THEORY AND PRACTICE OF SAMPLING AND RECONSTRUCTION FOR MANIFOLDS WITH BOUNDARIES

by

GOPI MEENAKSHISUNDARAM

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill

2001

Approved by:

_____

Advisor: Jack Snoeyink

_____

Reader: Herbert Edelsbrunner

_____

Reader: Shankar Krishnan

ABSTRACT

GOPI MEENAKSHISUNDARAM:

Theory and Practice of Sampling and Reconstruction for Manifolds with Boundaries

(Under the direction of Prof. Jack Snoeyink)

Surface sampling and reconstruction are used in modeling objects in graphics and digital archiving of mechanical parts in Computer Aided Design and Manufacturing (CAD/CAM). Sampling involves collecting 3D points from the surface. Using these point samples, a reconstruction process rebuilds a surface that is topologically equivalent and geometrically close to the original surface. Conditions are imposed on sampling to ensure correct reconstruction. For a special case of manifolds, there are theoretically sound algorithms for sampling and reconstruction. The sampling conditions for such algorithms impose a minimum required sampling density (maximum distance between close samples) to ensure correct reconstruction.

In this dissertation, I study the sampling and reconstruction of *manifolds with boundaries*. For this class of surfaces, I show that the conditions on minimum required sampling density are *not sufficient* to ensure correct reconstruction if only the point samples are given as input to the reconstruction process. Additional information like the smallest *boundary size* in a model, though sufficient to ensure correct reconstruction, imposes uniform sampling density throughout the model. In this dissertation, I propose a novel way to use the variation in the sampling density across the surface to encode the presence of a boundary. A sampling condition is proposed based on this approach, and the reconstruction process requires no additional information other than the input set of sample points. The reconstruction algorithm presented in this dissertation for reconstructing manifolds with or without boundaries is shown to be correct, efficient, and easy to implement.

PREFACE

Many thanks to

My parents, for their love and good judgment while raising me.

The Link Foundation for funding my research.

Herbert Edelsbrunner, whom I admire for his humility and knowledge, for introducing me to topology, giving the direction to this thesis, encouraging and guiding me throughout this process.

Jack Snoeyink, for guiding me with patience through the excruciating thesis writing process.

Dinesh Manocha, for all his moral support throughout my graduate life.

Greg Welch, for wonderful research discussions and for being a good friend.

Henry Fuchs, for giving me an opportunity to work with the STC group when I was deciding between staying for PhD and leaving the program.

Shankar Krishnan, my closest friend, for everything he has done for me, starting from my first day at the US when he accommodated me in his house to my last day in the graduate school when he is one of my committee members.

Many fellow grad students with whom I spent so much time, especially Ramesh Raskar, Ajith Mascarenhas, Ashes Ganguly, Anand Srinivasan, Chandna Bhatnagar, Deepak Bandhopadhyaya, Mike Brown, Praveen Patnala, Ruigang Yang, Gentaro Hirota, and many others.

UNC's graphics research groups, for providing an open, exciting and cooperative research environment.

The staff of the UNC computer science department, for helping me in innumerable ways and for being a pleasure to work with, and finally,

My wife Aditi Majumder without whose help, support, love, affection, and extraordinary determination I would not be seeing this day of completing my PhD.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The field of Computer Graphics since its birth has been striving to achieve *photo-realism* in computer generated images, and *physical realism* in its simulation of environments. *Object modeling*, where real-world objects in all their intricate complexity are represented in computers, is required to achieve *photo-realism*. *Physical modeling*, where physical phenomena are accurately modelled in computers, is required to achieve *physical realism*.

Object modeling is the process of representing a physical object in a suitable representation that is useful for the intended application. There are two broad application areas where object modeling is used – the Compute Aided Design and Manufacturing (CAD/CAM) industry, and the Computer Graphics industry. In the CAD/CAM industry, the computer (re)presentation of objects is used to manufacture physical models. In the computer graphics field, in general, existing physical objects are used to get a computer *re*presentation. Object modeling tools that have been developed over many years are more successful in aiding the user to *present a model* (for the CAD/CAM industry) than to *represent an object* (for the computer graphics industry). This difference is due to substantially different skill levels needed by the user for the tasks involved in presenting and representing the models.

Recently, technology to directly capture physical objects, both their geometry and texture (color and reflectance properties), has become commonplace, and this has given a tremendous boost to the modeling in computer graphics. Devices using this technology include touch probe sensors, laser scanner, 3D stereo systems, etc. With minimal human intervention, these devices sample the surface of the object in the form of points in 3D.

In addition to modeling for computer graphics applications, the technology of sampling objects is widely used for ‘digital archiving’. Digital archiving is a process of storing the models of objects

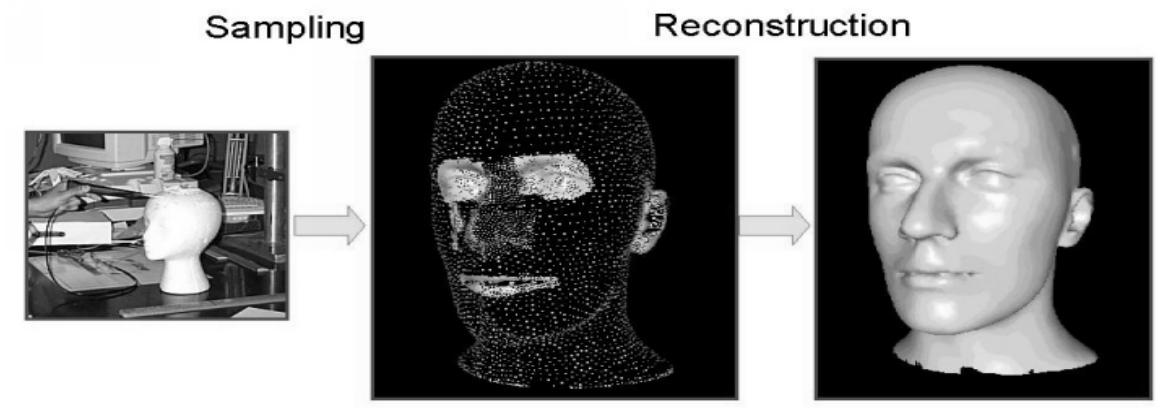

Figure 1.1: *Surface Reconstruction Problem. Left: Sampling the surface, Middle: Sample points, Right: Reconstructed surface.*

and their corresponding documentation in computers, rather than storing them as physical models, to save space and retrieval/reference time.

The set of points from the sampling process, by itself, is one form of representation of the underlying object. But this representation is not very useful for both *photo-realism* (visualization) and *physical realism* (simulation). *Visualization* requires the representation of a surface to handle texture mapping, to provide correct occlusion properties, to represent multiple levels of detail, and to bring out visual realism. *Physical simulation* and analysis requires a continuous surface representation. For example, detecting collision between two models represented using points, instead of a continuous surface, would produce unacceptable results. A continuous-surface representation, achieved by connecting these points appropriately using triangles or other primitive surface pieces, is a natural representation for visualization, physical simulation, and for applications involving surface analysis.

This problem of connecting the sample points appropriately to reconstruct the surface is commonly known as *surface reconstruction* in the computer graphics community. (Note that in the computer vision community, the process of sampling the object to get a point set is termed as surface reconstruction [Faugeras93].) A well developed area in the field of signal processing is sampling and reconstructing signals. The word "reconstruction" in *surface reconstruction* finds its root in the similarity between the surface reconstruction and signal reconstruction problems.

This dissertation addresses the problem of converting a discrete set of points into a continuous surface. Figure 1.1 shows the steps of this conversion. The figure on the left shows one of the discrete data capturing device, the touch sensor, which is used manually to sample data points from a real world model. The middle picture shows the data points sampled by this capture device. The right

picture shows a continuous representation of this data by reconstructing the surface using these sample points. It can also be seen from the images that the continuous data reflects more faithfully the original real-world object than the discrete data points. The rest of this dissertation describes in depth one such method to reconstruct continuous surfaces from discrete data points.

1.1 Surface Reconstruction: Problem Definition and Issues

In this section I define the surface reconstruction problem. Various terms used in this section are defined and elaborated in Chapters 2 and 3.

The surface reconstruction problem can be stated as follows: *Given a set of points S that are sampled from a surface M embedded in $\mathbf{R}^3$, construct a surface F such that the points of S lie on F, and F approximates M geometrically and is equivalent to M topologically*. A variation of this *interpolatory* definition, where F passes through the points in S, is one where F passes "close" to the set of points S. In this dissertation we consider the interpolatory version of the surface reconstruction problem.

The choice of underlying mathematical and computational representation of the reconstructed surface is important for its applicability. The most common choices are triangular and polygonal mesh representations. A triangular mesh allows us to express the topological properties of the surface, and it is the most popular model representation for visualization and rendering applications. Hence, in this dissertation, I will use only the piecewise linear triangular representation of the surface. Further, I assume that the points input to the surface reconstruction algorithm are sampled from a manifold with or without boundaries. There are various challenges for surface reconstruction algorithms including reliability, robustness, versatility, efficiency, and quality of reconstruction. Here in this section, I briefly discuss some of the issues that I address in this dissertation.

A proper reconstruction of surfaces is possible only if they are "sufficiently" sampled. However, sufficiency conditions are difficult to formulate and as a result, most of the existing reconstruction algorithms ignore or do not specify their requirements on sampling for reliable reconstruction. Hence, these algorithms can be classified as surface reconstruction heuristics as opposed to algorithms. Exceptions include the work of [Attali97, Bernardini97, Amenta98b]. These algorithms that can *reliably* reconstruct surfaces (without boundaries) provide sufficiency conditions for sampling. Assuming that the set of sample points satisfies these sampling conditions, correctness of these algorithms are ensured by theoretical guarantees based on the sampling conditions.

In this dissertation I show that the sampling conditions required to reconstruct the surfaces *without* boundaries are not sufficient to reconstruct surfaces *with* boundaries. As a result, none of the reconstruction algorithms available today can guarantee correct reconstruction of surfaces with boundaries. I provide the conditions for sampling surfaces with and without boundaries and prove the correctness of a reconstruction algorithm based on these sampling conditions.

Speed of reconstruction is another issue this dissertation will address. Currently, most of the surface reconstruction algorithms that guarantee a "good" quality triangulation and are theoretically sound produce higher dimensional simplices, like tetrahedra, or multi-sided polygons([Amenta01]). A second stage of these algorithms removes interior facets or triangulates each of the multi-sided polygons to produce the final triangulation. Therefore, these algorithms usually take in the order of a few minutes to run on data sets of moderate sizes (about 20,000 to 30,000 points).

Finally, the quality of triangulation has to be addressed when using triangulation representations. In many applications, like graphics rendering and finite element analysis, "fat" triangles with large vertex angles are preferred. The minimum angle in the triangulation is one measure of the quality of the triangulation. The reconstruction algorithm presented in this dissertation tries to maximize the minimum angle in its triangulation.

1.2 Thesis Statement and Results

The central claim of this research is:

> Conditions on minimum required sampling density are not sufficient to ensure reliable reconstruction of surfaces with boundaries from an input set of points and normal vectors at those points.

There are a few ways to define sufficient conditions to reconstruct surfaces with boundaries. These conditions are discussed in detail in Section 5.4.1. Following is an additional claim of this research.

> Conditions on relative minimum and maximum sampling densities are sufficient to ensure reliable reconstruction of surfaces with boundaries that takes a set of points and normal vectors at those points as input.

In summary, I present in this dissertation the following:

1. Sampling conditions and their properties to sample surfaces with and without boundaries.

2. An efficient reconstruction algorithm called the *Localized Delaunay Triangulation* algorithm that generates good quality triangulations of the surfaces with and without boundaries from the data points satisfying the conditions in 1.

3. A definition of *Geometric Triangulation*, and its properties.

4. Properties of multicovering functions between simplicial complexes and manifolds. In particular, the relationship between multicovering and the geometric triangulation.

5. A proof of correctness of the Localized Delaunay Triangulation by showing that the reconstructed triangulation is a geometric triangulation of the original surface.

6. An implementation of the Localized Delaunay Triangulation.

Apart from the above contributions, I also present a new normal estimation technique at sample points using other spatially close points, and a fast Voronoi neighbor computation algorithm.

1.3 Outline of this Dissertation

In Chapter 2, I provide the necessary background material for better appreciation of this dissertation. Following this, in Chapter 3, the work done till now in the field of surface reconstruction is briefly described.

In Chapter 4, I describe in detail my algorithm for surface reconstruction. This algorithm, called the Localized Delaunay Triangulation, is efficient, and can reconstruct surfaces both with and without boundaries. Further, in this chapter, I elaborate on the implementation of the above triangulation algorithm, analyze its performance and show the results of the algorithm on various moderate-sized models. The following chapters explore the theoretical foundation for this reconstruction algorithm, and prove that this algorithm produces a correct reconstruction of the surface.

In Chapter 5, the theory of sampling surfaces is explained. The basic difference between sampling surfaces with boundaries and without boundaries is elucidated with examples. The shortcomings of the familiar sampling techniques using medial axes, when sampling surfaces with boundaries, is also illustrated using examples. In this chapter, I also explore the design space of reconstruction algorithms that uses additional information other than the input set of sample points.

Finally, I use the variation in the sampling density as an implicit information to encode the presence of boundaries, and I prescribe this approach to design algorithms that uses only the sample points to reconstruct surfaces with boundaries.

In Chapter 6, I define the sampling conditions for surfaces both with and without boundaries. Results from Chapter 5 are used to develop the required mathematical formulation of the sampling conditions. The sampling conditions formulated in this chapter ensure that the Localized Delaunay Triangulation is a correct triangulation.

In Chapter 7, I develop the theory of triangulation, first by defining triangulations, and later by enumerating the properties of a triangulation. In this context, I introduce the concept of a *Geometric Triangulation*, as opposed to a conventional *Topological Triangulation*, to avoid multi-covering triangulation of surfaces. Under certain assumptions about the model, I prove the necessary and sufficient conditions for a triangulation to be a geometric triangulation. Finally, I prove that the Localized Delaunay Triangulation is a geometric triangulation, and hence a “correct” triangulation.

Finally, in Chapter 8, I conclude and list a few open problems and future directions.

CHAPTER 2

PRELIMINARIES

In this chapter I define surfaces, curvature, surface boundaries, samples, Voronoi diagrams, Delaunay triangulations, medial axes, and other concepts that are required to understand this dissertation. The definitions provided in this chapter are specific to the needs of this dissertation. These concepts have more general applicability than the situations they are used here, and in a general setting, some have broader definitions than the ones provided here.

2.1 General Geometric Concepts

A few of the following definitions are taken from [O'Neill97].

Definition: A *vector* $\vec{v} \in \mathbf{R}^3$ is defined by a starting point $\mathbf{p} \in \mathbf{R}^3$, commonly known as the origin, and an ending point $\mathbf{q} \in \mathbf{R}^3$. A *vector* is represented using a 3-tuple, $\vec{v} = \mathbf{q} - \mathbf{p}$, where subtraction of points is defined as componentwise difference.

Definition: The *dot product* of vectors $\vec{v} = (v_1, v_2, v_3)$ and $\vec{w} = (w_1, w_2, w_3)$ in $\mathbf{R}^3$ is the number

$$\vec{v} \cdot \vec{w} = v_1 w_1 + v_2 w_2 + v_3 w_3$$

If the dot-product is zero, then the two vectors are said to be *orthogonal* to each other, or *orthogonal vectors*.

Definition: The Euclidean *norm* of a vector $\vec{v}$ is the number

$$\|\vec{v}\| = (\vec{v} \cdot \vec{v})^{1/2}$$

Definition: The *Euclidean distance* between the points $\mathbf{p}$ and $\mathbf{q}$ of $\mathbf{R}^3$ is the norm of the difference vector $\|\mathbf{p} - \mathbf{q}\|$.

$$d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|$$

Definition: If **p** is a point in $\mathbf{R}^3$ and $\epsilon > 0$ is a number, then the *ϵ-neighborhood* of $\mathbf{p} \in \mathbf{R}^3$ is the set of all points **q** of $\mathbf{R}^3$ such that $d(\mathbf{p}, \mathbf{q}) < \epsilon$.

Definition: A subset B of $\mathbf{R}^3$ is *open* if each point of B has an *ϵ-neighborhood* that is a proper subset of B.

Open sets are used in topology and differential geometry concepts.

2.2 Topology for Surface Reconstruction

The topics in topology required for this dissertation are limited to the concepts that relate surfaces and its triangulation. Many of the following definitions are taken from [Edelsbrunner01].

2.2.1 Simplices and Complexes

Definition: The *n-dimensional half-space* is

$$\mathbf{H}^n = \{x = (x_1, x_2, \ldots, x_n) \in \mathbf{R}^n \mid x_1 \geq 0\},$$

or a space that can be transformed to $\mathbf{H}^n$ by a rigid transformation.

Definition: An n-dimensional polytope is the intersection of a finite number of n-dimensional half-spaces with non-empty interior in $\mathbf{R}^n$. The *faces* of an n-dimensional polytope are $(n-1)$-dimensional polytopes.

By definition, a polytope is convex. That is, if points **a** and **b** are in the polytope then the line segment joining **a** and **b** completely lies inside the polytope. On the other hand, a polyhedron, whose faces are also $(n-1)$-dimensional polytopes, need not be convex.

Definition: The *convex hull*, *convS*, of a set of points S is the smallest polytope that contains all the points.

Definition: An *n-simplex*, σ, is the convex hull of $n+1$ points in non-degenerate positions in $\mathbf{R}^m$, $m \geq n$. If S is the set of these $n+1$ points, then we write $\sigma =$conv$S$.

Thus an empty set $\emptyset$ is a -1-simplex, point is a 0-simplex, a line is a 1-simplex, a triangle is a 2-simplex and so on.

Definition: Let S be a set of $n+1$ points in non-degenerate position in $\mathbf{R}^m$, $m \geq n$, and $\sigma =$conv$S$ be an n-simplex. The l-simplex $\tau =$conv$T$, $T \subseteq S$ consisting of $l+1$ points, is called an *l-face* of σ and this relationship is denoted by $\tau \leq \sigma$.

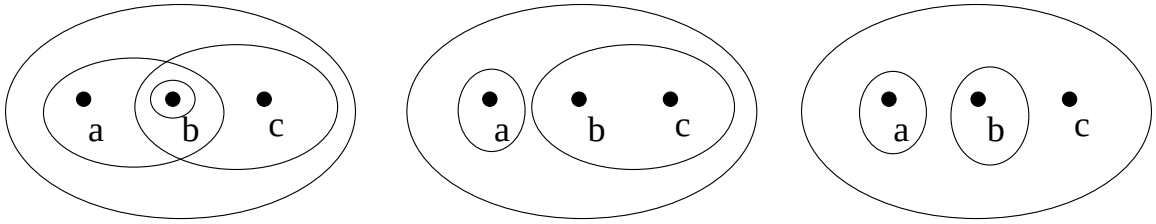

Figure 2.1: *Let* $\mathbf{X} = \{a, b, c\}$. *To the left,* $\Gamma = \{\emptyset, \mathbf{X}, \{a, b\}, \{b\}, \{b, c\}\}$. *Similarly,* Γ *is defined in the middle and the right. Note that* Γ *is a topology to the left and to the middle but not to the right as it lacks the set* $\{a, b\}$.

Definition: A *simplicial complex*, K, is the collection of faces of a finite number of simplices, any two of which are either disjoint or meet in a common face. More formally, it is a collection K such that

(a) if $\sigma \in K$ and $\tau \leq \sigma$, then $\tau \in K$, and

(b) if $\tau \in K$ and $\sigma \in K$, then $\tau \cap \sigma \leq \tau, \sigma$.

Note that $\emptyset$ is a face of every simplex and thus belongs to K by Condition (a). A tetrahedron, or a "book" where the pages are triangles and all pages are connected along one edge at the "rib" of the book, are examples of simplicial complexes. A collection of triangles along with their edges and vertices define a simplicial complex as long as no two triangles form an improper intersection.

2.2.2 Topological Spaces

Definition: A *topology* on a set $\mathbf{X}$ is a collection Γ of open subsets of $\mathbf{X}$ having the following properties:

(1) $\emptyset$ and $\mathbf{X}$ are in Γ.

(2) The union of the sets in any subcollection of Γ is in Γ.

(3) The intersection of the sets in any finite subcollection of Γ is in Γ.

Definition: A set $\mathbf{X}$ for which a topology Γ has been specified is called a *topological space*. Examples of topologies shown in Figure 2.1 are taken from [Munkres75]. Further, a relevant fact is that a simplicial complex is a topological space.

Definition: A *topological subspace* of the pair ($\mathbf{X}$, Γ_X) is a subset $\mathbf{Y} \subseteq \mathbf{X}$ together with the *subspace topology* Γ_Y consisting of all intersections between $\mathbf{Y}$ and open sets of Γ_X , $\Gamma_Y = \{\mathbf{Y} \cap A | A \in \Gamma_X\}$.

Definition: Let K be a simplicial complex in $\mathbf{R}^d$. Its *underlying space* is the union of its simplices together with the subspace topology inherited from $\mathbf{R}^d$,

$$|K| = \{x \in \mathbf{R}^d | x \in \sigma \in K\}.$$

Definition: A topological space is called a *Hausdorff space* if for each pair x_1, x_2 of distinct points of $\mathbf{X}$, there exist open neighborhoods U_1 and U_2 of x_1 and x_2, respectively, that are disjoint.

Definition: A *separation* of a topological space $\mathbf{X}$ is a pair U, V of disjoint nonempty open subsets of $\mathbf{X}$ whose union is $\mathbf{X}$.

Definition: The space $\mathbf{X}$ is said to be *connected* if there does not exist a separation of $\mathbf{X}$.

Another way of formulating the definition of connectedness is as follows. A space $\mathbf{X}$ is connected if and only if the only subsets of $\mathbf{X}$ that are both open and closed in $\mathbf{X}$ are the empty set and $\mathbf{X}$ itself.

Definition Given points x and y of the space $\mathbf{X}$, a *path* in $\mathbf{X}$ from x to y is a continuous map $f : [a, b] \to \mathbf{X}$ of some closed interval in the real line into $\mathbf{X}$, such that $f(a) = x$ and $f(b) = y$. A space is said to be *path connected* if every pair of points of $\mathbf{X}$ can be joined by a path in $\mathbf{X}$. Every path connected space is connected.

2.2.3 Homeomorphisms and Triangulations of Spaces

In this section we will see how two topological spaces can be compared using a mapping between the spaces. We will also discuss the topological definition of triangulation using these mappings.

Definition: A function $f : X \to Y$ is said to be *continuous* if the preimage of every open set in Y is open in X. A continuous function is called a *map*.

Definition: A map $f : \mathbf{X} \to \mathbf{Y}$ between topological spaces is said to be a *homeomorphism* if it is a one-to-one and onto continuous function whose inverse is also continuous.

Definition: Two topological spaces $\mathbf{X}$ and $\mathbf{Y}$ are considered of *the same type* or *homeomorphic* if there exists a homeomorphism between them. This is denoted by $\mathbf{X} \approx \mathbf{Y}$.

Definition: A topological space $\mathbf{X}$ is a *k-manifold* if every $x \in \mathbf{X}$ has a neighborhood homeomorphic to $\mathbf{R}^k$.

Definition: A space $\mathbf{X}$ is a *k-manifold with boundary* if every point $x \in \mathbf{X}$ has a neighborhood homeomorphic to $\mathbf{R}^k$ or to $\mathbf{H}^k$. The *boundary* is the set of points with neighborhood homeomorphic to $\mathbf{H}^k$, and is denoted by $\mathbf{bdX}$. The boundary is always either empty or a $(k-1)$-manifold.

A k-manifold in $(k+1)$-dimensions has two sides, one in the positive direction of the normal vector and the other in the negative direction. For example, if we consider a penny as a 2D disk, being in our 3D world, it has two faces.

Definition: If there exists a path in $\mathbf{X}$ from $x \in \mathbf{X}$ to itself such that the starting point is on one side and the ending point is on the other side of the manifold, then the manifold is said to be *non-orientable*. If no such path exists, then the manifold is said to be *orientable*.

Definition: A map $f : U \to \mathbf{R}^m$, where U is open in $\mathbf{R}^m$, is *differentiable of class* C^r (or a C^r*-map*) if f has continuous partial derivatives of order up to r. It is *smooth*, or of *class* C^∞, if it has continuous partial derivatives of all orders. A map $f : X \to \mathbf{R}^m$, where $X \subset \mathbf{R}^n$ is arbitrary, is of class C^r if it can be extended to a C^r map on a neighborhood of X.

Definition: A differentiable map $f : M \to N$ between differentiable manifolds is an *immersion* if the derivative df has maximal rank at every point $p \in M$. If, in addition, f is a homeomorphism onto its image, it is called an *embedding*, and we say M is *embedded* in N. In particular, an embedding is a one-to-one immersion.

No 2-manifold embedded in 3D can be non-orientable. That is, there does not exist a map from a non-orientable 2-manifold to $\mathbf{R}^3$ whose restriction to the image is a homeomorphism. On the other hand, there are non-orientable 2-manifolds with boundaries that can be embedded in $\mathbf{R}^3$.

Definition: A (topological) *triangulation* of a topological space $\mathbf{X}$ is a simplicial complex K whose underlying space is homeomorphic to $\mathbf{X}$, $\mathbf{X} \approx |K|$.

In this dissertation we are concerned only with triangulations of surfaces (defined formally in the next section) that are Hausdorff, orientable, connected 2-manifolds with or without boundaries embedded in 3D.

2.3 Differential Geometry for Surface Reconstruction

In this section, we review certain concepts in differential geometry that will be useful in describing the conditions for sampling a smooth surface.

Surface reconstruction requires definition of surfaces. In our case, as we are considering surfaces with boundaries, which are space curves in 3D, we need a definition of curves also.

Definition: A real-valued function f on $\mathbf{R}^3$ is *smooth* if all partial derivatives of f , of all orders, exist and are continuous.

Definition: A *curve* in $\mathbf{R}^3$ is a differentiable function $\alpha : I \to \mathbf{R}^3$ from an open interval I into $\mathbf{R}^3$.

Definition: A differential function $f : \mathbf{R}^n \to \mathbf{R}^m$ is called a *mapping* from $\mathbf{R}^n$ to $\mathbf{R}^m$.

