Edge Ratio

This filter is also applied on every triangle and tetrahedron with its neighbor perspective. If the Min/Max Edge ratio is greater than the average of its neighbor then it is kept in the data otherwise discarded. In similar to its global counterpart it also assesses the tetrahedrons initially as the single entity but break them further down if the Min/Max is less than the average of its neighbors into triangles and then discard that triangle which has the value lesser than the average. Figure 2.21 shows the visualization results of the data before and after the implementation of this filter and the improvement is clearly depicted.

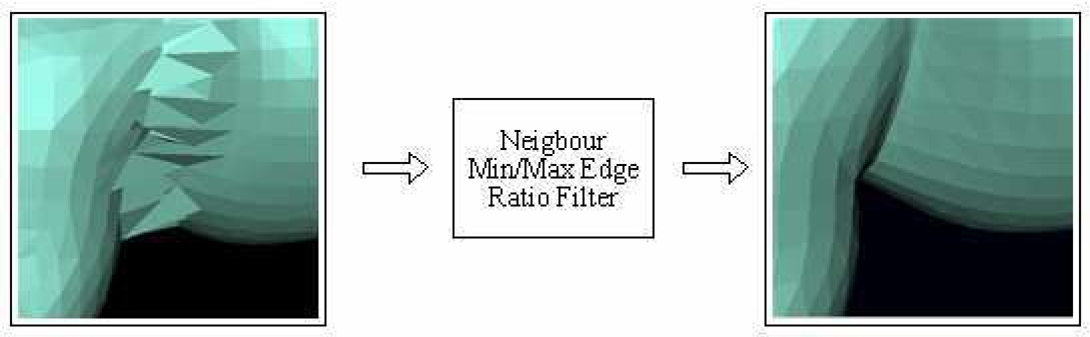


Figure 2.21: Effects of Min/Max Edge Ratio filter on the visualization of the data.

By the term local density we tend to refer the concentration of the points in a given area. It is calculated by the length of the edges in a region. Shorter the length of edges in the region implies that the vertices are closely located and this proves the our point of higher density values.

2.3.2.8 Calculating Neighbor Count

After the first 3D Delaunay triangulation of the original data lot of tetrahedrons and triangles are formed. So a single vertex must have two neighbors at least but there is no upper bound for that and a vertex may have more than twenty neighbors. The neighbor count is very important for a vertex. The more neighbor it has higher is its importance value. Additionally the neighbor count also helps in defining a certain area of the data. For calculating the neighbor count of the data we have developed an algorithm which was based upon vtk libraries. While counting neighbors of a tetrahedron we traverse all of its triangles and then find the cell count. If it is two then it is registered as the other cell Id. We use the vtk library function Unstructured Grid GetCellNeighbors().

By keeping an account of both the local density and the Neighbor count and implementing it iteratively for each vertex we finally are able to divide any 3D data into different regions. Through utilizing the radius of the circumsphere of the constructed tetrahedrons and the value of global (unweighted) Alpha of the data we deduce a new alpha value for that particular region. Hence we calculate different Local (weighted) Alpha for various regions. Each region is then retriangulated separately. At the end, all regions are combined to form the complete visualization of the whole data.

We have applied the unweighted Alpha values on the Surface data of Head which is being shown in figure 2.13 and on volumetric data of hand, which is shown in figure 2.22 and 2.23. The figure 2.13 depicts the implementation of the weighted Alpha on the surface data, starting from the value of Alpha zero the result of which can intuitively guessed as point data and then progressively increasing its value till all traverses all the data.

Similarly 2.22 and 2.23 manifests the weighted Alpha implementation on the volumetric data of hand, the Figure 2.14 explains the different values of weighted Alpha applied to different regions whereas in 2.15 the same routine as of 2.13 is taken that is to start from zero and then a progressive application of weighted Alpha till whole of the data is traversed.

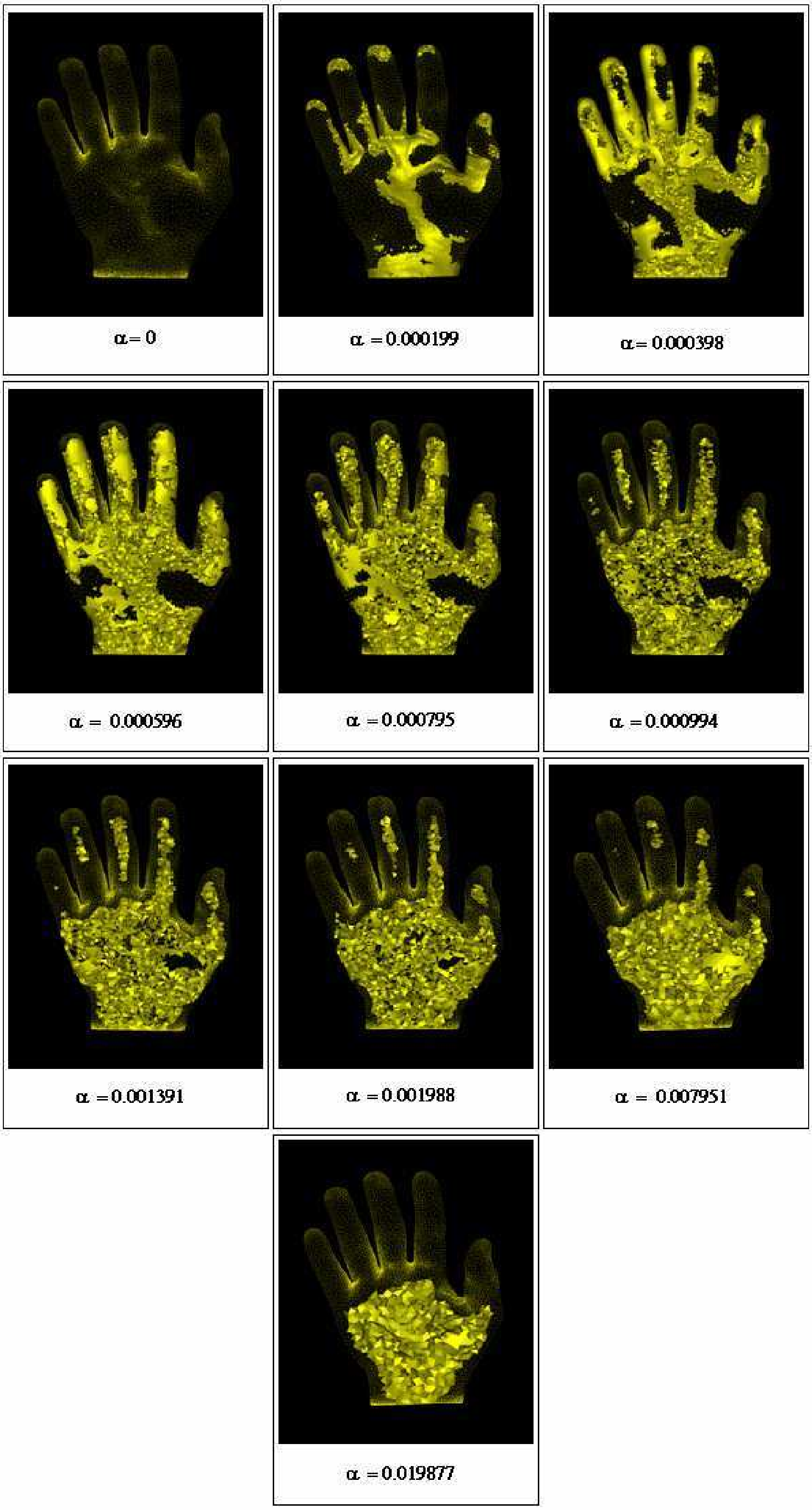


Figure 2.22: Starting with the weighted value of Alpha by zero we have just shown the different values of weighted Alpha being applied to different regions according to the density of points.

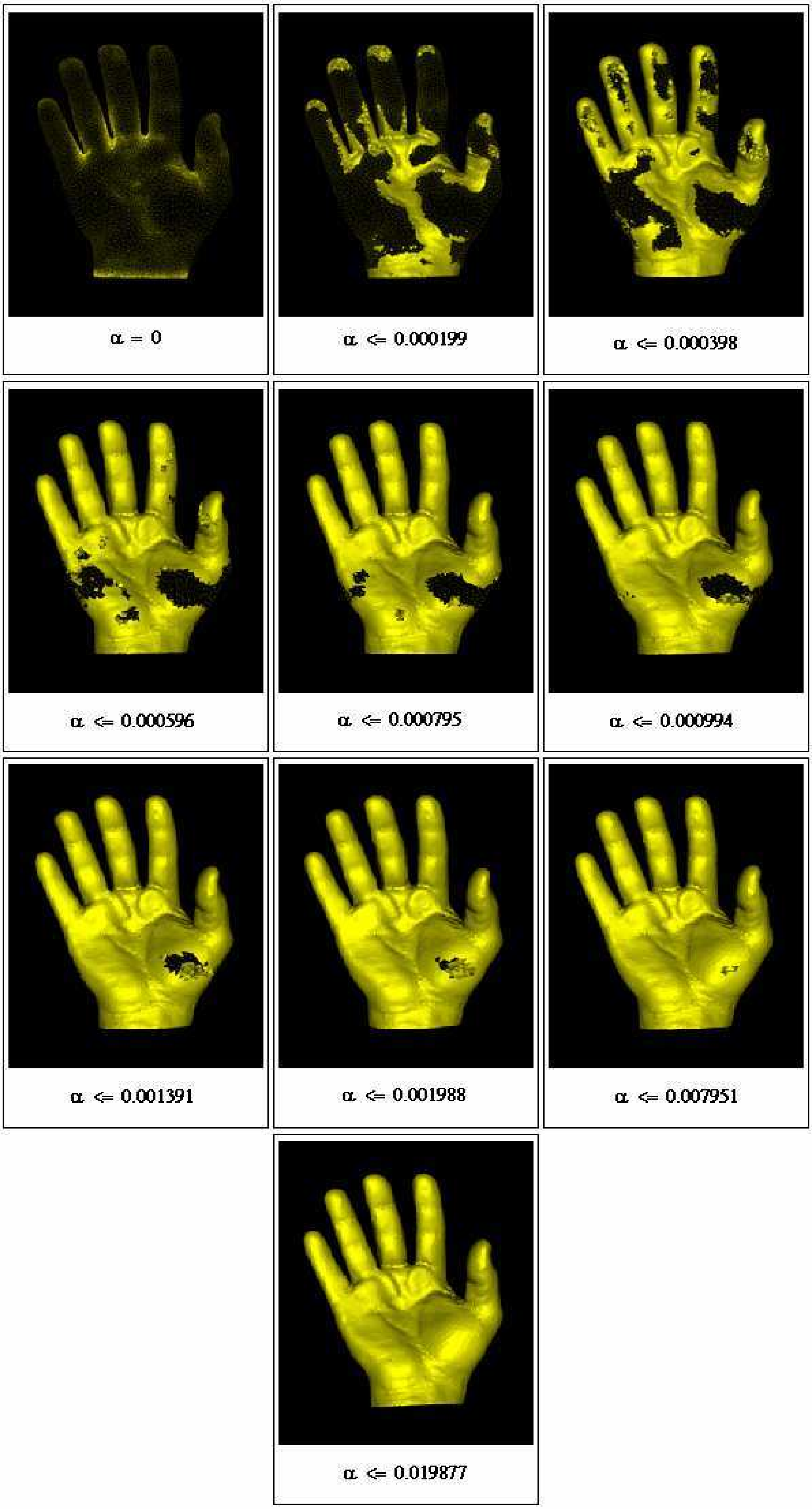


Figure 2.23: Similarly starting from the zero value of the weighted Alpha value progressively adding the last levels till whole of the data is being triangulated.

3. DATA DECIMATION

Hierarchical representation [12] of meshes provides rendering at various LODs, implying quality scalability in order to render the object at a reduced quality or to reduce the rendering time of objects and allows progressive/scalable transmission or storage of the object geometry. A hierarchical mesh model that is compressed in an embedded format thus enables different users to employ meshes of varying level of detail to represent the same content depending on the computational resources and channel/storage bandwidth available to them.

3.1 Hierarchical Mesh Representation

The Hierarchical mesh compression method is an algorithm which implements both the decimation techniques and we perform it iteratively till the quality of the mesh is maintained within some bounds.

The idea is to perform decimation from fine to coarse as shown in Figure3.1. The different levels of hierarchy are clearly depicted in Figure 3.1 when implemented on SkelHand data. It can be observed that at level 0 the visualization is very fine and then the decimating steps are being visualized from that very fine level 0 to the coarsest level 2.

The method can be explained graphically if we take example of a 2D mesh as in Figure 3.2 and we can observe that by applying both of the techniques we have reduced the amount of vertices in the data without making an imperceptible visual change.

