Examensarbete
LITH-ITN-MT-EX--07/006--SE

Computational Geometry and Surface Reconstruction from Unorganized Point Clouds

Ebbe Strandell

2007-02-05

Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Science and Technology
Linköpings Universitet
SE-601 74 Norrköping, Sweden

Institutionen för teknik och naturvetenskap
Linköpings Universitet
601 74 Norrköping

LITH-ITN-MT-EX--07/006--SE

Computational Geometry and Surface Reconstruction from Unorganized Point Clouds

Examensarbete utfört i medieteknik
vid Linköpings Tekniska Högskola, Campus Norrköping

Ebbe Strandell

Handledare Dr. Fang-Pang Lin
Examinator Björn Gudmundsson

Norrköping 2007-02-05

Avdelning, Institution
Division, Department

Institutionen för teknik och naturvetenskap

Department of Science and Technology

Datum
Date

2007-02-05

Språk
Language

- [ ] Svenska/Swedish
- [x] Engelska/English
- [ ] _______________

Rapporttyp
Report category

Examensarbete
- [ ] B-uppsats
- [ ] C-uppsats
- [x] D-uppsats
- [ ] _______________

ISBN

ISRN **LITH-ITN-MT-EX--07/006--SE**

Serietitel och serienummer
Title of series, numbering

ISSN

URL för elektronisk version

Titel
Title

Computational Geometry and Surface Reconstruction from Unorganized Point Clouds

Författare
Author

Ebbe Strandell

Sammanfattning
Abstract

This thesis addresses the problem of constructing virtual representations of surfaces only known as clouds of unstructured points in space. This problem is related to many areas including computer graphics, image processing, computer vision, reverse engineering and geometry studies. Data sets can be acquired from a wide range of sources including Computer Tomography (CT), Magnetic Resonance Imaging (MRI), medical cryosections, laser range scanners, seismic surveys or mathematical models. This thesis will furthermost focus on medical samples acquired through cryosections of bodies.
In this thesis report various computational geometry approaches of surface reconstruction are evaluated in terms of adequateness for scientific uses. Two methods called γ-regular shapes and the Power Crust are implemented and evaluated. The contribution of this work is the proposal of a new hybrid method of surface reconstruction in three dimensions. The underlying thought of the hybrid solution is to utilize the inverse medial axis transformation, defined by the Power Crust, to recover holes that may appear in the three dimensional γ-regular shapes.

Nyckelord
Keyword

Computational geometry, surface reconstruction, 3D point clouds, Voronoi diagram, Delaunay triangulation, medical visualization

Computational Geometry and Surface Reconstruction from Unorganized Point Clouds

Master Thesis

Ebbe Strandell

Department of Science and Technology, ITN
Linköping University, campus Norrköping

February 2007

Abstract

This thesis addresses the problem of constructing virtual representations of surfaces only known as clouds of unstructured points in space. This problem is related to many areas including computer graphics, image processing, computer vision, reverse engineering and geometry studies. Data sets can be acquired from a wide range of sources including Computer Tomography (CT), Magnetic Resonance Imaging (MRI), medical cryosections, laser range scanners, seismic surveys or mathematical models. This thesis will furthermost focus on medical samples acquired through cryosections of bodies.

In this thesis report various computational geometry approaches of surface reconstruction are evaluated in terms of adequateness for scientific uses. Two methods called "γ-regular shapes" and "the Power Crust" are implemented and evaluated. The contribution of this work is the proposal of a new hybrid method of surface reconstruction in three dimensions. The underlying thought of the hybrid solution is to utilize the inverse medial axis transformation, defined by the Power Crust, to recover holes that may appear in the three dimensional γ-regular shapes.

Keywords: Computational geometry, surface reconstruction, 3D point clouds, Voronoi diagram, Delaunay triangulation, medical visualization.

Acknowledgments

I want to dedicate a special thank you to everyone at NCHC for their hospitality and curiosity regarding myself and my research. Especially I would like to express my gratitude towards Dr. Fang-Pang Lin who made this collaboration possible and who helped me conduct the research. I also want to thank Kevin, James and Howard for guiding me during this time. It would not have been the same without your friendship and support.

Table of Contents

Part I Introduction

Chapter 1 Introduction

This chapter will provide background and motivation of this research, point out goals, explain how the research was conducted and the structure of this report.

1.1 Research Background And Motivation

The National Center for High-Performance Computing, NCHC, is a governmental sponsored research institute with headquarters in Hsinchu, Taiwan. The center's core technologies include network infrastructure and computer information systems, computational grids and clusters, supercomputing, engineering, physics and chemistry, biotechnology and visualization.

Within visualization the center actively works with what is called a Tiled Display Wall, TDW, that consists of multiple high resolution projectors and is used to display passive three dimensional images on a huge semi-transparent film. The TDW is used to display NCHC's research within visualization including high definition image processing, stereo video streaming, geographical information and medical imaging.

The center wishes to further develop their toolbox to build and display three dimensional models for use in, for example, medical studies. This process can roughly be divided into three stages; *data acquisition*, *surface reconstruction* and *visualization.* As NCHC possesses well-defined methods for the first and last stage, this research is primarily focused on surface reconstruction.

Surface reconstruction has, except for visual demonstrations, many practical uses. The technique has been successfully used to scan art work to reveal details that are impossible to see with the naked eye. Leonardo da Vinci's Mona Lisa and Michelangelo's David are famous examples of art that has been scanned and reconstructed as high-resolution three dimensional models. In the prior case the purpose was to monitor the state of conservation and to enable in-depth studies of techniques and materials used by the artist. Surface reconstruction can also be used in so called Automatic Optical Inspections, AOI, where circuit boards are scanned for errors. Another possible usage is within CAD-techniques where laser range scanners are used to create models and blueprints of buildings etc.

1.2 Problem Statement

A common use of computer graphics is to visualize models of real-world objects in artificial environments. Such models can be created by artists using 3D modeling software. As in the process of creating a painting it is up to the artist to understand and recreate the object, adding artistic effects and simplifications to the reality. Few people possess the skills needed to create a painting as detailed as a photograph.

An alternative way to create three dimensional objects is by what is called surface reconstruction. Here, the artistic skills of the user are of little importance, instead it requires precise equipment. The first step of reconstructing a model is to acquire data of the real, physical object. This can be done using various scanning techniques and the output is a collection of points describing the original surface. The resolution and correctness of the final model is highly dependent on the density and distribution of the points. The data is used to recreate a virtual model, or surface, of the object. An example of the intestines of a mouse is shown in figure 1.

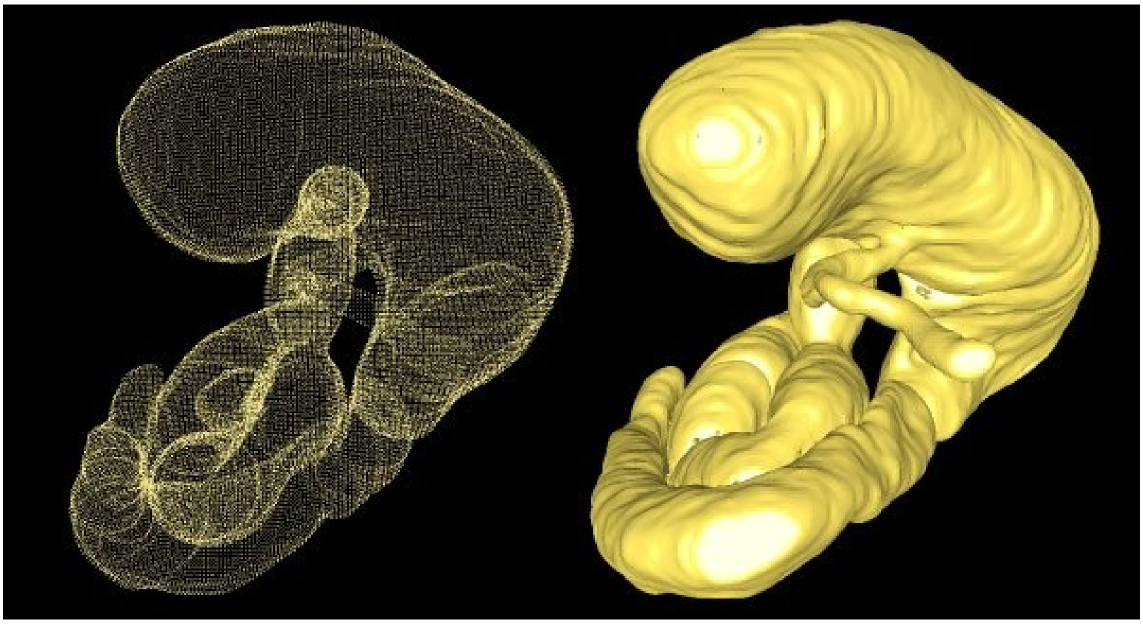


Figure 1: Left: Sample points of the intestines of a mouse. Right: The corresponding reconstructed surface.

The real challenge of surface reconstruction is that the information provided about the unknown surface is very limited. Except for the set of points, where each point is described by its position in space, no other information such as topology or connectivity is given. Furthermore data sets with non-uniform sampling density and noise must be considered. As discussed in this report it is therefore hard to recreate surfaces that contain the same details and topology as the referenced objects.

1.3 Thesis Aim

The goal of this thesis project is to design and implement a method for automatic reconstruction of three dimensional surfaces from sets of point data. The input to the reconstruction method is an unstructured point cloud, containing only the position of each point in $\mathbb{R}^3$. The density of sampling points must be assumed to be non-uniform, vastly increasing the complexity of the task. The surface reconstructing method should also

- work for large data sets (up to one million sample points),
- have a high level of correctness when used in medical applications,
- be computationally efficient,
- be semi, or preferably fully, automatic.

1.4 Sources

This thesis is based on a fairly large body of research papers within the surface reconstruction domain. Most of the articles date from the middle of 1990 to early 2000 when plenty of related research was conducted. During the literature review several PhD dissertations, such as, [Men01],[Hra03] and [Pap04] were used as they contain broad state of the art sections.

1.5 Methodology

This chapter will explain how this thesis was planned and carried out. Basically the research was carried out in four phases which are outlined below.

1.5.1 State of the Art

The first step in this research was to evaluate existing surface reconstruction methods. The goal was to find one or two methods that could be used as foundation for further evaluation and implementation. Each surface reconstruction method was evaluated bearing the goals of this thesis in mind. As numerous methods exist it would not have been feasible to test each implementation, instead many methods had to be ruled out based only on the results presented in the papers.

1.5.2 Implementation and Evaluation of Existing Solutions

The literature review provided information on how various algorithms can be expected to work. Based on this information the two methods called "γ-Regular Shapes" and "The Power Crust" were implementation and further evaluated. The implementation phase was the most time-consuming part of the research but necessary to get in-depth knowledge of the chosen algorithms. In practice the two methods corresponded well to expectations raised in related documentation. Unfortunately that implies that none of the reviewed methods were capable to fully match the goals of this research.

1.5.3 Development of a Hybrid Solution

A natural way to conclude the research was to make an effort of developing a novel surface reconstruction method. The new method is a hybrid solution, in which strengths from the two other methods is combined. This method proved to be a step in the right direction but no guarantees of correctness can be given with the current algorithm. This part of the research has more novelty value and is less derived from existing research.

1.6 Programming Environment And Hardware

The software implementation is carried out in a Linux environment to be compatible with the massive hardware that NCHC possesses. During implementation ordinary, although quite powerful, workstations have been used. Surface reconstruction of larger data sets is quite memory demanding, therefore the workstations used under this project are equipped with 1GB of memory. The existing framework and all external source codes are written in C whilst new additions are written in object oriented manners using C++.

1.7 Thesis Structure

This thesis report is divided into four parts. The first part provides background to the thesis itself and motivation to further research within surface reconstruction. The second part provides technical background to the research by explaining the pipeline in which surface reconstruction can be applied as well as a state of the art summary. The third part contains details of the code framework and the implementation related to this research which is followed by results and discussion in part four.

1.8 Target Group

This final thesis is a part of the graduate program *Master of Science in Media Technologies and Engineering* provided by Linköping University in Sweden. Due to the technical nature of surface reconstruction this work is foremost aimed for readers with background within computer science and visualization. The reader is expected to be familiar with linear algebra and to well understand the vocabulary and concepts explained in chapter 1.9.

1.9 Conceptual Framework

Throughout this report the reader is expected to be familiar with the following terminology and denotations. The terms explained below are general and commonly used in related documentation.

- The *dimension d* of the space in which a function or an object resides is denoted as $\mathbb{R}^d$.
- *Sample points* or *data points* are derived from the data acquisition process where the physical object is scanned in some manner.
- *Vertices* are used to represent virtual points. Within computer graphics vertices are considered to be of dimension zero, $\mathbb{R}^0$, as it has no length, area or volume. Vertices are also used as basic elements in all dimensioned elements such as lines ($\mathbb{R}^1$), triangles ($\mathbb{R}^2$), tetrahedra ($\mathbb{R}^3$) etc.
- A *voxel* is an analog to the three dimensional vertex. In contrast to the vertex the voxel has volume and can be seen as a box with edges of constant length. Each voxel is defined by 8 vertices. Voxels are fundamental within volumetric rendering and construction methods.
- A *simplex* is the geometric *d*–dimensional analog of a triangle; line in $\mathbb{R}^1$, triangle in $\mathbb{R}^2$, tetrahedron in $\mathbb{R}^3$. A so called *d*–simplex is a simplex of dimension *d* and is defined by $d+1$vertices.
- A *circumcircle* is defined as a circle that passes through all vertices of a triangle. *Circumspheres* and *circum-hyperspheres* are the extensions of the circumcircle to the third and *d*:th dimension respectively.
- The term *correct surface* will here be used to describe a surface that is topologically correct, watertight and that goes through all its sample points under some sort of sampling criteria.
- The terms *boundary* and *hole* are in this report used interchangeably. While holes are due to errors in the reconstruction algorithm boundaries are more general and can also exist on the edge of an open manifold.