We can extend the concepts of open intervals and curves to open regions in $\mathbf{R}^2$ and surfaces in $\mathbf{R}^3$. Before going into the definition of a surface, I would like to informally define tangent vectors. Tangent vectors to $\mathbf{R}^m$ have two components: a vector and a point of application of that vector.

Definition: Let $f : \mathbf{R}^n \to \mathbf{R}^m$ be a mapping. If **v** is a tangent vector to $\mathbf{R}^n$ at **p**, let $f_*(\mathbf{v})$ be the initial velocity of the curve $t \to f(\mathbf{p} + t\mathbf{v})$. The resulting function f_* sends tangent vectors to $\mathbf{R}^n$ to tangent vectors to $\mathbf{R}^m$, and is called the *tangent map of* f.

Definition: A mapping $f : \mathbf{R}^n \to \mathbf{R}^m$ is *regular* provided that at every point **p** of $\mathbf{R}^n$ the tangent map f_* is one-to-one.

Definition: A *coordinate patch* $\mathbf{x} : D \to \mathbf{R}^3$ is a one-to-one regular mapping of an open set D of $\mathbf{R}^2$ into $\mathbf{R}^3$.

Definition: A coordinate patch **x** is a *proper patch* if the inverse $\mathbf{x}^{-1}$ is also continuous over the range of **x**.

Definition: A *surface* in $\mathbf{R}^3$ is a subset M of $\mathbf{R}^3$ such that for each point **p** of M there exists a proper patch in M whose image contains a neighborhood of **p** in M.

This differential geometry definition of a surface implicitly defines a parametrization of the surface. The domain D of the patch **x** is the parameter space. Further, this definition of M allows us to consider the parameter space to be open.

As the coordinate patch **x** is a regular patch, the tangent space of D is mapped into the tangent space on M. The tangent space at a point **p** of $M \subset \mathbf{R}^3$, denoted by $T_p(M)$, is a plane passing through **p** called the *tangent plane*. The vector orthogonal to all the vectors in this tangent plane is called the *normal vector* to M at $\mathbf{p} \in M$.

As two topological spaces are compared using a map, in general, two surfaces are compared using a mapping function between them.

Definition: A map π is said to be a *diffeomorphism* if π is differentiable, one-to-one, and onto, and its inverse, π^{-1} is also differentiable.

The topological analogue of diffeomorphism is *homeomorphism*. The map needs to be continuous to be a homeomorphism, but for it to be diffeomorphism, the map needs to be differentiable also. As every differentiable function is continuous, every diffeomorphism is a homeomorphism.

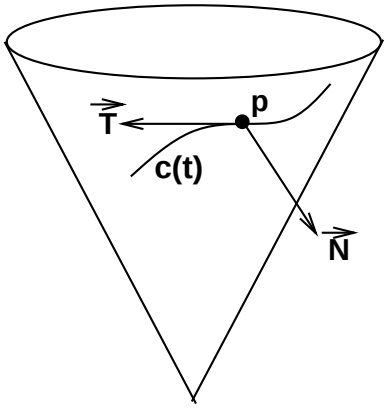

Figure 2.2: *The Darboux Frame*

2.3.1 Frenet and Darboux Frames, and Principal Curvatures

Consider a point **p** on a surface M as shown in Figure 2.2. Let the normal vector to M at **p** be $\vec{N}$. Given a unit vector $\vec{v}$ on the tangent plane at **p**, define a curve $c(t) : [-\epsilon, \epsilon] \rightarrow M$ such that $c(0) = \mathbf{p}$ and $c'(0) = \vec{v}$, where c' is the derivative of c. The **Darboux frame** at **p** is defined as the orthonormal differential frame $\vec{T} = \vec{v}$, $\vec{B} = \vec{N} \times \vec{T}$, and $\vec{N}$. Figure 2.2 shows the Darboux frame on the surface of a cone. It is easy to see that for surfaces with a well defined tangent plane everywhere, every point on the surface has a unique Darboux frame associated with it in a given direction $\vec{v}$ in the tangent plane.

Frenet Frame: The Darboux frame is defined for curves on a manifold. If the curves are space curves that are not restricted to be on the surface, then the normal $\vec{N}$ of the Darboux frame is not defined. Hence a different but consistent frame has to be defined for space curves. The derivative of the tangent vector $\vec{T}$ at a point on a curve is orthogonal to the tangent vector. This orthogonal vector is defined as the normal $\vec{N}$ of the space curve. The bi-normal $\vec{B} = \vec{N} \times \vec{T}$, is as in the Darboux frame. In our application of surface reconstruction, the boundary curves of the surface can either be considered as a space curve or a curve restricted by the surface. So we have an option of choosing either Darboux frames or Frenet frames for our computation. In this dissertation, I consider the boundary curves as part of the surface and use Darboux frames wherever there is a need for a coordinate frame.

Surface curvature: Associating a local differential frame at every point on the surface allows us to measure some geometric invariants on the surface. If we walk infinitesimally along a direction $\vec{v}$, the change of the surface normal in the direction $\vec{v}$ is called the *normal curvature*, κ_v. As we move along different directions in the tangent plane, the *normal curvature* varies. The directions with minimum and maximum normal curvatures are called *principal directions* and the corresponding curvatures in these directions are called *principle curvatures*. These principal directions are orthogonal

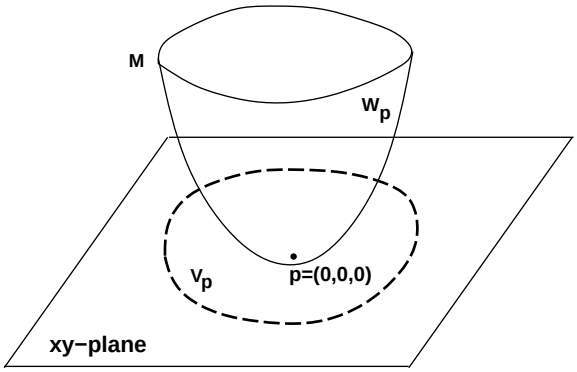

Figure 2.3: *Quadratic approximation of a surface*

to each other. In the rest of this dissertation, I will refer to the principal curvatures as κ_1 and κ_2 (or κ_{min} and κ_{max}).

2.3.2 Surface Approximation using Power Series

Consider a 2-manifold $M \subset \mathbf{R}^3$ and a point $\mathbf{p} \in M$ as shown in Figure 2.3. Without loss of generality, we make the following assumptions about **p** and M.

- **p** is the origin,
- the tangent plane $T_p(M)$ of M at **p** is the $z = 0$ plane, and
- the two principal directions of M at **p** are the coordinate axes $e_1 = (1, 0, 0)$ and $e_2 = (0, 1, 0)$.

It is easily seen that these conditions can be achieved by a rigid transformation of M. In order to proceed further, I make use of a result from classical differential geometry, namely the implicit function theorem, which I state here without proof.

Theorem 1 *[O'Neill97] There exists a small neighborhood W_p of $\mathbf{p} \in M$ such that the map $\pi : (x, y, z) \rightarrow (x, y)$ is a one-to-one map with its image being an open set $V_p \subset \mathbf{R}^2$. Moreover the map π is a diffeomorphism.*

The point $(x, y, z) \in M$ is a point in the local neighborhood of **p** and is defined in a local coordinate system at **p**.

The fact that π is a diffeomorphism implies that π^{-1} exists and that π and π^{-1} are smooth mappings. Therefore, we can approximate the surface in the neighborhood of **p**, denoted by W_p, using a smooth height function h as

$$W_p = \{(x, y, h(x, y)) : (x, y) \in V_p\}$$

Further, the tangent plane of $W_p \subset M$ at $\mathbf{p}= (0,0,0)$ is given by the basis vectors $(\partial W_p/\partial x)_{\mathbf{p}} = (1, 0, \frac{\partial h}{\partial x}(0,0))$ and $(\partial W_p/\partial y)_{\mathbf{p}} = (0, 1, \frac{\partial h}{\partial y}(0,0))$. Since we assumed that the tangent plane at $\mathbf{p}$ is the $z = 0$ plane, $\frac{\partial h}{\partial x}(0,0) = \frac{\partial h}{\partial y}(0,0) = 0$.

Shape operator is a familiar concept in differential geometry. For an elaborate study on shape operators, refer to Chapter 5 of [O'Neill97]. The shape operator at a point $\mathbf{p} \in M$ (denoted by S_p) is a linear operator that maps an element of $T_p(M)$ to another element in $T_p(M)$. If v_{p1} and v_{p2} are a set of basis vectors for $T_p(M)$, $S_p(av_{p1} + bv_{p2}) = cv_{p1} + dv_{p2}$ (a, b, c and d are real valued scalars). For the special case of a vector v being a principal direction, $S_p(v) = \kappa v$, where κ is the principal curvature. In our particular case, the tangent plane is spanned by e_1 and e_2. The shape operator applied to the vectors e_1 and e_2 are given by ([O'Neill97], Page 207)

$$\begin{aligned} S_p(e_1) &= \frac{\partial^2 h}{\partial x^2}(p)e_1 + \frac{\partial^2 h}{\partial x \partial y}(p)e_2 \\ S_p(e_2) &= \frac{\partial^2 h}{\partial x \partial y}(p)e_1 + \frac{\partial^2 h}{\partial y^2}(p)e_2 \end{aligned} \tag{2.1}$$

Since e_1 and e_2 are the principal directions, we can conclude that $\frac{\partial^2 h}{\partial x \partial y}(0,0) = 0$ and that $\frac{\partial^2 h}{\partial x^2}(0,0)$ and $\frac{\partial^2 h}{\partial y^2}(0,0)$ are the principal curvatures (denoted by κ_1 and κ_2 respectively).

We now use Taylor's formula to expand $h(x, y)$ around the origin $(0, 0)$. Thus,

$$\begin{aligned} h(x,y) &= h(0,0) + x\frac{\partial h}{\partial x}(0,0) + y\frac{\partial h}{\partial y}(0,0) \\ &\quad + \frac{1}{2}(x^2\frac{\partial^2 h}{\partial x^2} + 2xy\frac{\partial^2 h}{\partial x \partial y} + y^2\frac{\partial^2 h}{\partial y^2}) \\ &\quad + \text{higher order terms} \\ &= \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2) + \text{higher order terms} \\ &\cong \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2) \end{aligned}$$

This shows that the shape of M near $\mathbf{p}$ is *approximately the same as that of the surface*

$$W_p^{'}(x,y) = \left(x, y, \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2)\right),$$

and $W_p^{'}$ is called the *quadratic approximation of M near* $\mathbf{p}$.

If we use the polar form for the height function $h(r, \theta)$, where, $r = \sqrt{x^2 + y^2}$ and θ is the angle the vector (x, y) makes with the x-axis, then

$$h(r,\theta) \cong \frac{r^2}{2}(\kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta) \tag{2.2}$$

I use the quadratic approximation of the surface and the derivatives of height functions in describing and proving certain properties of my sampling conditions in Chapter 6 of this dissertation.

2.3.3 Euler Equation

The *normal curvature* κ_v at a point **p** on the surface in a given unit direction $\vec{v}$ on the tangent plane is defined as the curvature of the intersection curve of the surface with the plane formed by the vectors $\vec{v}$ and the surface normal at **p**. Using the *shape operator*, it can be written as

$$\kappa_v = S_p(\vec{v}) \cdot \vec{v} \tag{2.3}$$

We can represent $\vec{v}$ in terms of the principal directions ($\vec{v}_1$ and $\vec{v}_2$) as $\vec{v} = \cos\theta\vec{v}_1 + \sin\theta\vec{v}_2$, where θ is the angle $\vec{v}$ makes with $\vec{v}_1$. Therefore

$$\begin{aligned} \kappa_v &= S_p(\cos\theta\vec{v}_1 + \sin\theta\vec{v}_2) \cdot (\cos\theta\vec{v}_1 + \sin\theta\vec{v}_2) \\ &= (\kappa_1 \cos\theta\vec{v}_1 + \kappa_2 \sin\theta\vec{v}_2) \cdot (\cos\theta\vec{v}_1 + \sin\theta\vec{v}_2) \\ &= \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta \end{aligned} \tag{2.4}$$

The above equation is also known as the *Euler equation*. It expresses the normal curvature at a point on the surface in terms of the principal curvatures at that point. Using this result on the expression for $h(r,\theta)$ in Equation 2.2, we get

$$h(r,\theta) \cong \frac{\kappa_v r^2}{2} \tag{2.5}$$

I use these equations also in Chapter 6.

2.3.4 Frenet-Serret Equations for Darboux Frame

I will now present the equations that govern the behavior of the Darboux frame for curves on a surface M passing through **p**. These equations are a modified form of the well-known Frenet-Serret equations [Koenderink89, O'Neill97].

$$\begin{pmatrix} \frac{\partial\vec{T}}{\partial s} \\ \frac{\partial\vec{B}}{\partial s} \\ \frac{\partial\vec{N}}{\partial s} \end{pmatrix} = \begin{pmatrix} 0 & g & \kappa_v \\ -g & 0 & t \\ -\kappa_v & -t & 0 \end{pmatrix} \begin{pmatrix} \vec{T} \\ \vec{B} \\ \vec{N} \end{pmatrix} \tag{2.6}$$

The entries in the above matrix define the geometrical invariants at the point in consideration where κ_v is the component of the acceleration along the curve tangent (*normal curvature*), g (or *geodesic curvature*) is the component of the acceleration in the tangent plane, and t (also known as *geodesic torsion*) measures the twist along the $\vec{N} - \vec{B}$ plane.

It is now possible to describe the behavior of the normal vector along a curve on a surface M passing through a point **p**. The equation below can be derived from 2.6.

$$\partial\vec{N} = -\kappa_v \partial s \vec{T} - t\partial s \vec{B} \tag{2.7}$$

$$|\partial\vec{N}| = \sqrt{\kappa_v^2 + t^2}\partial s \tag{2.8}$$

The quantity $\sqrt{\kappa_v^2 + t^2}$ is called the *total curvature* of the space curve through **p**. I simplify the above equation for the special case of curves through **p** for which t is always zero. Then the total curvature becomes κ_v, the normal curvature. (Note that the torsion is zero for the planar curve used in the definition of *normal curvature* in the beginning of Section 2.3.3). Hence the modified rate of change of normal along a particular direction is given by,

$$|\partial\vec{N}| = \kappa_v \partial s \tag{2.9}$$

Let me briefly relate the concepts explained here with the sampling of a surface. The condition for sampling is related to the rate of change of normal vector, in other words the curvature, around a point on the surface. Samples should be close to each other in high curvature regions and can be spread out in low curvature regions. Such a sampling bounds the change of normal between two close samples on the surface. One way to describe this sampling using the concepts discussed here is as follows. Define a region around every point on the surface as all points that satisfy the inequality,

$$\kappa_v \partial s < constant \tag{2.10}$$

and enforce "uniform sampling" across these regions, by which we would find equal number of samples in every region. The regions defined by the above inequality are small in high curvature areas of the surface and large in the low curvature areas. Uniform sampling across these regions satisfies our requirement on sampling density mentioned above.

These concepts will be elaborated upon in Chapter 6 devoted to sampling conditions.

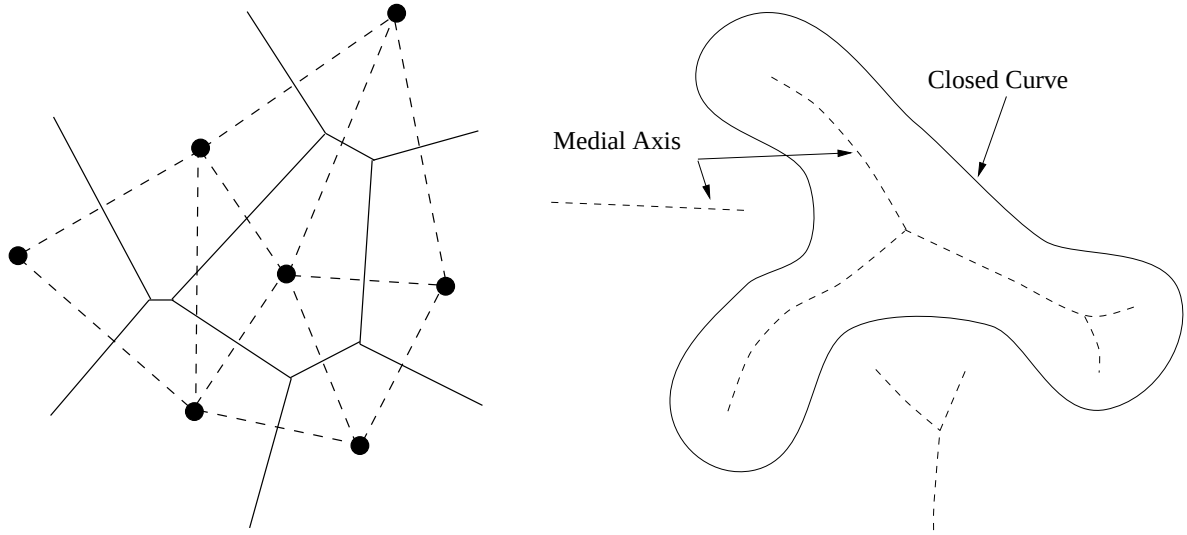

Figure 2.4: *Sampling: Left – Solid lines:Voronoi diagram, Dashed lines: Delaunay triangulation, Right – Medial axis of a closed curve.*

2.4 Voronoi Diagram and Delaunay Triangulation

The Voronoi diagram of a collection of disjoint geometric objects or *sites* is a partition of space into cells, each of which consists of the points closer to a particular set of sites than to any others. In the simplest case, these sites are points in space. Since distances have to be measured and compared, the concept of Voronoi diagram requires a definition of a distance function. There are various Voronoi diagrams depending on the distance metric used. In this dissertation, we discuss only the Voronoi diagram due to the Euclidean metric in 2D or 3D. The boundaries of the Voronoi cell are made up of Voronoi vertices, Voronoi edges, 2-dimensional Voronoi faces (in 3D and higher dimensions), and higher dimensional polytopes in higher dimensional spaces. Figure 2.4 shows an example of 2D Voronoi diagram for a set of point sites.

The Delaunay triangulation is the dual of a 2D Voronoi diagram. In a set of point sites, if two sites are separated by a Voronoi edge then they are connected by a Delaunay edge. When no four points are co-circular, the Delaunay triangulation is a triangulation of the given set of points. Figure 2.4 shows an example of Delaunay triangulation in 2D. In 3D, the Delaunay triangulation of a set of point sites is a tetrahedralization of the points.

2.4.1 An Incremental Algorithm for Delaunay Triangulations

Here I describe one well-known incremental algorithm to compute the Delaunay triangulation of a set of points. As the Delaunay triangulation is the dual of Voronoi diagram, for every Voronoi vertex there

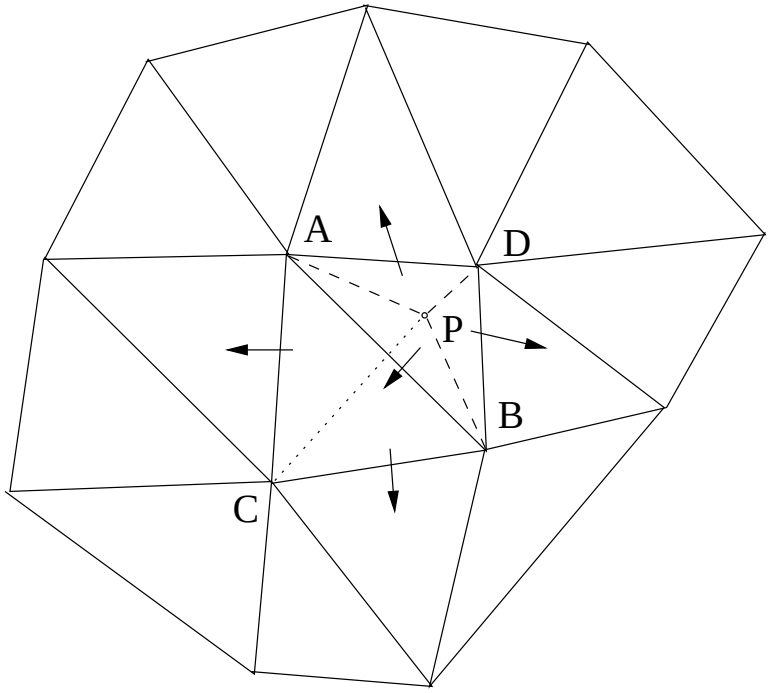

Figure 2.5: *Recursion over suspect edges.* **p** *is the new point.* **p**A*,* **p**B*, and* **p**D *are the new (dashed) edges. In this figure, edge* AB *failed the* Incircle *test and hence this edge has been swapped (dotted lines). Arrows indicate the propagation of suspect edges.*

is a corresponding Delaunay triangle, and for every Voronoi edge, a Delaunay edge. A Voronoi vertex in 2D is a point which is equidistant from three point sites. In other words, a maximal empty circle centered at a Voronoi vertex will touch three point sites. These three points form the vertices of the dual Delaunay triangle. Therefore, a triangle is a Delaunay triangle if and only if its circumcircle does not contain any vertex. More specifically, it can be shown that [Edelsbrunner01],

Theorem 2 *Given a triangulation of* n *sites such that for every pair of adjacent triangles* abc *and* bcd*,* a *is not in the circumcircle of* bcd*, then that triangulation is the Delaunay triangulation.*

This forms the basis of the incremental algorithm.

Let us assume that we are given a function *Incircle(a,b,c,d),* that returns true if the point d is inside the circumcircle defined by the triangle abc and false otherwise. Let us also assume the point sites are in non-degenerate positions, that is, no four points lie on a circle. The basic idea behind this algorithm is to introduce point sites one by one into an already existing Delaunay triangulation and then correct the resulting triangulation to make it Delaunay again. To start with, assume a big enough "dummy" triangle that contains all the input points. At the end of this algorithm, this dummy triangle can be removed and the remaining triangulation of only the input point set would still be Delaunay.

In the Delaunay triangulation any edge (except the boundary edges, which are dummy edges in our case) has two triangles incident on it. The vertices of these two triangles, in order, form a quadrilateral. Consider the recursive function below (and refer to Figure 2.5). This function checks if an edge AB of the quadrilateral $\mathbf{p}ACB$ is a valid Delaunay edge based on Theorem 2. If this edge is not a valid Delaunay edge, then it is flipped to connect $\mathbf{p}C$ of the quadrilateral $\mathbf{p}ABC$. It can be proved that by such repeated flipping, Delaunay triangulation can be constructed.

```
void CorrectSuspect(p, A, B)
{
    %COMMENT: Triangles incident on AB: pAB and ABC
    C = Vertex of the triangle incident on the edge AB
    If (Incircle(p, A, B, C)){
        Remove edge AB.
        Form edge pC.
        CorrectSuspect(p,A, C);
        CorrectSuspect(p,C, B);
    } else {
        return;
    }
}
```

Let **p** be the new point introduced, and let it lie inside the triangle ABD of the existing Delaunay triangulation. Form edges $\mathbf{p}A$, $\mathbf{p}B$, $\mathbf{p}D$. It can be proved [Edelsbrunner01] that these are Delaunay edges. The edges AB, BD, and DA need not be Delaunay edges anymore, and they are called *suspect edges*. To "acquit" them from this suspicion, the function *CorrectSuspect* is called for each suspect edge AB, along with the newly introduced point **p**. When the function returns, the triangulation is again Delaunay. Again, it can be proved that each edge is considered no more than once during this recursive flipping. Hence the running time of this algorithm is $O(n)$ for **p** and $O(n^2)$ in total, where n is the number of points.

In Chapter 4, I present a different algorithm to construct Delaunay triangulation locally by identifying the Voronoi neighbors. This algorithm assumes that all possible Voronoi neighbors of a point are known and they are ordered around the point by angle.

2.5 Medial Axis

The Medial Axis of a geometric object in 3D is defined as the set of the centers of spheres that touch at least two points of the surface of the object. More formally, the *medial axis* of a shape in Euclidean space is the locus of centers of maximal inscribed spheres. In general, the 3D medial axis is a collection of two dimensional surfaces in 3D. Refer to Figure 2.4 for an example of 2D medial axis for a curve.

2.5.1 Medial Axis and Voronoi Vertices

In the Voronoi diagram of a set of sites, the points in the interior of a Voronoi cell are closer to one site than the other. The points on the boundary between two or more Voronoi cells are equidistant from two or more sites. Let us consider a 2D curve. If every point on the curve is considered as a site, then every point on the boundary of Voronoi cells of these sites will be equidistant from at least two points on the curve, which is exactly the definition of a medial axis. In the limit, the Voronoi vertices which are equidistant from three points on the curve will lie on the medial axis. This fact is used in curve reconstruction algorithms presented in the literature [Amenta98a].

In the Voronoi diagram of the sample points of a surface in 3D, there might be Voronoi vertices close to the surface and away from the medial axis. This is because of the presence of "slivers". A sliver is a tetrahedron with bad aspect ratio yet a reasonably small circumradius to shortest edge ratio, such as the tetrahedron formed by four nearly equally spaced vertices around the equator of a sphere. The Voronoi center of a sliver can lie arbitrarily close to the surface.