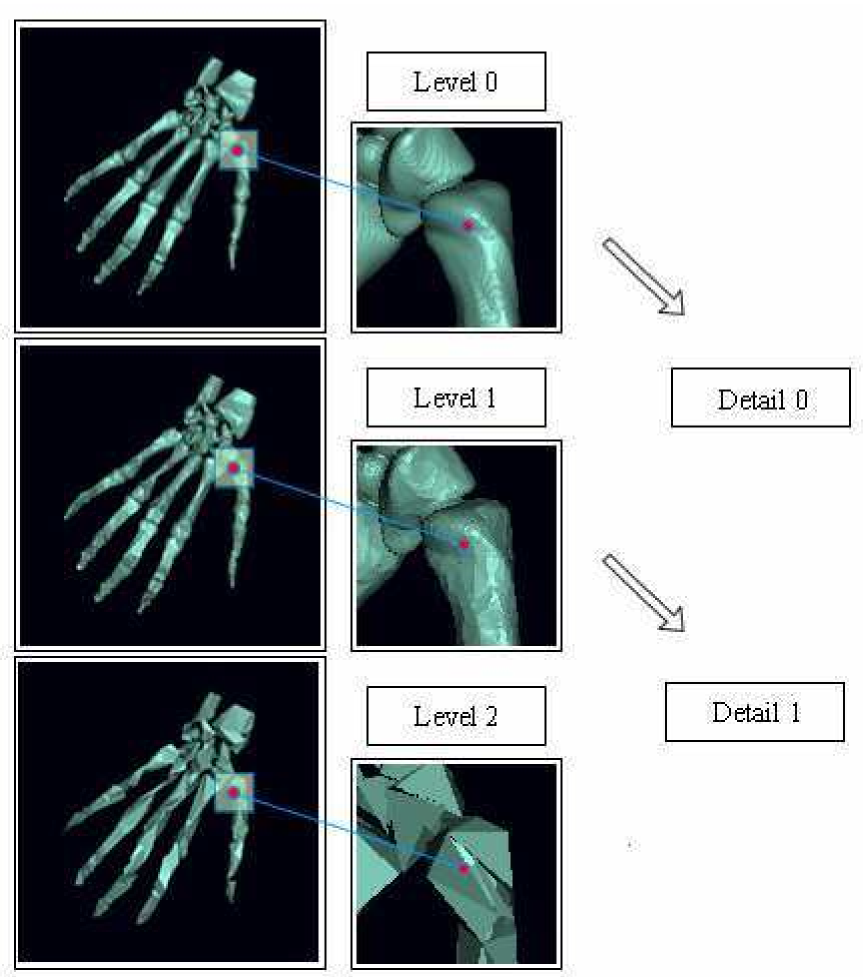


Figure 3.1: Block Diagram of Hierarchical Mesh Modeling

The size reduction is achieved by removing an independent set of vertices at each level starting from the finest resolution mesh. An independent set is a set of vertices among which no two vertices are adjacent to each other. While an independent point is removed the neighboring points are marked as not to be removed. In the following figures a fine resolution mesh is decimated two times by removing independent set of points.

In the first figure mesh M_i has the finest resolution; the independent points are situated in the regions enclosed with dark edges. Being these points independent at the second hierarch they can be removed from the mesh and the sub sampled mesh will construct the second mesh M_{i+1}. After determining the independent points for the mesh M_{i+1} the second sub sampled figure will be formed.

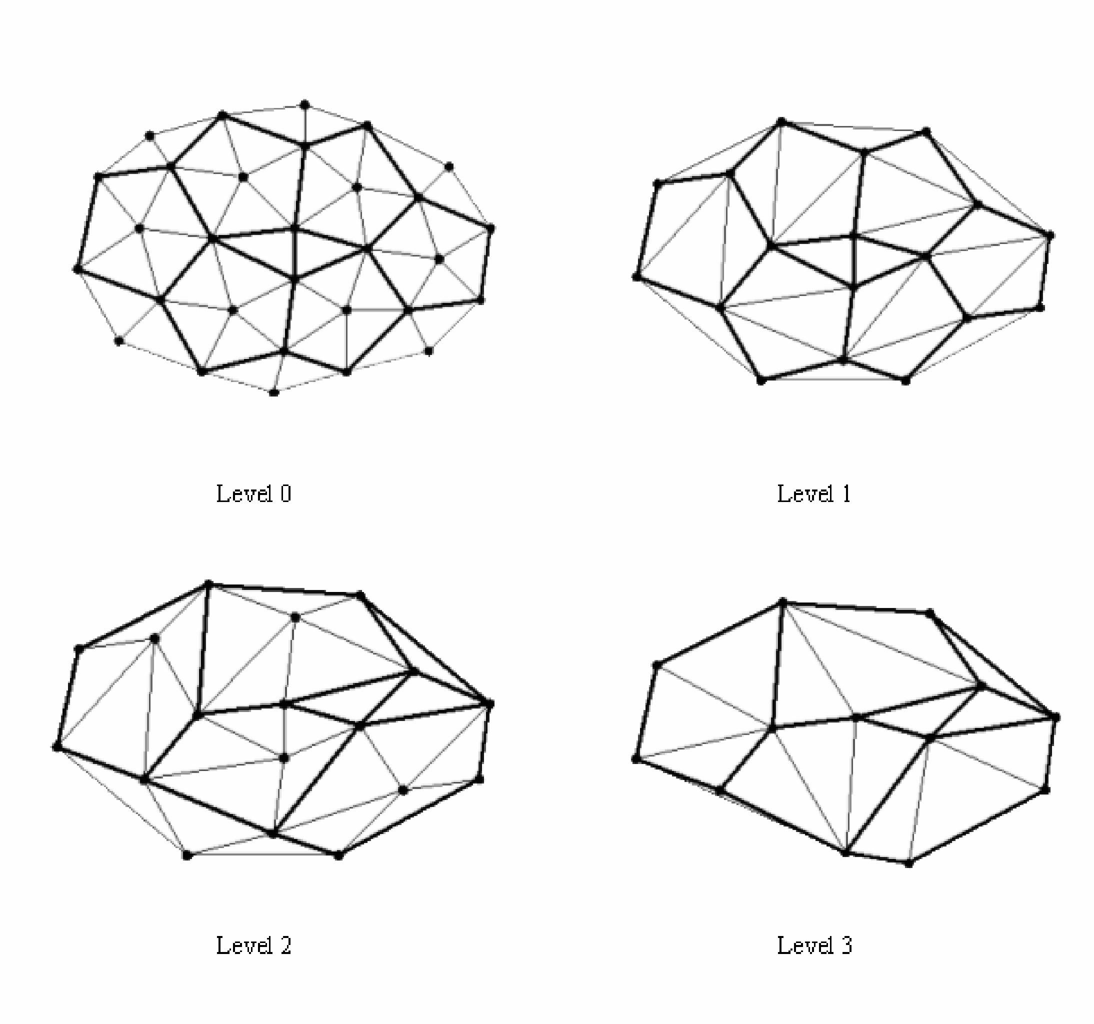


Figure 3.2: Expressing the data decimation

3D data decimation is a special type of compression technique in which the original data is reduced by reducing the number of vertices with different constraint being implemented. To start with decimation we have to first differentiate between the boundary and the inner vertices. Since for proper visualization of the data the boundary bear far more importance than the inner. Therefore we employ some constraint on the decimation and resultant is a smaller data with the least harm to its visualization.

3.2 Extraction of Boundary Vertices and Inner Vertices

While implementing Delaunay triangulation [8] created tetrahedrons generally do not intersect each other, so the boundary vertices can be determined by adding the space angles of each node (i.e. vertex). T(t_0, t_1, t_2, t_3) being a tetrahedron, the space angle of a node t_i is the projection of the remaining surface, which is a triangle of the remaining nodes, on the unit sphere centered at t_i.

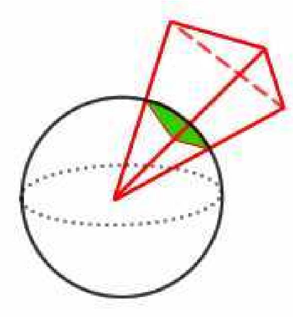


Figure 3.3: the calculation of space angle of a vertex

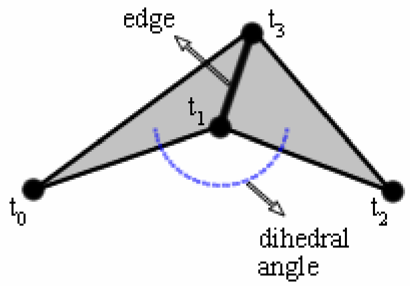
We have deduced following results and constraints for separating the vertices.

- If the sum of angles is less than 4π then it is a boundary vertex.
- If the sum of angles is equal to 4π then it is an inner vertex.
- The sum of angles can never exceed 4π.

The space angel of a tetrahedron T (t_0, t_1, t_2, t_3) can be calculated for each node by using the following formulas:

$$\psi_{t_0} = \Theta_{t_0t_1} + \Theta_{t_0t_2} + \Theta_{t_0t_3} - \pi$$
$$\psi_{t_1} = \Theta_{t_1t_0} + \Theta_{t_1t_2} + \Theta_{t_1t_3} - \pi$$
$$\psi_{t_2} = \Theta_{t_2t_0} + \Theta_{t_2t_1} + \Theta_{t_2t_3} - \pi$$
$$\psi_{t_3} = \Theta_{t_3t_0} + \Theta_{t_3t_1} + \Theta_{t_3t_2} - \pi$$

Here Θ represents the dihedral angle between two surfaces intersected on a common edge.



Dihedral angle can be calculated by using the following formula:

$$\Theta_{t_0 t_1} = \pi - \left| \cos^{-1}(face_{t_0 t_1 t_2} \bullet face_{t_0 t_1 t_3}) \right|$$

3.3 Extraction of Boundary and Inner Vertices Using vtk Library

Apart from the algorithm described above we have also employed the vtk library functions for the extraction of boundary vertices. This is only applied when the data is volumetric. This algorithm traverses all the tetrahedrons which have formed due to the 3D Delaunay triangulation. Then it observes the neighbor of each triangle of the tetrahedron. If any of the triangles has neighbor count less than one then it is a triangle which is located at the surface of the data. And all the points of that triangle are then registered as the surface vertices. We can see the effects in Figure 3.5 which

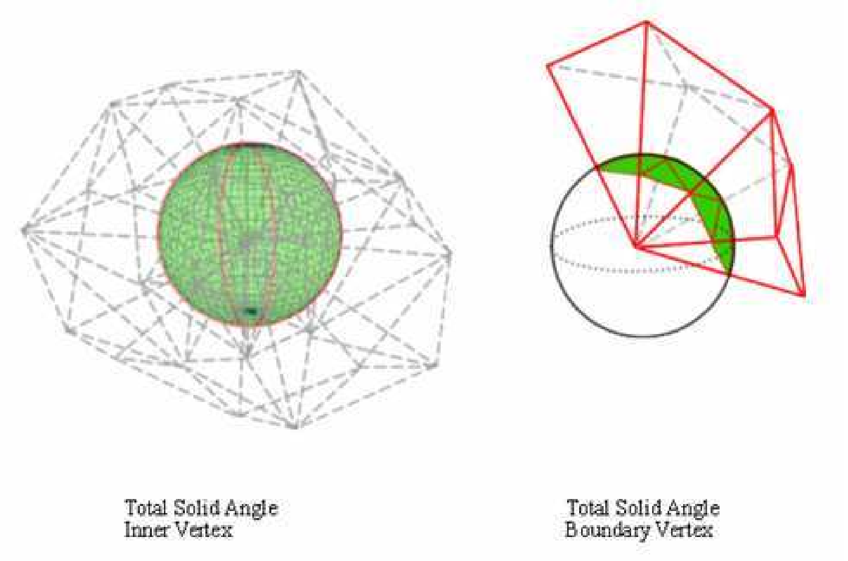


Figure 3.4: Figure showing the Extraction of Boundary and Inner Vertex.

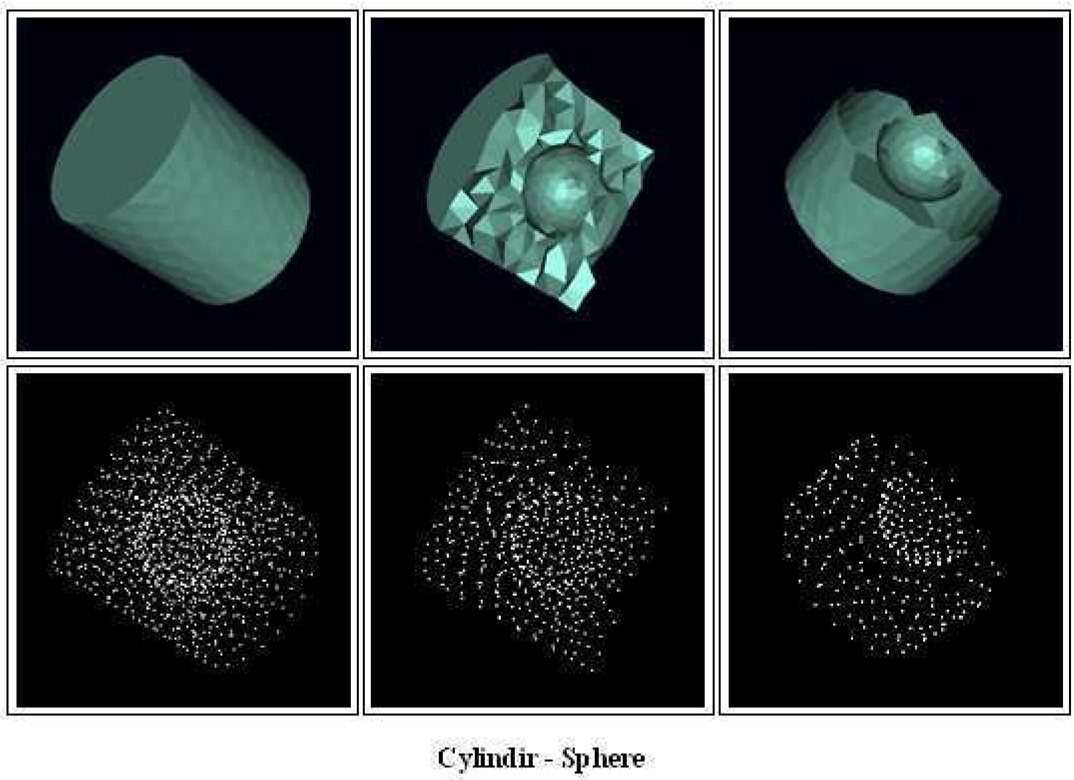


Figure 3.5: Implementation of Boundary Extraction Algorithm

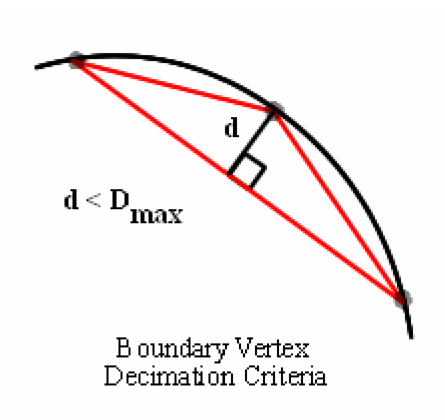
is showing the implementation on the Cylinder-Sphere data. It has two surfaces, one at the outside and the other at the inside of the data. After applying the boundary extraction algorithm on the data both of the boundary surfaces can clearly be noticed. For clarifying the visual results the horizontal and vertical cross sections are also taken of the data.