- A *degenerated* simplex is a simplex of dimension *d* that can be described with using a (*d*–1)simplex. For example a triangle ($\mathbb{R}^2$) with one side length close to zero can be described with a line ($\mathbb{R}^1$). Such a triangle is referred to as degenerated.
- Simplices that are are almost degenerated are called *slivers*. These include almost flat tetrahedra and and very narrow triangles. The word sliver does not have a final definition but in general a simplex is considered sliver if it contains a very small angle between two edges.

Part II Technical Background

Chapter 2 The Reconstruction Pipeline

The reconstruction process includes capturing surface properties of a real-world, physical object to create a model to use in an interactive scene, enabling the user to elaborate with a virtual copy of the object. We can speak about the process as three independent stages; data acquisition, surface reconstruction and visualization as shown in figure 2.

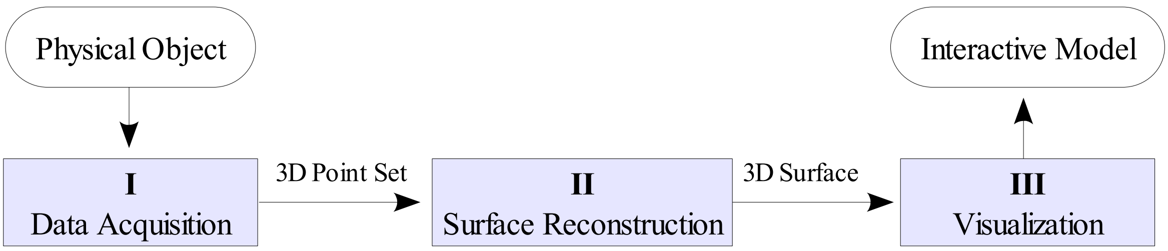


Figure 2: The reconstruction pipeline.

In this chapter each stage will be briefly explained so that the entire process of constructing virtual views from real objects can be understood.

2.1 Data Acquisition

The first stage of the reconstruction process is called data acquisition and concerns the means in which properties of physical objects are captured and saved as data sets. The most commonly used methods to acquire the data include automatic medical scanning techniques such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), image data extracted from cryosections (thin slices) of the object (fig 3, left), or laser range scanners used to scan static objects (fig 3, right).

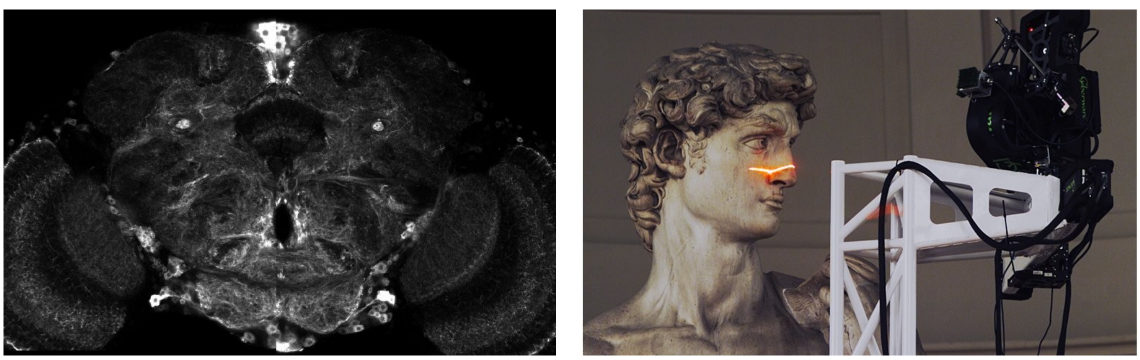


Figure 3: Data acquisition. Left: Cryosection of the brain of a fruit fly. Image in courtesy of NCHC. Right: Michelangelo's David is being laser scanned. Image courtesy of The Digital Michelangelo Project [TDMP].

As mentioned, the input to the reconstruction methods discussed in this thesis is purely a set of points, described with the position of each point in $\mathbb{R}^3$. Therefore the captured data must be converted before it can be used for surface reconstruction. For example the captured data may consist of 2D images (cryosecting methods), or 3D image data (CT, MRI) or range data (laser scanners).

2.2 Surface Reconstruction

The second stage of the reconstructions pipeline appears purely computational and it is based completely on software, well hidden from the end user. The set of points from the acquisition process is used to rebuild a surface as illustrated in figure 4. As discussed in chapter 3 there exist several approaches on how to deal with this problem. Given the limited information about the object, non-uniform sampling density and possible noise, there no method today that can guarantee to always reconstruct the surface correctly.

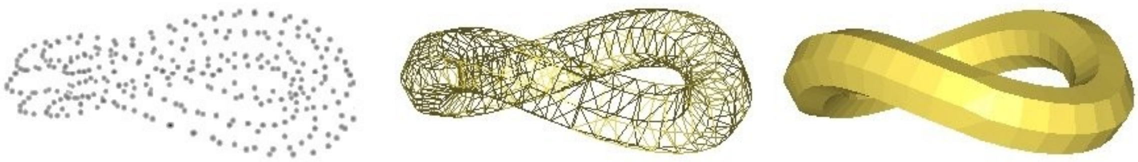


Figure 4: Left: Sample points of an object. Middle and right: Wireframe and solid representations of the corresponding reconstructed surface.

2.3 Mesh Visualization

The final part of the visualization pipeline, the actual rendering, involves inserting the model into an interactive scene and to draw the scene on the computer screen. This part is well defined and there exists standard solutions such as OpenGL and DirectX to utilize. OpenGL is a widely used standard that is supported by all major platforms such as Microsoft Windows, Linux/Unix and Mac OS.

NCHC's existing toolbox is called IvIsee and is based on VTK, the Visualization Tool Kit, and OpenGL. IvIsee adds functionality by supporting 7 different modes of stereo view for passive virtual reality as well as supporting various file formats commonly used to store triangle surfaces. IvIsee is also designed to be used with the Tiled Display Wall mentioned in the introduction.

Chapter 3 State Of The Art

Reconstruction of surfaces from unstructured points in $\mathbb{R}^3$ is a well studied problem and its complexity has spawned many approaches on how to generate surfaces with sufficient level of correctness. Early research in surface reconstruction acts as fundamental platforms that today often are used as platforms for case-sensitive hybrid solutions. Reconstruction of medical surfaces is challenging because of high requirements of correctness and details of the final surface. To make the task worse the data is commonly both noisy and of non-uniform sampling density. None of the reviewed methods seems to be developed primarily for medical use.

Most of the reviewed methods are stated to work well when certain sampling criteria are fulfilled. Detailed discussion about these criteria is, however, often left out. The methods mentioned here should be considered as possible platforms from where further research for our purpose has to be done. The existing techniques are grouped as proposed by L. Papaleo in [Pap04]:

- *Computational Geometric Approaches*: These approaches take into account the structure of the object to reconstruct and they are based on computational geometric concepts such as Voronoi Diagrams, Delaunay Triangulation or Medial Axis. These concepts are essential in this research and will be discussed in detail in chapter 3.1. Various Computational Geometric Approaches approaches are discussed in chapter 3.2.
- *Volumetric Approaches*: Methods in this category build the surface by analyzing the volume it is captured in. The representation may be a 3D grid of values representing the distance to the surface or it may be a mathematical representation of the distance function [Hra03]. Volumetric approaches are exemplified in chapter 3.3.

3.1 Triangulations And Space-Filling Graphs

A natural way to create an initial shape of a set of points is by *triangulation*. In a triangulation all sample points are connected to each other to form non-overlapping edges, triangles and tetrahedra, resulting in a decomposition of the space into simplices with vertices in the sample points. The result is illustrated in figure 5 where a set of points has been used to create a triangulation (left and middle) from which a subset of triangles are selected as the surface (right).

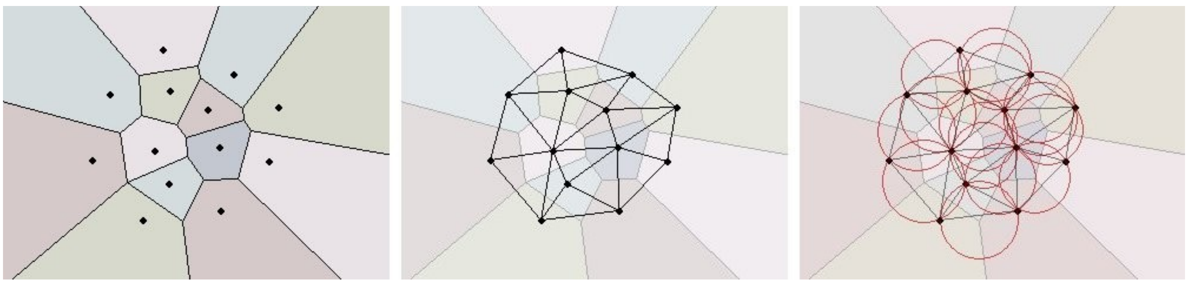
Figure 5: Wireframe (left) and solid (middle) representation of a Delaunay triangulation. Right: The corresponding surface.

The term triangulation is general and various ways of creating triangulations exist. As a triangulation may contain badly shaped simplices (degenerates and slivers) it becomes necessary to consider the quality of the triangulation. One triangulation that has proved to be of high quality in practical applications is the Delaunay triangulation. Before investigating the Delaunay triangulation, which is very essential to this research, it is necessary to understand its duality with the Voronoi diagram.

3.1.1 The Voronoi Diagram

Within the Voronoi domain the space containing the sample points is subdivided into Voronoi cells. Per definition each cell contains exactly one generating sample point so that any arbitrary point within the cell will be closer to the generating point than to any other sample points in terms of Euclidean distance. This implies that there are more than one closest point along the edges and in the vertices of the Voronoi diagram. The result is a partition of the space into regions for which each closest neighbor defines an edge in the Voronoi diagram. This is shown in figure 8a where each sample point defines a Voronoi cell.

As noted in [LFT98] and illustrated in figure 8b the straight-line duality of the Voronoi diagram is a triangulation. This triangulation is called Delaunay triangulations as it was first proposed by Boris Delaunay in 1934.



(a) Voronoi diagram (b) Delaunay triangulation (c) Empty Circum-circles

Figure 6: (a) The Voronoi diagram of a set of sample points. (b) The corresponding Delaunay triangulation of the sample points. (c) The empty circum-circle criteria imply that each Delaunay circle is empty of any other sample point.

3.1.2 The Delaunay Triangulation

The Delaunay triangulation provides a mathematically well-defined way to reduce the number of bad simplices in a triangulation. Each Delaunay simplex is bound to fulfill the empty circum-hypersphere criteria as shown in figure 8c. This means that the circum-hypersphere “touching” all vertices of a simplex does not contain any other sample point. Only if this criterion is fulfilled a $d+1$subset of points can form a Delaunay simplex. An interesting observation is that center points of the Delaunay circum-hyperspheres coincides with the Voronoi vertices so that each Delaunay center point also will be a Voronoi vertex.

As noted by Edelsbrunner and Mücke in [EM94] the Delaunay triangulation is guaranteed to be unique as long as the points are in general position. General position implies that no $d+1$ points are on the same hyperplane (for where no triangulation exists) and no $d+2$ points are on the same hypersphere (for when the triangulation is not unique).

The gain of the Delaunay constraint is that the triangulation always maximizes the minimum angle of the triangles, resulting in a triangulation that has the lowest possible amount of degenerated simplices. This property is desirable as triangles and tetrahedra with area and volume close to zero makes it hard to calculate for example circum-hyperspheres and the plane containing the degenerated simplex.

3.1.3 Regular Triangulations and Power Diagrams

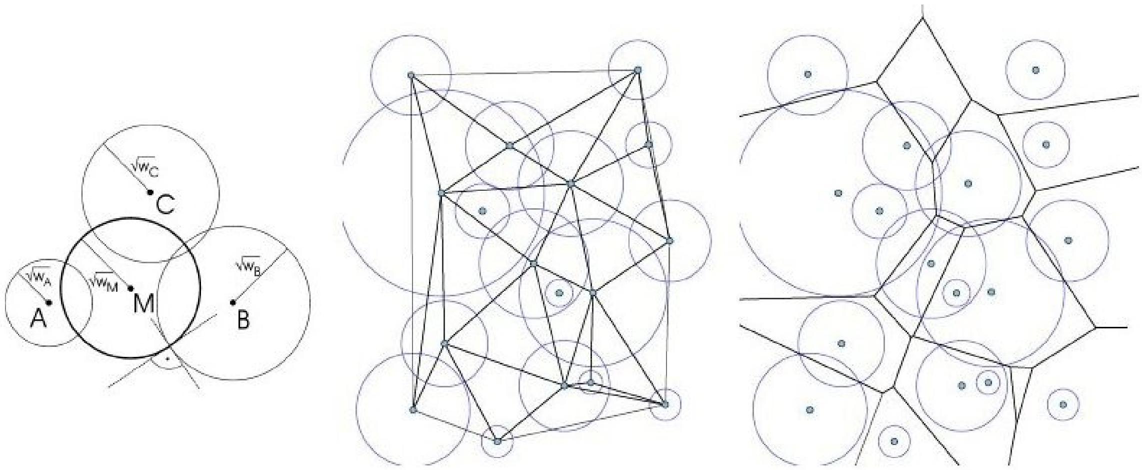
The regular triangulation is an extension of the Delaunay triangulation where input points are associated with weights. The intuitive explanation of the weighting is to see each input point **p** with the weight $\mathbf{w_p}$ as a ball centered at **p** and with a radius $r=\sqrt{W_p}$.