2.5.2 Resting and Passing Circles

Let us consider the 2D curve illustrated in Figure 2.6. The circle of curvature (also called the *osculating circle*) of a point **a** is the smallest circle that touches more than one point in the small neighborhood around **a** including **a** (refer to Figure 2.6). I call a circle that touches points in a small neighborhood a "resting circle". Every circle smaller than the circle of curvature at **a**, touches only **a**. Hence their centers cannot lie on the medial axis. But there are exceptions. There may be a circle that touches points that are not in a small neighborhood, but distributed over different parts of the curve or curves. I call such a circle a "passing circle". Its center lies on the medial axis, and it may be smaller than the circle of curvature at its points of contact. These cases are shown in Figure 2.6. It can be seen that the *resting circles* capture the local features and the *passing circles* capture the global features. The medial axis captures both local and global features of a curve as it consists of centers of both resting

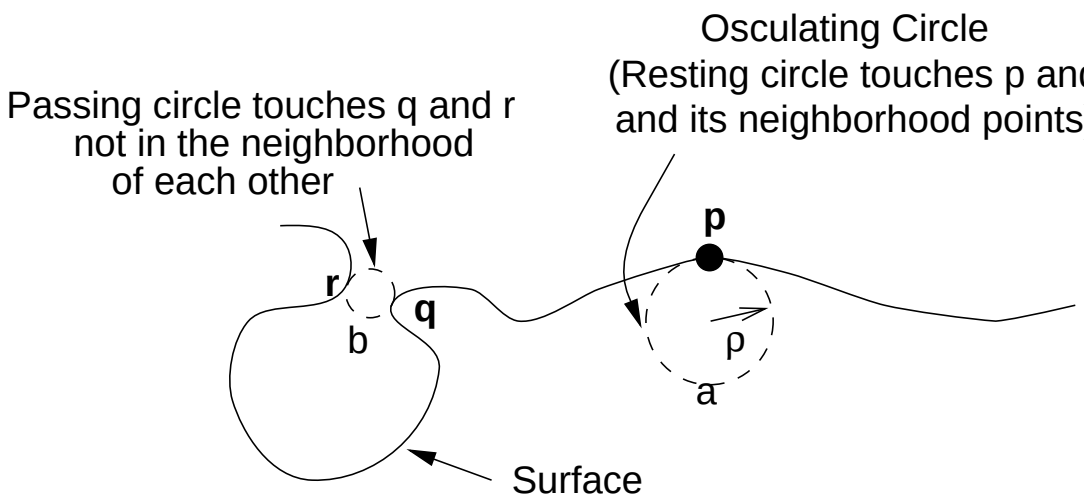

Figure 2.6: *Circle a is the circle of curvature at* **p**, *and its radius is ρ. Centers of a, and b are in the medial axis.*

and passing circles. The distance of a point to the centers of its resting and passing circles, and hence to the medial axis, is an indicator of the 'feature size' at that point. During the sampling process, the smaller the feature size, the higher should be the density of samples to get a good approximation of the medial axis by Voronoi boundaries of these samples. These concepts are used by Amenta *et al.* [Amenta98a, Amenta98b] as described in Chapter 3. We also use these concepts in Chapter 6 to compare our sampling with that of [Amenta98a, Amenta98b].

2.6 Summary

In this chapter I introduced various concepts in topology and differential geometry, and the concepts of Voronoi diagrams, Delaunay triangulations, and the medial axis. The important properties of topological spaces like connectedness, Hausdorff spaces etc. were defined. Tools to compare two topological spaces using homeomorphism were provided. Intuition was given on how the concepts in differential geometry can be used in sampling surfaces. The properties of surfaces, including orientability, were defined. In the section related to Voronoi diagrams and Delaunay triangulation, I presented a well-known algorithm to construct 2D Delaunay triangulation. Further, I also detailed the relationship between the Voronoi vertices and the medial axis and its potential use in designing robust surface reconstruction algorithms.

CHAPTER 3

PREVIOUS WORK

The problem of surface reconstruction has received significant attention from researchers in computational geometry and computer graphics. In this chapter, I discuss the previous work done in the field. I categorize the reconstruction methods as algorithms if they guarantee correct reconstruction when the input point samples satisfy some specified conditions, and the heuristics if they do not. The reconstruction methods, both algorithms and heuristics, use a combination of techniques to do the reconstruction. A few of the common techniques are described below.

A few surface reconstruction methods use *spatial subdivision* techniques. In these techniques, the space bounding the input set of points is subdivided into disjoint cells. Commonly used cell subdivision techniques include Delaunay tetrahedralization and voxelization. Once subdivided, the points in the adjoining cells are connected to arrive at the surface. These algorithms differ in the cell selection strategies for final reconstruction. The approaches of [Algorri96, Hoppe92, Edelsbrunner94, Bajaj95, Attali97] fall under this category.

A few methods use *distance functions* to decide on the final connectivity between points. The distance function might measure the distance of points from a local approximation of a surface [Hoppe92, Hoppe93], from the medial axis [Roth97], or from any other convenient shape.

A few methods use *direct surface reconstruction* techniques that interpolate or approximate the surface directly from the input set of points, without any intermediate representation like the medial axis or a spatial subdivision like a tetrahedralization or voxelization. Works in this category include [Boissonnat84, Muller93] and the algorithm presented in the dissertation. I describe below a few of the methods in the literature.

3.1 Reconstruction Algorithms with Sampling Conditions

The sampling process chooses sample points from the given object as input to the reconstruction algorithm. Sampling conditions are the restrictions on the sampling process so that the reconstruction algorithm would work reliably on the given input set of points. Only a few algorithms provide such sampling conditions.

Voronoi Filtering: One elegant sampling condition provided by Amenta *et al.* [Amenta98a, Amenta98b, Amenta99] is based on the medial axis. The underlying technique used in this method is called *Voronoi filtering*. Amenta, Bern, and Eppstein [Amenta98a] showed that the Voronoi vertices of a dense sampling of a curve in $\mathbf{R}^2$ approximates the medial axis of the curve. The closest distance of a point on a curve to the medial axis of the curve is called the *feature size* of that point. The sampling condition for the reconstruction algorithm states that for any point **a** on the curve, there should exist a sample point **p** at a distance that is less than a fraction of the feature size of **a**. If the value of the fraction is ϵ, then the sampling is said to be an ϵ-*sampling* of the curve. Their reconstruction algorithm first computes the Voronoi vertices of the given sample points. Then the Delaunay triangulation is computed for the set of points that is the union of the input set of points and the set of its Voronoi vertices. Those edges in this triangulation whose end points are in the input set of points form the reconstructed curve.

This idea does not extend directly to 3D, because the Voronoi vertices of points in 3D do not necessarily lie near the medial axis of the object. Amenta *et al.* [Amenta98b] developed heuristics to remove the Voronoi vertices that lie close to the surface, assuming that those are the Voronoi vertices that are not near the medial axis. Further, this sampling and reconstruction is applicable only to objects without boundary. In Chapter 5, I will show that in the case of objects with boundary, not all sections of medial axis are represented by Voronoi vertices. Improvements to [Amenta98b] are presented in [Amenta00].

Reconstruction of Curves with Boundaries: There are a few algorithms for curve reconstruction which guarantees correct reconstruction. One of them is based on the Voronoi filtering as explained earlier. The work that is relevant to us is the reconstruction of curves with boundaries, that is, curves with endpoints. Dey *et al.*[Dey00] developed the *conservative crust* algorithm and justify their reconstruction of curves with endpoints. The sampling condition is again based on the feature size as in the Voronoi filtering algorithm. Instead of prescribing a sampling condition for correct reconstruction, this method proves that there exists a sampling condition for the curve reconstructed by

the algorithm. In other words, there need not exist a sampling condition for the required reconstruction, but there exists a sampling condition for the curve reconstructed by the *conservative crust* algorithm. Further, this method reconstructs a family of curves with endpoints from the given sample points. Conservative crust algorithm can be made to reconstruct the correct (required) and only the correct curve by providing additional information as input to the reconstruction algorithm. This additional information can be the value of ϵ if the sampling is an ϵ-sampling of the curve with endpoints. This confirms the thesis presented in this dissertation regarding the sampling requirements for manifolds with boundaries (refer to Chapter 5).

Power Crust: A new surface reconstruction algorithm called the Power Crust by Amenta *et al.* [Amenta01, Amenta] uses the relationship between the Voronoi vertices and medial axis in a different way. The medial axis is the locus of the centers of largest empty spheres touching any point on the surface. The power diagram [Edelsbrunner93] is the weighted Voronoi diagram of a set of weighted points. The weights of points represent the radius of the spheres centered at those points. Combining these two concepts, the Power Crust method extracts the reconstructed surface as a power diagram of the largest empty spheres centered at the Voronoi vertices of the given set of sample points. As before, certain Voronoi vertices which are found to lie close to the surface are considered to be not on the medial axis and are removed. This is one of the most elegant techniques for reconstructing surfaces as it covers the space between the algorithms designed to reconstruct interpolatory surfaces that assume that the sample points do not have noise, and the ones to reconstruct approximating surfaces that assume that the sample points are probabilistic estimates of the underlying surface. This property enables the Power Crust algorithm to tolerate noise in the input data set and at the same time reconstruct the surfaces as faithfully as possible from the given set of points. The drawbacks of this algorithm are its dependence on heuristics to achieve this goal, its slow speed, and its need for triangulating the polygons generated by the power diagram. Again, the major limitation of this method is that it can be used only for surfaces with no boundaries.

Alpha Shapes: The alpha shapes introduced by Edelsbrunner and Mueke [Edelsbrunner94] were primarily to define shapes of molecules. But this concept was effectively used in surface reconstruction also. Given a set of points in 3D, a Delaunay tetrahedralization is constructed for this input set of points in the first step. In the second step, all tetrahedra, triangles, and edges whose radius of the smallest circumsphere is greater than a fixed parameter α are removed to arrive at the α-shape. In the third step, the triangles that belong to the desired surface are extracted out of the α-shape using the following rule.

Consider the two possible spheres of radius α through all three points of a triangle of the α-shape. If at least one of these does not contain any other point of the point set, then triangle belongs to the surface. Since α is a global parameter the user is not swamped with many open parameters, but the drawback is that a variable point density in sampling is not possible without loss of detail in the reconstruction and hence requires uniform sampling.

There are many reconstruction algorithms based on α shapes that inherit the sampling requirement from α shapes. Guo *et al.* make use of α-shapes for surface reconstruction but they propose a so-called *visibility* algorithm for extracting those triangles that represent the surface out of the α-shape. Teichmann *et al.* [Teichmann98] use density scaling and anisotropic shaping to improve the results of reconstruction using α-shapes. The Ball Pivoting Algorithm presented by Bernardini *et al.* [Bernardini99] again requires uniform sampling of the object, as it uses an algorithm similar to α-shapes. This algorithm assumes a normal at every vertex, and does not mention how to handle two close surfaces.

Normalized Meshes: As for α-shapes, again the Delaunay tetrahedralization is used for spatial decomposition in this approach of Attali [Attali97]. The normalized mesh introduced in this work, which is contained in the Delaunay graph, consists of the edges, faces, and tetrahedra whose dual Voronoi element intersects the surface of the object. In two dimensions, given the sample point set S taken from a curve c, the normalized mesh of c consists of all edges in the Delaunay triangulation whose Voronoi dual intersects c. If the curve belongs to a class of curves of bounded curvature called the r-regular shapes then, a bound on the sampling density can be given so that the normalized mesh retains all the topological properties of the original curve. During the reconstruction of c, all edges whose two Delaunay circumcircles intersect at an angle less than $\pi/2$ are chosen to be part of the reconstructed mesh. The idea behind this approach is that the Delaunay circumcircles tend to become tangent to the boundary of the object. If the sampling density is sufficiently high, then the reconstructed mesh is the same as that of the normalized mesh, which is the same as c topologically.

Though the concept of normalized mesh, also called as the restricted Delaunay triangulation, can be extended to 3D, the algorithm to construct normalized mesh cannot be extended. The reason is the same as the one encountered in the Voronoi filtering method of Amenta *et al.* [Amenta98b]. A few Delaunay spheres can intersect the surface without being approximately tangent to the surface, as in the case of sliver tetrahedra. Two heuristics are presented in [Attali97] to accommodate 3D sample points. The first heuristic is to find the normalized mesh, and then add the Delaunay triangles

that share two or three edges with the normalized mesh. This approach does not always provide the expected solution, as mentioned in [Attali97]. The second heuristic is a volume based approach where the Delaunay tetrahedra are added based on two rules. The first rule prevents adding the tetrahedron that removes a triangle belonging to the normalized mesh, and the second rule prevents adding of a tetrahedron that isolates a point inside a volume. The boundary of the volume is considered as the desired surface. This heuristic has been shown to work reasonably well on a number of models.

Note that the observation that (most of the) Delaunay circumspheres in three dimensions intersect the surface tangentially is used by Amenta *et al.* [Amenta01, Amenta] in their *power crust* technique for surface reconstruction.

Stable Voronoi Edges: This curve reconstruction algorithm by Weller [Weller97] uses the concept of *stable Voronoi edges*. Let S be a finite sequence of points on a plane. The point set S' is an *ϵ-perturbation* of S if $d(\mathbf{p}_i, \mathbf{p}'_i) \leq \epsilon$ holds for all $\mathbf{p}_i \in S,\ \mathbf{p}'_i \in S'$, $i = 1, \ldots, n$. An edge $\mathbf{p}_i\mathbf{p}_j$ is called *stable* if the perturbed endpoints $\mathbf{p}'_i$ and $\mathbf{p}'_j$ in every ϵ-perturbation are also connected by an edge of the Delaunay triangulation. It turns out that for a sufficiently sampled curve in the plane, only the edges connecting the adjacent points on the curve are the stable edges. The stability of an edge can be checked in time $O(|\text{Voronoi Neighbors}|)$.

The extension to this approach to three dimensions is not straightforward again due to the presence of sliver tetrahedra – an experience similar to that of Attali [Attali97], and Amenta *et al.* [Amenta98b].

3.2 Reconstruction Heuristics without Sampling Conditions

Reconstruction algorithms that do not specify the required sampling or prove the validity of the reconstructed surface are termed as surface reconstruction heuristics. In this section, I give a brief survey of existing reconstruction heuristics.

Algorri and Schmitt [Algorri96]: This method is based on voxel based spatial subdivision. In the first step, the rectangular bounding box of the given data set is subdivided by a regular voxel grid. In the second step, the algorithm extracts those voxels that are occupied by at least one point of the sample set. In the third step, the "outer" quadrilaterals of the selected voxels are taken as a first approximation of the surface. These quadrilaterals are subdivided into triangles, and the vertices of these triangles are warped appropriately based on the position of the sample points, to get a more pleasing surface. Clearly, identifying an "outer" quadrilateral is the most important step in this method. Unless the

voxels that contain the sample points are next to each other and bound a volume, "outer" cannot be defined. Further, we can see that this method is applicable only to data points that lie on fixed grid points, such as medical imaging data, and the data points have to be sufficiently close to each other and the size of the voxel appropriately chosen so that the set of voxels bound a volume.

Hoppe *et al.* **[Hoppe92, Hoppe93]** also use a voxel-based subdivision method. This method computes the signed distance of the vertices of the voxel from an estimated surface and chooses those voxels which have vertices of opposite signs. The surface passes through these voxels. The surface is then reconstructed using marching cubes algorithm. The distance of the voxel vertices from the surface is estimated as follows. First the tangent plane at every input point is estimated using the k-nearest neighbors of the point. Then these normal vectors are consistently oriented by constructing a *Riemannian graph* and propagating the direction of the normal vector from one vertex to another in the graph. Once the tangent planes with appropriate directions of the normals have been estimated, the signed distance of a query point from the surface is estimated to be the distance to the tangent plane at the closest data point.

There are a few algorithms that decompose the space into cells, remove those cells that are not in the volume bounded by the sampled surface and create the surface from the selected cells. Such works include [Bajaj95, Boissonnat84, Veltkamp95].

Boissonnat's Volume Oriented Approach [Boissonnat84] starts with the Delaunay tetrahedralization of the input set of points. Then the tetrahedra that satisfy certain conditions are removed from the boundary of the convex hull. The tetrahedra with two faces, five edges and four points, or one face, three edges and three points on the boundary of the current polyhedron are eliminated. The algorithm stops when all points lie on the surface. The order of removal of tetrahedron is based on decreasing *decision value* which is the maximum distance of a face of the tetrahedron to its circumsphere. The reconstruction of models with genus greater than zero using this method is not reliable. The method proposed by **Isselhard, Brunnett, and Schreiber [Isselhard97]** improved Boissonnat's method to reconstruct objects with non-zero genus, by allowing a few more types of tetrahedra to be removed. The method introduced by **Bajaj** *et al.* **[Bajaj95]**, which uses α*-solids*, is a combination of this Boissonnat's method and Edelsbrunner *et al.*'s α-shapes.

Boissonnat's Surface Contouring Algorithm [Boissonnat84], which is an example of a direct reconstruction technique, also uses a locally estimated tangent plane for the surface reconstruction like Hoppe *et al.* The algorithm starts with an edge, usually the shortest edge among all possible edges

between the input set of points. Then a triangle is attached to this edge by choosing the third vertex for this triangle. The choice of the third vertex is made by projecting the points in the neighborhood of this edge onto the estimated tangent plane. The point that makes the maximum angle when connected to the end points of this edge is chosen to be the third vertex of the triangle incident on this edge. This process continues till there is no free edge. This method can reconstruct only surfaces with no boundaries. Another method similar to the one described above is the **Spiraling-Edge triangulation** proposed by Crossno and Angel [Crossno99]. This approach works by creating a star-shaped triangulation between a point and its neighbors. Like other heuristics described in this section, this method also does not provide any theoretical foundation for the actual triangulation computed or express any sampling requirement a for correct triangulation.

Other notable approaches include Veltkamp's [Veltkamp95] γ-indicator, Bittar *et al.*'s [Bittar95] surface reconstruction by medial axis, Roth and Wibowoo's [Roth97] voxel based method, Mencl and Muller's [Mencl95, Muller93, Mencl98a, Mencl98b] method based on the Euclidean minimum spanning tree, and Fua and Sander's [Fua91, Fua92a, Fua92b] method on clustering of points. The approach of Curless and Levoy [Curless96] is fine-tuned for laser range data. This method is also well suited for handling very large organized data sets.

Warping-based reconstruction technique deforms an initial surface to give a good approximation of the input point set. This technique is particularly suited if a rough approximation of the desired shape is already known. Terzopoulos *et al.* [Terzopoulos88] use *deformable superquadrics* to fit the input data points. A different approach to warping with *oriented particles* was suggested by Szeliski *et al.* [Szeliski92] . By modeling the interaction between the particles (sample points), they construct the surface using forces and repulsion.

The work presented in this dissertation can be considered as a direct method for surface reconstruction. I also provide the sampling condition to reconstruct surfaces, and most importantly, surfaces with boundaries. Sampling conditions not only ensure topologically correct reconstruction, but also ensure minimum geometric deviation of the reconstructed surface from the original model. Hence, the sampling condition would at least demand a minimum sampling to ensure topologically correct reconstruction, and any additional sampling over this minimum requirement would make the reconstructed model a better geometric approximation of the original model. This is true in case of models without boundaries. In Chapter 5, we will see more about this philosophy and prove that this need not be true when considering models with boundaries.

CHAPTER 4

LOCALIZED DELAUNAY TRIANGULATION

In the previous chapter we discussed various surface reconstruction algorithms in the literature. In this chapter I present my algorithm for surface reconstruction called the Localized Delaunay Triangulation. Unlike existing algorithms, this algorithm works on a manifold with or without boundaries. I show the efficiency of this algorithm using empirical results. Further, the required sampling conditions to ensure correct surface reconstruction using this algorithm is detailed in Chapter 6. The algorithm detailed in this chapter and a preliminary version of the sampling conditions are presented in Gopi *et al.*[Gopi00].

4.1 Overview of the Algorithm

Our surface reconstruction algorithm takes as input, a set of unorganized 3D points S sampled from a manifold M with or without boundary, with no other additional information like normal vector . The output of the algorithm is a set of triangles, which defines a manifold L with or without boundary, passing through the input set of points. In Chapter 7, I prove that, under certain assumptions, L is homeomorphic to M.

This reconstruction algorithm uses a projection-plane technique where the neighbors of a vertex **p** in the final triangulation are computed by projecting **p**'s spatially close points onto the tangent plane at **p**, $T_{\mathbf{p}}$. This can be classified as a local triangulation technique as the algorithm *does not* take into account *all* the input points at the time of finding the neighborhood of one sample.

The surface reconstruction algorithm goes through six major steps: collect candidate points for normal estimation, normal estimation, candidate point selection, projection of candidate points onto the tangent plane, Delaunay neighbor computation, and finally the triangulation. This section gives a brief description of each the above steps.

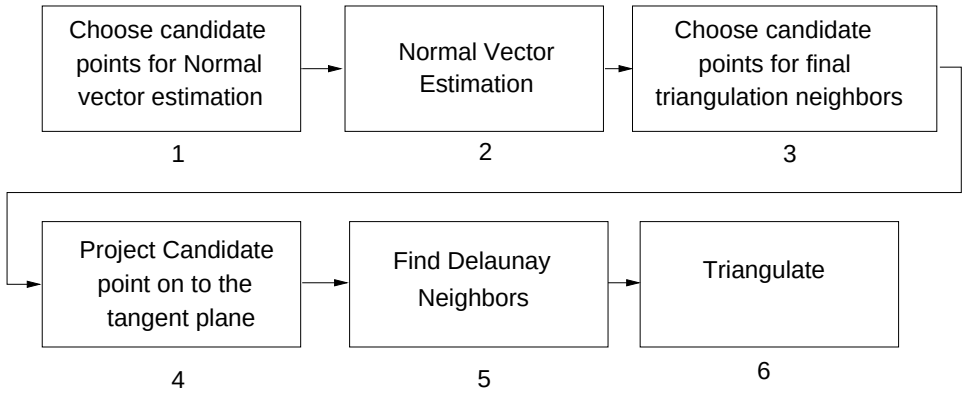

Figure 4.1: *Pipeline of the operations of the Localized Delaunay Triangulation algorithm.*

Candidate Points for Normal Estimation: (Step 1 in Figure 4.1) The first step is to find spatially close neighbors to every sample point. Using these neighbors, the normal vector at every sample is estimated.

Normal Estimation: (Step 2 in Figure 4.1) The next step is to estimate the unit normal at all sample points in S. This step also consistently orients the normal vectors of the sample points to get an orientable manifold. Steps 1 and 2 are performed only if the normal vector information is not part of the input.

Candidate Points Selection: (Step 3 in Figure 4.1) This step chooses those points that might be *possible* neighbors to a vertex in the final triangulation. The candidate point set for every sample **p** is denoted by $P_{\mathbf{p}}$.

Projection of Candidate Points: (Step 4 in the Figure 4.1) Each of the candidate points in the set $P_{\mathbf{p}}$ is mapped onto the tangent plane at **p**, $T_{\mathbf{p}}$. The mapping is done by a simple rotation of the vector from **p** to a candidate point about a well defined axis. The axis of rotation for a candidate point **q** is the line on $T_{\mathbf{p}}$ that is perpendicular to the vector from **p** to **q**. The set of these mapped candidate points on the tangent plane is denoted by $P^T_{\mathbf{p}}$.

Delaunay Neighbor Computation: (Step 5 in Figure 4.1) In this step, the 2D local Delaunay neighbors of every sample **p** is computed from the set $P^T_{\mathbf{p}}$ in its tangent plane, $T_{\mathbf{p}}$.

Triangulation: (Step 6 in Figure 4.1) The final surface triangulation is determined from the Delaunay neighborhood relationship around each sample determined in the previous step . Care is taken to avoid certain inconsistent configurations to arrive at the final triangulation.

4.2 Estimation of Vertex Normal

The first step in our algorithm is to find the unit normal, and thus the tangent plane of the surface, at every sample point **p**, if the normal vector information is not part of the input. The normal vector is computed using the k-*nearest* neighbors of **p** (Step 1 in Figure 4.1). The Euclidean metric is used to measure distances to collect the k-*nearest* neighbors. Given only a set of points around **p**, we need to define a vector that is a good representative of the normal to the surface M at **p**.

Definition: The normal vector $N_{\mathbf{p}}$ is the unit vector that minimizes the variance of the dot product between itself and the vectors from **p** to its k-*nearest* neighbors. If D_i is the dot product between $N_{\mathbf{p}}$ and the vector from **p** to the ith closest neighbor, then $N_{\mathbf{p}}$ minimizes

$$\frac{\sum_{i=1}^{k} (D_i - \frac{\sum_{i=1}^{k} D_i}{k})^2}{k}$$

Let me justify the above definition of the normal. Let us call the point **p** at which the normal has to be found as a *reference point*. There are two extreme configurations of sample points in the local neighborhood around the reference point **p**. One configuration makes **p** a vertex with very high curvature, and the other configuration makes **p** a vertex with zero curvature. If the reference point with high curvature is the apex of a cone and the neighbors are on the circular rim of the base, the expected normal at the apex is the direction of the cone axis. If the reference point with zero curvature is part of a plane then the reference point's normal is the normal to the plane. Under both these cases, the above definition of normal vector defines the expected normal. Let us see how to compute such a normal (Step 2 in Figure 4.1) .

If the k-*nearest* neighbors are $\mathbf{q}_1$ to $\mathbf{q}_k$, then the vectors from **p** to its k-*nearest* neighbors are $\vec{V_i} = \mathbf{q}_i - \mathbf{p}$, $1 \leq i \leq k$. We want to find $N_{\mathbf{p}}$ such that it minimizes

$$\frac{\sum_{i=1}^{k} (D_i - \frac{\sum_{i=1}^{k} D_i}{k})^2}{k} \tag{4.1}$$

where $D_i = N_{\mathbf{p}} \cdot \vec{V_i}$. Removing the scale factor $1/k$, we get

$$\min \left(\sum_{i=1}^{k} \left(N_{\mathbf{p}} \cdot \vec{V_i} - \frac{\sum_{i=1}^{k} N_{\mathbf{p}} \cdot \vec{V_i}}{k} \right)^2 \right) \tag{4.2}$$

$$= \min \left(\sum_{i=1}^{k} \left(\left(\vec{V_i} - \frac{\sum_{i=1}^{k} \vec{V_i}}{k} \right) \cdot N_{\mathbf{p}} \right)^2 \right) \tag{4.3}$$

The vectors $\vec{V_i}$ can be viewed as the coordinates of the k-*nearest* neighbors with **p** as the origin. If **p** is the origin, the centroid of the k-*nearest* candidate points is $C = \left(\sum_{i=1}^{k} \vec{V_i}\right)/k$. Thus, the above expression can be rewritten as

$$\min\left(\sum_{i=1}^{k}((\vec{V_i} - C)\cdot N_{\mathbf{p}})^2\right) \tag{4.4}$$

If A is a $k \times 3$ matrix where $\vec{V_i} - C$ defines the row vectors, then the above expression reduces to (ignoring the square)

$$\min(\|AN_{\mathbf{p}}\|_2) \tag{4.5}$$

Thus this minimization problem can be solved using singular value decomposition [Golub89]. The eigenvector which corresponds to the smallest eigenvalue of A is the normal vector that minimizes the above equation. Hoppe *et al.* [Hoppe92] proposed the use of principal component analysis of a covariance matrix to determine the normals. Even though our formulation is different, it turns out that the resulting normal vectors computed by both methods are the same.