3.4 Decimation Algorithms

As we have mentioned that since boundary vertices are more important for a data to be visualized then an inner vertex. So we have implemented different algorithms for each type of the vertices.

3.4.1 Decimation of Boundary Vertices

The algorithm was designed to implement the decimation of the boundary vertices of the data. It begins with classifying each vertex in the mesh and is inserted into the queue regarding the priority, which is based upon the error to delete the vertex and retriangulate the hole. Those vertices which cannot be deleted are skipped while analyzing. The candidate vertices are queued according to the priority values are assigned to them. The process continues until all the vertices in the queue are processed. In the second phase all the remaining vertices are processed, and the mesh is split into separate pieces along sharp edges or at non-manifold attachments points and reinserted into the priority queue. The vertices are reprocessed if the desired result is not obtained in the first phase.



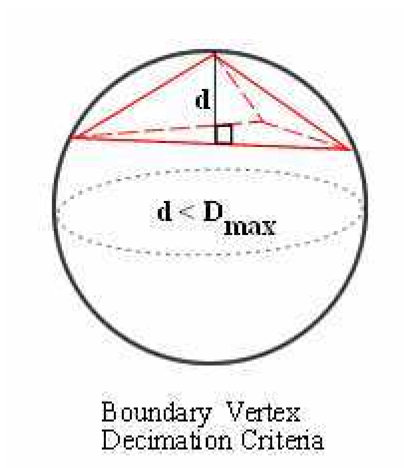


Figure 3.6: Representing the distance measurement technique for $D_{\max}$.

In the decimation of boundary vertices [8] the points are deleted according to decimation criteria from the vertex cloud. It is a simple error measurement technique which not only preserves the general shape but also helps in the reduction of the insignificant vertices. The error measurement is actually a distance measuring algorithm known as $D_{\max}$.

For each boundary vertex a candidate edge segment is drawn between an initial boundary vertex and the vertex under consideration, the chord vertex, the distance *d* from the vertex under consideration to the candidate edge segment is computed. Each vertex, whose distance *d* is less than $D_{\max}$, is a candidate for removal. The first vertex for which *d* is greater than $D_{\max}$ becomes the new initial vertex. This procedure is repeated until all the vertices have been processed.

We have implemented the $D_{\max}$ criteria in three different ways which are

- Constant $D_{\max}$
- Linearly Increasing $D_{\max}$
- Adaptive $D_{\max}$

3.4.1.1 Constant $D_{\max}$

By the term constant implementation we refer to the method that the value of $D_{\max}$ is kept constant in each decimation level. The value of $D_{\max}$ is calculated over the original data and then is implemented at each level of hierarchical decimation.

3.4.1.2 Linearly Increasing $D_{\max}$

By the term linearly increasing $D_{\max}$ value we imply to the fact that after the calculation of $D_{\max}$ on the original data we increase the value of $D_{\max}$ at each hierarchical decimation step. This is with the reference to the fact that since after the decimation some of the vertices are discarded by the program so ultimately the average distance between the boundaries tends to extend so implementing a linear $D_{\max}$ seems more logical than the Constant $D_{\max}$.

3.4.1.3 Adaptive $D_{\max}$

Since both the constant and the linear $D_{\max}$ are a parameters which employed in a general terms and the case can be really thought as in similar equivalence with Global (weighted) Alpha. So a best choice would be to calculate a separate value of $D_{\max}$ for each region and for each decimation step. This seems to be the most logical choice of constraining parameter for the decimation of the boundary vertices.

3.4.2 Decimation of Inner Vertices

The simplification of interior nodes of the mesh at a given level of the hierarchy was performed by using an iterative optimization algorithm [22] with the constraint that only non-neighboring and less important nodes shall be removed. Once a node is selected for removal, its neighbors are marked and not removed at that level of the hierarchy. The approach for selection of the nodes to be removed, together with this constraint, poses a drawback for the interior-node simplification. To this effect, we propose a dynamic programming-type optimization for the simplification of the interior nodes at each level of the hierarchy. The aim is to remove the maximum number of independent nodes from a mesh, while retaining a set of nodes that is optimal in the sense that it contains most important information going from one hierarchical level to the next coarser level, the goal is to retain the nodes that are expected to be important in mesh-based three dimensional data visualization, as well as highly connected nodes, removal of which would constrain subsequent simplification stages. The flow of the algorithm is first initialization to form layers then cost calculation for each nodes and placement of removed and retained nodes on a layer basing on which we calculate the placements of nodes on the next neighboring layer. Then the optimization algorithm is employed which has three different algorithms. According to the maximum of the ratio of the total cost of remained nodes over the total number of remained nodes, we choose the algorithm according to the performance.

3.4.2.1 Initialization

First, we form layers of nodes in the mesh, where each layer is defined by the nodes that are immediately adjacent to a node in a previously defined layer. Layer formation can be given as:

a) The first layer of nodes is defined by the nodes on the mesh boundary;

b) The next layer of nodes is defined by all nodes in the interior of the mesh that are immediately adjacent to a node in the previous layer;

c) Successive layers are formed by repeating step b).

Further, we assign a cost value to each node of the mesh, in order to determine an optimal removal strategy. The set of interior nodes to be removed while going from one hierarchy level to the next is determined using topology-based criteria. The degree determines how much this node is connected to its surrounding region. The average degree of a Delaunay mesh is approximately. The latter requirement is imposed by the desire to remove a maximum number of nodes from one hierarchical level to the next under the independence constraint, while preserving local connectivity. The set of interior nodes to be removed while going from one hierarchy level to the next can be determined using connectivity based criteria. Therefore, we define the cost of removing a node n by the following measure of its importance (IP), which we call the importance value function. A higher importance value will indicate the indispensability of the vertex for visualization. The general formula for calculating that constraint is defined as the multiplication of its degree with the ratio of the sum of its neighbor's volume to its volume.

$$IP(n) = \frac{MaxDegree - Degree(n)}{MaxDegree} \bullet \frac{TotalLayerNumber - NodeLayer(n)}{TotalLayerNumber} \quad (3.1)$$

Where,

MaxDegree	=	maximum number of neighbors of any node
Degree(*n*)	=	number of neighbors of traversed node
TotalLayerNumber	=	the number of total layers according to depth of the data
NodeLayer	=	the layer of the node being traversed

The degree of the vertex is the number of edges or vertices connected to it. The number of connectivity for a vertex indicates the importance of that node. It is same for the volume ratio. If the volume of a tetrahedron including this node is smaller than the volumes of the neighboring tetrahedrons it means that this tetrahedron belongs to a detailed region in the shape, so removing a node in this tetrahedron will cause us to less information to be lost. So we have the parameter MaxDegree which denotes the integral value of node connected with maximum number of vertices, we take the difference of the degree of the traversed node, and then normalize it with the maximum degree. This ensures the routine that if a node has large number of neighbors attached to it then its IP value will be increased and it will be ultimately a candidate node to be discarded. The other parameter that controls the IP function is the depth of the node. The proximity to the surface of a node is responsible for the tetrahedrons which are forming at the surface which in turn becomes responsible for the shape of the data. So closer the node to the surface lower is its IP function and deeper the location of the node higher will be its IP function. Hence this parameter transforms the nodes at the depth with higher probability of being removed from the data.

3.4.2.2 The Optimization Algorithm

The dynamic programming-type interior-node simplification algorithm can be explained as below.

1) For each separate layer, a set of nodes which are candidate to be removed is obtained as follows. Iteratively label the node with the smallest IP value in the layer as to be removed and label its neighbors in that layer as to be fixed. Nodes labeled as to be fixed may not be considered for removal later in this stage. Continue labeling nodes on this layer until a node with importance value equal to IP is found. Note there may be free nodes left that are labeled neither to be removed nor to be fixed.

2) Each node layer obtained in step 1) is now considered for employing the three different importance value assigning algorithms, which are:

- Forward Optimization Algorithm
- Backward Optimization Algorithm
- Greedy Optimization Algorithm

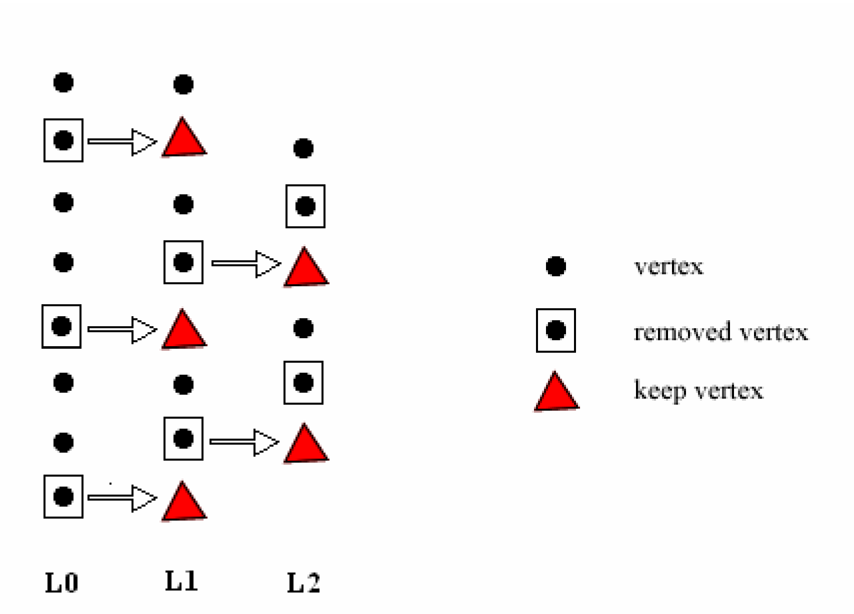


Figure 3.7: Illustration of forward optimization algorithm.

In the forward optimization algorithm, considering the decimation of layer 1, the neighbors of those which are decimated from boundary layer are fixed, the IP values are calculated according to the given formula in equation 3.2, the candidate vertex are then labeled. Similarly then moving on to layer 2, the neighbors of those vertices which are labeled as the candidates for removal are fixed and the same process is employed on the rest of the nodes, as clearly depicted in Figure 3.7. In this manner we proceed till the last or the inner most level and end up removing the best possible vertices according to this algorithm.

In the backward optimization algorithm, we start from the inner most layer, label the vertices as candidate for removal according to IP function calculated as above and proceed till the layer 1. Now in layer 1, we end up having two types of constraints, one those vertices which are the neighbors of the candidate nodes from layer 2 and secondly those vertices which are neighbors of the candidates for removal in the boundary layer, as shown in Figure 3.8. In this way end up assigning the IP function from inner to boundary layer that is why it is termed as backward algorithm.

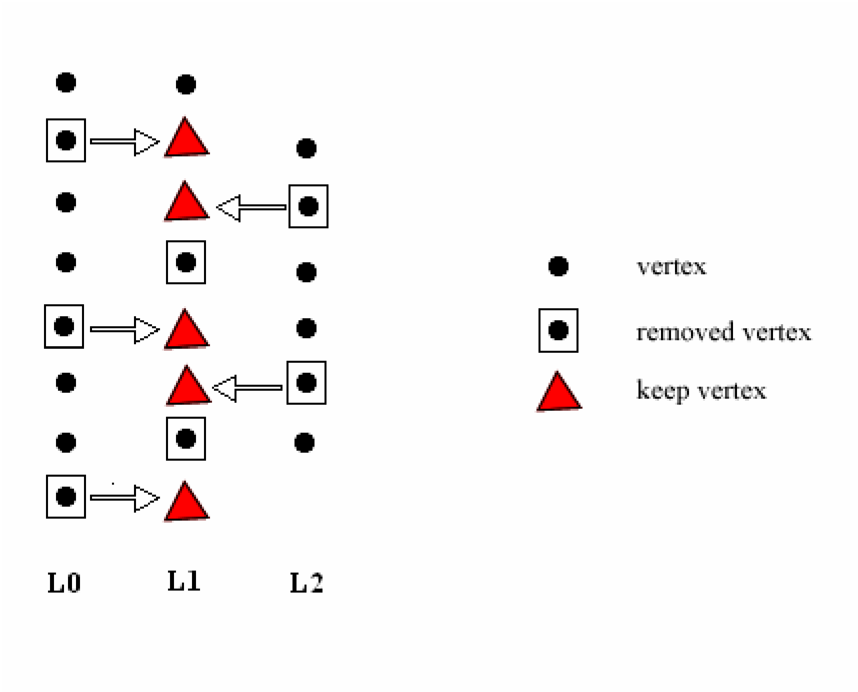


Figure 3.8: Illustration of backward optimization algorithm.