In unweighted Delaunay triangulations the Euclidean distance is used to measure the distance between sample points and to determine whenever a point lies inside a Delaunay cirucum-hypersphere. Regular triangulations work in similar manners using the so called *power distance*, **π**, between two balls **p** and **q**, which is given by the formula

$$\pi=|p-q|^2-W_p-W_q$$

where the first part of the equation describes the squared Euclidean distance between the centers of the balls from which the total weight of the two balls is subtracted.

A central concept within regular triangulations is *orthogonality*. Two weighted points, or balls, are called orthogonal if and only if the power distance between them is zero which happens when the tangent hyper-planes of the two balls are orthogonal at the points of intersection as illustrated between point M and B in figure 7a.



(a) Orthogonal circles (b) Regular triangulation (c) Power Diagram

Figure 7: (a) The orthogonal center M of the simplex with weighted vertices A,B and C. Image courtesy of [BSD+05]. (b, c) Regular triangulation and the corresponding power diagram of a set of weighted points. Image courtesy of [VPC02].

In a regular triangulation the orthogonality is used to define the *orthosphere*. For each *d*-simplex there exists exactly one, minimum, weighted point that is orthogonal to all *d*+1 weighted vertices of the simplex, we call that point *orthogonal center* or orthosphere [ES92]. This is shown in figure 7a where M is the orthogonal center of the triangle formed by vertices A, B and C.

In the unweighted Delaunay environment the circum-hypersphere of each simplex must be empty of any other sample point to fulfill the empty circle criteria. Similarly, the triangulation is called regular if all orthospheres are empty of other input points. This implies that the power distance between the orthocenter and any other input point is non-negative [BSD+05].

A remark to the above is that if all weights of a simplex are equal to zero the orthosphere of the simplex is it's circum-hypersphere. Similarly, if all input points have zero weight the regular triangulation is the Delaunay triangulation of the input points.

As with the duality of the Delaunay triangulation and the Voronoi diagram the dual of the regular triangulations is called the power diagram. This straight-line duality is shown in figure7b and c. Each weighted point of the regular triangulation defines a cell in the power diagram. Within each power cell the generating point is closer than any than any other input point in terms of power distance.

A consequence of the weighting is that not all input points will be a part of the regular triangulation. This is due to the fact that a point at insertion time may not be the closest point to any arbitrary point in terms of power distance, and therefore it will not define any power cell. If so, the new point does not invalidate any existing simplex. We call such points *redundant* as we do not include them in the triangulation.

3.2 Computational Geometric Approaches

Computational geometric approaches construct surfaces based on geometric properties of the point set using concepts such as Voronoi diagrams, Delaunay triangulation, medial axis etc. As we will see in this chapter this enables both interpolated and approximated surface reconstruction.

3.2.1 α-Shapes

Edelsbrunner and Mücke [EM94] introduced in 1994 the concept of α-shapes in three dimensions. α-shapes is a well-known approach to filter triangulations by removing elements that do not fit inside a circle or sphere with the radius α.

Algorithm overview:

- Compute the Delaunay triangulation of the sample points.
- Extract the α-shape by removing Delaunay elements with a surrounding sphere greater than a value α.
- Output a surface from each α-shape triangle with at least one α-ball empty of other sample points.

The Delaunay triangulation is used to create a tetrahedral decomposition of the convex hull, created from the input sample points. A second hull, called the α-shape, is created from the triangulation by removing each tetrahedra, triangle and edge with a surrounding sphere with radius greater than α. The output is a subset of the Delaunay triangulation, meaning that it still contains tetrahedra.

To extract a surface each triangle of the α-shape is evaluated. Each triangle have exactly two balls with radius α that touch the three vertices of the triangle. Such balls are called α-balls. A triangle belongs to the surface if and only if at least one its α-balls is empty of other sample points. The resulting hull consists of triangles and edges with a maximum curvature equal to α.

The obvious problem with this method is to find an α-value that can extract the correct surface. If α is too small gaps will appear and solid shapes can be disconnected. A too large α-value will instead lead to loss of details in the final shape. Therefore α has to be chosen experimentally. The method is also unstable in case of non-uniform density of the sample points when there might not exist any α-value to correctly reconstruct the surface.

3.2.2 Anisotropic Density-Scaled α-Shapes

The main shortcomings of Edelsbrunner and Mücke's approach have been addressed by Marek Teichmann and Michael Capps in [TC98] who propose an approach called anisotropic density-scaled α-shapes. This approach is built on the previous method with the addition of two new criteria on how to select faces of the surface.

The original α-shape approach would fail when the distance between two surfaces is small in comparison to α. The anisotropic test is used to distinguish such disconnected surfaces. The solution requires the direction of the face normals for each point and is based on the assumption that the variation of the normal direction of neighboring points remains small for smooth surfaces. This implies that two triangles on two disconnected surfaces will have face normals with somewhat opposite directions.

The density scaled α-ball is used to improve the result for point sets with non-uniform sampling density. It allows the α-ball to "shrink" wherever the density is high so that detailed information of the surface is available.

According to the paper the new approach significantly improves Edelsbrunner and Mücke's method but it is meant to be used in an interactive setting where the user controls parameters for the shape and scale of the α-ball.

3.2.3 γ-regular shapes

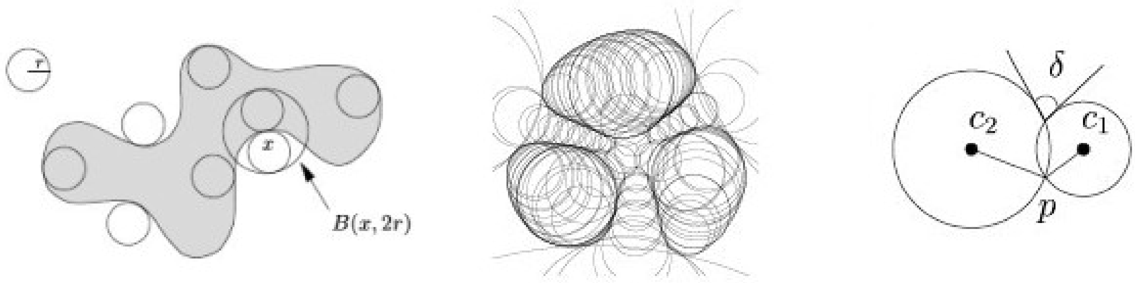
Another fundamental approach called γ-regular shapes was proposed by Attali in 1997 [Att97]. She defines the reconstruction problem as finding the *normalized mesh* of the unknown surface **S**. The normalized mesh is a particular union of cells with vertices at the sample points and per definition included in the Delaunay triangulation of the sample points, and therefore formed by its elements; edges, faces and simplices.

Algorithm overview:

- Compute the Delaunay triangulation of the sample points.
- For each face calculate the angle of intersection, δ, between the two Delaunay spheres passing through the face. Output faces with $\delta < \Pi/2$ as faces of the normalized mesh.
- In 3D, apply some post processing method to recover holes in the mesh.

To simplify the search for the normalized mesh Attali makes the assumption that **S** is the boundary of a γ-regular shape. As illustrated in figure 8a this implies that the surface is morphologically open and closed with respect to a disc with radius $\gamma > 0$, so that each point at the boundary of the shape have a tangent and a radius of curvature greater or equal to γ.

In $\mathbb{R}^2$ it is guarantied that every disk passing though three distinct boundary points will have a radius greater than γ. Furthermore the discs tend to converge to infinity when the sample density heads towards zero, eventually approximating the tangent of the boundary as illustrated in 8b.



(a) γ-sample (b) Delaunay disks of an 2D sample (c) The angles δ

Figure 8: (a) The γ-sample is morphologically open and closed with respect to a disc with radius γ.(b) Delaunay discs become tangent to the surface. (c) The angles δ formed by two Delaunay disks. Image courtesy of [Att97].

Each face of the Delaunay triangulation is intersected by two Delaunay discs with radii greater than γ passing through its endpoints. To decide which faces belong to the normalized mesh the angle $\boldsymbol{\delta}$ between the two Delaunay disks that passes through the face is calculated, fig. 8c. Let $\mathbf{S}_{\delta_0}$ denote the set of simplices with an intersection angle $\boldsymbol{\delta} \leq \boldsymbol{\delta}_0$. While the intersecting discs for internal simplices have relatively small radii (which results in a big angle), boundary edges tend to have at least one disc that approximates the tangent of the surface, resulting in a small angle between the discs.

Attali defines the *sampling path* $\boldsymbol{\epsilon}$ of **S** as the smallest possible radius of a sphere placed on **S** so that the sphere will always include at least one sampling point. As the sampling path tends to zero the normalized mesh will converge to the surface. One of the main findings in Attali's paper is that she gives a very precise sampling criterion by proving that when the sampling path $\boldsymbol{\epsilon} < \sin(\Pi/8)\boldsymbol{\gamma}$ the union of all edges or faces in $\mathbf{S}_{\Pi/2}$ is the normalized mesh of **S**.

In other words, for a good sample, the union of all *d*-simplices for which the angle between the two intersecting Delaunay hyper-spheres is less than $\Pi/2$ is indeed the normalized mesh.

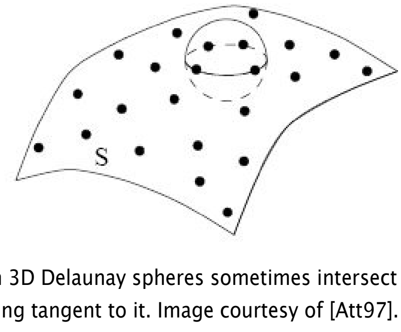


Figure 9: In 3D Delaunay spheres sometimes intersect the surface without being tangent to it. Image courtesy of [Att97].

Unfortunately not all nice properties of Attali's approach extend to three dimensions where holes are likely to appear even in areas with sufficient sampling density. This is due to the fact that Delaunay spheres may intersect the boundary without being tangent to it as shown in figure 9. This results in faces that are discarded due to large intersection angles between Delaunay spheres which leads to holes in the normalized mesh.

Attali proposes two alternatives to recover the holes, one is based on *border triangulation* and the other on *volume tessellation*. The first solution is very general as it can handle both open and closed surfaces as well as it is independent on the distribution of the sample points. The approach may, however, yield unexpected results as it simply works by recursively adding triangles that have more than one edge on the border. The volume tessellation approach is slightly more complicated. Based on a set of rules, Delaunay tetrahedra are merged until a satisfactory number of independent objects (manifolds) are obtained. This results in a set of closed surfaces of arbitrary topology. Unfortunately this solution is very computationally expensive ($O(n^4)$ in 3D) and perhaps more importantly, it leaves no guarantees of correctness.

3.2.4 The Power Crust

Amenta et al. have proposed a series of surface reconstruction approaches, starting with an approach called the “The Crust”, presented in [ABK98]. The Crust is based on an approximation of the surface's *medial axis* which can be thought of as the skeleton of the unknown surface (fig. 10a). The Crust is capable of generating topologically correct surfaces for “good samples”.

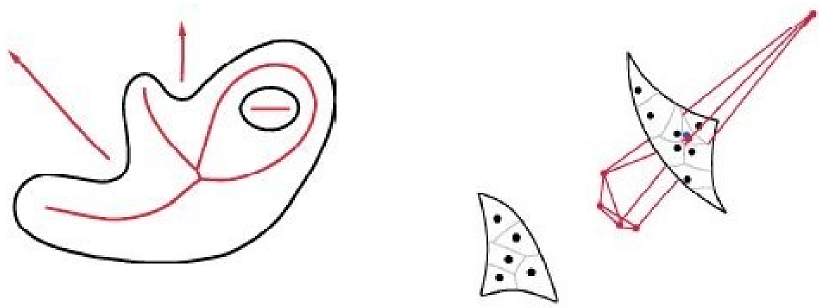
The later refined approach, called “The Power Crust”, as described in [ACK01], introduce various refinements such as the ability to always return *watertight* surfaces. This means that, even with very loose requirements on the input samples, the method will construct a mesh free of holes. The new approach also introduced some boosts in terms of calculation efficiency.

Algorithm overview:

- Compute the Delaunay triangulation of the sample points.
- For each sample point select two *poles* as the two, from the sample point, furthest away Voronoi vertices on opposite sides of the unknown surface. (These poles approximate the medial axis of the surface.)
- Compute each pole's weight and use the poles to compute a new, weighted, Delaunay triangulation.
- Build the power diagram of the poles and use it to classify each pole as being located either inside or outside the volume bounded by the surface.
- Output a surface defined by power diagram vertices of weighted Delaunay tetrahedra with both inside and outside vertices.

The Power Crust approach is based on the Delaunay triangulation and the approximation of the surface's medial axis. The medial axis is defined in all virtual points with more than one closest point on the surface. Each such point is the center of a, so called, *medial ball*. The medial ball is the largest possible hyper-sphere centered on the medial axis and empty of samples of the surface.