4.2.1 Propagation of Normal Direction

The normal vector found by the above process is correct only up to sign. To find a consistent orientation of the surface, we fix the orientation of one of the normals and propagate this information to the rest of the points.

We use the technique proposed by Hoppe *et al.* [Hoppe92] to do the propagation. Hoppe *et al.* pose this problem as a minimum spanning tree problem, where the vertices of the model are the vertices of the graph, and the edges of the model are the edges in the graph. The weight of the edge between the vertices **i** and **j** is assigned to be $(1 - |N_{\mathbf{i}} \cdot N_{\mathbf{j}}|)$, where $N_{\mathbf{i}}$ and $N_{\mathbf{j}}$ are the normals at the vertices **i** and **j** estimated using the method given in the previous section. The minimum spanning tree of this graph would give the propagation sequence of normals for the consistent orientability of the model. An arbitrary vertex of the graph is assumed to be the root and the normal is propagated to its children recursively. When the normal direction is propagated from vertex **i** to vertex **j**, if $N_{\mathbf{i}} \cdot N_{\mathbf{j}}$ is negative, then the direction of $N_{\mathbf{j}}$ is reversed; otherwise it is left unchanged.

4.3 Candidate Point Selection

This step is similar to clustering steps used by other triangulation schemes [Hoppe92, Heckel98]. The candidate points chosen for a reference point **p** are possible neighbors of **p** in the final triangulation. Incorrect choice of candidate point set might lead to incorrect triangulation. The sampling conditions presented in Chapter 5 ensure that the candidate points chosen as described in this section is correct. This section describes a method to *efficiently* choose a *correct* candidate point set.

In the first stage, we use L_∞ metric to prune our search for candidate points in the spatial proximity of **p**. In this step, our algorithm takes an axis-aligned box of appropriate dimensions centered at **p** and returns all the samples inside it. The way we decide the size of this bounding box, and the data structure used for this pruning are explained later Section 4.3.1.

The second stage uses either a Euclidean metric or another distance function Q to further prune the result of the first stage to arrive at the final set of candidate points of **p**, $P_{\mathbf{p}}$.

Distance Function Q: The distance of **q** from **p**, $Q_{\mathbf{p}}(\mathbf{q})$, is defined as

$$Q_{\mathbf{p}}(\mathbf{q}) \equiv E^2(\mathbf{p}, \mathbf{q}) + T^2_{\mathbf{p}}(\mathbf{q})$$

where $E(\mathbf{p}, \mathbf{q})$ is the Euclidean distance between **p** and **q**, and $T_{\mathbf{p}}(\mathbf{q})$ denotes the distance of **q** from the tangent plane at **p** (as determined by the normal vector at **p**). This distance function is explained in more detail in Section 6.3.2.

Clearly, the normal vector information is required to compute Q. If the normal vector is not given as part of the input, then the Euclidean metric (refer to Chapter 6) is used to measure the distance between two sample points, otherwise Q is used for the purpose. All points less than a particular distance around **p** measured using the Euclidean metric will be inside a *sphere of influence*, and if Q was used to measure distances, then these points will lie inside an *ellipsoid of influence*. As we discussed earlier, the second stage of pruning removes points that lie outside the sphere (ellipsoid) of influence around **p** from the candidate set got after the first step of pruning. The following section discusses the implementation details and the dimensions of the axis aligned bounding box (first stage of pruning) and the sphere (ellipsoid) of influence (second stage of pruning).

4.3.1 Implementation of Candidate Point Selection

Our data structure for the first stage of pruning is a depth pixel array similar to the *dexel* structure proposed in [Hook86]. The *dexel* data structure consists of a 2D pixel array where in each pixel

multiple points can be stored in a sorted order. Given the axis aligned bounding box of the complete input sample set, the dexel array can be considered as the discretization (pixelization) of the face perpendicular to the *z*-axis. All data points are orthographically projected in the positive *Z* direction. The points mapped on to the same pixel are sorted by their depth (*z*) values.

By using this data structure, the search for candidate points is limited to the pixels around the pixel where **p** is projected. The size of the bounding box for choosing the candidate points using L_∞ metric and the radius of the *sphere* (or *ellipsoid*) *of influence* used in the second stage of pruning are determined by the distance to the closest point from **p**. Let s be the distance to the closest sample point from **p** as measured by a distance function, then the bounding box and sphere(ellipse) should enclose all points that are at a distance less than $m \times s$ from **p**. The constant m is 2.0 if Euclidean metric is used, and is $\sqrt{5.83}$ (refer to Theorem 7, Equation 6.27) if Q is used, and this determines the dimension of the sphere(ellipsoid) of influence. The dimension of the axis aligned bounding box (cube) centered at **p** is $2ms$.

The points that are inside the sphere (ellipsoid) of influence are possible Delaunay neighbors of **p** in the tangent plane of **p**, $T_{\mathbf{p}}$.

Note: If the distance function Q (Chapter 5) is used, the candidate set of **p** is pruned further after rejecting the points that are outside the ellipsoid of influence around **p**. A sample point **q** is rejected from the candidate set of **p** if the distance $Q_{\mathbf{p}}(\mathbf{q})$ is greater than $L^2_{\mathbf{q}} + L^2_{\mathbf{q}}\sin^2(\phi) + \frac{3}{2}L^2_{\mathbf{p}} + 2L_{\mathbf{p}}L_{\mathbf{q}}(\sin(\phi) + \cos(\phi))$, where $L^2_{\mathbf{p}}$ and $L^2_{\mathbf{q}}$ are distance to the closest samples from of **p** and **q** respectively, and ϕ is the angle between the normal vectors at **p** and **q** (refer to Equation 6.23).

After the second stage of pruning, there might be a case where a sample **p** is in the candidate set of a sample **q**, but **q** is not in the candidate set of **p**. Consistency between the candidate sets is essential for the consistency of results of subsequent steps in this reconstruction algorithm (also refer to Theorem 7). To make the candidate sets $P_{\mathbf{p}}$ and $P_{\mathbf{q}}$ consistent, the sample **q** is added to $P_{\mathbf{p}}$. The final candidate set of each sample point is got after this sharing of candidate set information between the sample points.

4.4 Projection of Candidate Points

The next step is to project the candidate set $P_{\mathbf{p}}$ of each vertex **p** found in Section 4.3 onto the tangent plane at **p**, $T_{\mathbf{p}}$. The tangent plane $T_{\mathbf{p}}$ is defined by the normal vector estimated in Section 4.2. Let V_i be the vector from **p** to the ith candidate point. The mapping of candidate points onto $T_{\mathbf{p}}$ is done by

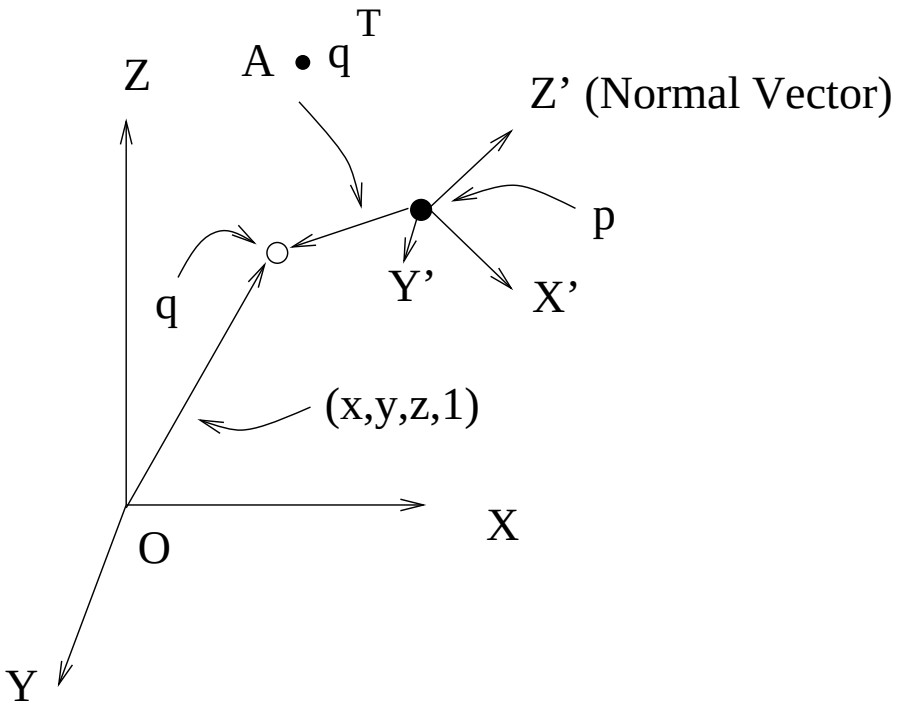

Figure 4.2: *Let the homogeneous coordinates of* **q** *be* $(x, y, z, 1)$ *with respect to the coordinate system at the origin* O*. The coordinate of* **q** *with respect to the local coordinate system at* **p** *is given by* $A \cdot (x\ y\ z\ 1)^T$*, where* A *is a 4x4 matrix computed as shown in Section 4.4.1.*

rotating the vector V_i about **p** with the line on $T_{\mathbf{p}}$ that is perpendicular to V_i as the axis. The vector V_i will sweep a sector of a disk on the plane defined by the normal $N_{\mathbf{p}}$ and V_i, due to this rotation. The rotation of V_i retains its length on the tangent plane. The set of all candidate points of **p** projected in this way onto its tangent plane is denoted by $P^T_{\mathbf{p}}$.

This rotation operation can be considered as flattening the surface around **p** onto $T_{\mathbf{p}}$ such that the distances of points on the surface from **p** remains the same after the flattening. The inverse of this mapping is called the *exponential map* in the Riemannian Geometry literature [O'Neill97]. The Voronoi diagrams and Delaunay triangulation are defined for Reimannian surfaces [Leibon00]. The rotation operation (inverse exponential map), instead of an orthogonal projection, makes the Voronoi diagram of the points in $P^T_{\mathbf{p}}$ and **p** on the tangent plane a better approximation of the projection of the Voronoi diagram of the candidate points of **p** on the Riemannian surface described in [Leibon00]. To be precise, consider the Voronoi curves V_C on the surface around **p** separating **p** and the points in $P_{\mathbf{p}}$, and their mapping onto $T_{\mathbf{p}}$ using the inverse exponential map. Let the mapped curves on $T_{\mathbf{p}}$ be V^T_C. The Voronoi edges on the tangent plane between **p** and points in $P^T_{\mathbf{p}}$ are tangents to V^T_C.

4.4.1 Implementation of Projection of Candidate Points

Let the global coordinate system in which the input set of points is represented be (X, Y, Z) (refer to Figure 4.2). The first step is to find the local coordinate system, say (X', Y', Z'), at the reference point **p**. The normal vector at **p** is the Z'-axis. Any vector on the tangent plane at **p** can be chosen as the X'-axis, and based on this choice, the Y'-axis is computed as the cross product of Z'-axis and X'-axis. The X'-axis is computed as follows.

Let the unit Z'-vector be $(z[0], z[1], z[2])$ with respect to the global coordinate system. Let $d = 1/\sqrt{z[0]^2 + z[1]^2}$. The X'-vector $(x[0], x[1], x[2]) = (z[1] \times d, -z[0] \times d, 0)$. It can be seen that X' is a unit vector, and $X' \cdot Z' = 0$.

Any point **q** with homogeneous coordinates $(x, y, z, 1)$ in the global coordinate system can be represented in the local coordinates by a linear transformation. The transformation matrix A is given by

$$A = \begin{pmatrix} x[0] & x[1] & x[2] & -(X' \cdot \mathbf{p}) \\ y[0] & y[1] & y[2] & -(Y' \cdot \mathbf{p}) \\ z[0] & z[1] & z[2] & -(Z' \cdot \mathbf{p}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.6}$$

The local coordinates of **q** are $(x', y', z', 1) = A \cdot (x, y, z, 1)^T$.

Rotating the vector $\vec{\mathbf{pq}}$ to the tangent plane of **p** is same as projecting **q** onto the tangent plane of **p** orthogonally and then scaling the vector from **p** to the projected point to the length equal to $|\vec{\mathbf{pq}}|$. The local coordinates of the orthogonally projected point of **q** onto the tangent plane of **p** is $(x', y', 0)$. The image of **q** after mapping onto the tangent plane by rotation is given by

$$\mathbf{q}' = \sqrt{\frac{(x'\ y'\ z') \cdot (x'\ y'\ z')}{(x'\ y'\ 0) \cdot (x'\ y'\ 0)}}(x'\ y'\ 0)$$

4.5 Delaunay Neighborhood Computation

The next step is to find the Delaunay neighbors of each vertex that are preferred neighbors in the final triangulation of the surface. To compute the Delaunay neighbors, first the projected candidate points $P^T_{\mathbf{p}}$ on the tangent plane are ordered by angle around **p**. Then a fast Delaunay neighbor computation is done using this ordering. These two steps are described below.

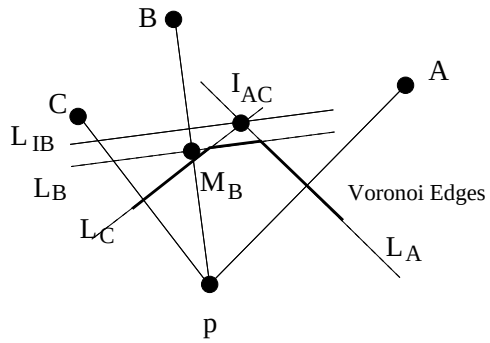

Figure 4.3: *Finding Delaunay neighbors. Given points A, B and C, check whether B is a Delaunay neighbor of* **p**. *The thick edges are the local Voronoi edges around* **p**.

4.5.1 Fast Ordering by Angle

The candidate points are transformed to the local coordinate system and are mapped to $T_{\mathbf{p}}$ as explained in the previous section. These points have 2D coordinates, with an implicit definition of coordinate axes. The set $P_{\mathbf{p}}^T$ is partitioned on the basis of the quadrant where the points lie on $T_{\mathbf{p}}$ in the local coordinate system. The points within each quadrant are ordered by angle using a simple method as explained below.

The square of the sine of the angle can be computed without any use of square-root or trigonometric functions, as $x'^2/(x'^2+y'^2)$. Points in each quadrant in the projection plane are ordered separately using the square of the sine values and finally merged to get the complete ordering of points around **p**.

4.5.2 Delaunay Neighbor Computation

This section describes the details of finding the neighbors for the reference point **p** from the candidate point set using 2D local Delaunay triangulation. The Delaunay neighbors of **p** is found from the angle-ordered candidate vertices around **p** as follows. The basic function of the algorithm takes the reference point and three consecutive points in angle ordered set, A, B, and C, to check whether the middle point (B) could be a Delaunay neighbor of **p** in the presence of A and C. Figure 4.3 explains this algorithm pictorially.

bool CheckDelaunay(**p**, A, B, C)

{

L_A = Perpendicular bisector of the line segment $\mathbf{p}A$.

L_B = Perpendicular bisector of the line segment $\mathbf{p}B$.

L_C = Perpendicular bisector of the line segment $\mathbf{p}C$.

I_{AC} = Intersection point of L_A and L_C.

L_{IB} = Line parallel to L_B, and passing through I_{AC}.

M_B = Mid point of the line segment $\mathbf{p}B$.

If both **p** and M_B lie on the same side of L_{IB}, then

B is a local Delaunay neighbor to **p** when compared with A and B.

return $TRUE$.

else

B is not a Delaunay neighbor to **p**.

return $FALSE$.

}

If $\mathbf{q}_1 \ldots \mathbf{q}_n$ are the ordered projected candidate points of **p**, then the above function is called for every triplet $\mathbf{q}_{i-1}, \mathbf{q}_i, \mathbf{q}_{i+1}$. If the test passes, then the next triplet $\mathbf{q}_i, \mathbf{q}_{i+1}, \mathbf{q}_{i+2}$ is tested. But, if the test fails, then $\mathbf{q}_i$ is rejected, and the algorithm backtracks with the call $\mathbf{q}_{i-2}, \mathbf{q}_{i-1}, \mathbf{q}_{i+1}$ to re-evaluate $\mathbf{q}_{i-1}$ for its validity. As the ordering of the vertices is by angle, and hence is cyclic, any point can be chosen as the first point $\mathbf{q}_1$. We choose the closest point to **p** as $\mathbf{q}_1$, as it is always a Delaunay neighbor. We also use $\mathbf{q}_1$ as a terminating condition for the backtracking step. The implementation of the above algorithm is optimized for speed, and each call to the above function takes less than 35 mathematical operations.

4.6 Triangulation

The Delaunay neighbors of each vertex are found and stored in a list ordered by angle. Vertices A, B, and C form a triangle if and only if $\{B, C\}$, $\{C, A\}$, and $\{A, B\}$ are consecutive Delaunay neighbors in the ordered neighbor lists of A, B, and C respectively (refer to Section 7.3.1 for justification and more detail). There are a few degenerate cases and problems arising out of improper sampling, and due to the projection onto different, though close, tangent planes.

In the triangulation stage of our algorithm, assume that A, B, C, and D form a quadrilateral as shown in Figure 4.4. There might be a case where both BD and AC are Delaunay edges forming overlapping triangles. There might also be a case where neither BD nor AC is a Delaunay edge forming a hole (refer Figure 4.4).

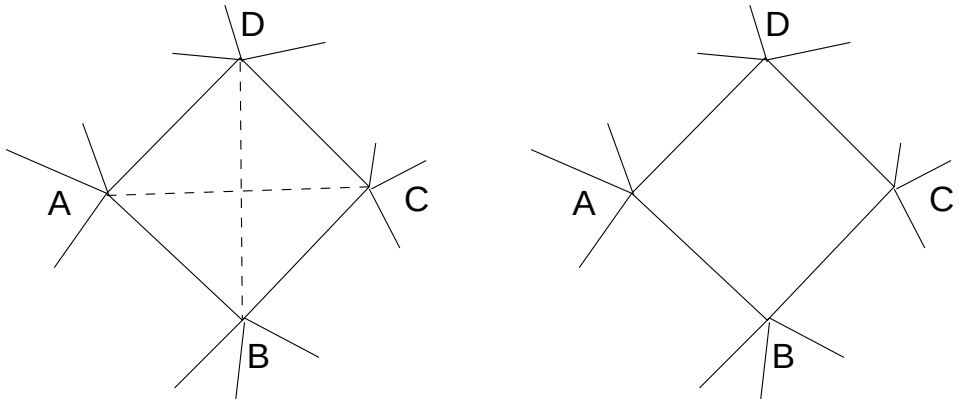

Figure 4.4: *Left: AC and BD are Delaunay neighbors of each other. Right: Neither AC nor BD are Delaunay neighbors of each other.*

In the first case, a simple contention detection and removal method is used to disambiguate the triangulation. For example, in our implementation, the triangulation around a vertex with lowest index (say A) is given preference to complete the triangulation, and hence triangles ABC and ACD are chosen.

In the second case where the hole is left due to lack of Delaunay edges, we have to justify that the hole is due to the inconsistent Delaunay neighborhood relationship rather than an actual hole in the model. The sampling conditions described in Chapter 6 ensures that the non-adjacent points on the boundary are not candidate neighbors of each other. If the boundary points of the hole are candidate neighbors of each other, then the hole is due to the inconsistent Delaunay neighborhood relationship. Such holes are filled using a simple polygon triangulation algorithm after projecting these points along the boundary to a plane defined by the average of normal vectors at these points.

As the Delaunay triangulation is computed locally in 2D, a dimension lower than the one in which the surface exists, this method of constructing the triangulation is called the *Lower Dimensional Localized Delaunay Triangulation*.

4.7 Results

We ran the implementation of this algorithm on various models, and the results have been documented in Table 4.1. All timings are measured on an SGI-Onyx with an R10000 processor running at 194 MHz. The color plates at the end of this dissertation show the results of our algorithm on a few models. The graph in Figure 4.5 shows that the performance of our algorithm is nearly linear with respect to the number of points in the input set. The computation of normal takes around half the total

Model	No. of points	No. of Triangles	Normal Comp.Time (in secs)	Triangulation Time (in secs)	Total Time (in secs)
Face	12837	25438	1.85	2.55	4.42
Club	16864	33643	2.40	2.30	4.75
Foot	20021	39862	2.88	3.30	6.22
Lower Jaw	25362	49061	3.50	2.70	6.26
Upper Jaw	27830	54458	3.91	2.95	6.92
Oil Pump	30937	61772	4.37	7.30	12.80
Bunny	34834	69630	5.03	5.00	10.64
Skidoo	37974	75364	5.25	4.58	9.85
Phone	83034	165981	12.18	13.40	25.67

Table 4.1: *Performance of our algorithm*

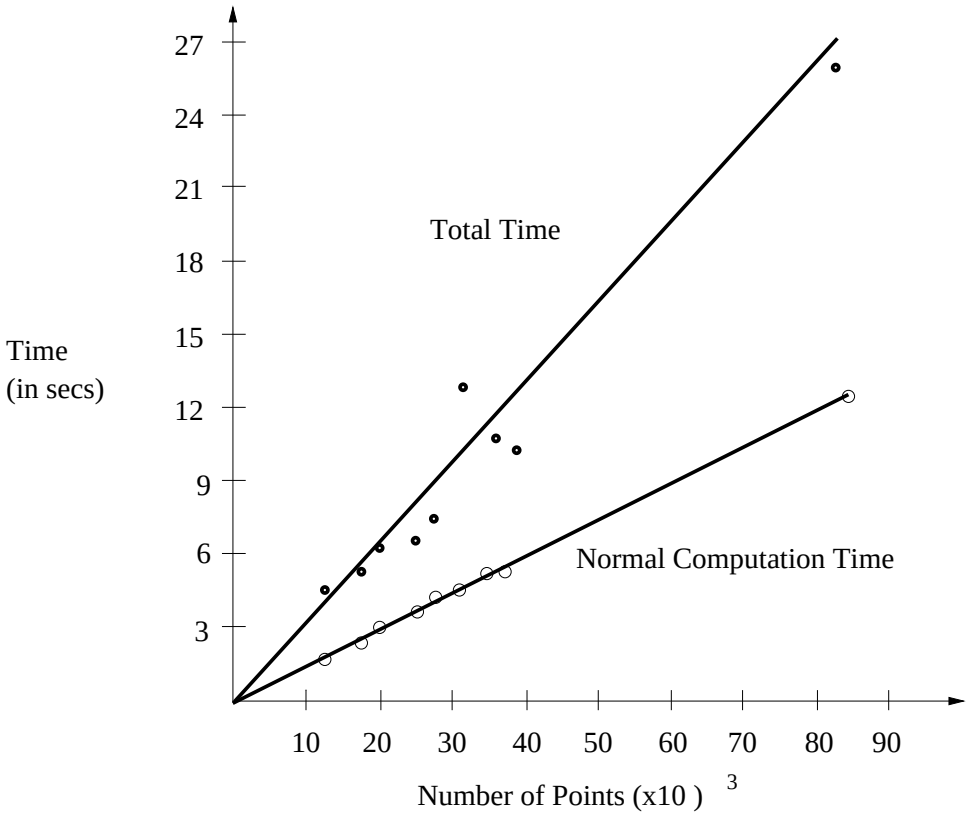

Figure 4.5: *Graph shows the total computation time and the computation time of the normal vector with respect to the number of points in the input model.*

time. The candidate point selection process using the dexel array is dependent on the depth complexity of the model in the projection direction. This has had its effect on the total time (refer to Oil-pump model) of reconstruction.

The total reconstruction time of our algorithm is in terms of number of seconds, whereas the running time on a similar platform for other implementations of algorithms that guarantee correct reconstruction is in terms of number of minutes. Further, our algorithm and implementation are modular and are suitable for parallelization with minor changes.

It is important to note that none of the models are sampled specially for our algorithm. Specifically, the inside-looking-out laser scan data of the Car and Room models (Courtesy Lars Nyland) have a lot of noise and perturbations in the location of the samples. Further, the Car model is a merged data set of four different laser scans. These models have been reconstructed satisfactorily, though with a number of small boundaries.

4.8 Summary

I have presented a fast and reliable surface reconstruction algorithm, the Localized Delaunay Triangulation algorithm. This algorithm can reconstruct surfaces both with and without boundaries. Further, using empirical results, this algorithm is shown to be efficient and have a near linear performance. In the ensuing chapters I develop the theory of sampling, and the sampling required by this algorithm to produce correct results. Further, I prove that the Localized Delaunay Triangulation presented in this chapter is a correct triangulation.

CHAPTER 5

THEORY OF GEOMETRIC SAMPLING

5.1 Introduction to Sampling

Sampling is a process of discretizing a continuous surface or a curve into discrete samples. The reconstruction process uses these samples to reconstruct the original continuous surface. To ensure correct reconstruction of the original surface, certain restrictions are imposed on the discretization process. These restrictions are called *sampling conditions* or *criteria*.

For example, in signal processing, one of the well known sampling criterion is the *Nyquist criterion* to sample a continuous signal, and this is used in digitization of audio, video, and is applicable to images also. According to the Nyquist theorem [S.84, B.89], the discrete time sequence of a sampled continuous band limited signal $V(t)$ contains enough information to reproduce the function $V(t)$ exactly provided that the sampling rate is at least twice that of the highest frequency contained in the original signal $V(t)$. To develop such a sampling condition for geometric sampling of curves and surfaces, we have to identify the parameter in curves and surfaces that is equivalent to frequency in signals.