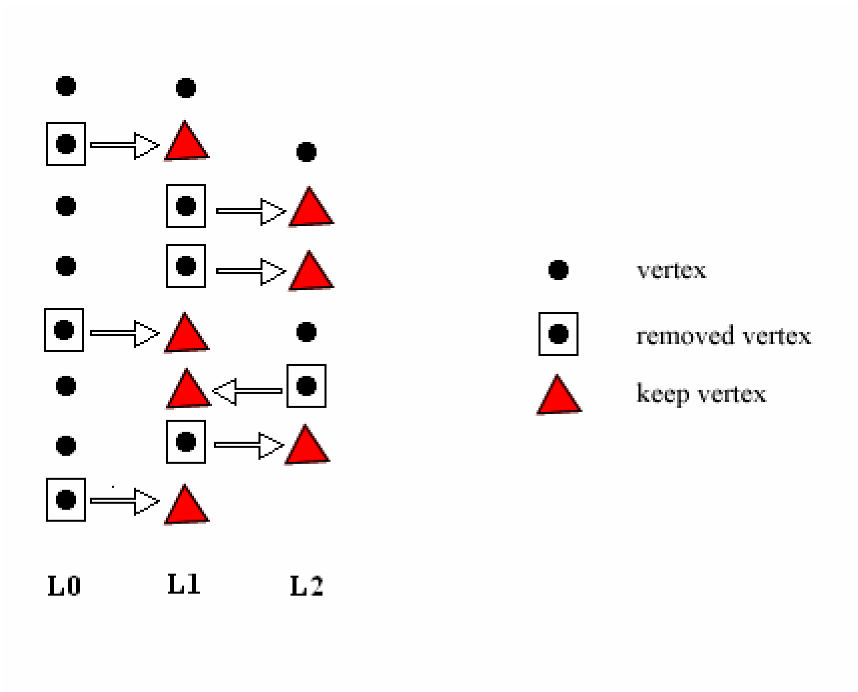


Figure 3.9: Illustration of greedy optimization algorithm.

In the greedy optimization algorithm, we only consider those vertices of layer 1, the neighbors of those which are decimated from boundary layer are fixed, and the IP values of rest of whole of the vertices are calculated according to the given formula in equation 3.2. The vertices are then decimated according to the threshold provided. The process is clearly explained in Figure 3.9.

This means we use the node labeling of the central layer as a constraint on the labeling of nodes in other layers. The neighbors of the nodes labeled as to be removed must be labeled to be fixed. This process continues until the last layer in forward and backward direction is processed. The importance values of all nodes that are labeled fixed in this stage are averaged.

3) The process in step 2) is performed using each layer as a central layer, each time computing an average IP value for that layer as described above. This average value for layer is denoted by IP.

4) The overall labeling for the layer with the maximum IP is retained; any free nodes left are labeled to be fixed. Finally, the nodes labeled to be removed in the optimal labeling are now actually removed from the mesh so that a new mesh for the next coarser level is formed.

3.5 Quality of Tetrahedron and Triangles Meshes

Since we reduce the amount of the vertices in each decimation level so it is inherently necessary to put a constant check and constraint over the quality of newly generated. Tetrahedron and triangle meshes. There are numerous tests [17] for accomplishing that target. The important ones for triangles are discussed below:

3.5.1 Angle Test for Triangles

Since an equilateral triangle, if formed while implementing the Delaunay triangulation is considered the healthiest for visualizing. So we have introduced an angle test which traverses the whole data, calculate its minimum angles and at the end we employ the following formula which is taking an average of whole data.

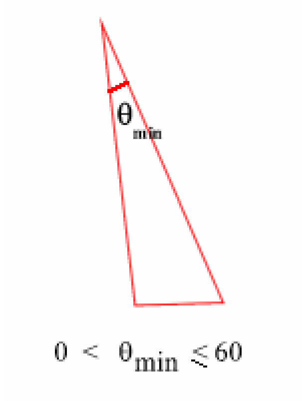


Figure 3.10: Angle Test for Triangles.

$$\mathrm{Quality\_1} = \mathrm{Sin}\left(\frac{\min(\mathrm{angle})}{2} * 3\right)$$

$$\mathrm{Avg\_quality\_1} = \sum_{\mathrm{n\_tetra}} \frac{\mathrm{Sin}\left(\frac{\min(\mathrm{angle})}{2} * 3\right)}{\mathrm{n\_tetra}}$$

3.5.2 Min/Max Edge Ratio Test For Triangles

This was another test which was introduced for analyzing the overall quality of the formed triangles which were constructed after the implementation of 3D Delaunay triangulation. The test is actually a ratio between the shortest and the longest edge of the triangle. Since the desired triangle while implementing the algorithm is equilateral when the ratio becomes 1. But in worst case scenarios it is very low. So the results are considered satisfactory if the average value remains unchanged after the decimation is implemented.

$$\mathrm{Quality\_2} = \frac{\mathrm{min\_edge\_length}}{\mathrm{max\_edge\_length}}$$

The average quality is given by

$$\mathrm{Quality\_2} = \sum_{\mathrm{n\_tetra}} \frac{\frac{\mathrm{min\_edge\_length}}{\mathrm{max\_edge\_length}}}{\mathrm{n\_tetra}}$$

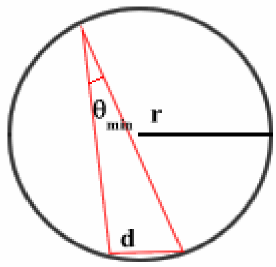


Figure 3.11: Min/Max Edge Ratio Test For Triangles

3.5.3 Radius Ratio Test

There are two kinds of different spheres which are formed regarding our decimation technique, the inner sphere of the tetrahedron which is actually a sphere connecting the tangents of the tetrahedron edges from inside and an outer sphere which is the sphere formed by connecting the points of the tetrahedron. The quality of the generated tetrahedron is considered optimal if the ratio of the radius of outer sphere is equal to three times of the radius of the inner sphere.

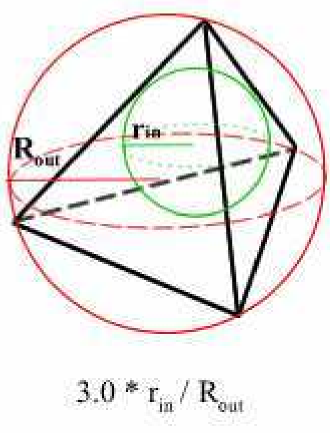


Figure 3.12: Manifesting the Radius Ratio Test

Quality_1 $= 3.0 * r_{in} / R_{out}$

The average quality of the total mesh of the tetrahedron formed is given by their sum as under:

$$\text{Avg\_quality\_1} = \sum_{n\_tetra} \frac{3.0 * r_{in} / R_{out}}{n\_tetra}$$

Where,

n_tetra = number of tetrahedron
r_{in} = radius of the inscribed sphere
R_{out} = radius of the circumscribed sphere

3.5.4 The Inner Radius to Maximum Edge Length Test.

Similarly the second test is again a ratio between the radius of the inner sphere to that of the length of the longest edge. According to this test the length of the longest edge of the tetrahedron should be equal to 2√6 times the radius of inner sphere.

$$\text{Quality\_2} = 2\sqrt{6} * r_{in} / L_{\max}$$

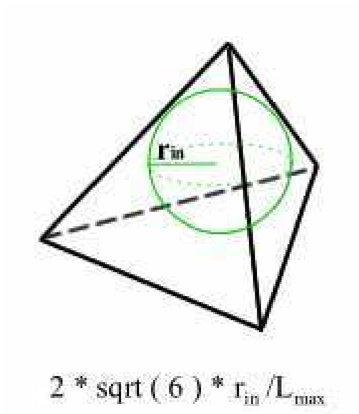


Figure 3.13: Manifesting Inner Radius to Maximum Edge Length Test

Whereas the average quality test can be given as the ratio of the sum of the both the quantities in the generated tetrahedron mesh.

$$\text{Avg\_quality\_2} = \sum_{n\_tetra} \frac{2\sqrt{6} * r_{in} / L_{\max}}{n\_tetra}$$

where

n_tetra = number of tetrahedron
r_{in} = radius of the inscribed sphere
$L_{\max}$ = length of longest side of the tetrahedron

3.5.5 Volume to Edge lengths Ratio

The third test is also a ratio between the volume of the tetrahedron and the sum of squared sum of all the length of edges of the tetrahedron. The generated tetrahedron may be considered as an optimal if this ratio is closer to 1.

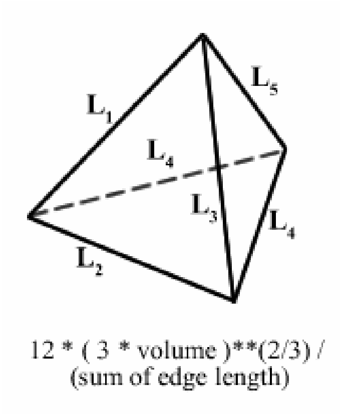


Figure 3.14: Showing the Volume to Edge lengths Ratio

$$\text{Quality\_3} = \frac{12*(3*Volume)^{\frac{2}{3}}}{\sum_{6} L_n^{\ 2}}$$

For considering data as a whole, it is the ratio of sum of whole volume of all the tetrahedrons to their respective length of edges.

$$\text{Avg\_quality\_3} = \sum_{n\_tetra} \frac{\dfrac{12*(3*Volume)^{\frac{2}{3}}}{\sum_{6} L_n^{\,2}}}{n\_tetra}$$

n_tetra = number of tetrahedron

$Volume$ = volume of the tetrahedron

L_n = n^{th} edge length

3.5.6 Minimum Solid Angle Test

In this method for testing the shape of tetrahedron, the minimum solid angle of the tetrahedron is calculated and then Sin of half its angle gives the value.

$$\text{Quality\_4} = \sin\left(\frac{\min(SolidAngle)}{2}\right)$$

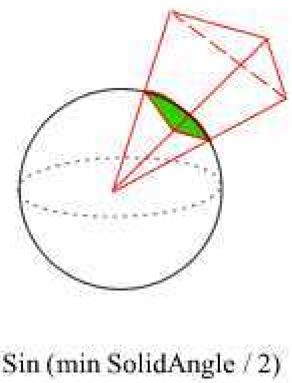


Figure 3.15: Manifesting Minimum Solid Angle Test

For calculating the tetrahedral mesh quality the average of the Sin of all minimum solid angles is calculated for all the tetrahedrons

$$\text{Avg\_quality\_4} = \sum_{n\_tetra} \frac{\sin\left(\dfrac{\min(SolidAngle)}{2}\right)}{n\_tetra}$$

n_tetra = number of tetrahedron

3.6 Data to be sent To Decoder.

When whole of the data is tested and processed then it should be coded and sent to the decoder. Since we have processed different types of data, different type of information is needed for decoding each data type. We have two cases for data types.

- The Surface Data
- The Volumetric Data

3.6.1 The Surface Data

If the surface data is to be sent then we require following information to be sent to the decoder:

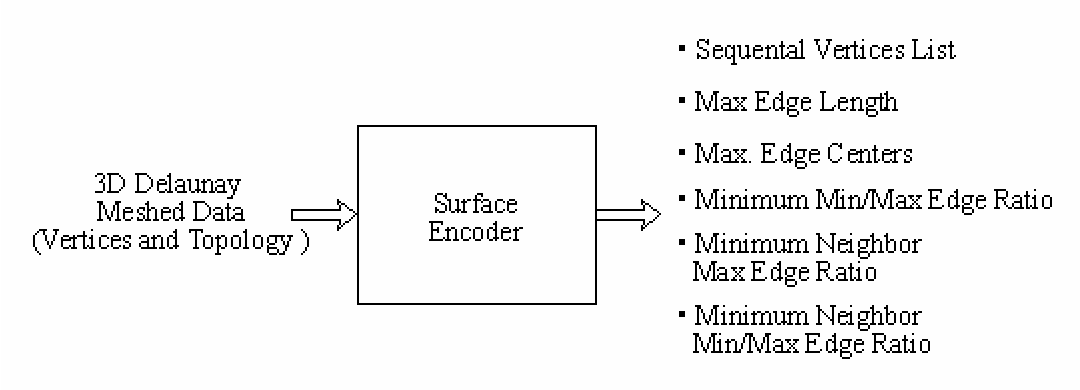


Figure 3.16: Block Diagram of Surface Encoder

- List of vertices in the data.
- The Maximum Edge length.
- The maximum edge formed by each vertex is calculated and then KNN algorithm is applied to which calculates the different classes for the increasing alpha values of the data and those cluster centres are needed to be sent. The vertices are also sent according to the alpha values they are attached to.
- Minimum min/max Edge Ratio.
- Minimum Neighbor Maximum Edge Ratio of the data.

- Minimum Neighbor Minimum/Maximum Edge Ratio.

If the data is open type surface data then apart from the information above a list of all those vertices which are found at the boundary of the surface is also sent.

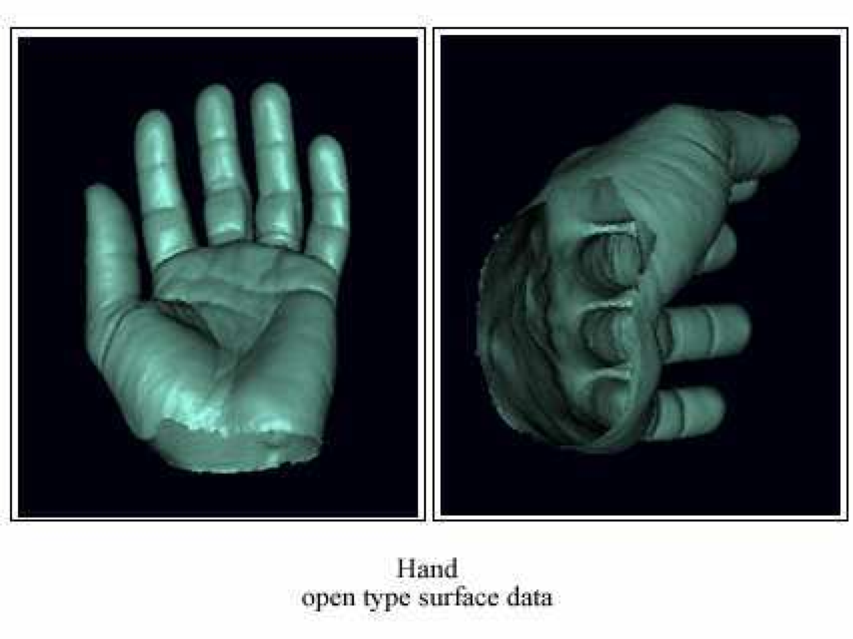


Figure 3.17: An Example of Open type surface data.