The crust utilizes the medial axis to create an “inside” and “outside” relationship in respect to the unknown surface. As two opposite medial balls will at most intersect shallowly, the basic idea is to define the unknown surface where these areas ($\mathbb{R}^2$) or volumes ($\mathbb{R}^3$) intersect.



(a) The medial axis. (b) Voronoi cell in 3D.

Figure 10: (a) The red curves indicate the medial axis of the black curves. (b) Some Voronoi vertices appear close to the surface and far away from the medial axis. Image courtesy of [ABK98].

The approximation of the medial axis is included in the Voronoi diagram, in fact, $\mathbb{R}^2$ Voronoi vertices converge towards the medial axis as the sampling density increases. As with the γ-regular shapes these nice properties does not extend to $\mathbb{R}^3$. In three dimensions Voronoi vertices can, and often will, appear close the surface and so far away from its medial axis, fig 10b.

This problem is addressed by associating each sample point with two *poles*. Voronoi cells tend to grow long and narrow as the sampling density increases. This implies that at least some Voronoi vertices of each cell will be located far away from the sample point and so be close to the medial axis. Figure 11 demonstrates the set of poles of a sample describing a knot. Poles are selected as the two opposite Voronoi vertices furthest away from the sample. The poles are on opposite sides of the sample if the vectors between the sample and the poles have a negative projection on each other. The union of the poles of each sample is indeed an approximation of the surface's medial axis.

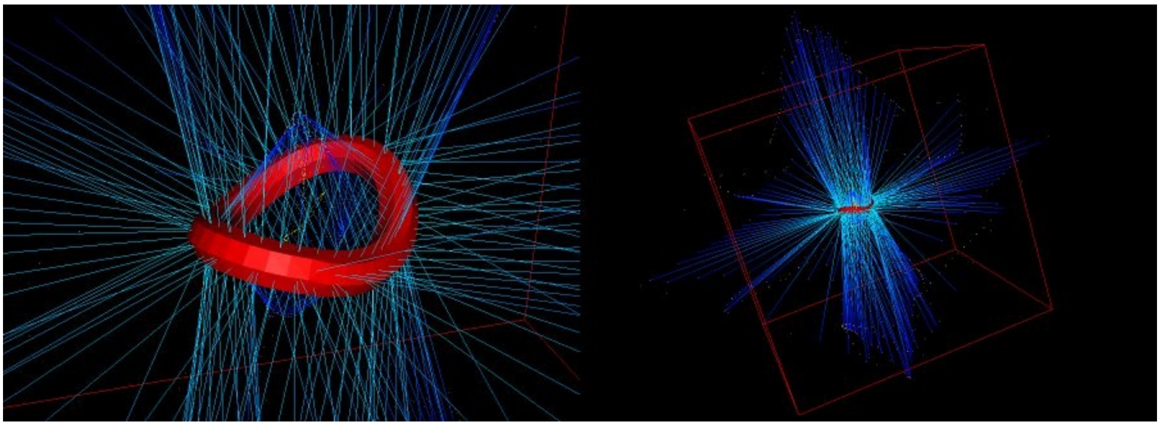


Figure 11: Crust poles of an set of data points describing the red knot. Dark-blue poles indicate outside poles, light-blue poles indicate inside poles.

Next, the power diagram of the medial balls, or *polar* balls, is calculated and utilized to build a neighboring structure of the poles. As explained in chapter **3.1.3** the power diagram is the dual of the weighted Delaunay triangulation and for each pole the squared radius of the polar ball is used as weight when calculating the power diagram.

So far the medial axis has been approximated as the union of the poles and a neighboring structure of these poles has been created. A data structure containing opposite relations of the poles is also created. The remaining steps are to label each pole as either inside or outside as well as output a surface that goes through the intersection of the polar balls.

The labeling algorithm is a key process of the Power Crust method and it starts with labeling poles outside a small bounding box as outside. For each classified pole its opposite and neighboring poles are classified by evaluating the angle between the polar balls. As shown in figure 12 two opposite poles are likely to have opposite labels if the angle between them is big, that is if they intersect shallowly. In similar manners, two neighboring poles are likely to have same labels if they intersect deeply which results in a small angle.

The final step of the Crust algorithm is to output “the Crust”. One, for the Power Crust, essential property of the power diagram, is that the power vertices will be positioned close to the original sample points due to the assignment of weights of the poles. The set of vertices defining the final surface is a subset of the power vertices, meaning that every vertex of the surface is a vertex of the power diagram but not vice versa. Faces are created “around” each edge of the power diagram with endpoints (poles) of opposite labels. Each face is formed by the Voronoi vertices from all second pass Delaunay tetrahedra that share the edge.

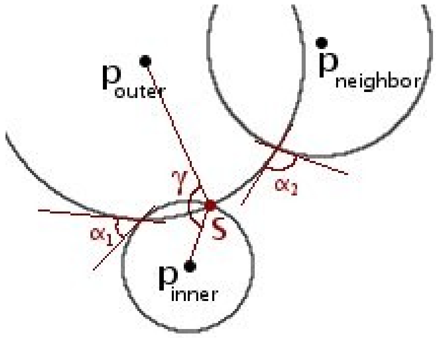


Figure 12: A sample **S** with its two corresponding poles and the angles α_1 and γ. Notice that α_2 is larger than α_1 as the neighboring balls are more deeply intersecting .

The final result of the crust algorithm, given a good sample, is an approximation of the medial axis as well as of the surface of the sample points. The output is watertight and topologically correct but does unfortunately not go through the original sample points.

3.3 Volumetric Approaches

In contrast to computational geometrical methods where the geometry of the sample points is evaluated, volumetric approaches are used to evaluate the space in which the points are captured. Volumetric approaches in general follow the method proposed by Hoppe et al and in this report his work will be used to explain volumetric approaches as a whole.

The method, normally referred to as "*Hoppe's Algorithm*" was presented in 1992 and published in [HDD+92]. The underlying thought in Hoppe's approach is to use a grid of voxels to represent the space in which the sample points are captured.

Algorithm overview:

- Divide the volume containing the sample points into a cubical lattice.
- For each grid-vertex calculate the distance to the closest local approximation of the unknown surface.
- Extract the surface using a marching cubes algorithm.

A voxel is a cubical element with vertices defining its corners and two adjacent voxels share exactly four vertices. Let **V** denote the set of vertices in the cubical lattice. The first task in Hoppe's approach is to calculate the signed distance from each vertex $\mathbf{V}_i$ to a local approximation of the unknown surface **S**. Each sample point $\mathbf{P}_i$ of **S** is associated with a tangent plane calculated from the *k*-neighborhood of closest points and centered in the sample point itself. The tangent planes of a subset of **S** act as a local approximation of **S.**

The distance function calculates the distance between $\mathbf{V}_i$ and its projection in the tangent plane with center $\mathbf{P}_j$ closest to $\mathbf{V}_i$. The sign corresponds to the orientation of the tangent plane so that it is positive or negative, depending on if $\mathbf{V}_i$ is positioned inside or outside the approximated surface.

The final step is to apply a *marching cubes* algorithm to extract an ISO-surface using the distance value of each cube vertex. Each cube with both external and internal vertices with respect to the ISO-value (positive and negative distance values), is examined and the contour is found within the tetrahedral decompositions of the cubic cell.

Part III Project Work

Chapter 4 Findings From Earlier Work

This chapter will highlight findings from the literature review in chapter 3 to motivate the selection of methods for further evaluation and implementation. While conclusions and assumptions in the previous chapter origins from already published articles this chapter is more focused on finding strengths and weaknesses in terms of the goals of this thesis.

4.1 Volumetric Approaches

The general volumetric approach works by examination of the space in which the sample points are captured. The volume is divided into a cubical grid. The surface is created by first approximating a local surface which is used to calculate the shortest distance each grid-point to the local surface. The ISO-surface is then extracted by visiting all the cubes that have both inside and outside corners with respect to the ISO-value.

As noted by both [ABK98] and [Hra03] volumetric approaches approximate the surface rather than interpolate it. This means that the surface will not run through the original points but instead through another set of generated points. This approximation has a low-pass filtering effect on the data, making small details disappear. Whilst this effect can be good on noisy data it is probably undesired in a medial application where the highest possible level of detail is required.

Faces of the final surface are defined by the vertices of the cubical lattice, not of the original sample points. This implies that a high resolution cubical lattice is required to preserve fine details. One consequence of this is excessive triangles in areas with low curvature which leads to a second challenge of volume rendering; how to select the grid resolution. Unless an automatic system for selecting a suitable grid resolution can be constructed it has to be chosen experimentally.

Even though volumetric approaches appears promising for applications where only the visual representation matters, they can not give strong enough guarantees of correctness to be used as base for this research.

4.2 Computational Geometric Approaches

The computational geometric approaches reviewed in this thesis are all based on the Voronoi diagram and its duality with the Delaunay triangulation. The Delaunay constraint is used as it minimizes the number of degenerated simplices in the triangulation.

4.2.1 α-Shape Based Algorithms

The α-shape based method offers stable means to extract surfaces from unstructured point clouds by removing Delaunay elements with circumsphere larger than a constant α. The concept of α-shapes is fundamental within surface reconstruction and most other computational geometric approaches are directly or indirectly based on its ideas. The obvious problem with this approach is to find a good α-value. If α is too small the surface will suffer from holes and if α is too big, small details will disappear. Working with sample points with non-uniform density an α-value to construct a correct surface might not exist.

The approach using Anisotropic Density-Scaled α-Shapes, as proposed by Marek Teichmann and Michael Capps, takes α-shapes closer to the goal of this research. Their method is less sensitive to non-uniform sampling density and to surfaces that are close to each other with respect to α. Unfortunately the method requires parameters to be defined manually and in this application a higher level of automation is desired. Due to these shortcomings this research will not be based on α-shapes.

4.2.2 γ-Regular Shapes

Attali's method provides strong mathematical guarantees of constructing the correct surface given a good sample as input. Dr. Lin et al provided in [LFT98] a study of the γ-regular shapes where several $\mathbb{R}^2$ data sets has been used to evaluate the method (fig. 13). One of their main conclusions was that the γ-regular shape method performs very well even for data sets with limited points scattered in the place of interest.

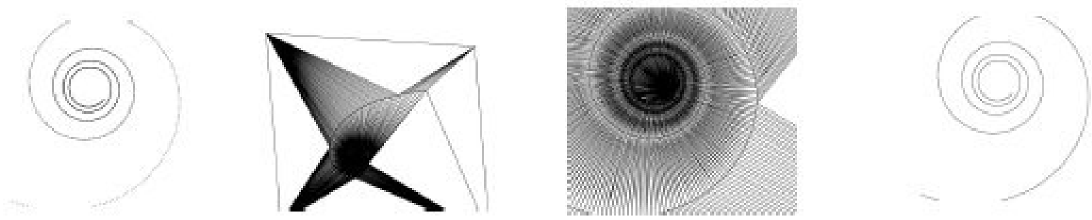


Figure 13: Left: 2D sample points. Middle left: Delaunay triangulation of sample points. Middle right: Close up of the Delaunay triangulation. Right: Surface reconstructed using Attali's γ-regular shapes. Image courtesy of [LFT98].

Due to the fact that, in $\mathbb{R}^3$, Delaunay spheres are likely to intersect the surface without being tangent to it, holes exist even in areas with good sampling density. To overcome the problem with holes Attali proposes two post-processing methods. Unfortunately neither of these comes with any guarantees of correctness. Given a refined approach to recover holes, Attali's approach is very likely to yield great results. It is furthermore fully automatic and straightforward to implement. The computation complexity comes down to calculating the Delaunay triangulation of the sample points and little extra work is required to find the normalized mesh.

4.2.3 The Power Crust

The Power Crust, developed by Amenta et al, is another approach with interesting properties. The approach is based on the approximation of the object's medial axis from which a watertight surface is defined in the intersection between inside and outside medal balls. In areas where the sample fulfills certain sampling criteria the approach is guaranteed to generate topologically correct surfaces. As pointed out by Varnuska et al in [VPK05] the major advantages of the Crust algorithms is that they are insensitiveness to non-uniform sampling density. The Crust algorithm suffers in big under- and oversampled areas, along boundaries, at sharp edges and it is also somewhat sensitive to noise. Many of these problems are better taken care of in the new approach, the Power Crust.

In terms of scientific applications the main concern with the Power Crust approach is that the final surface does not go through the original sample points. The offset between surface and sample point is small where the sampling quality is good but increases where curvature increases more than the sampling density. Another problem is that new, arbitrary points are introduced, resulting in a surface with many excessive points and triangles.

One consequence of the second pass weighted Delaunay triangulation is that poles with small weights (small radii of the medial ball) sometimes become redundant. This has a filtering effect which is necessary for the stability of the medial axis approximation where small perturbations in the sample can lead to large features in the medial axis.

Another issue with the Power Crust is that it is considerably more computationally expensive than other computational geometric approaches. The extra cost comes from the calculation of the second pass Delaunay triangulation of the poles. These are close to twice as many as the sample points, resulting in many extra Delaunay tetrahedra.

4.3 Geometric Skeletons

The concept of geometric skeletons has been touched by both Attali and Amenta. In this line of research the skeleton has the important property that it is a simplified representation of the geometry containing all its topological properties.