The reconstruction of a signal that is sampled based on the Nyquist frequency is dependent on Fourier transforms. By this transform and appropriate band-pass filters, it can be proved that the reconstructed signal is exactly the same as the original signal, and not an approximation. This dissertation deals with geometric sampling of a surface and reconstruction of the surface from the sample points. The reconstructed surface is a correct surface if it is *topologically equivalent* and *geometrically close* to the original surface. In other words, we are not reconstructing exactly the same original surface, though such reconstruction might be possible under various assumptions about the surface. Because of these reasons, and the natural differences between signals and surfaces, we cannot directly relate the Nyquist sampling and Fourier signal reconstruction, with the sampling and

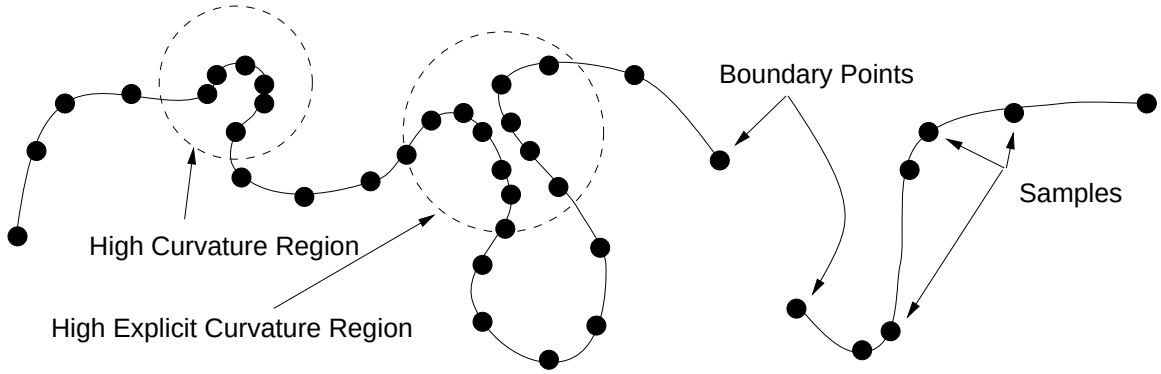

Figure 5.1: *Sampling of a Space Curve.*

reconstruction of a surface. Nevertheless, I try to draw an analogy between signals and surfaces. To simplify further, I use curves instead of surfaces in this analogy.

5.1.1 Signals and Curves

Curvature has been used successfully in differential geometry to study the shape of an object. Hence to get a *geometrically close* approximation of the curve, we need more samples in high curvature region and less samples in low curvature region.

Curvature is an implicit property and does not capture an explicit feature, like self-intersection, that is due to the space in which the curve is immersed or embedded. Though we assume that the mapping is an embedding, which eliminates self-intersection, we have to address the issue of capturing certain explicit features for *topologically correct* reconstruction.

Reconstruction of a curve connects adjacent samples. Hence adjacency relationships between samples have to be established unambiguously for correct curve reconstruction. In the case of signals there is an implicit temporal ordering of samples. In the case of curves, in the absence of explicit ordering of samples, various heuristics are used to find an ordering using spatial proximity of samples. When two different parts of the curve are spatially close to each other as in Figure 5.1, these heuristics might fail if the distance between adjacent samples is greater than the distance between different parts of the curve. Spatial proximity of different parts of the curve is measured using *explicit curvature*.

Definition: *Explicit curvature at a point* **p** *on the curve* is defined as the reciprocal of the positive minimal radius of a circle centered at **p** whose intersection with the curve is not homeomorphic to real line. Note the similarity of this definition with the definition of (implicit) *curvature* which is the reciprocal of the radius of the osculating circle.

The sampling density should be proportional to the explicit curvature as well as implicit curvature at any region of the curve. In summary, the 'frequency' of a signal can be considered to be an analogy to the explicit and implicit curvatures at the points on the space curve.

5.2 Medial Axes and Curvatures

One of the best indicators of local explicit and implicit curvatures is the medial axis of the curve/surface, since it unifies both these concepts. The closest distance of a point on the surface to the medial axis is less than the minimum of the reciprocals of the implicit and explicit curvatures at that point. This fact is exploited by Amenta *et al.* [Amenta98a, Amenta98b] for providing the sampling condition for reliable reconstruction of curves and surfaces. The sampling condition of Amenta *et al.* says that for every point **a** on the curve or surface, there should exist a sample point at a distance less than a fraction of the closest distance of **a** to the medial axis. This sampling condition yields a sampling density that is proportional to the maximum of the implicit and explicit curvatures.

During the reconstruction process, the medial axis of a curve or a surface cannot be computed with just a set of sample points. In their reconstruction algorithm, Amenta *et al.* use the fact that the Voronoi vertices of the samples lie close to the medial axis of the densely sampled curve (refer to Chapter 3). A few more algorithms [Amenta01, Amenta, Amenta00] have been designed recently on the same lines, but all these algorithms use the same underlying relationship between the medial axis and the Voronoi vertices. In the next section, I show that there is a class of surfaces for which these algorithms cannot be used, as the relationship between the medial axis and Voronoi vertices ceases to exist in these surfaces.

5.3 Medial Axes and Voronoi Vertices

Algorithms that use medial axis for sampling rely on the estimation of medial axis for reconstruction. The 3D medial axis *computation* algorithms are so uncommon that there are only a few for polygonal models [Culver00] and none if only the sample points of the model are given. The only *estimation* technique for medial axis from sample points uses Voronoi vertices, and is applicable only when samples are taken from closed models. In this section I show that this estimation technique is not reliable when the sample points are taken from objects with boundaries.

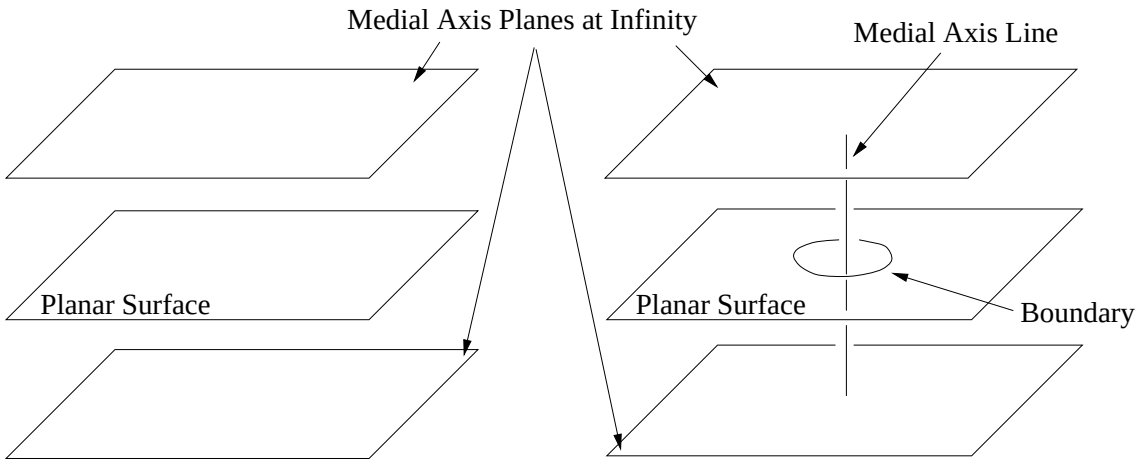

Figure 5.2: *Medial axis of a plane: Left – Without Boundary, Right – With Boundary.*

All Voronoi vertices of a samples of a 2D curve lie close to the medial axis (MA) if the curve is sufficiently sampled. For samples of a 3D object, it is shown in the literature that there might be Voronoi vertices close to the surface of the model and far away from the MA. Based on some heuristics, such Voronoi vertices are removed by the reconstruction algorithms. These algorithms assume that the underlying surface is a closed surface, and all sections of the MA are well represented by the Voronoi vertices (Figure 5.2).

When we consider a surface with boundary, it is not only true that not all Voronoi vertices lie close to the MA, but also that not all sections of the MA are represented by the Voronoi vertices. For example, consider a planar region and its sampling (Figure 5.2). The MA of this plane are two parallel planes at infinity, one above and another below the given plane. If there is a circular hole in the plane, then a line of the MA passes through the hole connecting the two planes at infinity. However dense the sampling is along the boundary and the given plane, there will be no Voronoi vertex in this connecting line. Even if there are a few Voronoi vertices on this connecting line due to perturbation of points in the boundary, the reconstruction algorithms cannot distinguish these vertices from the ones that lie close to the surface and far away from the MA, and hence will be removed.

Hence, for a surface with boundary the relationship between the Voronoi vertices of samples and the medial axis of the surface ceases to exist, and any algorithm relying on this relationship cannot reliably identify or reconstruct the boundary of the surface. The next section further explores the class of surfaces with boundaries.

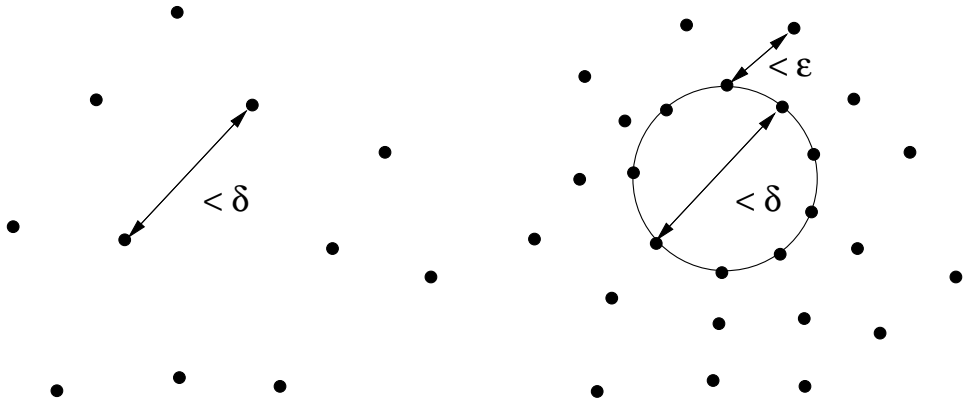

Figure 5.3: *Sampling: Left – Without Boundary, Right – With Boundary.*

5.4 Sampling Surfaces with Boundaries

Conditions for sampling a surface depend on the reconstruction algorithm used. The few algorithms that specify sampling conditions focus on surfaces with no boundaries. In this section I show that sampling surfaces with no boundaries is mathematically different from sampling surfaces with boundaries.

Almost all of the reconstruction algorithms designed till today have the "Nyquist philosophy" that the denser the sampling, the closer the approximation of original surface with the reconstructed surface. Hence the sampling conditions directly or indirectly specify the maximum distance between close samples, or the minimum required sampling density.

For different reconstruction algorithms there will be some minimum distance between sample points on a surface boundary for which the algorithm will correctly reconstruct the boundary. We call this the boundary size, and its definition is based on the reconstruction algorithm used. For example, in terms of the sampling conditions prescribed by [Amenta98a, Amenta01], the boundary size is the minimum distance of boundary points to the medial axis. In terms of alpha shapes, the boundary size is the minimal value of alpha that cannot reconstruct a topologically correct and geometrically close surface.

Theorem 3 *Conditions on the minimum required sampling density are not sufficient to design reliable algorithms that reconstruct surfaces with boundaries using only the point samples and normal vectors as input.*

Proof: Let us assume we are given a hypothetical surface reconstruction algorithm A that claims to reliably reconstruct surfaces both with and without boundaries. Let A be based on a sampling

condition C that directly or indirectly specifies just the maximum distance between samples in a given region. Let δ be the maximum distance between the closest samples demanded by C in a specific region on M (Figure 5.3). If we consider a boundary B on a surface M of size less than δ (and greater than another constant, say, ϵ), then the sampling demanded would not be able to capture the feature B. Hence C is forced to demand a denser sampling in the region around the boundary B. Let the new limit on maximum distance between closest samples be ϵ which is less than δ. Since, by the new sampling condition, the distance between closest samples is less than both ϵ and δ, the sampling satisfies the conditions required to sample M *both with and without the boundary* B. With no additional information about the presence or absence of boundary, the algorithm A has no way to find out whether the given set of sample points are of a surface with or without boundary. Hence conditions on just the minimum required sampling density is not sufficient for reliable reconstruction of M with B when only the sample points are used during reconstruction. $\diamondsuit$

5.4.1 Analysis of the Algorithm Design Space

Reconstruction algorithms can reliably reconstruct only manifolds (without boundaries) if only the conditions on minimum required density were used for sampling. Consider the diagram in Figure 5.4. We need to design algorithms to reconstruct all instances of the class of orientable manifolds with and without boundaries.

Let us analyze the design space of reconstruction algorithms. There are two stages to the reconstruction process, the sampling stage and the reconstruction stage. To handle surfaces with boundaries, either the sampling condition should be strengthened in the sampling stage, or more information should be provided apart from just the sample points during the reconstruction stage. In this dissertation we do not allow any additional information other than the sample points to the reconstruction process, and hence take the approach of strengthening the sampling conditions. But let us first analyze the option of providing more information to the reconstruction stage.

Trivial (and maximum) additional information for correct reconstruction is the complete connectivity information, which I do not consider in this dissertation for obvious reasons. A "medium-sized" additional information can be in the form of tags for all the boundary points. All tagged points can be used as input to a curve reconstruction algorithm that would reconstruct closed boundary curves. The rest of the connectivity can be found using present (closed) surface

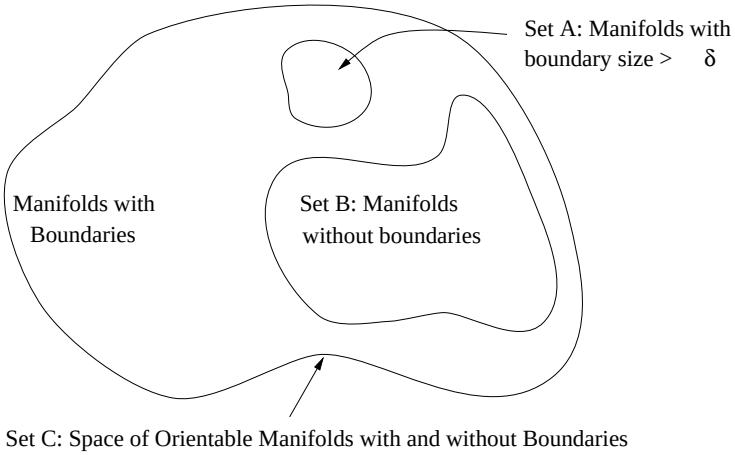

Figure 5.4: *Solution space for different sampling conditions.*

reconstruction algorithms with minor modification. This approach needs further investigation which this dissertation will not address.

The minimum additional information that can be provided is one number, and let that be the smallest boundary size, say δ, of the model. A reconstruction solution can use existing (closed) surface reconstruction algorithms and, as a post processing, remove all edges with length larger than δ. The sampling should be dense enough such that the required edges are ensured to have length less than δ. This approach can reconstruct the instances of surfaces both in sets A and B in Figure 5.4. Reducing the value of δ allows the reconstruction algorithm to handle set B and a superset of set A, but increases the required sampling density. In the limit, to reconstruct all instances of set C, infinite sampling is required, making it theoretically uninteresting. Nevertheless, this approach is suitable for data sets from uniform, minimally controlled, and automatic sampling processes, and where the reconstruction stage is well controlled. For example, large data sets of rooms and outdoors from laser scanners can be handled using this approach. Again, this option is not investigated in this dissertation.

In order to change the sampling to dynamically adapt to boundary-sizes, conditions on sampling should be strengthened. Since we are not allowing any other information other than the sample points for the reconstruction stage, we encode the presence of boundary using the variation in sampling density. To accomplish this, here I establish a relationship between the sampling density and the presence of boundary.

5.4.2 Sampling Condition for Surfaces with Boundaries

To represent boundaries, we have to prevent the addition of certain simplices (triangles) that close holes in the final triangulation. Intuitively, we would not connect two points **p** and **q** that are "far" away. The concept of "farness" is relative, and depends on how close other points are to **p** and **q**. This leads us to our proposed solution. The distance to the farthest point that can be connected to, say, **p** should be related to the distance to the closest point of **p** and vice-versa. If the distance to the farthest and the nearest neighbors are indicators of minimum and maximum sampling densities respectively, then this argument shows that *the boundary can be identified using sampling conditions which expresses relative minimum and maximum sampling densities*. Note in Section 4.3.1 a constant m, the ratio between the minimum and maximum sampling densities, was used to collect the candidate neighbors of a reference point. The sampling should ensure that this relation between minimum and maximum sampling densities is satisfied for correct candidate point selection, and hence a correct triangulation of the sampled surface.

Non-Monotonic Sampling: A sampling condition for signals or closed surfaces prescribes only the minimum required sampling density. Any additional sampling over this minimum requirement would not affect the reconstruction process. For a surface with boundary, let us assume that a sampling of a model as demanded by the sampling condition is provided. Any additional sample might violate the relationship between the distances to the nearest and farthest neighbor, and hence would be a *prohibited sample*. *Prohibited sampling* would lead to ambiguity in deciding the presence of boundary, and hence would trigger the need for more samples to disambiguate the situation. After the introduction of more points, the sampling again becomes a valid sampling. Since the state of the sampling may change between valid and prohibited by increasing the number of samples, I call this sampling condition a *non-monotonic sampling condition*. Sampling conditions for surfaces with boundaries require such non-monotonicity.

5.5 Classification of Surface Reconstruction Algorithms

Based on the above analysis of the design space of the surface reconstruction algorithms, here I classify these algorithms. The classification is based on the changes in the two stages, the sampling stage and the reconstruction stage, over its basic requirements. In the sampling stage, the basic requirement for a sampling condition is to specify the minimum required sampling density. In the reconstruction

Class	Change Sampling	Change Reconstruction	Characteristics/Features
A	No	No	Cannot reconstruct surfaces with boundaries
B	No	Yes	Imposes uniform sampling density
C	Yes	No	Non-monotonic sampling
D	Yes	Yes	*Not explored yet*

Table 5.1: *Classification of surface reconstruction algorithms.*

stage, the basic input is the set of sample points. Changes in these stages include strengthening of the sampling condition in the sampling stage and providing additional information to the reconstruction stage. The Table 5.1 enumerates the classification.

All classes of algorithms, except Class A, can reconstruct surfaces with boundaries. The algorithms of [Amenta98a, Amenta98b, Attali97] belong to Class A. The algorithms based on α-shapes belong to Class B. The value of α can be used as additional information to reconstruct surfaces with boundaries. The method used for reconstructing laser scan data of large areas also belong to Class B, as the boundary size is used as additional information. The algorithm presented in this dissertation is a Class C algorithm. Algorithms in Class D algorithms are yet to be explored.

5.6 Conclusion

In this chapter, I analyzed the failure of Voronoi vertices in its use in approximating the medial axis of surfaces with boundaries, and hence concluded that algorithms that use this relationship are less useful in reconstructing surfaces with boundaries. Further, I also proved that the conditions on minimum required sampling density are not sufficient to reconstruct surfaces with boundaries reliably. Analysis of the design space of the surface reconstruction algorithms led to the classification of these algorithms. In the process, I eliminated the methods that require additional information other than sample points for reconstruction. If no additional information is provided, a convenient parameter to encode the presence of boundary is the variation in the sampling density. This encoding can be achieved by specifying the relative minimum and maximum sampling as a sampling condition. A consequence of such a condition is that the sampling is non-monotonic. The next chapter discusses one such set of non-monotonic sampling conditions for a Class C algorithm presented in Chapter 4.

CHAPTER 6

SAMPLING CONDITIONS

6.1 Introduction

In the previous chapter I argued that sampling conditions for reliable reconstruction of surfaces with boundaries should not only mention the minimum required density of samples, but also should spell out the maximum allowed density of samples. If these minimum and maximum sampling densities are defined using absolute quantities, then the sampling will have two problems. First, the sampling will not be able to adapt to the changing feature size of the given model. Second, additional constraints are required to make sure that the minimum required density is actually less than the maximum allowed density for the conditions to be logically sound. An alternate approach is to describe these bounds in relative terms. In other words, the maximum allowed density, which decides how *close* two sample points can be in a region, should be described in terms of minimum required density which decides how *far* two sample points can be in that region. This approach eliminates the above two problems of absolute definition of sampling densities. In this chapter, I describe one such set of sampling conditions for sampling surfaces with boundaries.

6.1.1 Sampling Region and Cover Region

The sampling conditions presented in this dissertation are based on two neighborhoods, a *sampling region* and a *cover region*, defined around points on the given surface.

For every point **a** on the surface, I will define a *sampling region* $S(\mathbf{a})$. Each sampling region must contain at least one sample point. This indirectly restricts the maximum distance between samples and thus imposes a minimum required sampling density in a region.

For every sample point **p**, I will define a *cover region* $C(\mathbf{p})$, that must contain no other sample. Each point of the surface must lie in some cover region. This restricts the (relative) minimum distance between samples and hence the maximum allowed sampling density in a region of the surface.

The rest of the chapter formally defines the sampling and cover regions using various distance functions and establishes their properties. I first approximate the local region of the surface using quadratic patches and develop the required theory behind these distance functions. We rely on some of the differential geometry background introduced in Chapter 2. Refer to Table 6.1 for the list of terms and notations used in this chapter.

6.2 Quadratic Approximation F of a Surface M

Let **a** be a point on the surface M with a well defined normal and, hence, a tangent plane. The normal at **a**, together with the principal directions on the tangent plane at **a**, define a local coordinate system with **a** as the origin. I make use of the implicit function theorem (Theorem 1) to express the surface in the neighborhood of a point **a** as a height function (Refer to Equation 2.2) in terms of the principal curvatures (Refer to Section 2.3.2). Let x and y axes be the principal directions and κ_1 and κ_2 be the principal curvatures along those directions, respectively. I represent the height function from the tangent plane to the surface M in this local coordinate system as

$$\begin{aligned} z(x,y) \quad &= \tfrac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2) + \text{higher order terms} \\ &\approx \tfrac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2) \end{aligned}$$

Hence the surface M can be approximated at and around **a** by the image of

$$F(x,y) = (x, y, \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2)) \tag{6.1}$$

This is called the *quadratic approximation of the surface M at* **a**.

6.2.1 Computation of Normal Vectors

The unit normal at any point $F(x,y)$ on the quadratic approximation of the surface can be computed using the partial derivatives of F with respect to x and y as follows:

M	Surface
F	Quadratic approximation of M
S	Sample set
bd(X)	Boundary of X
cl(X)	Closure of X
a, **b**, **c**	Points on M
a, b, c	Three axes of an ellipsoid
p, **q**	Sample points on M
$S(\mathbf{a})$	Sampling Region around **a**
$C(\mathbf{p})$	Cover Region around $\mathbf{p} \in S$ in M
$C^R(\mathbf{p})$	Cover Region around $\mathbf{p} \in S$ in $\mathbf{R}^3$
$U^\theta(\mathbf{a})$	Normal Based Region around **a**
	of angle deviation $\leq \theta$
$U^E(\mathbf{a})$	Curvature Based Region around **a**
$U^S(\mathbf{a})$	Sheet Based Region around **a**
$U^n(\mathbf{a})$	Closest samples to **a** based on the distance function Q
$T_\mathbf{p}$	Tangent plane or tangent vector
	at **p** of M or **bd**(M)
$\vec{T}_\mathbf{p}$	Tangent vector at $\mathbf{p} \in$ **bd**(M)
N_p	Normal vector at **p** of M or **bd**(M). Since
	the Darboux frame is used, surface and curve
	normal vectors are same for a boundary point
D, Q	Distance functions
$B_X(\mathbf{a}, \alpha)$	Ball of radius α measured using
	the distance function X around the point **a**.
x, y, z	Coordinate Axes
κ_1, κ_2	Principal curvatures ($\kappa_1 > \kappa_2$)
κ_v	Curvature along the direction $\vec{v}$.

Table 6.1: *Terms used in this chapter.*

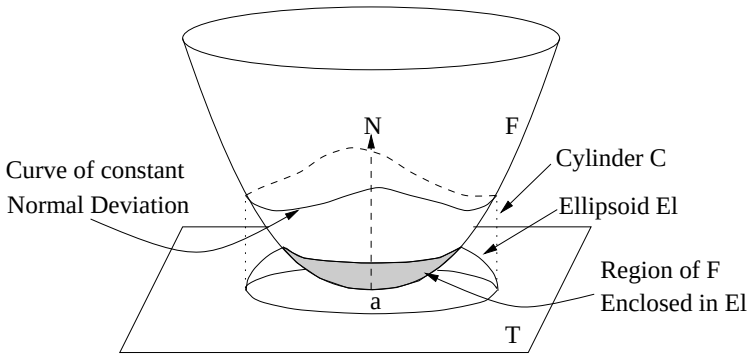

Figure 6.1: *The normal vector at* **a** $\in F$ *is* N *and the tangent plane is* T*. The curve of constant normal deviation is the intersection curve of an elliptical cylinder (*C*) and* F*. Two of the three axes of the ellipsoid* El *are defined such that* El $\cap\, T = C \cap T$*.*

$$\begin{aligned} F_x &= (1, 0, \kappa_1 x) \\ F_y &= (0, 1, \kappa_2 y) \\ F_x \times F_y &= (-\kappa_1 x, -\kappa_2 y, 1) \\ N(x, y) &= \|F_x \times F_y\| = \frac{(-\kappa_1 x, -\kappa_2 y, 1)}{\sqrt{\kappa_1^2 x^2 + \kappa_2^2 y^2 + 1}} \end{aligned} \tag{6.2}$$

6.2.2 Curve of Constant Deviation of Normal

We can compute the angle between the normal at the origin and the normal at a point $F(x, y)$ on F. The unit normal at the origin $F(0, 0) =$ **p** is $(0, 0, 1)$. The cosine of the angle, θ, between the normal at $F(x, y)$ and the normal at $F(0, 0)$ is given by the inner product of these vectors.

$$\cos(\theta) = \frac{1}{\sqrt{\kappa_1^2 x^2 + \kappa_2^2 y^2 + 1}} \tag{6.3}$$

Consider the locus of points on F with the same value of $\cos(\theta)$. The projection of this locus onto the xy plane is an ellipse given by

$$\frac{x^2}{\alpha^2/\kappa_1^2} + \frac{y^2}{\alpha^2/\kappa_2^2} = 1 \tag{6.4}$$

where $\alpha = \tan(\theta)$.

In other words, this ellipse on F is the curve of intersection of an elliptical cylinder with F (refer to Figure 6.1). The cross section of the elliptical cylinder is given by equation 6.4, and the axis of the cylinder is the z-axis (normal at the origin).

If we assume that $|\kappa_1| > |\kappa_2|$, the maximum height of $F(x, y)$ from the tangent plane above the ellipse occurs where this ellipse reaches the maxima at the major axis or minor axis. This can be computed as

$$z_{max} = \frac{1 - \cos^2(\theta)}{2\kappa_2 \cos^2(\theta)} = \frac{\alpha^2}{2\kappa_2} \tag{6.5}$$

Let us consider a *solid* ellipsoid, $El_\alpha(\mathbf{a})$ around **a**, defined by the equation

$$\frac{x^2}{\alpha^2/\kappa_2^2} + \frac{y^2}{\alpha^2/\kappa_1^2} + \frac{z^2}{\alpha^2/\kappa_1^2} \leq 1 \tag{6.6}$$

The half lengths of its three orthogonal symmetry axes are $a = \alpha/|\kappa_2|, b = \alpha/|\kappa_1|$, and $c = \alpha/|\kappa_1|$.