3.6.2 The Volumetric Data

If the volumetric data is to be sent then we require following information to be sent to the decoder.

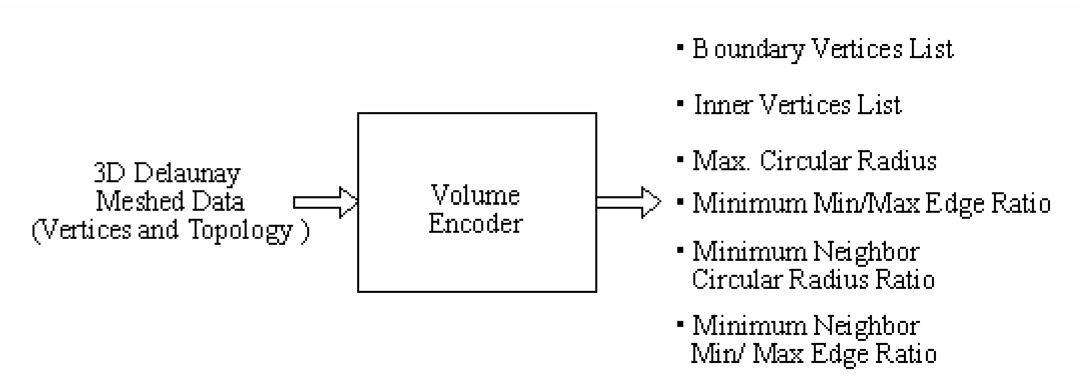


Figure 3.18: Block Diagram for Volume Encoder.

- List of Boundary Vertices.
- List of Inner Vertices.
- Maximum Circular Radius.
- Minimum Neighbor Circular Radius Ratio.
- Minimum Neighbor Minimum / Maximum Edge Ratio

The reason we are sending the information apart from vertices is to have the same level of visualization impact at the decoder or receiver level.

3.7 The Decoder.

To assess the performance of our algorithm we also have designed a decoder which employs whole of the process in the reverse manner. The decoder was designed for surface and volumetric data separately.

3.7.1 Surface Data Decoder.

- Initially 3D Delaunay triangulation is applied over the vertex data by using maximum edge length. That results in the formation of lines, triangles and tetrahedrons.
- All lines deleted.
- The triangles formed by every vertex are traversed and the boundary points are assessed with maximum bound of Maximum Edge Length, if lengthier than the upper bound is found then it will be deleted.
- If tetrahedrons are formed then the triangle of tetrahedron possessing maximum perimeter is deleted so the tetrahedron breaks into or three triangles. So that there is volumetric information residing within the 3D surface visualization.

- Different vertices are connected according to the alpha values that are assigned to them.
- Minimum Neighbor Maximum Edge ratio is taken as the lower bound and any triangle possessing less than this bound is deleted.
- Minimum Neighbor minimum/maximum edge ratio is taken as the lower bound again and any triangle having less than this bound is deleted.

The whole process is done with the Gap control method being activated so that we may not have any inconclusive end at part of visualization.

3.7.2 Volumetric Data Decoder.

- Initially 3D Delaunay triangulation is applied over the vertex data by using maximum circular radius. That results in the formation of lines, triangles and tetrahedrons.
- All lines and triangles are deleted.
- Then we register the temporary boundary vertices which are found while visualization. Those vertices of the tetrahedrons which are located at the boundary list and as well as in the list being sent by the encoder are never deleted. But if they are only found in the received list then those tetrahedrons are to be deleted. This protects all the vertices of the tetrahedron in the convex formation but help deleting those which are found at the concave side of the data.
- Minimum Neighbor circular radius ratio is taken as the lower bound and any tetrahedron possessing less than this bound is deleted.
- Minimum Neighbor minimum/maximum circular radius ratio is taken as the lower bound again and any tetrahedron having less than this bound is deleted.

4. RESULTS

The proposed decimation scheme was implemented on the different types of data. Since two different encoders were created, the surface and the volumetric so we had two different types of data those were surface and the volumetric.

4.1 Results for the Surface Data

To assess the performance of the surface data encoder we have tested three different types of surface data. Since the surface data only contains the boundary vertices so the D_{max} criteria was used to decimate the vertices. It has operated in the manner that, the percentage of the vertices to be deleted, were fixed initially and then the algorithm was implemented. The data is visualized at each decimation level from finest to coarsest. In addition with the visualization, the different tetrahedral mesh quality tests were also performed to continuously monitor the mesh quality at each decimation level. The QT 1 is the minimum/maximum edge ratio and the QT2 is the angle test.

4.1.1 Results for Torus Data.

The visualization results for the surface data of torus is given in figure 4.1. We can observe that the quality of the visualization is although getting coarser as we proceed from level0 to level5.

Table 4.1: Showing the number of vertices, global Alpha, and quality tests being performed at each level of hierarchy from finest to coarsest, Min/Max Edge Ratio, Neighbor Max Edge Ratio and Neighbor Min/Max Edge Ratio.

	NV	GA	QT1	QT2	Min/Max Edge Ratio	N.Max Edge Ratio	N. Min/Max Edge Ratio
Level 0	36450	0.0657	0.3628	0.8174	0.2832	0.9873	0.9540
Level 1	7290	0.0140	0.1625	0.1729	0.0085	0.1656	0.0192
Level 2	1822	0.0149	0.1613	0.1719	0.0038	0.2049	0.0128
Level 3	233	0.0155	0.1601	0.1715	0.0034	0.2956	0.0259

Table 4.2: The different values of local (weighted Alpha) which is denoted by maximum edge centre (MEC), at each level the value correspond to the MEC's of that level and MC denotes the member count associated with that MEC of the data given in table 4.1 at each level of decimation from level 0 to level 3.

	Level 0		Level 1		Level 2		Level 3	
MEC	Value	MC	Value	MC	Value	MC	Value	MC
1	0,047511	1918	0,098156	13	0,163432	15	0,364443	5
2	0,047842	4802	0,106771	238	0,184997	34	0,412204	4
3	0,048541	2120	0,110978	823	0,224053	89	0,555743	8
4	0,049645	2352	0,117257	332	0,240631	120	0,607612	16
5	0,051077	3068	0,122315	261	0,255658	180	0,656085	13
6	0,052735	2976	0,128220	206	0,269019	160	0,680123	6
7	0,054598	2464	0,133801	866	0,284826	147	0,720125	14
8	0,056549	1900	0,141004	450	0,301040	159	0,787808	12
9	0,058338	1620	0,146651	685	0,310429	112	0,839954	15
10	0,060368	1620	0,156322	600	0,319156	85	0,872387	21
11	0,062192	1668	0,170223	562	0,329356	87	0,904255	4
12	0,063640	1336	0,188140	560	0,342315	91	0,943009	7
13	0,064585	1436	0,211083	577	0,359056	110	0,996432	7
14	0,065369	2596	0,238747	461	0,378244	95	1,038096	12
15	0,065568	2880	0,271267	397	0,408759	124	1,105707	11
16	0,065783	1694	0,314320	178	0,461988	120	1,156051	15
17	----	----	0,371442	59	0,530700	51	1,191481	7
18	----	----	0,432584	10	0,618041	28	1,299084	14
19	----	----	0,452355	10	0,689759	4	1,406669	10
20	----	----	0,514226	2	0,764422	7	1,653243	7
21	----	----	----	----	0,841112	4	1,803431	6
22	----	----	----	----	----	----	2,009205	6

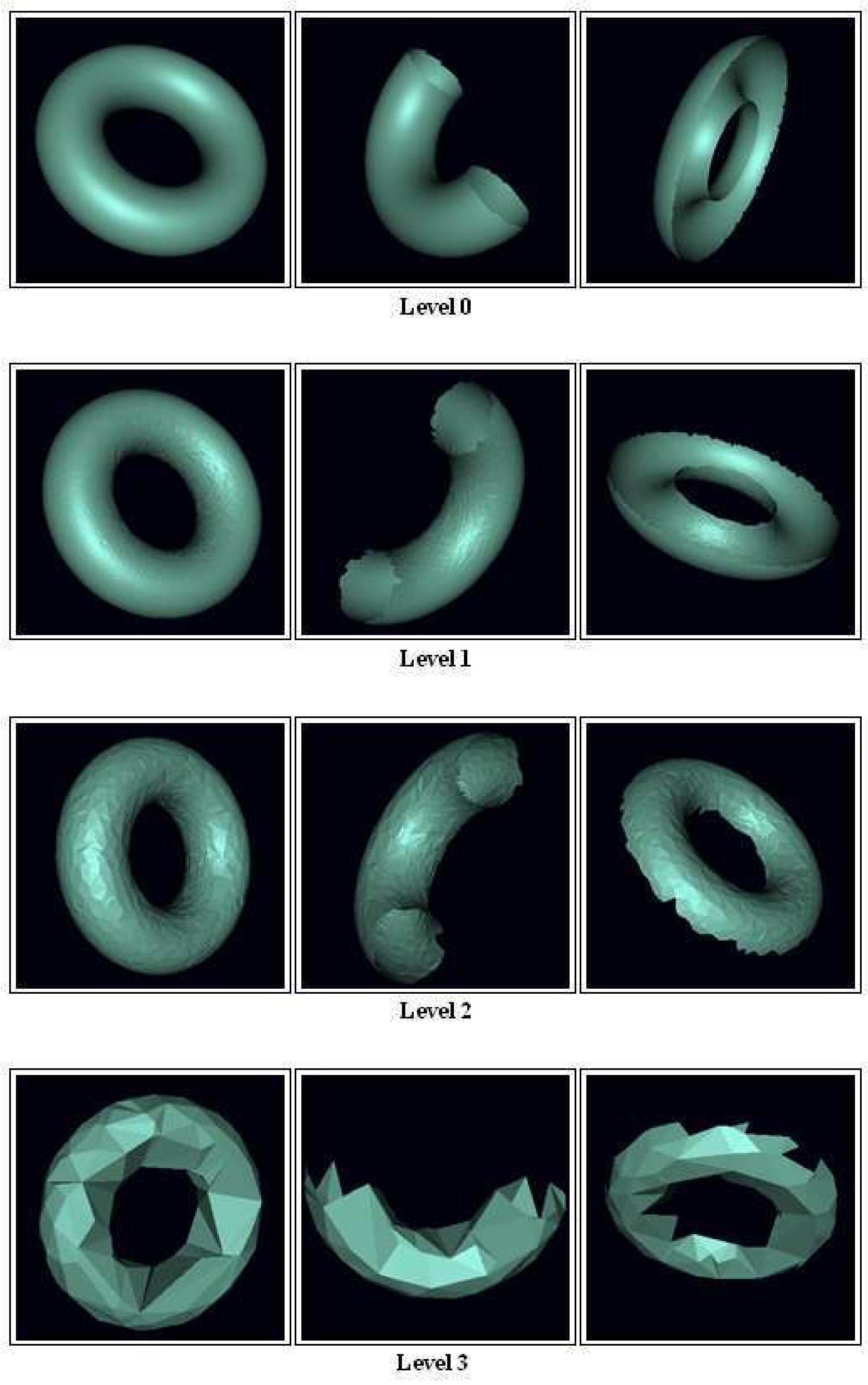


Figure 4.1: The different hierarchical levels of Torus Data.

4.1.2 Results for Cat Data.

The visualization results for the surface data of cat is given in figure 4.2. We can observe that the quality of the visualization is although getting coarser as we proceed from level0 to level3. The QT1 represents the minimum / maximum Edge Ratio and the QT2 the angle test being performed over the data.

Table 4.3: Showing the number of vertices, global Alpha, and quality tests being performed at each level of hierarchy from finest to coarsest, Min/Max Edge Ratio, Neighbor Max Edge Ratio and Neighbor Min/Max Edge Ratio.