Attali's method introduces the abstraction of the geometric skeleton by taking the inner medial balls of the Voronoi diagram into consideration. As mentioned it is, in $\mathbb{R}^3$, common that Delaunay spheres are intersecting the surface instead of being tangent to it. This has two consequences for Attali's method. The first consequence is that holes will randomly appear in the surface, the second that the skeleton will contain a lot of sharp triangles and therefore needs to be simplified. This process of finding and simplifying the skeleton is described in more detail in [AM97].

As noticed in [MBR06] another good representation of the skeleton is the medial axis of the object. To overcome the problem with Voronoi centers being close to the surface and far away from the medial axis the Power Crust approach involves poles to find the medial axis transformation, the MAT. By doing so Amenta et al ignores medial axis balls that are close to the surface, yielding a quite stable skeleton.

Finding geometric skeletons is an active area of research and utilized in many practical applications including computer animations, motion tracking systems, computer vision and recognition. In terms of surface reconstruction skeletons can be seen as a minimalistic representation of the topological properties of the surface. This can be utilized to create watertight surfaces as shown by Amenta et al.

Chapter 5 Implementation

The core of this research is development and implementation of surface reconstruction algorithms. This chapter will provide details about the programming framework and the implementation of the two methods proposed by Attali and Amenta et al.

5.1 The Surface Reconstruction Framework

This research is based on a rather large code framework that is originally written by Dr. Fang-Pang Lin at NCHC. The most essential components in the framework for this research is the Voronoi diagram and the γ-Regular Shape implementation. The main motivation of using this framework is that the Delaunay triangulation has proved to be robust and of good quality even for large data sets.

5.1.1 Data Structures

To facilitate more in-depth discussions about the implementation it is necessary to at least outline the main data structures in the framework. The most low-level element used is the vertex which is simply described with coordinates in space and a weight (necessary for regular triangulations). As with all other basic elements the vertices are stored in dynamic lists. Other basic elements such as triangles and Delaunay tetrahedra are defined by 3 respectively 4 indices to the vertex list to avoid redundancy. Elements from these two structures also keep indices to the 3 respectively 4 neighboring elements. Each Delaunay tetrahedra also contains an extra vertex describing the coordinates of its circum-center as well as the radius of the corresponding Delaunay sphere. The properties of the most important elements are summarized in table 1.

Table 1: Properties of Basic Elements used in the implementation.

Data Structure	*Data type*	*Name[size]*	*Description*
Vertex	double	x, y, z	Coordinates in $\mathbb{R}^3$.
	double	weight	Weight of vertex.
Triangle	long	n[3]	Forming points, indices the vertex list.
	long	v[3]	Neighboring triangles, indices to the triangle list.
Delaunay Tetrahedra	long	n[4]	Forming points, indices the vertex list.
	long	v[4]	Neighboring tetrahedra, indices to the list of Delaunay tetrahedra.
	Vertex	vc	Voronoi center.
	double	R	Radius of Voronoi Sphere.

5.1.2 The Voronoi Diagram

The Voronoi diagram implementation used in this framework is based on the work done by Bowyer. The method starts with a set of vertices in a big bounding box forming a small set of Delaunay simplices, triangles in ℝ2, tetrahedra in ℝ3. Points are incrementally inserted into the triangulation. For each insertion a set of simplices will be invalidated in terms of the empty circum-hypersphere criterion as shown in figure 14. This set of simplices is replaced by a new set that are all locally Delaunay.

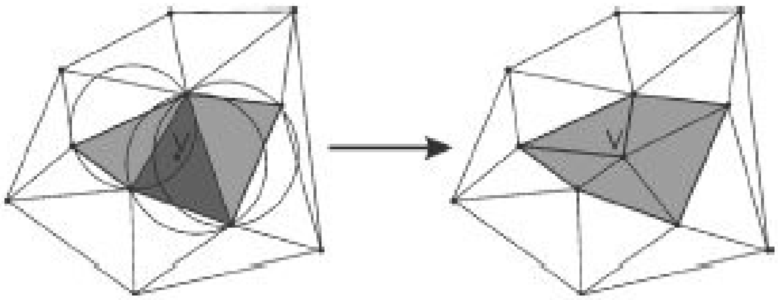


Figure 14: Bowyer Delaunay triangulations. Simplices invalidated by the inserted vertex v are replaced with new simplices formed with v. Image courtesy of [BSD+05].

It appears that given *one* invalidated simplex it is easy to find *all* invalidated simplices by. This is done by recursively checking neighbors of invalidated simplices for other invalidated simplices. This can be implemented efficiently using a small stack.

The challenge is to find the first invalidated simplex. In Dr. Lin's implementation this is done with a linear search which, for increased efficiency, is done in reverse order, starting with the last created simplex. In worst case this is no faster than $O(n^2)$ for three dimensions but in practice that is almost never the case ([ABK98]). And as noted by Dr. Lin et al in [LFT98] the implementation has proved to be both computationally effective and robust.

5.1.3 γ-Regular Shapes

Dr. Lin's implementation of the γ-regular shape is very straightforward. To find the normalized mesh the list of Delaunay simplices is traversed. For each face the angle between the two intersecting Delaunay spheres is evaluated. According to Attali's theory each face with an intersection angle **δ** (fig right) below 90^0 is a part of the normalized mesh.

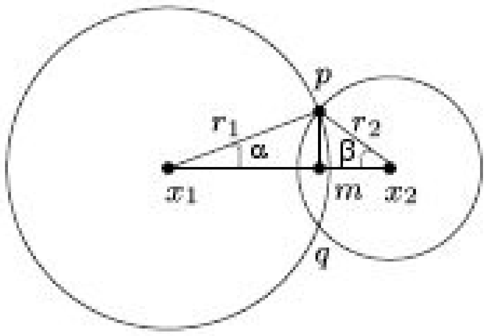


Figure 15: Finding the intersecting angle **δ**.

Following the notations in fig. 15 **δ** is calculated as following:

Per Attali's definition $\delta = \pi - \wedge(x_1,p,m) - \wedge(m,p,x_2) = \alpha + \beta$

$\delta < 90^0 \Rightarrow \cos(\delta) > 0 \Rightarrow \cos(\delta) = \cos(\alpha) * \cos(\beta) - \sin(\alpha)\sin(\beta) > 0$

$l = d(x_1,x_2) \ (= l_1 + l_2)$

$l_1 = d(x_1,m) = (r_1^2 - r_2^2 + l^2)/2l$

$l_2 = d(x_2,m) = (r_2^2 - r_1^2 + l^2)/2l$

$h = d(m,p) = sqrt(r_1^2 - l_1^2)$

$\cos(\alpha) = l_1/r_1$. $\cos(\beta) = l_2/r_2$. $\sin(\alpha) = h/r_1$. $\sin(\beta) = h/r_1$.

$\Rightarrow \cos(\delta) = (l_1/r_1) * (l_2/r_2) - (h/r_1) * (h/r_2)$

The only computationally expensive step of computing the γ-regular shape is to calculate the Delaunay triangulation of the sample points. The remaining step is to select faces that belong to the normalized mesh, an operation with complexity linear to the number of Delaunay tetrahedra.

5.2 File Format

One of the uses of this research is to build models to be visualized using NCHC's Tiled Display Wall (TDW). As mentioned, the TDW is a passive 3D stereo display meaning that it can display real 3D images for users wearing stereo, Polaroid, glasses. To power the TDW a software framework called Ivlsee has been developed by the NCHC crew. This software is built on the Visualization ToolKit, VTK, and supports many different stereo display modes as well as file formats. In this implementation the VTK file format has been used because it works well with Ivlsee as well as with other visualization frameworks. A basic VTK file can look as follows:

```
# vtk DataFile Version 2.0
Polygon Data Example
ASCII

DATASET POLYDATA
POINTS 316 float
 0.553709  -3.549647   1.515702
 0.329689  -3.545071   0.767544
-0.326118  -5.135679   1.360761
-0.065318  -4.809473   2.020700
 0.112076  -4.061206   2.156988
...

TRIANGLE_STRIPS 616 2464
3     225     218     224
3     315     314      13
3     213     206     212
3     206     199     205
3     184     178     177
...
```

Basically the file contains three sections; *the header*, *point data* and *triangle data*. The header essentially tells the program that reads the file what type of file it is and what type of data that is described in the file. The second section states that the following data is a fixed number of points (here 316) and that each data field is a floating number, followed by the coordinates in x,y,z of each point. The last section describes the set of polygons (here triangles) defining the surface. Each line starts by defining how many vertices the polygon consists of, followed by as many indices to the points in the list from the previous section. The obvious reason for describing surfaces like this is to avoid redundancy of having multiple points describing the same coordinate.

5.3 Implementation Of The Power Crust

Implementing the Power Crust algorithm into Dr. Lin's framework involved major efforts. In this chapter the choice of reimplementing the algorithm is motivated as well as some of the challenges of the implementation discussed.

5.3.1 Motivation

The Power Crust Algorithm source code by Amenta et al is released in the public domain, promoting the option to do the evaluation without extensive implementation. Nonetheless the Power Crust algorithm was reimplemented into the existing framework, a process that required restructuring most data structures and functions presented by Amenta et al. This rather lengthy process is motivated as follows.

- Dr. Lin's framework includes a Delaunay triangulation implementation that has proved to be robust for triangulation of large data sets. The original Power Crust implementation, on the other hand, is based on a Delaunay triangulation called "*The Hull*", written by Ken Clarkson. According to Amenta et al this triangulation is not stable for large data sets, therefore the Hull implementation needed to be substituted.
- An early thought was to compare the Crust algorithm with Attali's γ-regular shapes. Based on the same Delaunay triangulation aspects such as initial mesh quality and efficiency of mesh generation could be ruled out in such a comparison.
- A mutual computational ground would facilitate research and development of a hybrid solution.

5.3.2 Regular Triangulations

As mentioned the Delaunay triangulation is the most central component in most computational geometrical approaches, so also in the Power Crust. The Power Crust is furthermore dependent on a second, weighted, pass of Delaunay triangulation. As the existing Delaunay triangulation did not support weights it was necessary to adjust the method. In terms of code, this was a rather simple process that could be done with about 50 lines of code. However, the theory behind it is quite dense and not very intuitive. For example it becomes necessary to handle circum-spheres with negative radius, which is not easily imagined. Therefore even when grasping the regular triangulations in theory, it is hard to understand how they work in practice.

Before converting the existing Delaunay triangulation into weighted manners a substantial amount of research papers on the subject were studied, trying to establish necessary rules to create regular triangulations. Especially the papers by Edelsbrunner et al ([ES92]) and Vigo et al ([VPC02]) were used to formulate rules to keep a regular triangulation during insertion of new points. The following rules are defined using **S** and **T** to denote two neighboring Delaunay d-simplices of which **z** is the orthogonal center of **S**.

Regular triangulation rules:

- **S** is called *globally regular* iff $||\mathbf{q}-\mathbf{z}||^2-\mathbf{w}_z > \mathbf{w}_q$ where **q** is any other, non-redundant, point in the triangulation.
- Let Δ denote the (d-1) simplex shared by **S** and **T,** let **q** denote the vertex of T that *does not* belong to Δ. Then Δ is said to be locally regular iff $||\mathbf{q}-\mathbf{z}||^2-\mathbf{w}_z > \mathbf{w}_q$ and **S** is also locally regular iff all $\Delta\in$**S** are locally regular.
- Any point that does not invalidate any d-simplex is called redundant and is not included in the triangulation.

In practice only the local regularity of a simplex is evaluated. To perform global checks would be very computationally inefficient. The third rule points out an important contrast to the unweighted Delaunay triangulation in which every point will generate a Voronoi cell and therefore be a part of the triangulation. In a weighted environment, however, some points will not define any power cell as, in terms of power distance, the point will not be the closest point to any other arbitrary point. Therefore it will be excluded from the triangulation.

5.3.3 Floating point errors

Within the computational geometrical domain it is important to keep track of floating point round-off errors. The problem appears because the *double* precision in C/C++ is not always exact enough. For example the accuracy in the decision if a point lies within a circumsphere or not is crucial for the correctness of the Delaunay triangulation, especially in the approach suggested by Bowyer. Due to loss of precision when calculating the circum-centre, a point can be wrongly classified as either inside or outside. The results of these errors appear completely random which makes them hard to trace.

5.3.4 Findings

The efforts of implementing the Power Crust method into Dr. Lin's framework did not reveal any surprising details of the Power Crust that have not previously been published by the authors of the method. The efforts did, however, give in-depth knowledge of the Crust algorithm. This knowledge has been necessary to evaluate and, more importantly, further develop the algorithm.

Chapter 6 The Hybrid Solution

The contribution of this research is the introduction of a new method to reconstruct three dimensional surfaces. The approach is a hybrid solution based on Attali's γ-regular shapes and Amenta's Power Crust. The approach is intended to be used within scientific applications as it is based on two methods that come with strong mathematical guarantees under certain sampling criterion.

6.1 Background

The resulting surfaces produced by the two methods proposed by Attali and Amenta et al correspond well to expectations raised in the related documentation. As expected neither of the methods is able to generate correct surfaces in three dimensions.

Attali's method produces surfaces that go through the sample points and that are topologically correct wherever its sampling criterion is fulfilled. However the method is incapable of generating watertight surfaces even in areas that fulfill the sampling criteria stated by Attali. This is due to Delaunay spheres that may appear on the surface without being tangent to it. The result of such Delaunay spheres is holes in the surface.