When we compare Equations 6.4 and 6.6 we can see that the intersection of $El_\alpha(\mathbf{a})$ and the tangent plane at **a** is actually the projection of the constant-normal-deviation curve on the tangent plane. Further, every point in $El_\alpha(\mathbf{a}) \cap F$ has normal deviation less than θ from the z-axis. We get a family of such ellipsoids for various values of α. I call these ellipsoids *curvature ellipsoids* as they are defined based on the curvature at a point and their shape captures the curvature variation at **a**. They are elongated in the direction of low curvature and the minor axis is along the direction of high curvature.

6.2.3 Curvature Ellipsoids and Scaled Ellipsoids

Let us see more about ellipsoids in general, and curvature ellipsoids in particular.

Definition 1 *The eccentricity, e, of the an ellipse given by $x^2/a^2 + y^2/b^2 = 1$ is defined as $\sqrt{1 - b^2/a^2}$. For an ellipsoid given by $x^2/a^2 + y^2/b^2 + z^2/c^2 = 1$ there are two eccentricities, $\sqrt{1 - b^2/a^2}$ and $\sqrt{1 - c^2/a^2}$.*

Since, in the curvature ellipsoid, two of the axes have same lengths, we have to consider just one eccentricity,

$$e = \sqrt{1 - b^2/a^2} = \sqrt{1 - \kappa_2^2/\kappa_1^2}.$$

Note that the eccentricity is no longer dependent on α, but only on the principal curvatures at a point. If $\kappa_1 = \kappa_2$, $a = b = c$ and $e = 0$. For example, the curvature ellipsoids defined on points on a plane, points on a sphere, sharp corners, or at any umbilic point in general, have the eccentricity $e = 0$ as $\kappa_1 = \kappa_2$ at these points.

Now we have another way of defining an ellipsoid. Instead of representing the ellipsoids using the half lengths a, b, and c, we can represent the same ellipsoid with a and two eccentricities or, in

case of the curvature ellipsoid, a ($= \alpha/|\kappa_2|$), and e. Let a' be a variable $0 \leq a' \leq a$. For various values of a', we get a family of ellipsoids with the property that the ellipsoid with larger a' contains the smaller one. I call this family of ellipsoids *scaled ellipsoids* with varying scale factors.

Definition 2 *The* Scaled ellipsoid El^m *with scale factor* m*, is defined with the axes dimensions* $a' = m$*, and* $b' = c' = m\sqrt{1-e^2}$.

Note that the curvature ellipsoid is a scaled ellipsoid with $m = \alpha/|\kappa_2|$. In fact, the curvature ellipsoids and the scaled ellipsoids are two different representations of the same ellipsoid. In certain cases where the curvature is zero, curvature ellipsoid is not defined, whereas the scaled ellipsoid is defined. The scaled ellipsoids, like the curvature ellipsoids, capture the curvature variation at a point.

Curvature and Scaled Ellipsoids for Boundary Curves

We defined scaled and curvature ellipsoids around points in the interior of a surface. As we allow surfaces with boundaries, we have to define these ellipsoids for curves also. For a curve on a surface, the standard Darboux frame is used to define the normal. Refer to the definitions of tangent, normal, and bi-normal of a curve in Chapter 2. Let the derivatives of normal, tangent, and bi-normal of the boundary curve be $\dot{N}$, $\dot{T}$, and $\dot{B}$ respectively.

The dimensions of the curvature ellipsoid $El_\alpha(\mathbf{a})$ around a point **a** on the boundary of a surface, are defined by $a = \alpha/|\dot{B}|$ along the direction of $\dot{B}$ which is in the plane defined by T and N, $b = \alpha/|\dot{N}|$ along the direction of $\dot{N}$ which is in the plane defined by T and B, and finally $c = \alpha/|\dot{T}|$ along the direction of $\dot{T}$ which is in the plane defined by B and N.

The ellipsoid around a point on a boundary has two eccentricities as all its axes have different lengths. These eccentricities are $e_1 = \sqrt{1-b^2/a^2}$, and $e_2 = \sqrt{1-c^2/a^2}$. The *scaled ellipsoid* El^m with scale factor m, at a point on the boundary, is defined with the axes dimensions $a' = m$, $b' = m\sqrt{1-e_1^2}$, and $c' = m\sqrt{1-e_2^2}$. Again, the curvature ellipsoid for a point in the boundary is one of the scaled ellipsoids.

These definitions of scaled ellipsoids around points on the surface and its boundary are used in Section 6.4.2.

6.3 Distance Functions D and Q

In this section I define two distance functions D and Q. The distance function D is related to the curve around a point **p** with constant normal deviation, and to the *local feature size* which is based on the medial axis, introduced by Amenta *et al.* The distance function Q is easy to evaluate from the positional information of points and the normal vectors at these points. In this chapter I show that the function Q and D are related, and thus justify the use of Q in sampling and reconstruction.

6.3.1 Distance Function D

Let F be the quadratic approximation of the surface M, as before. Let **a** be the origin. The distance function D is defined for the points on F at close proximity to the origin **a**. Let **b** be a point close to **a**, and the direction vector from **a** to **b** be $\vec{v}$. The distance D of **b** from **a** is the product of the normal curvature $\kappa_v(\mathbf{a})$ in the direction of the orthogonal projection of $\vec{v}$ on the tangent plane, and the Euclidean distance E from **a** to **b**.

$$D(\mathbf{b}) \equiv |\kappa_v(\mathbf{a})E| \tag{6.7}$$

Note that both $\vec{v}$ and E are dependent on the position of **b** with respect to **a**. Further, we can relate this distance function with Equations 2.9 and 2.10.

Distance Function D and the Curve of Constant Normal Deviation

In this section I show the relationship between the distance function D and the curve of constant normal deviation explained in 6.2.2. I use a slightly modified distance function $D^{'}$ (defined below) instead of D for this purpose.

Let us approximate the distance E from **a** to **b** by r, which is the geodesic distance on F from **a** to **b**. Let this approximated distance function be represented as $D^{'}$. Spherical approximation of the curve from **a** to **b** with curvature κ_v would show that the arc length distance $r = \theta/\kappa_v$, and the Euclidean distance $E = 2\sin(\theta/2)/\kappa_v$. Substituting these values in the equations of the distance functions, $D^{'} = \theta$ and $D = 2\sin(\theta/2)$. For small values of θ, $D < D^{'} < 2D$. Hence both are equivalent functions. I use $D^{'}$ in this section to relate the curve of constant normal deviation and D.

Let us consider all the points on the surface that are at a distance δ from a point **a** on the surface as measured by $D^{'}$. This is a closed curve around **a**, say $C^{\delta}(\mathbf{a})$.

Lemma 1 *The deviation of the normal vector at points around* **a** *inside* $C^{\delta}(\mathbf{a})$ *from the normal vector at* **a** *is bounded.*

Proof: I use the equations 2.7–2.9 to show the relationship between $C^{\delta}(\mathbf{a})$ and the curve of constant normal deviation. I repeat those equations here for the sake of completeness. In these equations, $\partial\vec{N}$ denotes the change of normal between **a** and **b**, ∂s denotes the arc length distance between these points, t is the torsion factor, $\vec{T}$ is the tangent vector of the curve from **a** to **b**, and $\vec{B}$ is the bi-normal vector.

$$\partial\vec{N} = -\kappa_v \partial s \vec{T} - t\partial s \vec{B} \tag{6.8}$$

$$|\partial\vec{N}| = \sqrt{\kappa_v^2 + t^2}\,|\partial s| \tag{6.9}$$

$$|\partial\vec{N}| = |\kappa_v \partial s|. \tag{6.10}$$

Since $C^{\delta}(\mathbf{a})$ contains all points with distance measured using D' being less than δ from **a**,

$$|\partial\vec{N}| = |\kappa_v \partial s| < \delta. \tag{6.11}$$

Substituting this in Equation 6.8 and setting $t = 0$, as in Equation 6.10 for a planar curve, we get $\partial\vec{N} < \delta\vec{T}$ in the direction of $\vec{T}$. With some algebraic manipulation, we can find that the deviation of the normal of the points inside $C^{\delta}(\mathbf{a})$ from the normal at **a** is less than $\theta = \arccos(1/\sqrt{1+\delta^2})$. $\diamondsuit$

Further this shows that all points in $El_{\alpha}(\mathbf{a}) \cap F$ (Equation 6.6) are in the region bounded by $C^{\delta}(\mathbf{a})$ where $\alpha < \tan(\arccos(1/\sqrt{1+\delta^2}))$.

Distance Function D and the δ-Sampled Surface

The distance function D captures the curvature variation of the surface and, as explained at the end of Section 2.3.4, can be used in defining the required sampling of the surface. For example, if we enforce "uniform" sampling according to this distance function, then indirectly we are enforcing denser sampling in high curvature regions and sparse sampling in low curvature regions. Under such a sampling condition, in planar regions almost no samples are required, and in sharp edges infinitely many samples are required. This is similar to the behavior of the sampling condition of Amenta *et al.*[Amenta98b]. Conditions of Amenta *et al.* also require almost no samples on planar regions and infinitely many samples at sharp edges. The rest of this section is devoted to formalize this relationship between the conditions based on D and the sampling condition of Amenta *et al.*

Though I have not yet formally introduced any sampling condition, let me give the intuition behind the use of D by relating a sampling condition based on D with that of Amenta *et al.* The sampling condition of Amenta *et al.* is based on *local feature size*, and D is based on the radius of curvature ($1/\kappa$). So first I will relate the radius of curvature and the *local feature size*, before relating the sampling conditions.

The *local feature size, LFS(**a**),* of a point **a** on a surface is its closest distance to the medial axis (refer to Section 2.5). As we are trying to compare the radius (of circle) of curvature, (which is a *resting circle*), with the LFS, I consider only those curves in which the closest point on the medial axis to any point **a** on the curve is the center of a resting circle of **a**. I call such curves *resting curves,* and for all the arguments in this section, I consider only *resting curves*. For a resting curve, the center of the circle of curvature is in the medial axis, and hence, *LFS(**a**)*$= \rho$, where ρ is the radius of curvature at **a**. We will use this observation to argue that the distance function D has required properties to be used in sampling, by relating it to the sampling condition of Amenta *et al.*

Amenta *et al.*'s sampling: A *sampling* of a surface is a set of points sampled from that surface. A sampling of a surface is said to be a δ sampling if for every point **a** on the surface, there exists a sample point **p** such that the Euclidean distance between **a** and **p**, $E(\mathbf{a}, \mathbf{p}) < \delta$*LFS(**a**)*. The definitions of the *LFS* and δ-sampling are used in Amenta *et al.* and are applicable for curves also.

Let a hypothetical sampling condition based on D be the following.

Sampling Condition: For every point **a** on the curve, there exists a sample point **p** such that $D(\mathbf{p}) < \delta$ from **a**, where δ is a constant.

In the case of curves, the curvature κ at **a** is used in the definition of D instead of directional curvature κ_v. Though we use this hypothetical sampling condition just to relate the distance function with the medial axis, we use this condition in a slightly different form as our actual sampling condition later in this chapter.

Lemma 2 *A sampling of a* resting curve *that satisfies the sampling condition above, is a δ-sampling.*

Proof: Let **a** be a point on the curve, and **p** be a sample point satisfying the sampling condition around **a**. Let $E(\mathbf{p}, \mathbf{a})$ denote the Euclidean distance between **a** and **p**, κ be the curvature, and ρ be the radius of curvature at **a**. By the sampling condition, $\kappa E(\mathbf{a}, \mathbf{p}) < \delta$, so $E(\mathbf{a}, \mathbf{p}) < \delta\rho < \delta LFS(\mathbf{a})$. By this observation, and the definition of δ-sampling, we conclude that if the sampling satisfies the above sampling condition, then it is a δ-sampling. ♢

This shows that the above sampling condition based on D captures the *local features of a curve* in the

same way as the condition of Amenta *et al.*, which is based on the medial axis and local feature size. Later in this chapter, we define conditions that capture the *global features* also.

We proved the usefulness of D in sampling a curve. These arguments can be extended directly to surfaces, except that the distance function D has to be modified to use the *maximum principal curvature*, κ_1, instead of κ_v in its definition. But in this dissertation we use κ_v instead of κ_1 and hence the sampling conditions are "anisotropic" (different in different directions) as against "isotropic" (uniform in all directions) sampling conditions of Amenta *et al.*

6.3.2 Distance Function Q

If I use the distance function D in the sampling process, I might have to use D again in the reconstruction process to measure the distance between sample points. But this function has normal curvature components that are difficult to compute with just the sample points. Further, although D captures the curvature variation of the surface that is required for a good sampling condition, it is anisotropic and requires theoretically no sample point along the direction of zero curvature. So to alleviate these two problems I introduce another distance function Q, and establish its relationship with D.

The distance function Q is defined on a set of sample points with normal vector information at each point and not on other surface parameters, like curvature, that would have to be estimated or interpolated.

Let $T_{\mathbf{a}}(\mathbf{b})$ denote the distance of **b** from the tangent plane at **a**, $T_{\mathbf{a}}(\mathbf{b}) = |N_{\mathbf{a}} \cdot (\mathbf{b} - \mathbf{a})|$, where $N_{\mathbf{a}}$ is the unit normal vector at **a**. Let $E(\mathbf{a}, \mathbf{b})$ denote the Euclidean distance between **a** and **b**. The distance function $Q_{\mathbf{a}}$ is defined as

$$Q_{\mathbf{a}}(\mathbf{b}) \equiv E^2(\mathbf{a}, \mathbf{b}) + T^2_{\mathbf{a}}(\mathbf{b}) \tag{6.12}$$

If **a** is the origin then, on a quadratic approximation of a surface (equation 6.1) with $N_{\mathbf{a}}$ as the z-axis, $T_{\mathbf{a}}(\mathbf{b})$ is just the z-coordinate of **b**.

Hence,

$$Q_{\mathbf{a}}(\mathbf{b}) = (x^2 + y^2 + z^2) + (z^2) = x^2 + y^2 + 2z^2, \tag{6.13}$$

where $\mathbf{b} = (x, y, z)$.

Now I show how the distance functions D and Q are related and thus justify the use of Q instead of D in the reconstruction process.

Definition 3 *Let X and Y be two distance functions on a space V, and $B_X(x, \epsilon)$ denote the set of all points in V that are at a distance less than ϵ from x when measured using X. The functions X and Y are said to be* equivalent, *in the sense that they generate the same topology, if given any $x \in V$ and $\epsilon > 0$, there exists $\delta_1 > 0$ and $\delta_2 > 0$ such that, $B_X(x, \delta_1) \subseteq B_Y(x, \epsilon)$,and $B_Y(x, \delta_2) \subseteq B_X(x, \epsilon)$.*

Lemma 3 *Given any* **a** $\in F$ *and $\epsilon > 0$, there exists $\delta > 0$ such that,*

$$B_Q(\mathbf{a}, \delta) \subseteq B_D(\mathbf{a}, \epsilon)$$

Proof: Without loss of generality, let us assume that $|\kappa_1| > |\kappa_2|$. Define $\delta = \epsilon^2/\kappa_1^2$.

$$\begin{aligned}
\mathbf{b} \in B_Q(\mathbf{a}, \delta) &\Rightarrow \left(x^2 + y^2 + 2z^2\right) < \delta = \epsilon^2/\kappa_1^2 \\
&\Rightarrow \left(x^2 + y^2 + z^2\right) < \epsilon^2/\kappa_1^2 \\
&\Rightarrow \kappa_1^2 \left(x^2 + y^2 + z^2\right) < \epsilon^2 \\
&\Rightarrow |\kappa_1 \left(x^2 + y^2 + z^2\right)^{1/2}| < \epsilon \\
&\Rightarrow \mathbf{b} \in B_D(\mathbf{a}, \epsilon) \\
&\Rightarrow B_Q(\mathbf{a}, \delta) \subseteq B_D(\mathbf{a}, \epsilon)
\end{aligned}$$

For every $\epsilon > 0$, a $\delta > 0$ is defined, and the above derivation is true for any **a**. $\diamondsuit$

It can be proved that $B_D(\mathbf{a}, \delta) \not\subseteq B_Q(\mathbf{a}, \epsilon)$, especially when $\kappa_1 = 0$. We have to find a $\delta > 0$ such that all points in $B_D(\mathbf{a}, \delta)$ are contained in $B_Q(\mathbf{a}, \epsilon)$ for a given $\epsilon > 0$. This is not possible as $B_D(\mathbf{a}, 0)$ contains all points on the surface when $\kappa_1 = 0$.

Definition 4 *Let G be a distance function. For every point* **a** *on the surface, if there exists a sample* **p** *such that $G(\mathbf{a}, \mathbf{p}) < \delta$, then such a sampling is called a δ sampling of the surface using the distance function G.*

Theorem 4 *In a compact surface, δ-sampling using Q yields a δ-sampling of the surface using D.*

Proof: The proof follows directly from Lemma 3. One more question that remains to be answered is the countability of number of samples when Q is used, since a trivial solution of infinite sampling would prove this lemma. Define *represented region* of a sample **p** as all points on the surface from which **p** is at a distance less than δ. Assume the represented regions of the sample points are open and

non-empty. The union of represented regions cover the surface. As the surface is compact, any open cover of the surface has a finite subcover. Hence a finite number of samples are required to sample the surface using the distance function Q. ♢

The above theorem shows the usefulness of Q in sampling. Further, as Q can be computed between any two points in the input set with just their position and normal information, it is a distance function that can be applied directly on the input set without any further estimation of other parameters like curvature. Now, I show that Q is equivalent to our familiar Euclidean distance function E.

Theorem 5 *The distance functions Q and E are equivalent functions.*

Proof: For any positive constant ϵ, define $\delta_1 = \epsilon^2$ and $\delta_2 = \sqrt{\epsilon/2}$. It is now easy to show that

$$\begin{aligned} B_Q(\mathbf{a}, \delta_1) &\subseteq B_E(\mathbf{a}, \epsilon), \text{ and} \\ B_E(\mathbf{a}, \delta_2) &\subseteq B_Q(\mathbf{a}, \epsilon) \end{aligned}$$

♢

Lemma 4 $\text{El} = \{\mathbf{b} \mid Q_{\mathbf{a}}(\mathbf{b}) = \beta,\ \mathbf{b} \in \mathbf{R}^3\}$ *is an ellipsoid.*

Proof: Let **a** be the origin, and the points in space be represented as (x, y, z). From the equation 6.13, the set *El* contains all points that satisfy

$$\begin{aligned} Q_{\mathbf{a}}(\mathbf{b}) &= \beta \\ x^2 + y^2 + 2z^2 &= \beta \\ \frac{x^2}{\beta} + \frac{y^2}{\beta} + \frac{z^2}{\beta/2} &= 1 \end{aligned} \tag{6.14}$$

Equation 6.14 represents a "pan-cake" shaped ellipsoid whose cross-section along the xy plane is a circle. The diameter of this circle is twice its total thickness along the z-axis. ♢

Variations of Q for Distance Between Boundary Points

The distance function Q as given in Equation 6.12 can be used for points in the interior of the surface, since a tangent plane is defined at these points. For a point in the boundary of the surface, the following definition of Q is used.

$$\mathbf{a} \in \mathbf{bd}(M), \mathbf{b} \in \mathbf{R}^3,\ \ Q_{\mathbf{a}}(\mathbf{b}) = E^2(\mathbf{a}, \mathbf{b}) + \vec{T}^2_{\mathbf{a}}(\mathbf{b}) \tag{6.15}$$

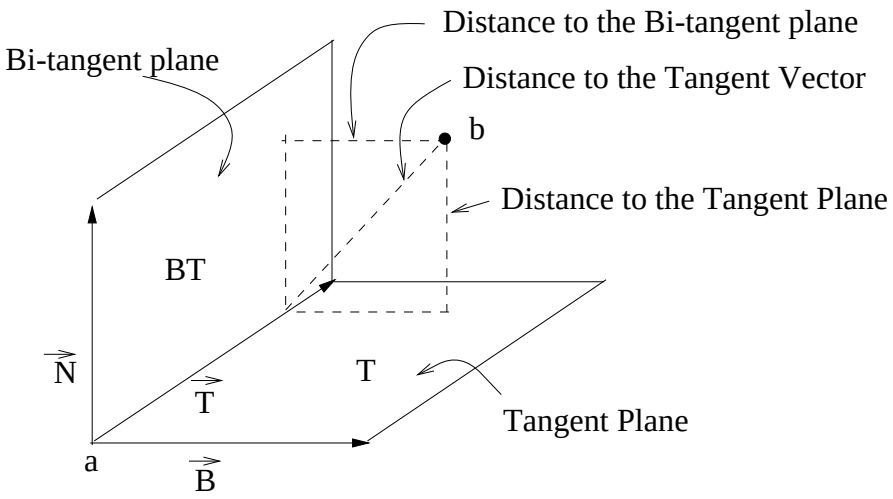

Figure 6.2: *Square of the distance from the tangent vector is the sum of the squares of the distance from the tangent plane and the distance from the bi-tangent plane.*

where $\vec{T}_{\mathbf{a}}$ is the tangent vector of the boundary curve at **a**. We have been using $T_{\mathbf{a}}$ to represent both the tangent plane and the tangent vector. But for this section, we use $T_{\mathbf{a}}$ to represent the tangent plane and $\vec{T}_{\mathbf{a}}$ to represent the tangent vector.

Let us compare the distance functions $Q_{\mathbf{a}}$ defined for $\mathbf{a} \in M$ and $\mathbf{a} \in \mathbf{bd}(M)$. Since $\vec{T}$, $\vec{N}$, and $\vec{B}$ form an orthogonal coordinate system, the tangent vector is the intersection of the tangent plane $T_{\mathbf{a}}$ defined by the normal vector $\vec{N}$, and the bi-tangent plane $BT_{\mathbf{a}}$, defined by the binormal vector $\vec{B}$. Hence the value of $\vec{T}^2_{\mathbf{a}}(\mathbf{b})$ can be further broken down into two components (refer to Figure 6.2) as

$$\vec{T}^2_{\mathbf{a}}(\mathbf{b}) = T^2_{\mathbf{a}}(\mathbf{b}) + BT^2_{\mathbf{a}}(\mathbf{b}).$$

Hence, the Q distance of a point **b** from a point $\mathbf{a} \in \mathbf{bd}(M)$ is always greater than equal to the Q distance from **a** if **a** were considered an interior point of the surface.

Now let us consider variations of Q when normal vector information or tangent vector information is not given or not known. If tangent vectors at the points on the boundary are not given, then the points on the boundary are treated as points on the interior of the surface for defining Q, and Equation 6.12 will be used. If the normal vectors at the points on the surface are not known then the Equation 6.12 reduces to

$$Q_{\mathbf{a}}(\mathbf{b}) = E^2(\mathbf{a}, \mathbf{b}).$$

6.4 Distance Functions and Regions

Let us summarize this chapter till now and see where we are going from here. In Section 6.3, we defined two distance functions D and Q. We also defined two ellipsoids in Equations 6.6 (the curvature ellipsoid) and 6.14 (ellipsoid due to distance function Q). We discussed three regions, the region of bounded normal deviation and two regions of bounded distance as measured using D and Q. We proved the relationship between these three regions (Lemma 1, Theorem 4). We also showed the relationship between the ellipsoids and these regions (Lemma 4, Section 6.2.2, Lemma 1) as follows. The region of the surface enclosed inside the curvature ellipsoid of Equation 6.6 is contained in a region of bounded normal deviation and bounded distance measured by D. Further, the region inside the ellipsoid of Equation 6.14 is a region of bounded distance as measured by Q.

The relationship between the three regions, bounded normal deviation region and the bounded distance regions of D and Q, were established only under the assumption that the underlying surface is a quadratic surface. As most of the real world surfaces deviate from this assumption, I treat these regions as independent regions and make use of the ellipsoids to define these regions on a surface.

In this section, I formally define the *Normal Based Region* U^θ, the *Curvature Based Region* U^E, and the *Sheet Based Region* U^S, with additional constraints on the topology of these regions. Using these regions, I define the *Sampling Region* $S(a)$, and the *Cover Region* $C(p)$, that are used to define the sampling conditions.

In all the following definitions, M is considered as a smooth surface with or without boundary, and $S \subset M$ is the set of sample points. The boundary of M is denoted by $\mathbf{bd}(M)$.

6.4.1 Normal Based Region

For every point **a** on M, the region around **a** based on the normal deviation is defined as follows.

$$U^\theta(\mathbf{a}) = \{\mathbf{b} \in M \mid \angle N_\mathbf{a} N_\mathbf{b} < \theta\}$$

where $N_\mathbf{a}$ and $N_\mathbf{b}$ are normal vectors to M at **a** and **b** respectively. If $\mathbf{a} \in \mathbf{bd}(M)$ then

$$U^\theta(\mathbf{a}) = \{\mathbf{b} \in \mathbf{bd}(M) \mid \angle N_\mathbf{a} N_\mathbf{b} < \theta, \angle T_\mathbf{a}\, T_\mathbf{b} < \theta\}$$

where $T_\mathbf{a}$ and $T_\mathbf{b}$ are tangent vectors to $\mathbf{bd}(M)$ at **a** and **b** respectively.

As we use the Darboux frame at the boundary curves, the normal to the curve and the normal to the surface are same. We fix θ to be less than 12° (0.2 radians) for various proofs in this chapter.

6.4.2 *Curvature Based Region*

Let us consider the scaled ellipsoids defined in Section 6.2.3, around a point **a**. Let us intersect these scaled solid ellipsoids, with various values of the scale factor m, with the surface M, $R = M \cap El^m(\mathbf{a})$. We would like R to contain points that are close to **a** in terms of having a similar curvature. Naturally, we expect R to have a single component and not have multiple disconnected components. For monotonic increase in the value of m, the number of components in R might increase and decrease in a non-monotonic way. A *maximal ellipsoid* with scale factor $m = l$ is one whose R is just one connected component, and for all scale factors $l + \epsilon$, with $\epsilon \to 0$, R has multiple disconnected components. There might be many such *maximal ellipsoids* in this family of ellipsoids for varying values of m. The smallest of them all is defined as follows. The shrink factor t used below is a fraction that shrinks the smallest maximal ellipsoid.