	NV	GA	QT1	QT2	Min/Max Edge Ratio	N.Max Edge Ratio	N. Min/Max Edge Ratio
Level 0	10000	0.3494	0.3097	0.6773	0.000027	0.0190	0.000042
Level 1	3001	0.3494	0.2738	0.6134	0.000065	0.0734	0.000235
Level 2	1501	0.3426	0.2554	0.5750	0.000065	0.0692	0.000262
Level 3	594	0.3647	0.2526	0.5735	0.001661	0.1000	0.004013

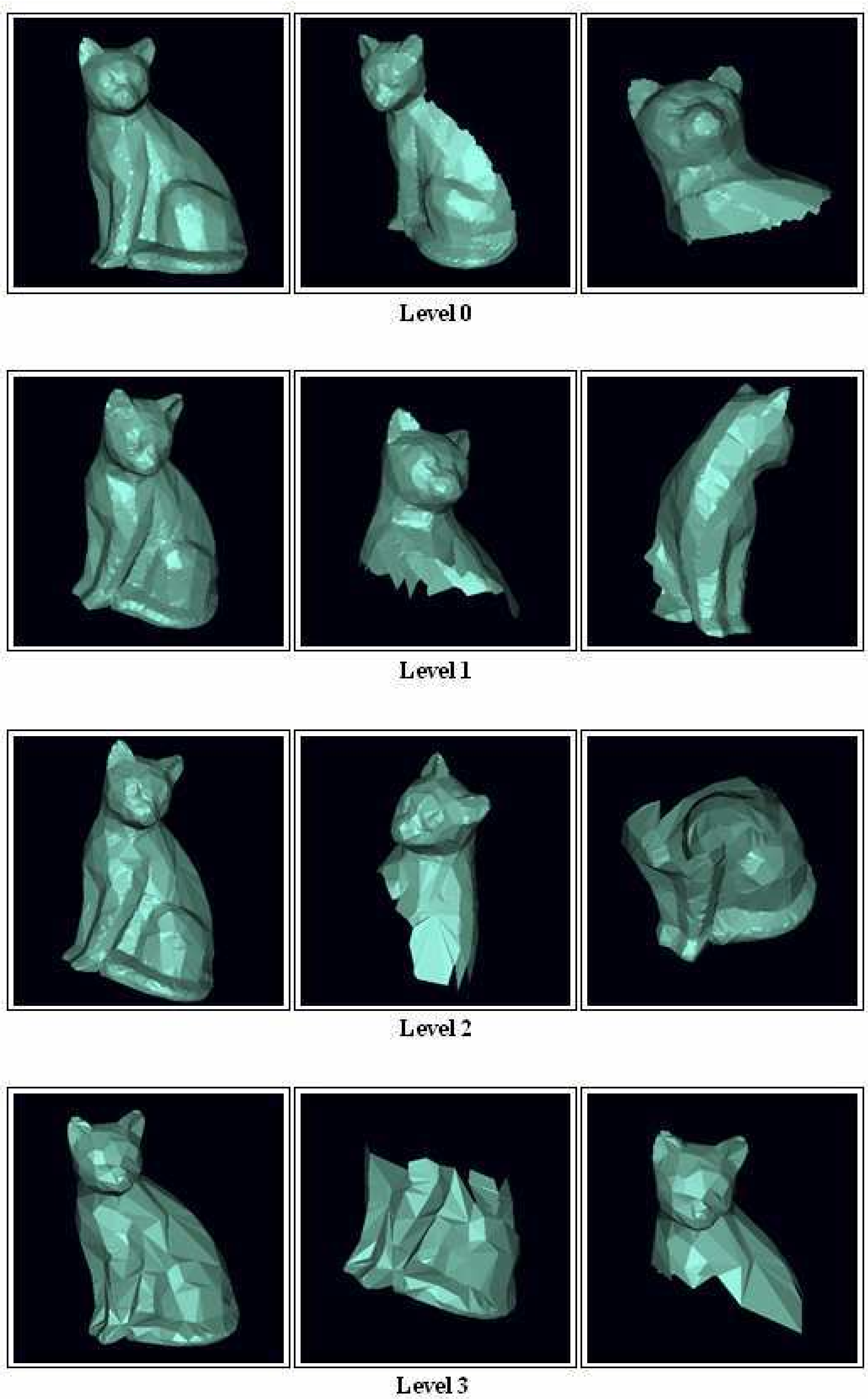


Figure 4.2: The different hierarchical levels of Cat Data.

Table 4.4: The different values of local (weighted Alpha) which is denoted by maximum edge centre (MEC), at each level the value correspond to the MEC's of that level and MC denotes the member count associated with that MEC of the data given in table 4.3 at each level of decimation from level 0 to level 3.

	Level 0		Level 1		Level 2		Level 3	
MEC	Value	MC	Value	MC	Value	MC	Value	MC
1	0.006010	43	0.010415	27	0.010991	19	0.016709	15
2	0.007545	148	0.012729	90	0.013585	47	0.025651	30
3	0.009425	404	0.016366	195	0.018121	93	0.034274	39
4	0.010746	504	0.019243	238	0.021328	86	0.042359	36
5	0.011606	500	0.022309	270	0.024661	114	0.050257	37
6	0.012373	400	0.025777	311	0.028381	96	0.058203	48
7	0.013192	581	0.029316	258	0.031852	84	0.067236	65
8	0.014185	742	0.032178	216	0.035157	66	0.076930	47
9	0.015133	756	0.035548	219	0.038610	78	0.084892	40
10	0.016081	767	0.038756	190	0.042280	63	0.091796	30
11	0.016984	740	0.042103	174	0.046618	75	0.100889	30
12	0.017875	666	0.047389	204	0.051436	91	0.114730	45
13	0.018829	666	0.055670	224	0.056305	86	0.131184	33
14	0.019852	630	0.068324	193	0.061108	70	0.151519	47
15	0.020924	606	0.087563	116	0.065503	69	0.184379	27
16	0.022157	536	0.117719	47	0.071436	73	0.220161	5
17	0.024019	531	0.163249	14	0.081037	85	0.245914	4
18	0.027739	538	0.209454	4	0.094794	79	0.299384	3
19	0.039588	191	0.259682	1	0.117719	65	0.314434	1
20	0.069436	13	0.326906	4	0.146024	34	0.367250	3
21	0.112405	9	0.351144	6	0.182911	16	-----	-----
22	0.147207	11	-----	-----	0.216677	4	-----	-----
23	0.174299	3	-----	-----	0.259682	1	-----	-----
24	0.208567	3	-----	-----	0.299549	2	-----	-----
25	0.259682	1	-----	-----	0.345010	5	-----	-----
26	0.307888	2	-----	-----	-----	-----	-----	-----
27	0.352843	9	-----	-----	-----	-----	-----	-----

4.1.3 Results for Bimba Data

The visualization results for the surface data of cat is given in figure 4.5. We can observe that the quality of the visualization is although getting coarser as we proceed from level0 to level3. The QT1 represents the minimum / maximum Edge Ratio and the QT2 the angle test being performed over the data.

Table 4.5: Showing the number of vertices, global Alpha, and quality tests being performed at each level of hierarchy from finest to coarsest, Min/Max Edge Ratio, Neighbor Max Edge Ratio and Neighbor Min/Max Edge Ratio.

	NV	GA	QT1	QT2	Min/Max Edge Ratio	N.Max Edge Ratio	N. Min/Max Edge Ratio
Level 0	74764	0.0101	0.7876	0.9834	0.016712	0.485188	0.023054
Level 1	22430	0.2456	0.4475	0.7573	0.001265	0.037042	0.001434
Level 2	11216	0.2918	0.3678	0.6872	0.000366	0.023206	0.000539
Level 3	4146	0.3647	0.2718	0.5984	0.000366	0.025547	0.000653

Table 4.6: The different values of local (weighted Alpha) which is denoted by maximum edge centre (MEC), at each level the value correspond to the MEC's of that level and MC denotes the member count associated with that MEC of the data given in table 4.5 at each level of decimation from level 0 to level 3.

	Level 0		Level 1		Level 2		Level 3	
MEC	Value	MC	Value	MC	Value	MC	Value	MC
1	0.004983	90	0.004674	5	0.003820	1	0.005364	22
2	0.004986	1018	0.005054	227	0.005226	117	0.006616	59
3	0.005125	2831	0.005480	1562	0.006522	694	0.011417	154
4	0.005287	3239	0.006083	1773	0.008979	405	0.014819	391
5	0.005415	3240	0.007094	694	0.010191	1235	0.019626	635
6	0.005517	3032	0.008433	703	0.012027	972	0.024901	723
7	0.005589	2663	0.009159	1515	0.014259	1597	0.030736	540
8	0.005651	2573	0.010168	2241	0.016393	1016	0.037278	394
9	0.005710	2590	0.011423	1437	0.018601	887	0.046305	296
10	0.005786	3819	0.012792	1121	0.021189	649	0.060204	303
11	0.005885	4737	0.013826	1232	0.024463	609	0.081452	225
12	0.005987	4647	0.014905	1072	0.028646	555	0.110047	206
13	0.006091	4313	0.016144	964	0.034357	648	0.148429	115
14	0.006197	3977	0.017313	882	0.042510	652	0.193008	76
15	0.006309	3683	0.018569	885	0.054269	565	0.219148	13
16	0.006427	3451	0.019950	974	0.070064	309	0.242711	13
17	0.006548	3082	0.021980	920	0.090145	176	0.291832	15
18	0.006674	2797	0.024821	1128	0.118588	69	0.325519	10
19	0.006813	2927	0.028748	1037	0.153479	25	0.350894	5
20	0.006982	3219	0.034931	840	0.195933	22	0.372977	6
21	0.007197	3468	0.043775	631	0.227370	3	-----	-----
22	0.007472	3573	0.057356	348	0.269340	3	-----	-----
23	0.007816	2881	0.076580	119	0.295324	7	-----	-----
24	0.008249	1797	0.104207	67	-----	-----	-----	-----
25	0.008822	884	0.136103	26	-----	-----	-----	-----
26	0.009398	193	0.171495	19	-----	-----	-----	-----
27	0.010191	40	0.195928	2	-----	-----	-----	-----
28	-----	-----	0.231336	4	-----	-----	-----	-----
29	-----	-----	0.245638	2	-----	-----	-----	-----

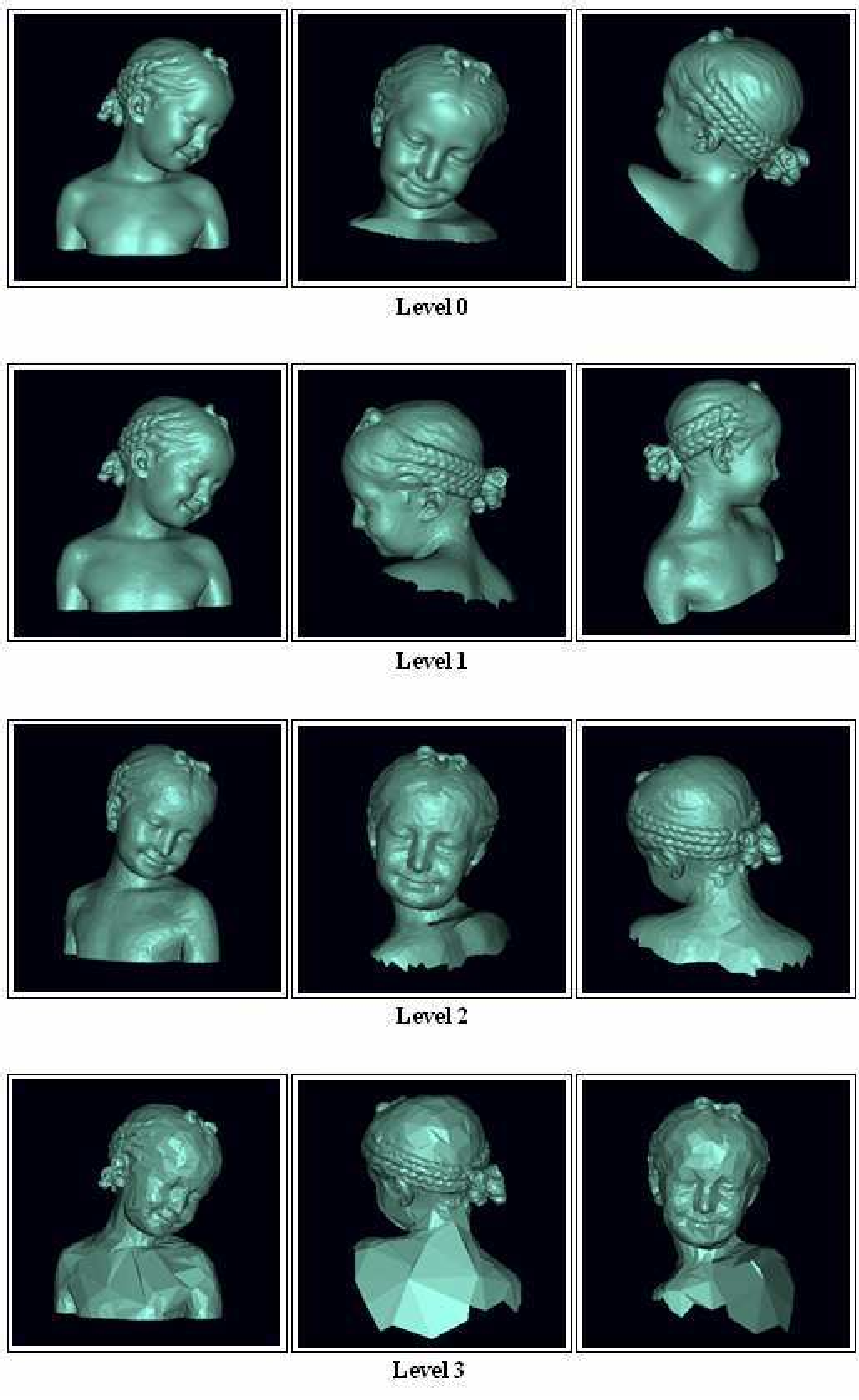


Figure 4.3: The different hierarchical levels of Bimba Data.

4.2 Volumetric Data.

To assess the performance of the volumetric data encoder we have tested three different types of volumetric data. Since the volumetric data contains the boundary vertices as well as interior vertices so both of the formulated algorithms were implemented. To implement the decimation of the boundary vertices, it has operated in the manner that, the percentage of the vertices to be deleted, were fixed initially and then the algorithm was implemented. The decimation of inner vertices is under taken in the manner which has been explained in the section 3.4.2. The volumetric nature of data specifies that we have different layers of data if considered from point of depth. Therefore, the data is divided into different layers to employ optimization algorithm. The data is visualized at each decimation level from finest to coarsest. In addition with the visualization, the different data that is required to be sent for proper visualization at the receiving end. The Global Alpha is also registered.