Amenta's method, on the other hand, produces topologically correct, watertight surfaces under certain sampling conditions. As shown these surfaces are very good visually speaking, however they do not suit well for scientific applications since they do not go through the sample points. Furthermore the resulting surface contains many excessive vertices and faces.

The novel hybrid approach proposed here combines the two methods. Attali's method is based on the assumption that if the sampling density is “good enough” the Delaunay triangulation will in fact contain the normalized mesh of the surface. The problem is that the sampling condition is likely to fail in some areas, in particular the ones with high curvature. The underlying thought of the hybrid approach is to recover these areas using the geometric skeleton of the sample.

The Crust provides a stable approximation of the skeleton (the medial axis). Given that the skeleton contains all the topological properties of the object and that the Crust surface is watertight it seems suitable to use the Crust surface to recover holes in the γ-regular shape.

6.2 Sampling Criterion

For scientific applications it is essential to state when the method can be expected to generate correct surfaces. The sampling criterion of the hybrid method is derived from the two methods on which it rely.

Attali's sampling criterion is very precise. As explained in chapter 3.2.3 one of the main findings in Attali's paper is that she proves that, given a sample with sampling path $\epsilon < \sin(\Pi/8)\gamma$, the Delaunay triangulation contains normalized mesh of the surface. Here ϵ corresponds to the radius of the smallest possible sphere, centered on the surface so that the sphere will contain at least one sample point. γ corresponds to the smallest sphere that is both morphologically open and closed in respect to the sample itself (fig.).

6.3 Implementation Details

This chapter provides information on how the proposed approach has been implemented. The algorithms presented are of course developed to work in Dr. Lin's framework but with few exceptions they are constructed in general manners and should be straightforward to implement in any framework.

Essentially the algorithm works by finding holes on the γ-regular shape and then creating and applying patches from the Crust surface as outlined below:

Algorithm overview:

- Calculate the γ-regular shape and the Crust of the sample points.
- Locate holes of the γ-shape.
- Create a patch from the Power Crust to cover each hole of the γ-shape.
- Fit the patch onto the γ-shape.

6.3.1 STL Data Structures

The implementation of the hybrid solution relies heavily on data structures provided by the Standard Template Library, STL, which is available in standard C++. These data structures are used because they are computationally efficient and very straightforward to use. Below is a short explanation of the data structures used in the implementation. More information about the STL can be found in [STL].

- **STL::Multimap** is a sorted associative container in which each data element is sorted on an associated *key value*. The multimap allows multiple entries with the same key value.
- **STL::Set** is a sorted associative container in which the data is sorted on its own value. The set does not allow multiple entries with the same key value.
- **STL::Stack** is a Last-In-First-Out data structure.

6.3.2 Locating Boundaries

The procedure of locating boundaries is rather straightforward and done by traversing the list of triangles. By nature each edge can be shared by at most two triangles and if defined by only one triangle the edge must be on the boundary of the surface.

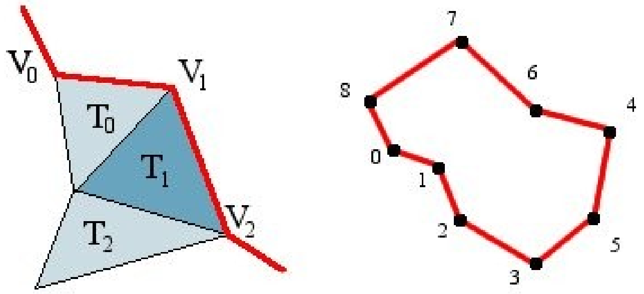


Figure 16: Left: Triangles close to a boundary. The red line mark the boundary of the surface. Right: A closed ring on the boundary defined by 9 vertices.

A multimap is used to store information about triangle edges, namely the indices to the two vertices defining the edge as well as the index to the triangle itself. When inserted into the multimap the smaller vertex-index is used as key value.

For every triangle each edge is taken into consideration. The multimap is searched for the edge, if the edge is found it is deleted from the map. Otherwise it is inserted into the map. This way every edge that is shared by two triangles will first be inserted and then deleted from the map. Edges that are left in the map after the triangles list is traversed is, per definition, on the boundary of the surface.

It is necessary to mention that the search for an edge is not strictly done in logarithmic time. As a vertex is a member of at least two edges it becomes necessary to implement a small linear search on top of the logarithmic search of the multimap. This causes the search cost to be $O(n^2)$ in worst case. In practice, however, the cost is close to $O(\log n)$ as the number of edges of which a vertex is a member is small.

The entire procedure is $O(n \log n)$ where *n* is the number of triangles on the surface. The result of the method is shown in figure 17 where boundary triangles are marked with blue color.

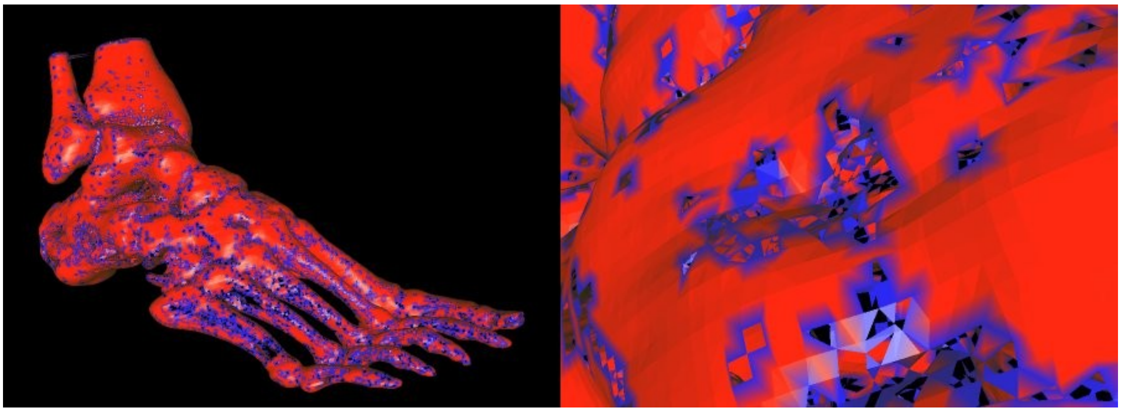


Figure 17: Boundary triangles of a γ-regular surface.

It becomes necessary to extract each ring of boundary edges as shown to the right in figure 16. These rings are found by recursively extracting edges from the multimap, starting at the first edge in the map. As edges are indexed by their *lower* index value of the vertices the following edge can be found by searching the map using the *higher* index value of the current edge. In the illustrated example, starting with edge (V_0,V_1) the following edge, (V_1,V_2), is found by searching the map using V_1 as needle. The process continues until the initial vertex V_0 is reached for when the ring is closed.

Unfortunately this method sometimes ends up in a deadlock. In the given example the algorithm will fail as it reaches edge (3,5) as the following edge, (4,5), cannot be found in the map when searching with key value 5.

The implemented solution to this deadlock is to keep a copy of the multimap where each edge is indexed by its higher index instead of its lower index. The solution is not optimal but has worked well in practice.

6.3.3 Creating Patches

The process of creating patches involves finding a patch on the crust that corresponds to and covers the hole of the γ-shape. Because the sometimes random correlation between the boundary of the γ-shape and the Crust surface this procedure turned out to challenging to generalize.

An early observation was that the offset between the two surfaces often is increased in areas with low sampling density/curvature rate which makes this process challenging. When extracting a patch it is important to make the patch big enough to completely cover the hole. It is, however, as important to keep the patch as small as possible as different parts of a patch otherwise can grow together in areas with complicated hole structures.

The approach tested in this research is done in two steps:

- Find crust triangles that covers the boundary of the γ-shape.
- Repeatedly add Crust triangles on the inside boundary until the patch is completely watertight.

The procedure starts by finding crust triangles that completely cover the boundary. For each boundary edge, $\mathbf{e}_i$, defined by the points $\mathbf{p}_s$ and $\mathbf{p}_t$ the goal is to find Crust triangles so that the edge is completely covered. This is done with a walking procedure.

The first step is to define a plane **P** in which the edge lies, so that the plane is perpendicular to the triangle $\mathbf{t}_i$ (of which $\mathbf{e}_i$ is an edge). The plane is used during the triangle walk to choose only triangles with vertices on each side of **P** . The motivation of this is that it is necessary to make sure that the entire edge is covered by Crust triangles. Without this constraint any walking method could deviate from the straight path leaving uncovered parts of the edge. The idea is shown in figure 18.

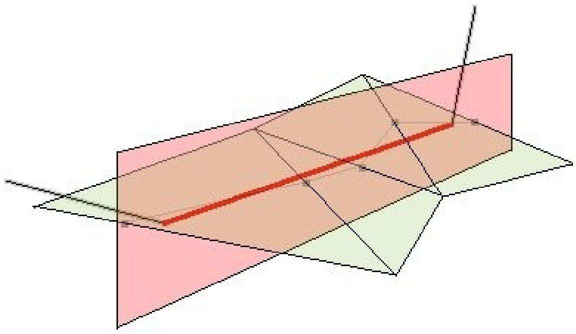


Figure 18: Triangulation along an edge. The thick red line illustrates an edge on the boundary with its perpendicular plane. The Crust triangles that covers the edge is marked in light-green.

The perpendicular plane is straight forward to find. If $\mathbf{p}_u$ is the third vertex of $\mathbf{t}_i$ the normal of the triangle, $\mathbf{n}_t$, is given by the cross product between the two vectors $\mathbf{v}_0=(\mathbf{p}_t-\mathbf{p}_s)$ and $\mathbf{v}_1=(\mathbf{p}_u-\mathbf{p}_s)$. The plane is defined in the so-called *Hessian normal form* as that enables a very efficient way of determining on which side of the plane an arbitrary point is located. The only data needed to define the plane on this form is the plane normal, $\mathbf{n}_p=\mathbf{n}_t \times \mathbf{v}_0$, and the offset from the origin, given by scalar product between $\mathbf{n}_p$ and the vector from the origin to the point $\mathbf{p}_0$.

The triangle walk is started with the, with respect to point $\mathbf{p}_0$, closest point on the crust for which the one-ring neighborhood of triangles is located. From this set of triangles we select the triangle with the smallest distance between centroid and $\mathbf{p}_1$ given it has vertices on both sides on **P**.

The walk is simply done by repeatedly visiting the neighboring triangle that has vertices on each side of **P** and stopped when the two vectors between the centroid of the current triangle and the two endpoints of the edge have a positive projection on each other.

When the walking procedure is completed for each edge the boundary *should* be completely covered by Crust triangles. Here a STL:set is used to store triangles on the boundary because it provides a simple way to make sure that no triangle exist twice in the container.

Before applying the patch to the surface the patch needs to be made watertight. This procedure is similar to the one proposed in [Att97] where Delaunay triangles are recursively added to the surface if they have at least two edges on border of the surface. In this approach, triangles from the Crust surface are added to the patch in similar manners. The procedure is repeated until no more valid triangles can be found.

6.3.4 Applying Patches

The final step is to apply the patch to the γ-shape. In general patches are slightly larger than the hole and contain a higher triangles density than the surrounding surface. The goal is to cover the hole without increasing the number of vertices in the surface. The attempted solution is to apply the patch using a variation of a Laplace smoothing operation as described in the following formula.

$$p = p + \frac{a}{n} * \sum \left(\frac{1.0}{(|d_i - d_{min}| + 1.0)^{16}} * (p - bp_i) \right)$$

The smoothing operation is constructed so that each vertex on the patch, **p,** is attracted by each point, $\mathbf{bp_i}$, on the boundary. The force in which **p** is attracted by a boundary point is determined by a constant, **a**, the number of points on the boundary, **n**, and distance, $\mathbf{d_{i,}}$,between the two points. The formula is adjusted so that boundary points close to **p** will have an larger attraction-force than points further away, this is done using an exponent that has been chosen experimentally.

The result of this operation is that the points on the patch will be moved slowly towards the closest points on the border. Eventually each point will get close enough to one point on the border to be merged. When this happens there is a chance that some triangles, of which both target points are members, become degenerated and have to be removed.

To make sure that the method is free from deadlocks the total distance between a point on the patch and its closest point on the boundary is used as an error measurement. Normally the error drops down to zero for when the patch is successfully applied. In rare occasions, however, the error does not decrease during an iteration and the process skips to applying the next patch.

Part IV Results and Discussion

Chapter 7 Results

This chapter will present results from the implementation of Attali's γ-regular shape, Amenta's Power Crust and the novel hybrid solution. Each method is evaluated in terms of visual results and correctness, performance and mesh redundancy. This chapter is aimed to show the strengths and weaknesses of each method and will be used as ground for the conclusions and discussions in following chapters.