Definition 5 *Let* $a = \alpha/|\kappa_2| = \tan(\theta)/|\kappa_2|$. *The* Limit ellipsoid *is a scaled ellipsoid with the scale factor*

$$\mathbf{a} \in M,\ l_1(\mathbf{a}) = \sup\{n \mid 0 \leq n \leq a, \forall m < \frac{n}{t}, \mathrm{El}^m_\alpha(\mathbf{a}) \cap M \approx \oplus\}$$

where $\approx \oplus$ *means homeomorphic to an open disk or half disk.*

Here we limit the maximum value of m to be a, the major axis of the curvature ellipsoid (refer to Equation 6.6). The value of θ is same as the one used in Section 6.4.1. We have to define the scale factor of the *Limit ellipsoid* for points on the boundary also. The scale factor is,

$$\mathbf{a} \in \mathbf{bd}(M),\ l_1(\mathbf{a}) = \sup\{n \mid 0 \leq n \leq a, \forall m \leq \frac{n}{t}, El^m_\alpha(\mathbf{a}) \cap \mathbf{bd}(M) \approx \oplus\}$$

where $\approx \oplus$ means homeomorphic to an open interval. The shrink factor t, $0 \leq t \leq 1$ will be fixed at the end of this chapter.

Definition 6 *The* Curvature Based Region U^E *is defined as*

$$U^E(\mathbf{a}) = \{\mathbf{b} \mid \mathbf{b} \in \mathrm{El}^{l_1}_\alpha(\mathbf{a}) \cap M\} \text{ if } \mathbf{a} \in M, \text{ and} \tag{6.16}$$

$$U^E(\mathbf{a}) = \{\mathbf{b} \mid \mathbf{b} \in \mathrm{El}^{l_1}_\alpha(\mathbf{a}) \cap \mathbf{bd}(M)\}, \text{ if } \mathbf{a} \in \mathbf{bd}(M). \tag{6.17}$$

6.4.3 *Sheet Based Region*

The region R_S around a point $\mathbf{a} \in M$ (or $\mathbf{bd}(M)$) is defined as follows.

$$R^m_S(\mathbf{a}) = \{\mathbf{b} \in M \mid Q_{\mathbf{a}}(\mathbf{b}) \leq m\}$$

Similar to the previous section, where we defined the limit ellipsoid as the smallest maximal ellipsoid containing just one component of the region, we again define such a limit region based on the distance function Q.

The *limit region* around a point $\mathbf{a} \in M$ is $R_S^{l_2}(\mathbf{a})$ where

$$l_2(\mathbf{a}) = \sup\{n \mid \forall m < \frac{n}{t}, R_S^m(\mathbf{a}) \cap M \approx \oplus\}$$

The value of t is the same as used in the definition of the *limit ellipsoid*, and will be fixed at the end of this chapter. Here again, M is substituted by $\mathbf{bd}(M)$ if $\mathbf{a} \in \mathbf{bd}(M)$.

Definition 7 *The* Sheet Based Region U^S *is defined as,*

$$U^S(\mathbf{a}) = \{\mathbf{b} \mid \mathbf{b} \in R_S^{l_2}(\mathbf{a}) \cap M\} \text{ if } \mathbf{a} \in M, \text{ and} \tag{6.18}$$

$$U^S(\mathbf{a}) = \{\mathbf{b} \mid \mathbf{b} \in R_S^{l_2}(\mathbf{a}) \cap \mathbf{bd}(M)\} \text{ if } \mathbf{a} \in \mathbf{bd}(M). \tag{6.19}$$

6.4.4 Sampling Region

Definition 8 *The* sampling region *is an open influence region around any point* $\mathbf{a} \in M$ *(or* $\mathbf{bd}(M)$*) and is defined as*

$$S(\mathbf{a}) = U^\theta(\mathbf{a}) \cap U^E(\mathbf{a}) \cap U^S(\mathbf{a}) \tag{6.20}$$

For the purpose of enabling our proofs, $\theta < 0.2$ radians.

The closure of $S(\mathbf{a})$ is denoted by $\mathbf{cl}(S(\mathbf{a}))$.

6.4.5 Cover Region

Given the sampling S of M, the *nearest neighbor set*, $U^n(\mathbf{a})$, of any point $\mathbf{a} \in M$ is defined as the set of closest samples $\mathbf{p} \in S - \{\mathbf{a}\}$ as measured using the distance function Q.

$$U^n(\mathbf{a}) = \{\mathbf{p} \in S - \{\mathbf{a}\} \mid \forall \mathbf{q} \in S - \{\mathbf{a}\}, Q_{\mathbf{a}}(\mathbf{p}) \leq Q_{\mathbf{a}}(\mathbf{q})\}$$

Note that this definition is valid for all samples in S also, and $U^n(\mathbf{p})$, for any $\mathbf{p} \in S$ is the set of closest neighbors of $\mathbf{p}$ in S.

The distance from $\mathbf{a}$ to $U^n(\mathbf{a})$, $Q_{\mathbf{a}}(U^n(\mathbf{a}))$, is the closest distance of $\mathbf{a}$ to any of the elements of $U^n(\mathbf{a})$.

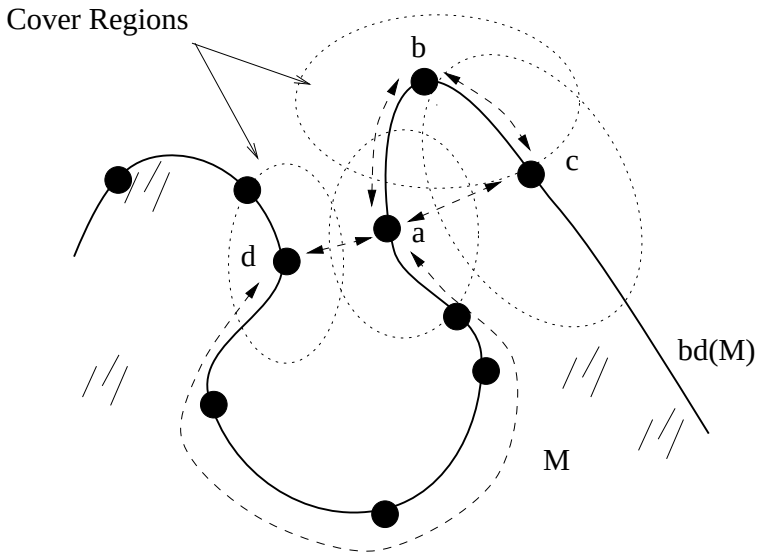

Figure 6.3: *The samples* **a**, **b**, *and* **c** *are visible from each other, whereas* **a** *and* **d** *are not visible from each other even though their cover regions intersect.*

Definition 9 *The* cover region, $C(\mathbf{a})$, *of a point* $\mathbf{a} \in M$ *is defined as*

$$C(\mathbf{a}) = \{\mathbf{b} \in M \mid Q_{\mathbf{a}}(\mathbf{b}) \leq Q_{\mathbf{a}}(U^n(\mathbf{a}))\}$$

Similarly, the closest region in $\mathbf{R}^3$ *that is closest to* **a** *is defined as*

$$C^R(\mathbf{a}) = \{\mathbf{b} \in \mathbf{R}^3 \mid Q_{\mathbf{a}}(\mathbf{b}) \leq Q_{\mathbf{a}}(U^n(\mathbf{a}))\}$$

Even though we have defined the cover region for any point in M, we will be primarily using cover regions for sample points in S. In the next section, the sampling and cover regions defined here are used to develop the sampling conditions.

6.5 Sampling Conditions

In this section, I develop the conditions for sampling surfaces with boundaries. These conditions exhibit the required properties for sampling surfaces with boundaries that were introduced in Chapter 5. In other words, the conditions restrict the maximum distance and relative minimum distance between samples in a region. Further, these conditions enable the reconstruction algorithm to reconstruct surfaces that are topologically equivalent, and geometrically close to the underlying sampled surface.

Given the samples that satisfy the sampling conditions, the reconstruction process constructs a piecewise linear interpolation of these points to approximate the underlying surface. There might

be cases when the reconstructed surface is topologically equivalent but is not in any way close to the surface both in geometry and appearance. Verification of geometric closeness can be done using operations like *containment* and *intersection*. For example, it is easy to check in 2D whether a line segment is contained completely within a region, or whether two line segments intersect. By repeating such operations we can verify whether the reconstructed surface is close to the original surface geometrically. These operations might not make sense in case of a 2D surface embedded in 3D. For example, containment of a line segment in a surface embedded in 3D, and intersection of two line segments in 3D are more of a chance than of a rule. Hence these operations have to be redefined for curves on surfaces (rather than line segments) for their applicability on 2D surfaces embedded in 3D. This redefinition makes these operations less useful for verification during the reconstruction process, as no definition of surface exists during this process. For this reason, I use projection of regions of the surface onto a 2D plane for one of the definitions below, and extensively in the reconstruction stage.

Definition 10 *Let* $P_{\mathbf{a}}(\mathbf{b})$ *denote the orthogonal projection of* **b** *onto the tangent plane at* **a**. *Sample points* $\mathbf{p}, \mathbf{q} \in \mathcal{S}$ *are said to be* visible from each other, *if the projection of the line segment* **pq**, $P_{\mathbf{p}}(\mathbf{pq}) \subset P_{\mathbf{p}}(C(\mathbf{p}) \cup C(\mathbf{q}))$ *and* $P_{\mathbf{q}}(\mathbf{pq}) \subset P_{\mathbf{q}}(C(\mathbf{p}) \cup C(\mathbf{q}))$.

Further, if $\mathbf{p}, \mathbf{q} \in \mathbf{bd}(M)$, then **p** and **q** are said to be visible if they are visible as per the above definition or if there is a path between **p** and **q** along $\mathbf{bd}(M) \cap (C(\mathbf{p}) \cup C(\mathbf{q}))$ that does not pass through any other sample point (Refer Figure 6.3).

Condition 1: Every sampling region has a sample:

$$\forall\, \mathbf{a} \in M, S(\mathbf{a}) \cap S \neq \emptyset$$

Condition 2: If the sampling region $S(\mathbf{a})$, $\mathbf{a} \in M$, has non-empty interior (that is, **a** is not a sharp corner) then the closure of $S(\mathbf{a})$ contains the nearest samples of **a**.

$$U^n(\mathbf{a}) \subset \mathbf{cl}(S(\mathbf{a}))$$

Condition 3: The cover regions of the samples cover the surface:

$$\bigcup_{\mathbf{p} \in S} C(\mathbf{p}) = M$$

Condition 4: If the 3D cover regions of boundary points intersect, then they are visible from each other (Refer Figure 6.3). That is, if $\mathbf{p}, \mathbf{q} \in \mathbf{bd}(M)$ and $C^R(\mathbf{p}) \cap C^R(\mathbf{q}) \neq \emptyset$ then **p** and **q** are visible from each other.

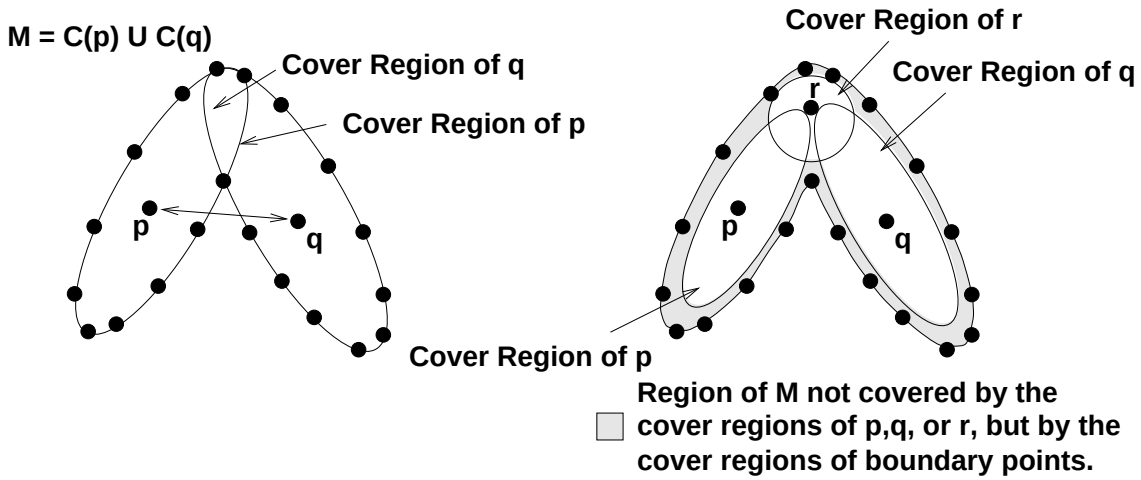

Figure 6.4: *In this example, the interior samples* **p** *and* **q** *are not visible from each other even though their cover regions intersect. Any triangulation that connects* **p** *and* **q** *might be a topologically correct triangulation, but might not be a visually pleasing triangulation. One solution is to introduce another point, say* **r***, as shown in the right.*

Condition 5: If the cover regions of two interior samples, **p**, **q**$\in M$, intersect, then **p** and **q** are visible from each other.

The condition 2 is useful when the objects have sharp edges between faces. By this condition, the nearest sample point of any point in one face is in that face or in the boundary (edge) between faces.

We know that all the given sample points belong to a 2-manifold. We can make use of the fact to avoid forming higher dimensional simplices while reconstructing the surface. This allows us to impose fewer restrictions on the intersections of the cover regions $C(\mathbf{p})$ for the interior of the surface. But for the boundary of the surface, if additional details, like the binormal or the tangent vector or a tag indicating whether a sample point belongs to boundary, are not given, then we have to impose additional restrictions on sampling the boundary, like Condition 4 above.

Let me illustrate a case where Condition 5 is used. Let us consider a hypothetical situation illustrated in Figure 6.4. The surface with boundary, M, is just the union of the two cover regions of **p** and **q**. There are additional samples along the boundary such that no two cover regions of two non-adjacent boundary samples intersect. Further, the samples **p** and **q** are not visible from each other. If any triangulation of M connects **p** and **q**, the edge would cut across the boundaries in the projection. Though this triangulation would be topologically equivalent to M, it would not have the same visual quality as that of M. So, to improve the geometric closeness of triangulation to M, I introduce Condition 5 as one of the sampling condition. Further, the reconstruction algorithm introduced in

Chapter 4 relies on the non-intersection of edges in the projection to arrive at a triangulation of M. Hence, Condition 5 can also be considered as an artifact of the reconstruction algorithm used in this dissertation.

If the sampling does not satisfy Condition 5, a solution would be to introduce new sample points in the region of intersection of the cover regions (of **p** and **q** in Figure 6.4) and still satisfying all the sampling conditions. For example, in Figure 6.4, after introducing the sample **r**, the cover regions of **p** and **q** shrink. The union of **p**, **q**, and **r** no longer covers M, but the union of cover regions of all samples including the boundary samples covers M, thus satisfying Condition 3.

6.6 Analysis of the Sampling Conditions

In this section, I analyze the effects of the definitions of sampling and cover regions, and the sampling conditions. In the previous chapter, I proved that to reliably reconstruct surfaces with boundaries, the sampling condition should specify the minimum required sampling and also the relative maximum allowed sampling. In other words, the conditions have to impose restrictions on maximum distance between samples and relative minimum distance between samples. In this section I argue that the definition of sampling region and Condition 1 of the sampling conditions restrict the maximum distance between samples, and the definition of cover region and Condition 3 restrict the relative minimum distance between samples.

6.6.1 Sampling Regions and the Maximum Distance Between Samples

Let us assume that the surface M is bounded and hence all regions, including the sampling and cover regions, are bounded. Consider a sample point **p** that lies inside the sampling region of $\mathbf{a} \in M$, thus satisfying Condition 1 of the sampling condition for the point **a** (refer to Figure 6.5). The point **p** might lie in many other sampling regions of points close to **a** on M. Let us consider all such points on M, in whose sampling region **p** lies. This defines a region around **p**, and let us call this the *represented region of* **p** (for which **p** is a representative sample point). Note that **p** belongs to the represented region of **p**. Since the sampling regions are bounded, the represented region is also bounded. Let us consider a point, say **c**, arbitrarily close to, but outside the represented region of **p**. To satisfy Condition 1, there exists a sample point **q** in the sampling region of **c**. Since both sampling and represented regions are bounded, the distance between **p** and **q** are bounded and restricted by the size of sampling regions. The size of the sampling region depends on the local curvature variation on M and hence the maximum

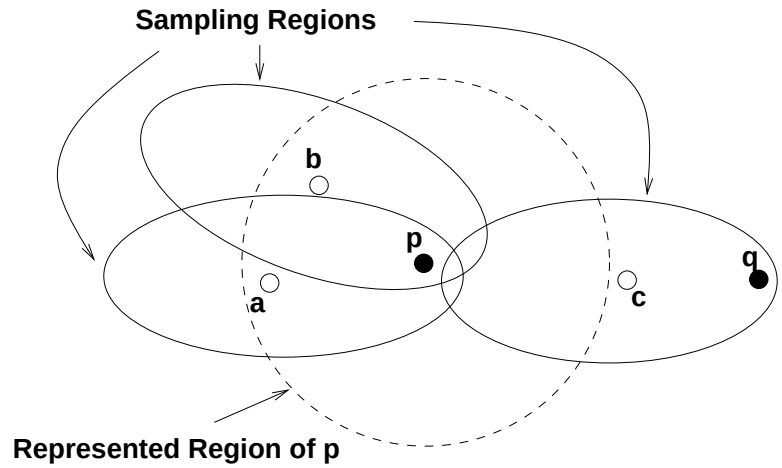

Figure 6.5: *The sample* **p** *lies in the bounded sampling regions of* **a** *and* **b**. *The represented region of* **p** *is bounded. The sampling region of* **c** *is represented by another sample* **q**. *Bounded sampling regions bounds the maximum distance between the samples* **p** *and* **q**.

distance between samples is determined by local curvature variation. This analysis again justifies our intuition behind the sampling conditions and the sampling region that in low curvature region the sample points can be spread out and in high curvature region the samples are close together.

6.6.2 Cover Regions and the Relative Minimum Distance Between Samples

Consider a 2D problem of sampling a curve as shown in Figure 6.6. Assume we have a set of samples satisfying the sampling condition as shown in the top of the figure. An additional sample point, **r**, arbitrarily close to **p** makes the cover region of **p** shrink, uncovering a portion of the surface. This violates Condition 3 which requires covering of the surface by cover regions. Due to this violation, the reconstruction algorithm does not connect samples **p** and **q** as their cover regions do not intersect, thus leaving them as boundary points.

Analysis of this situation leads to the following question: how close can the sample **r** be to **p** so that **q** is considered close enough to be a neighbor of **p** in the reconstructed curve? The answer depends on how far **q** is from **p**, and the sizes of cover regions of **p** and **q**. Whatever may be the answer, the question by itself reveals the fact that **r** cannot be arbitrarily close to **p**, and it has to maintain a minimum distance from **p** based on (or relative to) the distance between **p** and **q** and the sizes of the cover regions. Thus the Condition 3, along with the definition of cover region, impose restrictions on relative minimum distance between samples for correct reconstruction of the surface.

Recall from Section 5.4.2 that non-monotonic sampling is needed for surfaces with boundaries. The example illustrated using Figure 6.6 shows that the sampling conditions discussed in the chapter

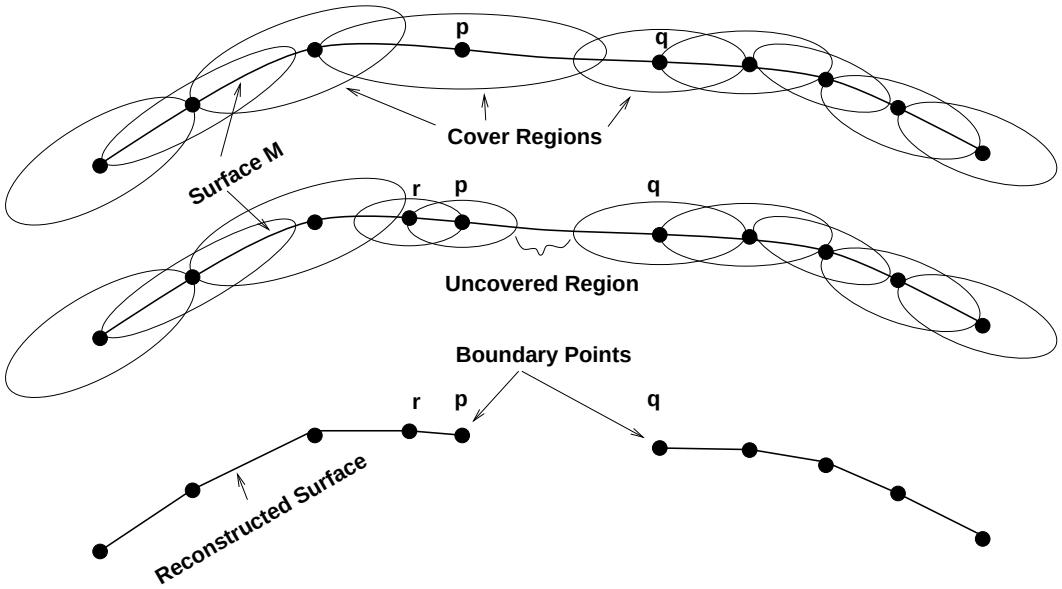

Figure 6.6: *The top picture shows that the cover regions of sample points cover the surface. An additional sample (***r***) arbitrarily close to* **p** *uncovers some of the surface and thus violates the sampling condition. Resulting reconstructed surface leaves* **p** *and* **q** *as boundary points.*

are *non-monotonic sampling conditions*. In summary, *any sampling condition for Class C surface reconstruction algorithms for reliable reconstruction of surfaces with boundaries will be characterized by non-monotonicity*.

6.6.3 Existence of Sampling Satisfying the Sampling Conditions

The sampling conditions presented in this chapter are valid only if there exists a sampling that satisfies these conditions. Let us assume that the surface M is compact and piecewise smooth. There are finite number of smooth patches in M and these patches are attached to each other on the boundaries. Hence the boundaries are considered to be part of both the patches. A "dent" on a smooth surface may be a sharp edge or a vertex in the interior of a smooth patch and not necessarily a closed curve between patches. The dents are considered to be in the "closure" of the smooth patch in which it resides. Each smooth patch is considered as a closed set along with their boundaries and "dents". For this setting, let us prove the following theorem.

Theorem 6 *There exists a finite sampling of M that satisfies the sampling conditions.*

Proof: This proof uses the definition of *represented region* given in Section 6.6.1. Each patch is a closed subset of M, and hence compact. Each patch is sampled satisfying condition 1, assuming

that the boundaries and dents are part of a smooth surface and not curves. Let us assume that the represented region of a point is an open set.

The union of all represented regions cover M. Since M is compact, there exists a finite number of represented regions that cover M. The corresponding sample points satisfy the sampling condition 1. Now the boundary curves, edges, and dents are sampled using the theory developed for curves in this chapter. This set of curves is a closed subset of M and hence compact, except that the constituent curves have lower dimension. Again the same argument can be used to choose a finite number of sample points to cover the boundary curves, edges, dents, and vertices. The union of these two finite sets of sample points is finite and they satisfy condition 1 of the sampling conditions. Let this finite set of samples be S_1.

Now the above sample points should satisfy condition 2 where for every point in M, the closest sample point is in the closure of its sampling region.

Define the *closest sampling region* of $\mathbf{a} \in M$ as $C_s(\mathbf{a}) = C(\mathbf{a}) \cap S(\mathbf{a})$, where $C(\mathbf{a})$ and $S(\mathbf{a})$ are the cover region and sampling region of **a** respectively. (Note even though we use cover regions for sample points, it is defined for any point on the surface).

Let $N = \{\mathbf{a} \in M \mid \nexists \mathbf{p} \in S \cap \mathbf{cl}(C_s(\mathbf{a}))\}$. The set N contains those points in M for which sampling condition 2 is not satisfied. Let $M_2 = \mathbf{cl}(\cup_{\mathbf{a} \in N} C_s(\mathbf{a}))$. M_2 is compact, and does not contain any sample point from S_1. Assuming $C_s(\mathbf{a})$ as the sampling region, finite number of samples are used to sample M_2 in the same way as M was sampled. Let this finite set of samples be S_2. The samples in the union of the sets $S_1 \cup S_2$ satisfy Conditions 1 and 2.

If the cover regions of the sample points $S_1 \cup S_2$ cover the surface then the proof is done. Assume that the cover regions of the above sample points do not cover the surface. Place a sample point **q** in the uncovered region. Since **q** is not in any cover region, no cover region is affected (no cover region shrinks). Point **q** has its own cover region. Repeat the above operation till no part of M is uncovered. Since cover regions are open, the union of cover regions cover M, and M is compact, there are finite number of cover regions that cover M, and hence finite number of sample points that sample M according to the sampling conditions 1-3. It is difficult to prove the finiteness of sample points with conditions 4 and 5 as they involve projections of the cover regions to the tangent planes. So I ignore the last two conditions in this proof. $\diamondsuit$

6.7 Tools for Reconstruction

In this section, I develop a few mathematical tools that can be used during the reconstruction process. In the Localized Delaunay Triangulation algorithm presented in Chapter 4, we collected the candidate neighbors for each sample point. Neighbors of a sample point in the final triangulation are chosen from these candidate points. Hence choosing candidate points is an important step in the algorithm. The following is the formal definition of candidate neighbors.

Definition 11 *If the cover regions of two sample points* $\mathbf{p}$ *and* $\mathbf{q}$ *intersect, then they are* candidate neighbors *of each other.*

Hence the core of the reconstruction algorithm is to detect intersection of cover regions. But the cover regions, and hence their intersection, are described using the surface definition, whereas, we have to use the intersection of cover regions to find the surface definition. To break this inter-dependency, we assume that if the ellipsoids that contain the cover regions, C^R, intersect then the cover regions, C, intersect. The other direction holds because $C(\mathbf{p}) \subset C^R(\mathbf{p})$.

Lemma 5 *If* $C(\mathbf{p}) \cap C(\mathbf{q}) \neq \emptyset$ *then* $C^R(\mathbf{p}) \cap C^R(\mathbf{q}) \neq \emptyset$.