4.2.1 Sphere Data

The visualization results for the volumetric data of sphere are given in figure 4.4. We can observe that the quality of the visualization is although getting coarser as we proceed from level0 to level3. The first column shows the complete hierarchy of the data, the central column shows the hierarchy of the cross section of the volume data and the last column depicts the hierarchical decimation of the surface of the volume data.

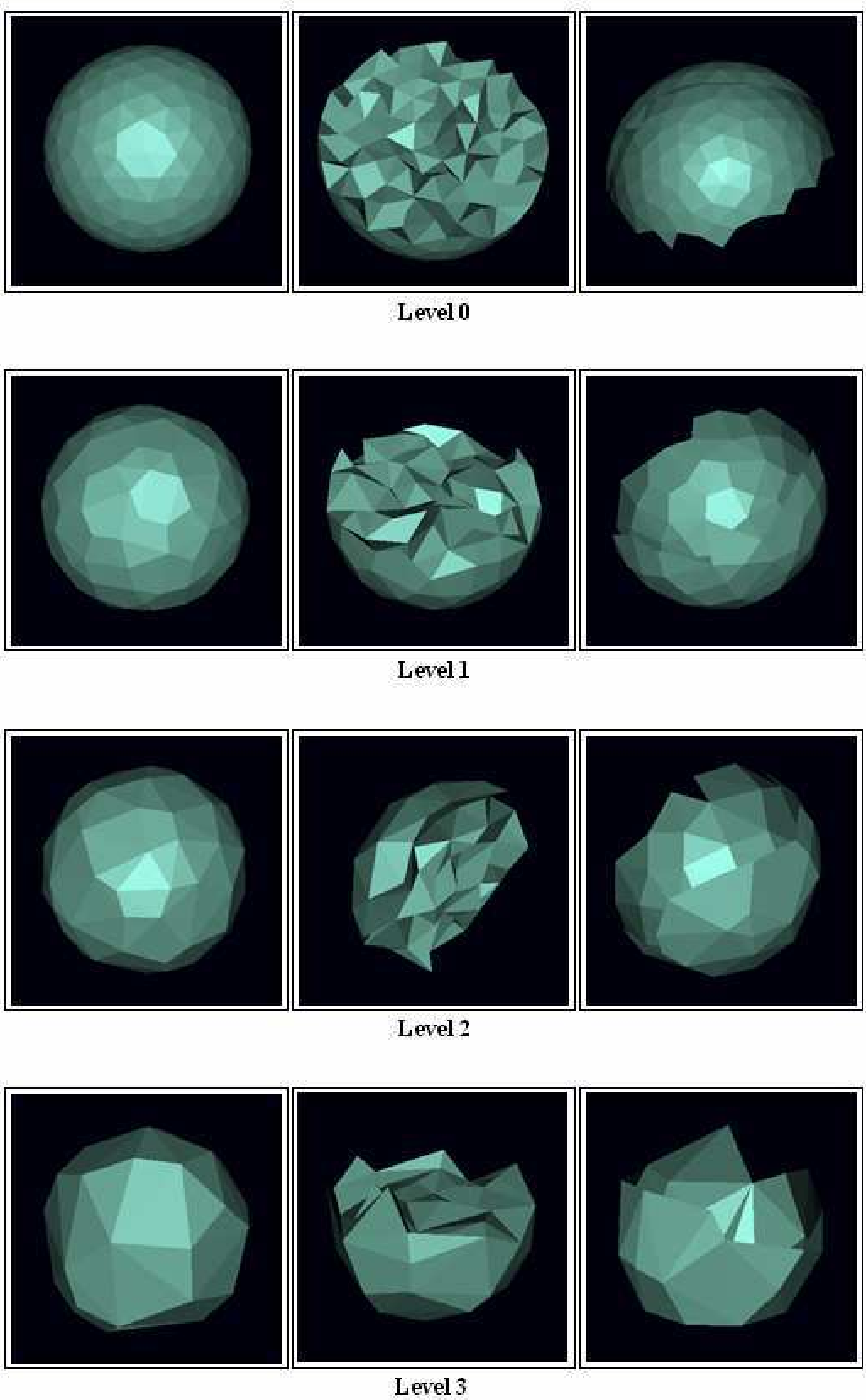


Figure 4.4: The different hierarchical levels of Sphere Data

Table 4.7: Showing the number of vertices VN, CN denotes the number of cells or tetrahedrons, L0 denotes boundary layer vertices, L1 the next to the boundary vertices as layer 1 and similarly L2, L3 and L4 as we proceed inside the sphere data at each successive level of hierarchy from finest to coarsest.

		VN	CN	L0	L1	L2	L3	L4
Level 0	Original	587	2775	247	171	110	53	6
Level 1	Forward	482	2374	173	165	101	42	1
	Backward	470	2325	173	164	98	35	-----
	Greedy	463	2316	173	290	-----	-----	-----
Level 2	Forward	396	2080	116	161	90	29	-----
	Backward	381	2010	116	158	88	19	-----
	Greedy	368	1947	116	252	-----	-----	-----
Level 3	Forward	346	1849	102	147	86	11	-----
	Backward	326	1736	103	149	66	8	-----
	Greedy	319	1688	102	217	-----	-----	-----

Table 4.8: The tables shows the quality test (QT1) 1 refers to Radius Ratio Test, test 2 (QT2) refers to The Inner Radius to Maximum Edge Length Test, test 3 (QT3) points to Volume to Edge lengths Ratio and test 4 (QT4) indicates Minimum Solid Angle Test results when three different methods Forward backward and greedy algorithm of optimization are employed on the successive hierarchy levels of sphere data as we proceed from finest to coarsest.

		QT1	QT2	QT3	QT4
Level 0	Original	0,858620	0,755284	0,879613	0,675845
Level 1	Forward	0,785068	0,684683	0,827727	0,580754
	Backward	0,760470	0,662699	0,808384	0,553689
	Greedy	0,743532	0,648058	0,794188	0,535661
Level 2	Forward	0,689000	0,602986	0,754335	0,478718
	Backward	0,669247	0,586601	0,736922	0,459515
	Greedy	0,659275	0,574947	0,726901	0,443711
Level 3	Forward	0,639740	0,560500	0,711294	0,422772
	Backward	0,616392	0,541770	0,691250	0,400480
	Greedy	0,623277	0,544040	0,693256	0,398846

The over all results indicate that the algorithm devised for volumetric decimation produce satisfactory and desired visualization.

4.2.2 Skull Data.

The visualization results for the volumetric data of sphere are given in figure 4.4. We can observe that the quality of the visualization is although getting coarser as we proceed from level0 to level3. The first column shows the complete hierarchy of the data, the central column shows the hierarchy of the cross section of the volume data and the last column depicts the hierarchical decimation of the surface of the volume data. The table 4.9 and 4.10 show the numerical results of the data.

Table 4.9: Showing the number of vertices VN, CN denotes the number of cells or tetrahedrons, L0 denotes boundary layer vertices, L1 the next to the boundary vertices as layer 1 and similarly L2 and L3 as we proceed inside the skull data at each successive level of hierarchy from finest to coarsest.

		VN	CN	L0	L1	L2	L3
Level 0	Original	37813	157600	23972	12557	1206	78
Level 1	Forward	29343	130051	16779	11659	887	18
	Backward	29089	128966	16779	11573	725	12
	Greedy	28977	127450	17258	11719		
Level 2	Forward	21947	105988	10064	11205	675	3
	Backward	21607	104314	10064	11057	486	
	Greedy	21201	100295	11213	9988		
Level 3	Forward	16393	86409	5026	10560	805	2
	Backward	16023	84357	5026	10405	592	
	Greedy	15255	77239	6720	8535		

Table 4.10: The tables shows the quality test (QT1) 1 refers to Radius Ratio Test, test 2 (QT2) refers to The Inner Radius to Maximum Edge Length Test, test 3 (QT3) points to Volume to Edge lengths Ratio and test 4 (QT4) indicates Minimum Solid Angle Test results when three different methods Forward backward and greedy algorithm of optimization are employed on the successive hierarchy levels of skull data as we proceed from finest to coarsest.

		QT1	QT2	QT3	QT4
Level 0	Original	0,808271	0,703310	0,838863	0,558288
Level 1	Forward	0,743310	0,646017	0,790561	0,499523
	Backward	0,734835	0,638544	0,783668	0,491047
	Greedy	0,718785	0,624066	0,771140	0,474772
Level 2	Forward	0,682997	0,598005	0,745149	0,450617
	Backward	0,673285	0,589578	0,736559	0,440920
	Greedy	0,645212	0,561909	0,710395	0,407650
Level 3	Forward	0,607511	0,542297	0,686888	0,394967
	Backward	0,596290	0,533254	0,677161	0,384799
	Greedy	0,582678	0,512419	0,657353	0,355133

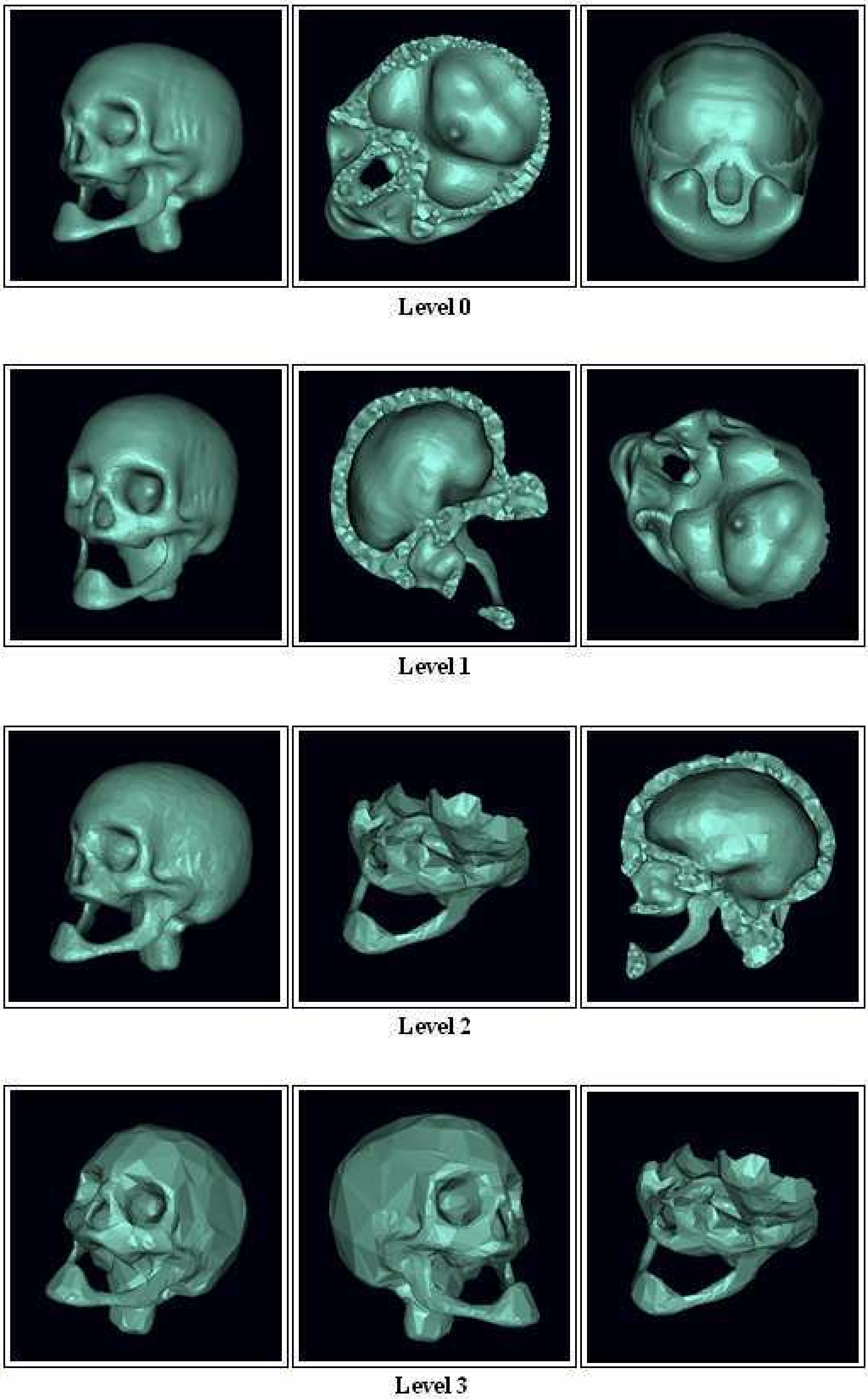


Figure 4.5: The different hierarchical levels of Skull Data

4.2.3 Hand Data.

The visualization results for the volumetric data of hand are given in figure 4.6. We can observe that the quality of the visualization is although getting coarser as we proceed from level0 to level3. The first column shows the complete hierarchy of the data, the central column shows the hierarchy of the cross section of the volume data and the last column depicts the hierarchical decimation of the surface of the volume data. The table 4.11 and 4.12 show the numerical results of the data.

Table 4.11: Showing the number of vertices VN, CN denotes the number of cells or tetrahedrons, L0 denotes boundary layer vertices, L1 the next to the boundary vertices as layer 1 and similarly L2, L3, L4 and L5 as we proceed inside the hand data at each successive level of hierarchy from finest to coarsest.