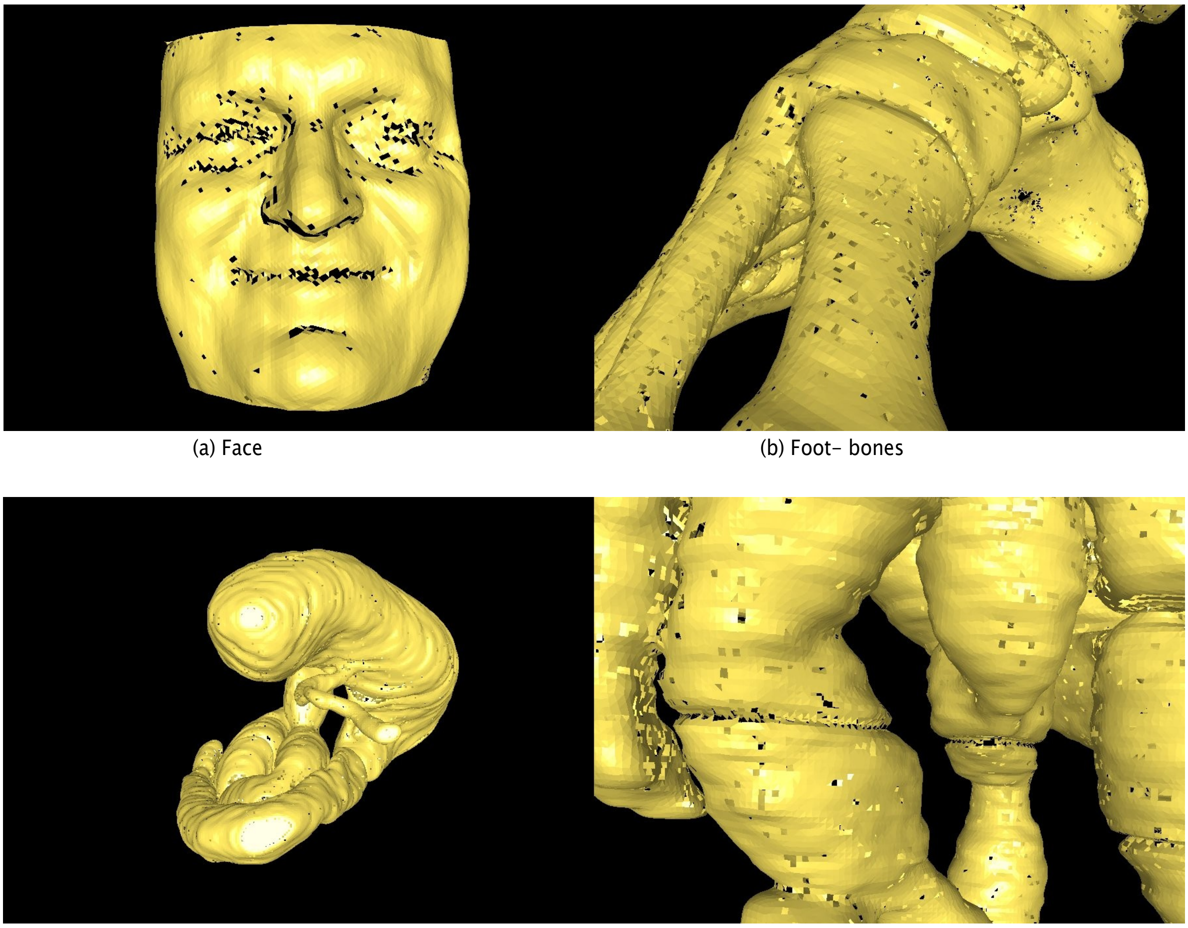
7.1 Visual Results And Scientific Adequateness

In this chapter each method will be discussed in terms of the visual result of the surfaces created using the method as well as how adequate the method is in pure scientific applications.

7.1.1 γ-Regular Shapes

Literature stated that Attali's method yields very accurate results in $\mathbb{R}^2$. The approach is unfortunately not as stable in $\mathbb{R}^3$ where it suffers from holes. One important observation, demonstrated in figure 19, is that holes in the γ-shape seem to appear quite randomly, but, are more likely to appear in regions with low rate of sampling density and curvature. The random holes are caused by properties of the three dimensional Voronoi diagram. That fact that holes are more common in areas with high curvature is natural as the method is designed having γ-Regular samples in mind. The method will therefore fail when the sampling density is too low.

Holes can cause visual cluttering, making it hard for the user to recognize and orient through the artificial environment. This is clearly unsuitable for most applications. As stated by Attali the approach is therefore not suitable to create 3D surfaces in its pure form. She is also encouraging development of methods to repair the holes to make her method more usable in $\mathbb{R}^3$.



(a) Face (b) Foot- bones

(c) intestines of a mouse (d) Close-up of the intestines

Figure 19: Example surfaces created with using Attali's γ-regular shapes. Notice that holes appear both in regions with high and low sampling density/curvature rate in (d).

7.1.2 Power Crust

The Power Crust algorithm was implemented in the framework written by Dr. Lin at NCHC. This framework is based on an incremental Delaunay triangulation similar to the one proposed by Bowyer. In contrast the original implementation is based on a flipping algorithm called "The Hull", written by Ken Clarkson. The main difference in the two approaches is that flipping algorithms recursively flip tetrahedra that are non-locally Delaunay until they fulfill the Delaunay criteria. This can be done without calculating the circumsphere of the tetrahedra. Flipping approaches are therefore less sensitive to floating point round-off errors.

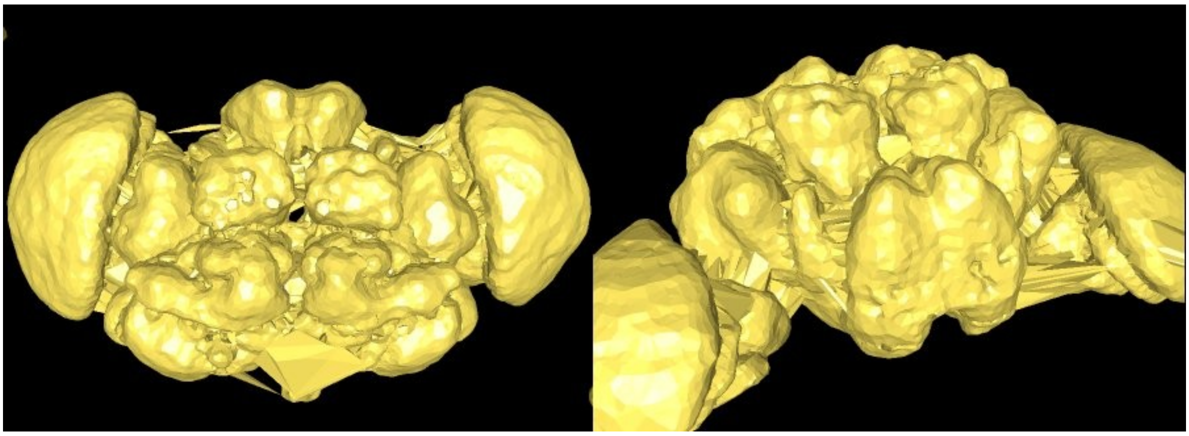
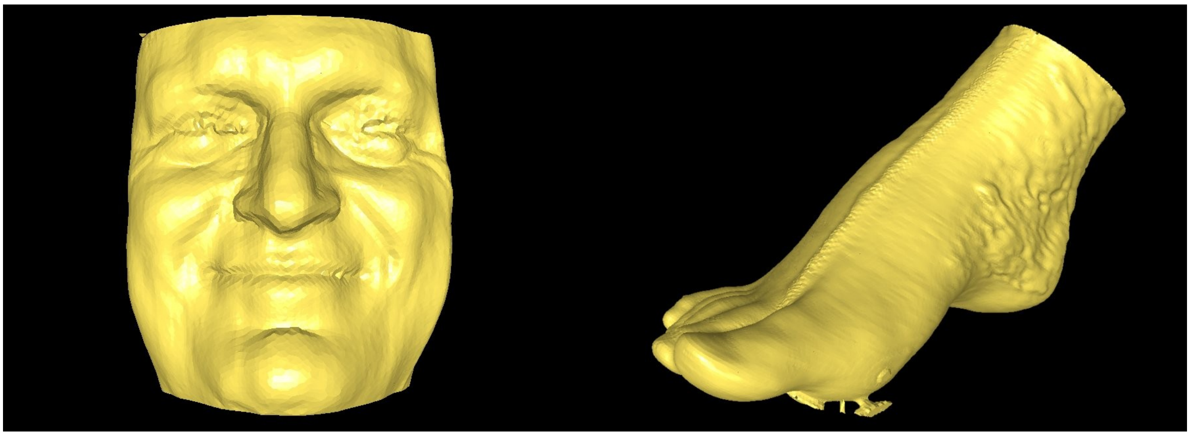


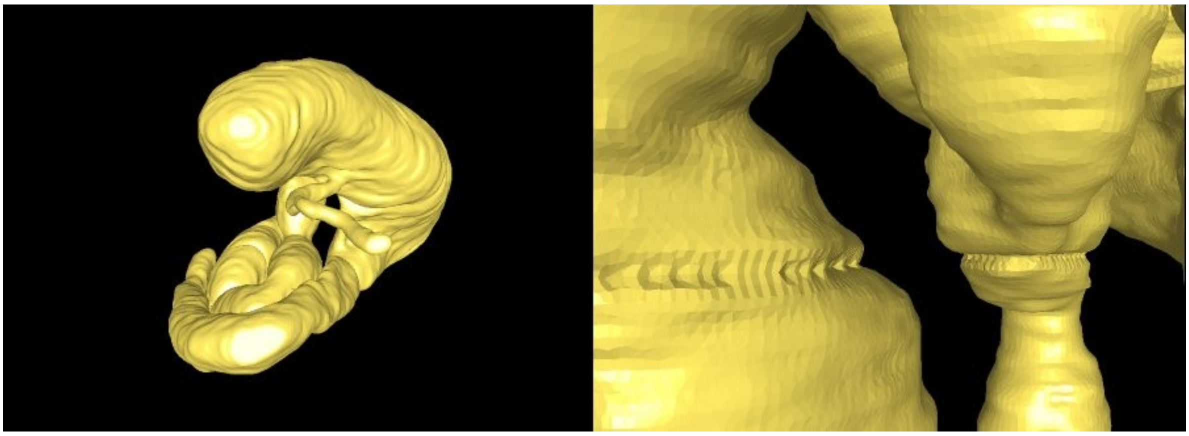
Figure 20: The surface of a sample set acquired through cryosections of a brain from a fruit fly. Left: Surface generated with Amenta's implementation of the Power Crust. Right: Surface generated with the new implementation. Notice that the surfaces is generated from the same sample set and that both implementations have similar problems with incorrect triangles in certain areas.

A data sample of the brain of a fruit fly is used to compare the visual results between the new and the original implementation of the Power Crust. This sample is used because it has a quite low sampling density/curvature rate. It is therefore not a “good sample”. The resulting surfaces are demonstrated in figure 20 with Amenta's implementation to the left and the new implementation to the right. In most areas the surface appear nicely reconstructed and topologically correct. Unfortunately, and a little surprisingly, the method suffers from problems in certain areas. It is likely that the origin of this problem is that some poles are wrongly labeled. This problem is present in both implementations but does only appear in rare occasions, for a few samples.

In the general case, however, the Power Crust algorithm constructs surfaces that are extremely pleasant to look at. A few examples is demonstrated in figure 21. The surfaces are watertight as promised and they appear topologically correct. In particular it is satisfying to see how well the method performs in complicated areas with low rate of sampling density/curvature. The reader is advised to compare the visual result in fig. 21d with the corresponding γ-shape visualized in fig 19d.



(a) Face (b) Foot

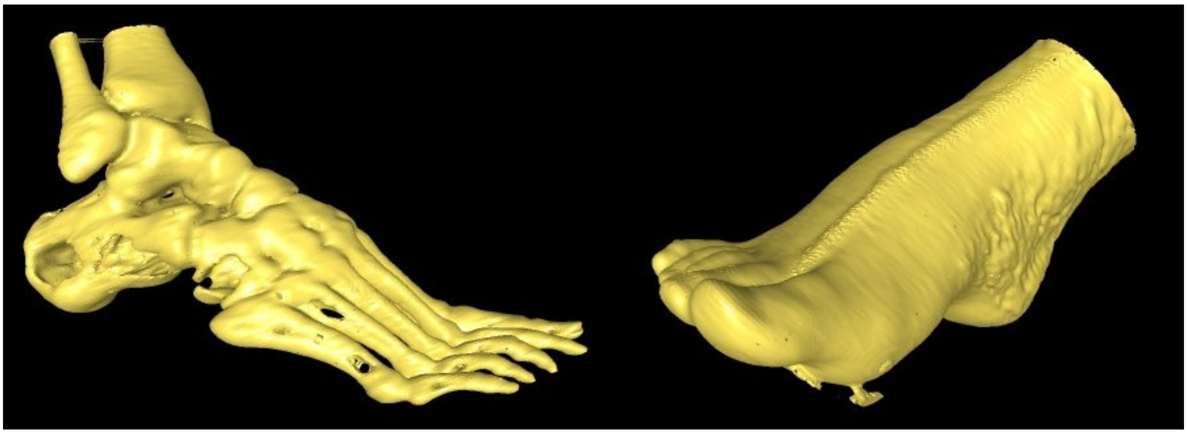


(b) Intestines of a mouse (d) Close-up of the intestines

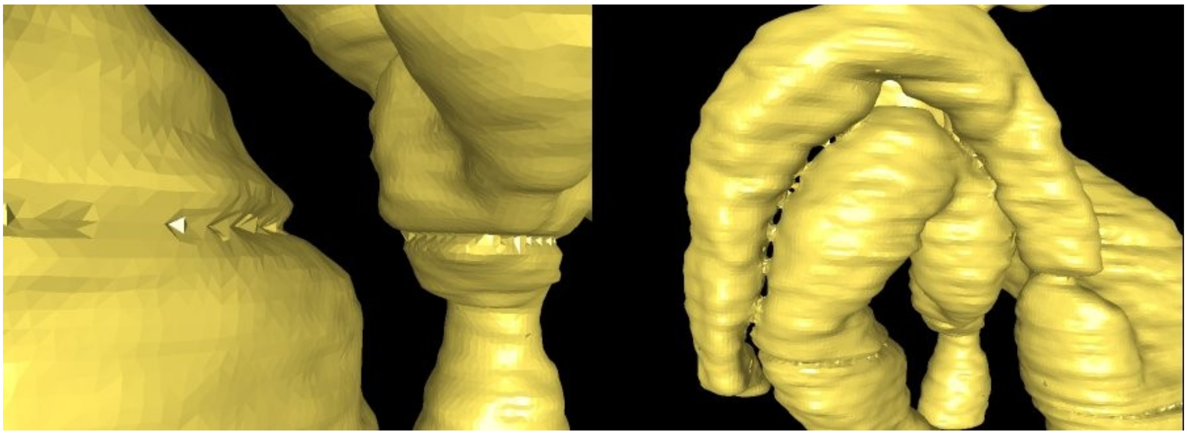
Figure 21: Example surfaces created with using Amenta's Power Crust. The method creates smooth, watertight surfaces that are, visually speaking, very nice. Notice that the errors on the sole of the foot in (b) are due to errors in the data set and not in the algorithm itself. The method performs well even in areas with low rate between sampling density as in (d).

7.2 The Hybrid Solution

The novel hybrid approach developed during this research has yielded quite good visual results, demonstrated in in figure 22. Simple holes are, with few exceptions, repaired in topologically correct manners and even complicated areas are repaired with satisfying results. Again the reader is advised to compare figure 22c with earlier figures of the same sample.



(a) Foot, bone (b) Foot, skin



(c) Intestines of a mouse , close-up (d) Intestines of a mouse

Figure 22: Example surfaces constructed with the novel hybrid approach developed during this research. The algorithm performs quite well in areas with low curvature as well as in more complicated areas (c).

7.3 Surface Properties

The size of the surface in terms of number of vertices and faces is very important as it directly effects how long time it takes to load the surface and if it can be seamlessly used in an interactive environment. Table 2 contains properties of surfaces created with the three surface reconstruction algorithms in this evaluation. Notice that the number of sample points for each surface is the same as the number of vertices in the γ-Regular Shape.

Table 2: Properties of surfaces created using the different surface reconstruction approaches

Data Set	*Method*	*Vertices*	*Faces*	*File Size (mb)*
Brain of a fruit fly				
	γ-Regular Shapes	34913*	52137	4.4
	Power Crust	185683	376515	29.8
	Hybrid Solution	40582	92348	7.1
Mouse intestines				
	γ-Regular Shapes	78452*	143205	11.6
	Power Crust	453706	907552	72.2
	Hybrid Solution	85028	160693	12.9
Foot, bone structure				
	γ-Regular Shapes	107238*	184795	15.1
	Power Crust	665523	1331119	105.9
	Hybrid Solution	120888	224897	18.1
Foot, skin				
	γ-Regular Shapes	227073*	409637	33.2
	Power Crust	1379879	2758825	219.6
	Hybrid Solution	247682	466431	37.5

* This is the same as the number of points in the data sample

As shown in the table Crust surfaces contain huge amounts of excessive vertices and triangles, making the data files large. On a normal personal computer it is possible, but hardly convenient, to load data files larger than 100MB into memory. The hybrid approach introduces a relatively small number of new points and triangles. The triangles are needed to cover the holes, the extra vertices, however, are not. In theory the number of extra vertices should be close to zero.

7.4 Computational Efficiency

An important aspect of surface reconstruction algorithms is how efficient they are. The process of creating a surface using the hybrid solution involves calculating both the γ-shape and the Crust surface. The measurement provided here is only based on one sample, the mouse intestines. As discussed in chapter 9 some processes implemented in this work need to be optimized, therefore a more in-depth evaluation is not be interesting at this moment. This measurement is, however, an indication on which subprocesses are the most time consuming. The sample consists of 78444 sample points of non-uniform density.

Table 3: Execution time for different processes of the hybrid surface reconstruction approach.

Process	*Duration (sec)*	*Total Duration (sec)*	*%*
Reading points from file	0.42	0.42	>0.01
Calculating Delaunay triangulation	680.32	680.74	22.9
Creating the normalized mesh (γ-shape)	1.20	681.94	>0.01
Creating The Power Crust			
Generating Poles	4.20	686.14	>0.01
Calculating Regular Triangulation	1935.10	2621.24	65.2
Building Power Diagram	2.12	2623.36	>0.01
Labeling Poles	1.30	2624.66	>0.01
Creating the Crust Surface	6.22	2630.88	>0.01
Creating Hybrid Surface *	337.08	2967.96	11.4

* Computation time for the solution when the fitting part is done without the Laplace smoother.

As shown in the table 3 above the total execution time consists of 3 posts that together constitute 99.4% of the execution time of 49.5 minutes. In total 88.1% of the time is spent on creating Delaunay triangulations and to be more specific finding initial tetrahedra that are invalidated by newly inserted vertices. This result shows the importance of an Delaunay triangulation that is efficient to compute for large data sets.

Chapter 8 Conclusions

The aim of this thesis was to develop a method to reconstruct surfaces based on sets of point data. This problem is complex as the input to the method is an unstructured, three dimensional, point cloud that may be subject to noise. The method was also required to be capable of handling large data sets, produce topologically correct and visually acceptable surfaces, be computationally efficient and finally to be semi or fully automatic.

The problem was approached by implementing two existing surface reconstruction algorithms from which a hybrid solution was developed. The three approaches was evaluated using medical data sets.

The largest data set used with all the three methods contain about 250.000 points. This is less that the original goal but must be considered as a fairly large data set. The γ-shape and Power Crust surfaces of a data set roughly twice as big has been successfully created. Unfortunately the data file of the Crust surface became more than 500mb in size, which is too much to load for any of the computers used in this research.

The time complexity of the three implemented methods is highly depend on the complexity of calculating the Delaunay triangulation. The current implementation could be improved by speeding up the search in which invalidated Delaunay simplices are found. In its current state the method is not optimized in terms of computational efficiency.

The quest to find an algorithms to construct *correct surfaces* continues. The works of Attali and Amenta are well-studied and both give deep mathematical background to their approaches. Both authors have also received much kudos from the surface reconstruction society. During this research errors in the Power Crust approach were noted, but only for samples with poor sampling density. Otherwise the two approaches perform as expected. Attali's theory is used as a base in the hybrid solution developed in this work. The idea was to repair holes on the γ-shape using topological properties from the Power Crust surface. In practice the method works quite well but unfortunately errors randomly appear. To be of real value the method would need to be better defined in theory so that, given certain sampling criteria, guarantees of correctness can be provided. In its current state the algorithm can only be used to create, visually speaking, acceptable surfaces.

The last requirement was fully satisfied. It is possible to manually set some parameters of the Power Crust algorithm but this was not needed during the research. Attali's method do not need manual input. The hybrid solution requires a good representation of the γ-shape and the Crust surface. It does not, however, need any manual input.

Chapter 9 Discussions

This chapter is dedicated to more open and personal thoughts about this research and serves to promote further research within the line of this research.

9.1 The Approach

Working in the geometrical domain good performance of the Voronoi diagram is the essential for good performance of the application. Theoretically speaking the Voronoi diagram can be defined very precisely but in practice different implementations are very likely to produce different – yet correct – diagrams. When benchmarking geometrical surface reconstruction methods it is important to take the quality of the triangulation into consideration.

Two existing surface reconstruction algorithms were implemented in the same Delaunay framework. The underlying thought was that, because of this, performance and quality aspects of the triangulation could be overseen when comparing the methods. Unfortunately the time spent on the implementation of the Power Crust algorithm exceeded all expectations and one could argue that it would have been more efficient to use the existing implementation instead. Despite this I believe that the efforts of the reimplementing the Power Crust paid off well as it facilitated the research of the novel hybrid approach.

The hybrid approach was developed as none of the implemented methods fully corresponded to the goals of this research. This part of the research had more novelty value. The task seemed somewhat trivial in the beginning but turned out to be quite challenging, mostly because of the random correlation between the surfaces in areas of poor sampling density. Time was too limited to make the new solution completely stable but I think that further research within the same line could yield very nice results.

9.2 Future Improvements

I find the results of the hybrid solution promising and I do believe that efforts in improving my work could be very fruitful. The overall goal is not only to create, visually speaking, pleasuring surfaces. It is also important to be able to state under what sampling criteria the method can be expected to work. At the moment no such guarantees can be given, therefore it is not yet suitable in scientific applications.

The weakest link of the hybrid solution is the process when patches are applied onto the γ-shape. The smoothing algorithm lacks both in robustness and in the topological results it provides. I believable, however, that this process could be improved quite easily. The process of creating patches also needs to be improved, especially in the way patches are made watertight. This fails for some patches. The process of finding edges is, however, quite robust.

In terms of optimizing the efficiency of the implementation it would be interesting to investigate what results can be achieved using alternative searching methods when searching for the first invalidated Delaunay tetrahedra. Devillers et al. propose the “stochastic visibility walk” in which, given a starting triangle or tetrahedra, the repeatedly walks towards the point until it is found. The method is, according to the authors, easy to code, performs well and does not suffer in case of degenerated simplices [DPT01].

Part V References

Bibliography

Publications

[ABK98] N. Amenta, M. Bern, M. Kamvysselis. A New Voronoi-Based Surface Reconstruction Algorithm. ACM Siggraph 1998 (pp. 415-421).

[ACK01] N. Amenta, S. Choi, R.K. Kolluri. The Power Crust. ACM Symposium on Solid and Physical Modeling archive Proceedings of the sixth ACM symposium on Solid modeling and applications. 2001.

[ACK201] N. Amenta, S. Choi, R.K. Kolluri. The Power Crust, Unions of Balls, and the Medial Axis Transform. Computational Geometry, Volume 19, Number 2, July 2001 (pp. 127-153).

[Att97] D. Attali. r-Regular Shape Reconstruction from Unorganized Points. ACM Symposium on Computational Geometry 1997 (pp. 248-253).

[AM97] D. Attali, A. Montanvert. Computing and Simplifying 2D and 3D Continuous Skeletons. Computer Vision and Image Understanding. Vol 67. No. 3, September. 1997 (pp. 261-273)

[DPT01] O. Devillers, S. Pion, M. Teillaud. Walking in a triangulation. ACM Symposium on Computational Geometry. 2001 (pp. 106 - 114).

[BSD+05] T. Beyer, G. Schaller, A. Deutsch, M Meyer-Hermann. Parallel dynamic and kinetic regular triangulation in three dimensions. Computer Physics Communications, Volume 172, Issue 2, 2005 (pp. 86-108).

[EM94] H. Edelsbrunner, E. P. Mücke. Three-dimensional Alpha Shapes. ACM Transactions on Graphics. 1994.

[ES92] H. Edelsbrunner, N. R. Shah. Incremental topological flipping works for regular triangulations. ACM Symposium on Computational Geometry 1992 (pp. 43-52).

[HDD+92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonals, W. Stuetzle. Surface Reconstruction from Unorganized Points. ACM *Siggraph* 1992 (pp. 71-78).

[Hra03] J. Hrádek. Methods of surface reconstruction from scattered data. Technical Report No. DCSE/TR-2003-02. March 2003.

[LFT98] Data Warehousing and Meshing for large Scale Dataset from Surveyed Profiles of Watercourses. Computational Geometry, 10, 1998 (pp. 239-247).

[MBR06] C. Menier, E. Boyer, B. Raffin. 3D Skeleton-Based Body Pose Recovery. Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization and Transmission. 2006.

[Men01] R. Mencl. Reconstruction of Surfaces from Unorganized Three-Dimensional Point Clouds. Dissertation zur Erlangung des Grades des Doktors der Naturwissenschaften der Universität Dortmund. Dortmund 2001.

[TC98] M. Teichmann, M. Capps. Surface reconstruction with Anisotropic Density-Scaled Alpha Shapes. IEEE Visualization 1998 (pp. 67-72).

[Pap04] L. Papaleo. Surface Reconstruction: Online Mosaicing and Modeling with Uncertainty. Ph.D. Thesis in Computer Science, University of Genova. March 2004.

[VPC02] M. Vigo, N. Pla, J. Cotrina. Regular Triangulations of Dynamic Sets of Points. Computer Aided Geometric Design archive, Volume 19. Issue 2. 2002 (pp. 127-149).

[VPK05] M. Varnuska, J. Parus, I. Kolingerova. Simple Holes Triangulation In Surface Reconstruction. Proceedings of Algoritmy 2005 (pp. 280-289).

[ZOF01] H. K. Zhao, S. Osher, R. Fedkiw. Fast surface reconstruction using the level set method. Variational and Level Set Methods in Computer Vision. Proceedings. IEEE Workshop. 2001 (pp. 194-201).

Internet Resources

[LR] R. Lattuada, and J. Raper. Applications of 3D Delaunay triangulation algorithms in geoscientific modelling. http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/sf_papers/lattuada_roberto/paper.html. Accessed 7th of January 2007.

[STL] SGI. Standard Template Library Programmer's Guide http://www.sgi.com/tech/stl/. Accessed 7th of January 2007.

[TDMP] The Digital Michelangelo Project. http://graphics.stanford.edu/projects/mich/. Accessed 4th of January 2007.

[WikDT] Wikipedia, the free encyclopedia. Delaunay Triangulation. http://en.wikipedia.org/wiki/Delaunay_triangulation. Accessed 4th of January 2007.