		VN	CN	L0	L1	L2	L3	L4	L5
Level 0	Original	28796	125127	15192	7395	4058	1709	432	10
Level 1	Forward	19980	89541	10633	5450	2828	985	83	-----
	Backward	19696	88432	10633	5450	2727	828	57	-----
	Greedy	19901	87714	10633	9267	-----	-----	-----	-----
Level 2	Forward	12941	57472	7392	3340	1712	485	12	-----
	Backward	12554	55415	7396	3285	1536	333	4	-----
	Greedy	13337	58407	7288	6049	-----	-----	-----	-----
Level 3	Forward	8813	39043	5135	2206	1225	247	-----	-----
	Backward	8429	36902	5137	2132	1032	12	-----	-----
	Greedy	8763	37802	4900	3861	-----	-----	-----	-----

Table 4.12: The tables shows the quality test (QT1) 1 refers to Radius Ratio Test, test 2 (QT2) refers to The Inner Radius to Maximum Edge Length Test, test 3 (QT3) points to Volume to Edge lengths Ratio and test 4 (QT4) indicates Minimum Solid Angle Test results when three different methods Forward backward and greedy algorithm of optimization are employed on the successive hierarchy levels of hand data as we proceed from finest to coarsest.

		QT1	QT2	QT3	QT4
Level 0	Original	0,836796	0,730324	0,863616	0,588864
Level 1	Forward	0,722918	0,629561	0,771002	0,482526
	Backward	0,708009	0,616411	0,758651	0,467963
	Greedy	0,697245	0,600177	0,724280	0,453327
Level 2	Forward	0,636595	0,553857	0,692750	0,395857
	Backward	0,620672	0,539570	0,677880	0,378585
	Greedy	0,619378	0,528213	0,654877	0,375240
Level 3	Forward	0,583259	0,506749	0,640646	0,341079
	Backward	0,564261	0,490312	0,622057	0,323103
	Greedy	0,562590	0,476880	0,601169	0,318494

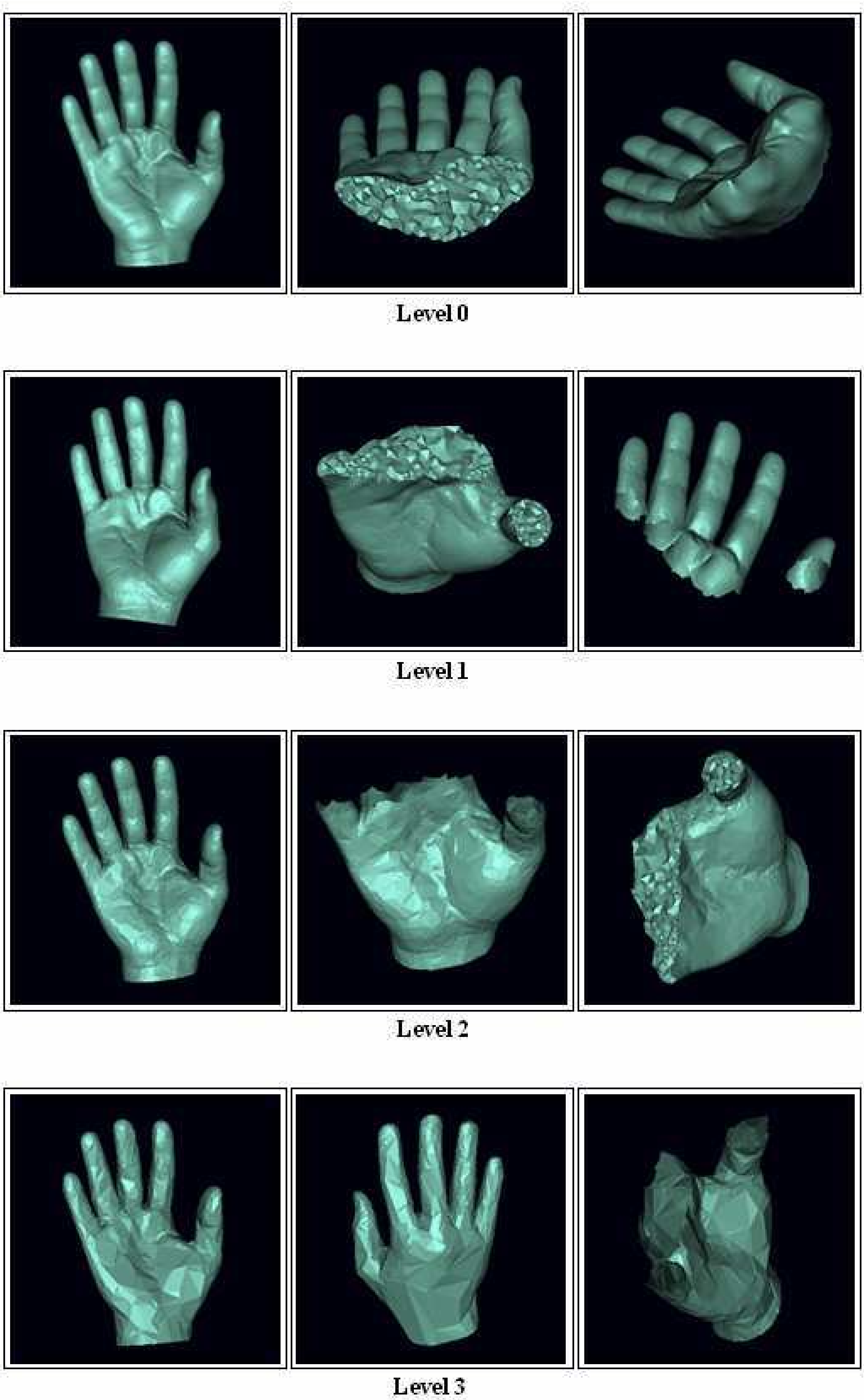


Figure 4.6: The different hierarchical levels of Hand Data

4.3 Conclusions

By thoroughly observing the results of implemented design it can be concluded that the hierarchical mesh decimation can prove a very fine alternative for better transmission of 3D Surface and Volumetric data over the internet and can also find its application in the field of telemedicine.

Although there are lot of degeneracies produced when 3D Delaunay method is implemented over Surface and Volumetric data, but a careful algorithm can deal with that and can produce satisfactory results.

The adaptive Alpha values facilitates in better visualization of 3D Surface and Volumetric data. It is also noticed that while decimation of the data, since the topology of the data is least harmed so the values of global alpha increases in the process.

The proposed method for Boundary and Inner vertex decimation produce satisfactory results in the sense that they not only reduce the data according to the partameters those are chosen, but also the quality of the tetrahedral mesh generated after each step of decimation remains within permissible limits, because the topology of the data is kept under control. It can also be noticed that even reducing the data to mere ten percent of its original size the general topology remains same and no perceptible change can be detected.

Forward optimization algorithm provides the best results in terms of quality of the tetrahedrons as well as with regard to decimation. It is due to the fact that while implementing the forward algorithm each layer is being awarded the IP value and is being chosen as a candidate for decimation with best constraints. While in backward algorithm the decimation of inner layers have least of the constraint whereas at first layer after the boundary layer have the maximum constraint so it results in lesser decimation. On the contrary greedy is complex and eager to remove the vertices so implementation of greedy results in more vertices to be decimated but the quality of tetrahedrons is being compromised.

For future work it is suggested that IP function may formulized which may be data and application dependent.

The adaptive Alpha and the Decimation criteria have transformed the implemented design into a system which is capable of sending the data with least amount of bits and with minimum error in visualization.

The tests implemented for constantly monitoring the tetrahedral mesh quality are in conformity with the visualization of data.

Since the system is based upon Delaunay triangulation whose dual is Voronoi diagram, therefore the designed system has the ability to implement the Robust Mesh Watermarking as proposed by Hoppe [15] by utilizing the concept of Adaptive Alpha and Voronoi diagram.

For evaluation we may compare our algorithm with Multiresolution analysis. MRA or wavelets provide useful and efficient tools for representing functions at multiple levels of detail. A new class of wavelets, based on subdivision surfaces, that radically extends the class of representable functions is proposed in [7].

1. The complexity of both algorithms is theoretically high.

2. Quality of triangles at successive hierarchy levels with respect to the original ones is approximately similar for both methods. Deviation angle is increased for some degrees in the multiresolution method while it is decreased in the proposed simplification method.

3. The main difference between these two methods is the type of meshes that they can be used for. For multiresolution based one mesh has to be conformable to 1 to 4 subdivisions, case which is not always possible. Hoppe [3] have developed and applied algorithms and methods to convert meshes of any topology to 1 to 4 subdivided meshes. This implies that one has to know both mesh vertices and triangles. In the proposed method only vertices have to be known, triangles are determined by Delaunay triangulation.

4. The main difficulty with Delaunay triangulation is that optimality for 3D meshes is not proven. There may be non connected regions since α value does not guarantee a connected mesh. This is overcome with use of local α values

5. Comparison in terms of removed number of vertices shows that for the multiresolution based method there is a reduction of 24.5%-25.5% of vertices when passing from one finer hierarchy level to the coarser one. Thus there is a removal of 75% of vertices for one decimation level. But the algorithm implemented can decimate from one percent to eighty percent of data in the successive hierarchy levels.

6. For the multiresolution approach based decimation method when transmitting a mesh, one needs to transfer beside vertices and triangles of the base mesh, wavelet

coefficients In the proposed method, one only needs the vertices and a very small amount of other data like global alpha, around ten values of local alpha and the information about the topology of the data is needed to be transmitted so a higher transmission rate is possible. This extra data ranges around 9-10 alpha values as given in Table 4.2, 4.4, 4.6, 4.8, 4.10, 4.12.

So the criteria for the formation of hierarchy of 3D meshes in the proposed method are to remove the maximum number of independent vertices. This will help in handling huge amount of volume data. The tradeoff between maximum removal of vertices and quality of the so formed hierarchy meshes is accomplished by allowing removal of maximum number of vertices so that the formed coarser mesh has an acceptable value for the mesh performance criteria.

1. REFERENCES

[1] **W. Schroeder, K. Martin, B. Lorenson,** 1998. The Visualization Toolkit: An Object-Oriented Approach to 3-D Graphics, 2nd Edition, Prentice Hall.

[2] **H. Borouchaki, S. H. Lo**, 1995. Fast Delaunay triangulation in three dimensions, *Comput. Methods Appl. Mech. Engrg.*, pp. 153–167.

[3] **H. Edelsbrunner,** 1992. Weighted alpha shapes. Technical Report UIUCDCS-R-92-1760, Dept. Comput. Sci., Univ. Illinois, Urbana, IL.

[4] **H. Edelsbrunner, E. P. Mücke,** 1992**.** Three-dimensional alpha shapes, in *Computer Graphics-Proc.* SIGGRAPH ' 92, 75-82.

[5] **J. R. Shewchuck,** 1998. Tetrahedral mesh generation by delaunay refinement, in *Proceedings of the fourteenth annual symposium on computational geometry*, Association for Computing Machinery, 86-95.

[6] **Heckbert P. S. and Garland M**., 1994. Multiresolution Modeling for Fast Rendering. *Graphics Interface'94-, Proc*, Banff, Alta.

[7] **Hoppe H.,** 1996. Progressive Meshes, *Computer Graphics-Proc.* SIGGRAPH ' 96.

[8] **Aysal C., Melkisetoğlu R., Celasun I., Tekalp A. M.,** 2003. Design of Hierarchical 3D Meshes for Object Representation and Progressive Transmission, CCCTY03, Atlanta, USA.

[9] **Van Beek .P.J.L, Celasun.I, and Tekalp A. M.,** 1998. “Hierarchical coding of 2D mesh geometry and motion”, VCIP98.

[10] **A. Liu, B. Joe,** 1994. Relationship between tetrahedron shape measures, Bit, 34, 268-287.

[11] **P. Labbé, J. Dompierre, F. Guibault, R. Camarero,** 1999. On element shape measures for mesh optimization, CERCA R99-23.

[12] **I. Celasun, E. Manavoğlu, B. Yeğen, A. Murat Tekalp,** 2001 . Hierarchical design of 3D meshes in representation/storage of volume data for multimedia applications, ELECO Bursa.

[13] **I. Celasun, A. Murat Tekalp,** 2000 . Optimal 2D Hierarchical Content-based Mesh Design and Update for Object-Based Video", IEEE CSVT .

[14] **Hearn D., and Baker M. P.,** 1997. Computer Graphic , 2nd ed. Englewood Cliffs, NJ: Prentice Hall.

[15] **E. Praun, H. Hoppe, A. Finkelstein**, 1999. Robust Mesh Watermarking, in: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques,* ACM Press/Addison-Wesley Pub., pp. 49--56.

[16] **Deering, M., (1995).** Geometry Compression. *Siggraph'95 Conference Proceedings*, pp.13-20.

[17] **John Burkardt,** http://www.scs.fsu.edu/~burkardt/

[18] **Michael Garland and Paul S. Heckbert**, 1997. Surface Simplification Using Quadric Error Metrics, Association for Computing Machinery (ACM)

[19] http://shapes.aim-at-shape.net/

[20] **William J. Schroeder, Jonathan A. Zarge, William E. Lorensen.** Decimation of Triangle Meshes, General Electric Company Schenectady, NY.

[21] **Wilson DR, Martinez TR** (1997) Improved heterogenous distance functions, *Journal of Artificial Intelligence Research 6*, pp. 1-34.

[22] **Celasun, I and Tekal AM** Optimal 2-D Hierarchical Content-Based Mesh Design and Update for Object-Based Video, *IEEE Transactions on Circuits and systems for video technology*, VOL. 10, No. 7, October 2000, pages 1135-1153

BIOGRAPHY

Mr Serkan ERÖKSÜZ borned on 26th of April 1979 at Ankara, Turkey. He received his early education from Merzifon Anatolian High School, Amasya between 1990-1997. In 2002 he graduated from in the Telecommunication engineering department of Electrical and Electronics Engineering of Istanbul Technical University. Since 2003 he has been working as research assistant at the Center for Satellite Communication and Remote Sensing (CSCRS), of Istanbul Technical University (ITU).