Assumption 1 *If* $C^R(\mathbf{p}) \cap C^R(\mathbf{q}) \neq \emptyset$ *then* $C(\mathbf{p}) \cap C(\mathbf{q}) \neq \emptyset$.

Now the problem is to find if two ellipsoids intersect. We can compute the intersection using classical variable elimination theory. By most of the techniques, the problem of finding whether two ellipsoids intersect boils down to finding the actual intersection curve, or at least a point of intersection. This is time consuming, and we prefer to find fast, even if approximate, way to determine if two ellipsoids intersect.

For the sake of brevity in notation, in the following theorem, I use $L_{\mathbf{p}}$ to denote $\sqrt{Q_{\mathbf{p}}(U^n(\mathbf{p}))}$, the distance to the closest sample point from $\mathbf{p}$.

Theorem 7 *Let* $\mathbf{p}$ *and* $\mathbf{q}$ *be two sample points with* $L_{\mathbf{p}}$ *and* $L_{\mathbf{q}}$ *as their distances to their respective closest sample point. These distances define the ellipsoids* $C^R(\mathbf{p})$ *and* $C^R(\mathbf{q})$. *If* $C^R(\mathbf{p}) \cap C^R(\mathbf{q}) \neq \emptyset$ *then*

$$Q_{\mathbf{p}}(\mathbf{q}) \leq \frac{3}{2}\left(L_{\mathbf{p}}^2 + L_{\mathbf{q}}^2\right) + 2\sqrt{2}L_{\mathbf{p}}L_{\mathbf{q}}, \text{ or} \tag{6.21}$$

$$Q_{\mathbf{q}}(\mathbf{p}) \leq \frac{3}{2}\left(L_{\mathbf{p}}^2 + L_{\mathbf{q}}^2\right) + 2\sqrt{2}L_{\mathbf{p}}L_{\mathbf{q}} \tag{6.22}$$

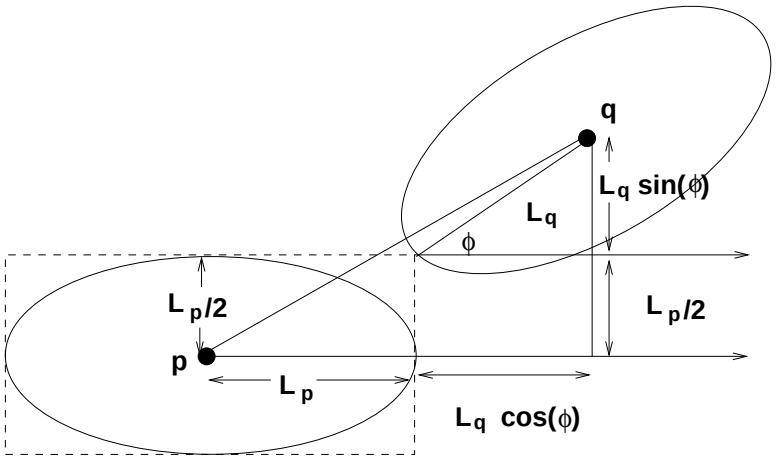

Figure 6.7: *Configuration to find the maximum $Q_{\mathbf{p}}(\mathbf{q})$ when $C^R(\mathbf{p})$ and $C^R(\mathbf{q})$ intersect.*

Proof: Let us assume a conservative configuration of the ellipsoids around **p** and **q** such that the distance $Q_{\mathbf{p}}(\mathbf{q})$ is maximized. For ease of computation, I consider a rectangular box around the ellipsoid around **p** and assume that the ellipsoid around **q** touches the box as shown in Figure 6.7. The $Q_{\mathbf{p}}(\mathbf{q})$ is maximized at this configuration for a particular value of ϕ. Without loss of generality, let us assume that $L_{\mathbf{p}} > L_{\mathbf{q}}$.

$$\begin{aligned} Q_{\mathbf{p}}(\mathbf{q}) &= T^2_{\mathbf{p}}(\mathbf{q}) \ + \ E^2(\mathbf{p},\mathbf{q}) \\ &= \left(L_{\mathbf{q}}\sin(\phi) + \frac{L_p}{2}\right)^2 + \left(\left(L_{\mathbf{p}} \ + \ L_{\mathbf{q}}\cos(\phi)\right)^2 \ + \ \left(L_{\mathbf{q}}\sin(\phi) + \frac{L_p}{2}\right)^2 \right) \\ &= L^2_{\mathbf{q}} \ + \ L^2_{\mathbf{q}}\sin^2(\phi) \ + \ \frac{3}{2}L^2_{\mathbf{p}} \ + \ 2L_{\mathbf{p}}L_{\mathbf{q}}(\sin(\phi)+\cos(\phi)) \end{aligned} \tag{6.23}$$

The right hand side of 6.23 maximizes at $\phi = \pi/4$, when

$$Q_{\mathbf{p}}(\mathbf{q}) \leq \frac{3}{2}\left(L^2_{\mathbf{p}} + L^2_{\mathbf{q}}\right) \ + \ 2\sqrt{2}L_{\mathbf{p}}L_{\mathbf{q}}. \tag{6.24}$$

$\diamondsuit$

Let the angle between the normal vectors at **p** and **q** be θ. The minimum value of θ is ϕ in the above equation. In the equation 6.23, if we substitute $\theta = \phi = 0.2$ radians, the value we assumed in Section 6.4.1, then we get

$$Q_{\mathbf{p}}(\mathbf{q}) < \left(1.22\left(L_{\mathbf{p}} \ + \ L_{\mathbf{q}}\right)\right)^2. \tag{6.25}$$

If the ellipsoid around **q** was actually touching the right end of the ellipsoid around **p**, instead of the tip of the rectangular box, then we get a better bound as,

$$Q_{\mathbf{p}}(\mathbf{q}) \leq \left(L_{\mathbf{p}} + L_{\mathbf{q}}\right)^2. \tag{6.26}$$

If the cover region was a circle on a plane, then we get exactly the same bound as above.

6.7.1 The Shrink Factor t

We used the shrink factor t in Sections 6.4.2 and 6.4.3 to shrink the *curvature based region* and *sheet based region* of M. The neighbors of a sample point in the final triangulation lie in these regions of M. These regions, and hence the neighbors, are found during the reconstruction process by estimating the distance to the farthest neighbor. Incorrect estimation of farthest distance leads to incorrect choice of neighbors, which in turn will lead to incorrect triangulation of M. The shrink factor t can be viewed as a safety factor in searching for neighbors such that even with the worst case estimate of distance to the farthest neighbor, the correctness of the triangulation is not affected. In this section I fix the value of t for its use in the sampling process.

Lemma 6 *The space covered by the set* $\left\{\mathbf{a} \mid \mathbf{a} \in M,\ Q_{\mathbf{p}}(\mathbf{a}) < L_{\mathbf{p}}^2/t\right\}$ *is homeomorphic to a disc.*

Proof: Follows from the sampling condition 2 and the definition of the sampling region. ♢

As we discussed in the previous section, finding the intersection between ellipsoids is expensive. Hence, we are going to assume that the converse of the Theorem 7 is true. In essence, we are assuming that if the distance of $\mathbf{q}$ from $\mathbf{p}$ is less than $\frac{3}{2}\left(L_{\mathbf{p}}^2 + L_{\mathbf{q}}^2\right) + 2\sqrt{2}L_{\mathbf{p}}L_{\mathbf{q}}$ then the ellipsoids $C^R(\mathbf{p})$ and $C^R(\mathbf{q})$ intersect. Such samples like $\mathbf{q}$ are possible neighbors of $\mathbf{p}$ in the final triangulation. Assuming $L_{\mathbf{p}} > L_{\mathbf{q}}$ as before, Equation 6.24 becomes

$$Q_{\mathbf{p}}(\mathbf{q}) \leq\ 5.83\ L_{\mathbf{p}}^2 \tag{6.27}$$

Lemma 7 *If a shrink factor* $0 < t < 0.17$ *is used in the sampling process, then the reconstruction will be reliable.*

Proof: All points within the distance of $5.83\ L_{\mathbf{p}}^2$ from $\mathbf{p}$ are possible neighbors in the final triangulation. Hence the region covered by them *need to be* homeomorphic to a disc. Further, by Lemma 6, all points within a distance of $L_{\mathbf{p}}^2/t$ *are* homeomorphic to a disk. Hence,

$$5.83\ L_{\mathbf{p}}^2 \ <\ L_{\mathbf{p}}^2/t \tag{6.28}$$

$$t \ <\ 0.17 \tag{6.29}$$

Further t has to be positive to avoid the need for infinite number of samples. ♢

6.8 Summary

In this chapter, I introduced the *curvature ellipsoids* at a point **a** on the quadratic approximation of a surface. The region of the surface enclosed within this solid ellipsoid has bounded normal deviation from the normal vector at **a**. Based on the curvature ellipsoids, I defined a family of ellipsoids called *scaled ellipsoids*. These ellipsoids were used in defining the *curvature based region*.

This was followed by the introduction of a distance function D. Relationship between the region of bounded normal deviation around **a** and the distance function D was established. Further, to justify this distance function, a hypothetical sampling condition based on D was shown to have the same properties as the sampling conditions due to Amenta *et al.*[Amenta98b].

Computation of D requires the knowledge of normal curvature and principal directions at every sample point. As this is not practical, another distance function Q was introduced and its relationship with D was established. The distance function Q was used to define three more regions, *the sheet based region*, *the sampling region*, and *the cover region*. Apart from the above four regions, *normal based region*, which is based on angle between normal vectors was also introduced.

Based on these five regions, five sampling conditions were developed and analysis of these sampling conditions were presented. Tools were developed for the reconstruction process to make use of the results and effects of the sampling conditions. These results ensure that the reconstruction process indeed produces geometrically close and topologically correct triangulation of the surface. The proof of this correctness is presented in the next chapter.

CHAPTER 7

THEORY OF SURFACE RECONSTRUCTION

In the previous chapter, I defined conditions for sampling that satisfied the requirements for sampling surfaces with boundaries presented in Chapter 5. In this chapter I define triangulation, properties of triangulation, and the relationship between the triangulation generated from the sample points and the surface.

In this chapter, the following notation is used. The surface that is sampled is denoted by M and the set of sample points by S. Let M be a 2-manifold for initial arguments. Later I will extend the discussion to 2-manifold with boundaries. Triangulation of M is denoted by $L \subset K$, where K is a simplicial complex whose vertices are in S. The tangent plane at a point **a** will be denoted by $T_{\mathbf{a}}$.

Outline of this chapter: In this chapter, I define triangulations, topological and geometric triangulations, of M. I also define a multicovering of M, and derive conditions when this multicovering can be a triangulation of M. Then I construct a function f from the Localized Delaunay Triangulation, L, to M that satisfies these conditions, to prove that L is homeomorphic to M. Finally, I relate a few concepts and results presented in the literature related to the triangulation with the concepts presented in this dissertation. This includes the cover region, *nerve*, and the extraction of triangulation.

7.1 Definitions

Definition: A *triangulation* L of a surface M is a piecewise linear representation of M using triangles, whose topology is same as that of M.

For L to have the same topology of M, there should exist a homeomorphism between L and M.

Definition 12 *A* topological triangulation *of* (M, S) *is a triangulation of* M *with vertex set* S.

For our application, the triangulation vertices are exactly the points sampled from the surface. This definition of topological triangulation allows certain representations of the surface that are undesirable as a faithful representation of the shape or geometry for its use in visualization applications. The following example shows such a case.

For the sake of simplicity, let us discuss an example in two dimensions. Consider the sampling of a circle, and let there be odd number of sample points. "Triangulation" of the circle is a piecewise linear representation of a circle, which is a closed polygon with the sample points as its vertices (refer to Figure 7.1a without the vertex V and its incident triangles). If we connect alternate sample points, we can go around the circle twice and still can construct a closed polygon, though not a simple polygon. This non-simple polygon is homeomorphic to the circle, making it a valid topological triangulation of the circle. The same example in 3D would be the triangulation that covers a torus more than once, and still is homeomorphic to a torus. As such a triangulation covers an object multiple times, it is called a multi-covering of an object. This is not a good representation of the shape of the object for visualization applications. I define triangulation in a different way, which I call a *geometric triangulation*, to eliminate such undesirable cases.

Definition 13 *Let L be a simplicial complex whose vertex set is S. If there exists a homeomorphism $g : |L| \rightarrow M$ such that $g(\mathbf{p}) = \mathbf{p}$ for all $\mathbf{p} \in S$, then L is a* geometric triangulation *of M.*

The known geometric relationship between L and M is that the vertices in L are the sample points on M. A *geometric triangulation* is a *topological triangulation* where the homeomorphism takes the vertices of L to itself, as they are also elements of M. In the rest of the chapter by "triangulation" we mean "geometric triangulation".

Definition 14 *A* multicover *is a simplicial complex L and a map $f : |L| \rightarrow M$ such that the restriction of f to the star of every vertex $\mathbf{p} \in S$ is an embedding.*

Definition 15 *A multicover (L, f) is called a* geometric multicover *if $f(\mathbf{p}) = \mathbf{p}, \forall \mathbf{p} \in S$.*

Definition 16 *Let* **abc** $\in L$ *be a triangle and* **d** $\in S,$ **d** $\neq$ **a**, **b**, **c** *be a vertex in L. Hence* **d** $\in M$. *If* **d** $\in f($**abc**$)$, *then* **d** *is called the internal vertex of the triangle* **abc**.

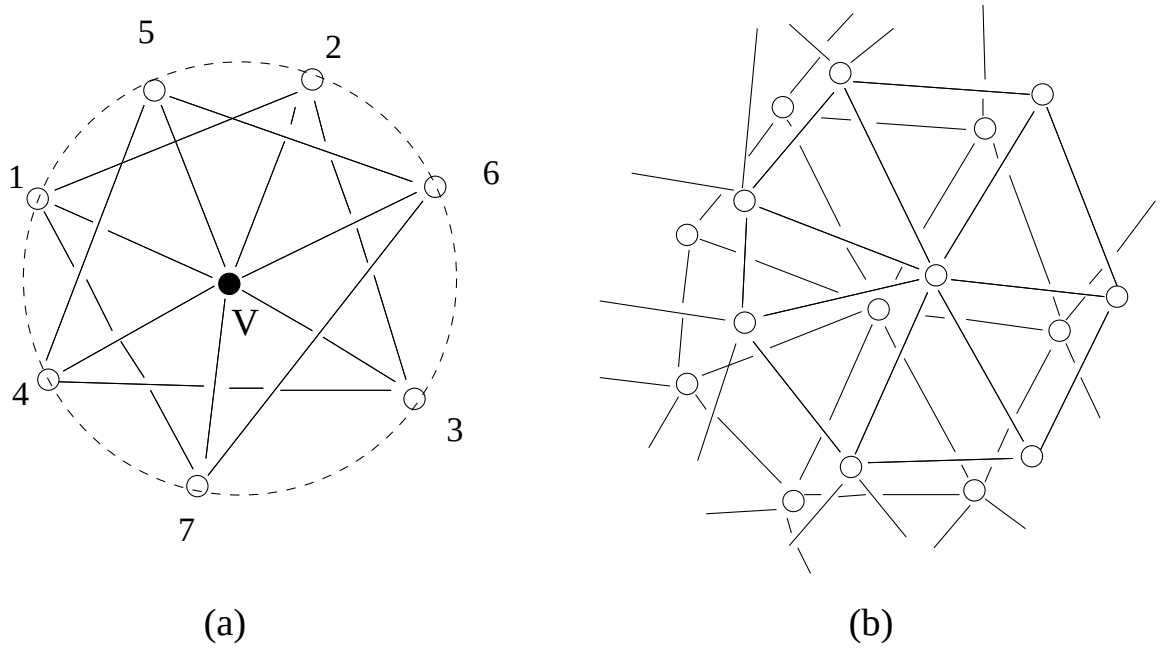

Figure 7.1: *(a) If a function f maps the star of the vertex V to the plane of the paper retaining the order of edges around V as shown above, then f is not a multicover as the mapping of a star is not an embedding. (b) Piece of multicovering triangulation of a sphere. If the mapping function is f as in* (a)*, note that many triangles have interior samples.*

7.2 Multicovering and Geometric Triangulation

In this chapter, we prove the following theorem.

Theorem: *If $f : |L| \rightarrow M$ is a geometric multicover with no internal vertex, then f is a homeomorphism, and hence L is a geometric triangulation of M.*

The following are the assumptions and notations used in this proof.

1. L is a finite simplicial complex, $|L|$ is a compact, connected manifold.
2. M is a compact, connected manifold.
3. $|L|$ and M are Hausdorff spaces. (Refer to Section 2.2.2 for the definition of Hausdorff spaces).
4. $f : |L| \rightarrow M$ is a continuous map.
5. $S \in M$ is a set of sample points and is the vertex set of L.
6. Points **x** and **y** are in $|L|$, points **a** and **b** are in M, points **p** and **q** are sample points belonging to both $|L|$ and M.

The following is the outline of the proof. I prove in Lemma 9 that there exists an open set $B_{\mathbf{a}}$ around every point **a** in the image of $|L|$ in M such that the number of pre-images of any point in the set is greater than or equal to the number of pre-images of **a**. In Lemma 11, I prove that within this open set $B_{\mathbf{a}}$, the set A containing the points with strictly greater number of pre images than of **a** is closed in $B_{\mathbf{a}}$. This leads to the result in Lemma 10 that the multicover map f from $|L|$ to M is onto. Then in Theorem 8, I prove that A is empty and hence all points in M have same number of pre-images. The multicover f is a homeomorphism if and only if the number of pre-images is one (Lemma 12). The relationship between the internal vertex, and the number of preimages is established in Lemma 13. All these results are put together to prove the final theorem, Theorem 9, that ties the concepts of geometric triangulation, geometric multicover, and the internal vertex.

7.2.1 Theorems and Proofs

Lemma 8 *Let $f : |L| \rightarrow M$ be a multicover. There exists $\epsilon > 0$ such that $\forall \mathbf{x} \in |L|$, $f(B(\mathbf{x}, \epsilon))$ is an embedding.*

Proof: For all $\mathbf{p} \in S$, Star(**p**) is open. Since f is a multicover, $f(\text{Star}(\mathbf{p}))$ is an embedding and hence open. For every $\mathbf{x} \in |L|$, there exists $\mathbf{p} \in L$ such that $\mathbf{x} \in$ Star(**p**), since the union of stars covers $|L|$. For every $\mathbf{x} \in |L|$, there exists $\delta_{\mathbf{x}} > 0$ such that $B(\mathbf{x}, 2\delta_{\mathbf{x}}) \subseteq$ Star(**p**) for some $\mathbf{p} \in L$, since a star is open. Further, the union of open sets $\cup_{\mathbf{x} \in |L|} B(\mathbf{x}, \delta_{\mathbf{x}})$ covers $|L|$. Since $|L|$ is compact, there exists finite number of **x** such that $\cup_{\mathbf{x} \in \text{finite set}} B(\mathbf{x}, \delta_{\mathbf{x}})$ covers $|L|$. Further, we can choose open sets such that no open set is contained in another. Let the chosen **x** be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. Let $\epsilon = \min_{\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}} \{\delta_{\mathbf{x}}\}$.

For every $\mathbf{y} \in |L|$, there exists $\mathbf{x}_i$ such that, $B(\mathbf{y}, \epsilon) \subseteq B(\mathbf{x}_i, \delta_{\mathbf{x}_i} + \epsilon) \subseteq B(\mathbf{x}_i, \delta_{\mathbf{x}_i} + \delta_{\mathbf{x}_i}) = B(\mathbf{x}_i, 2\delta_{\mathbf{x}_i}) \subseteq$ Star(**p**), for some $\mathbf{p} \in L$. Hence $\forall \mathbf{y} \in |L|$, $f(B(\mathbf{y}, \epsilon))$ is an embedding. $\diamondsuit$

Definition 17 *The* pre-image index *of* $\mathbf{a} \in M$*, denoted by* PI(**a**)*, is the number of distinct pre-images of* **a** *in* $|L|$ *under* f*. That is,* PI(**a**) = $|f^{-1}(\mathbf{a})|$ *where* $f^{-1}(\mathbf{a}) = \{\mathbf{x} \in |L| \mid f(\mathbf{x}) = \mathbf{a}\}$.

The above definition is valid only if the number of pre-images of any $\mathbf{a} \in M$ is finite which is true as L is a finite simplicial complex, and the image of a star is an embedding.

Lemma 9 *For every* $\mathbf{a} \in f(|L|)$*, there exists* $\delta_{\mathbf{a}} > 0$ *such that* $\forall \mathbf{b} \in B(\mathbf{a}, \delta_{\mathbf{a}})$, $PI(\mathbf{b}) \geq PI(\mathbf{a})$.

Proof: For any point $\mathbf{a} \in f(|L|)$ there is a finite number of pre-images. Let $f^{-1}(\mathbf{a}) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, so $\mathbf{x}_i \in |L|$ are the distinct pre-images of $\mathbf{a}$. By Lemma 8 there exists $\epsilon > 0$ such that $f(B(\mathbf{x}_i, \epsilon))$ is an embedding in M. Further, as $|L|$ is a Hausdorff space, there exists open sets $U_1, U_2, \ldots, U_n$ such that $\mathbf{x}_i \in U_i \subseteq B(\mathbf{x}_i, \epsilon)$, and for every $i \neq j$, $U_i \cap U_j = \emptyset$. The set $\cap_{1 \leq i \leq n} f(U_i) \neq \emptyset$ since it contains $\mathbf{a} = f(\mathbf{x}_i)$. This intersection set, $\cap_{1 \leq i \leq n} f(U_i)$, is open as there are finite number of U_is and $f(U_i)$ are open as f is an embedding. This shows that there exists $\delta_{\mathbf{a}} > 0$ such that $\forall \mathbf{b} \in B(\mathbf{a}, \delta_{\mathbf{a}}) \subseteq \cap f(U_i),\ PI(\mathbf{b}) \geq PI(\mathbf{a})$. Further, as the U_i's are disjoint, the pre-images of **b** are distinct. ♢

Lemma 10 *Multicover $f : |L| \rightarrow M$ is onto.*

Proof: The spaces $|L|$ and M are compact, connected manifolds. A topological space M is said to be *connected* if there does not exist nonempty disjoint open sets that cover M. Since $|L|$ is compact, and f is a map, B$= f(|L|) \subseteq M$ is also compact and hence closed and bounded. The set C$= M-$B is open since B is closed and M is compact. For all $\mathbf{b} \in$ B, $PI(\mathbf{b}) > 0$, and for all $\mathbf{c} \in$ C, $PI(\mathbf{c}) = 0$. By Lemma 9, for every $\mathbf{b} \in$ B$= f(|L|)$, there exists $\delta_{\mathbf{b}} > 0$ such that for all $\mathbf{d} \in B(\mathbf{b}, \delta_{\mathbf{b}})$, $PI(\mathbf{d}) \geq PI(\mathbf{b})$. Hence, for all $\mathbf{b} \in$B, $B(\mathbf{b}, \delta_{\mathbf{b}}) \cap$ C $= \emptyset$. The set D$= \cup_{\mathbf{b} \in \text{B}} B(\mathbf{b}, \delta_{\mathbf{b}})$ is open, B $\subseteq$ D, D $\cap$ C $= \emptyset$, and D $\cup$ C $= M$ implying that M is not connected, a contradiction. Hence either D $= \emptyset$ or C $= \emptyset$. As $|L| \neq \emptyset$ in general, $f(|L|) =$ B $\subseteq$ D $\neq \emptyset$ and hence C $= \emptyset$. Therefore, there exists no $\mathbf{c} \in M$ such that $PI(\mathbf{c}) = 0$, and every $\mathbf{c} \in M$ has a pre-image. The map f is onto. ♢

Lemma 11 *Let $\mathbf{a} \in M$ and $\delta_{\mathbf{a}} > 0$ such that for all $\mathbf{b} \in B(\mathbf{a}, \delta_{\mathbf{a}}), PI(\mathbf{b}) \geq PI(\mathbf{a})$ as given by Lemma 9. The set $A = \{\mathbf{b} \mid PI(\mathbf{b}) > PI(\mathbf{a}), \mathbf{b} \in B(\mathbf{a}, \delta_{\mathbf{a}})\}$ is closed in $B(\mathbf{a}, \delta_{\mathbf{a}})$.*

Proof: As given in the proof of Lemma 9, let $f^{-1}(\mathbf{a}) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \in |L|$ be distinct pre-images of **a**. Note that $f^{-1}(\mathbf{a})$ is non-empty for any $\mathbf{a} \in M$ as f is onto by Lemma 10. There exist open sets $U_1, U_2, \ldots, U_n$ such that $\mathbf{x}_i \in U_i \subseteq B(\mathbf{x}_i, \epsilon)$, $U_i \cap U_j = \emptyset$, $i \neq j$. Further, $B(\mathbf{a}, \delta_{\mathbf{a}}) \subseteq \cap_{1 \leq i \leq n} f(U_i)$. Consider the closed set $X = |L| - \cup_{1 \leq i \leq n} U_i$. The image of X, $f(X) \subset M$ is closed as the image of a closed set is closed under a continuous function. All points $\mathbf{c} \in B(\mathbf{a}, \delta_{\mathbf{a}})$ with $PI(\mathbf{c}) = PI(\mathbf{a})$ will now have $PI(\mathbf{c}) = 0$ in $f(X)$, and other points $\mathbf{b} \in A \subset B(\mathbf{a}, \delta_{\mathbf{a}})$ with $PI(\mathbf{b}) > PI(\mathbf{a})$ will now have $PI(\mathbf{b}) > 0$. By definition, $A = B(\mathbf{a}, \delta_{\mathbf{a}}) \cap f(X)$. The set $f(X)$ is closed, $B(\mathbf{a}, \delta_{\mathbf{a}})$ is open. Therefore, $B(\mathbf{a}, \delta_{\mathbf{a}}) - f(X)$ is open, and $B(\mathbf{a}, \delta_{\mathbf{a}}) - (B(\mathbf{a}, \delta_{\mathbf{a}}) \cap f(X))$ is open. As $B(\mathbf{a}, \delta_{\mathbf{a}}) - A$ is open and $A \subset B(\mathbf{a}, \delta_{\mathbf{a}})$, A is closed in $B(\mathbf{a}, \delta_{\mathbf{a}})$. ♢

Theorem 8 *Let L be a finite simplicial complex, and $|L|$ and M be compact, connected manifolds. If $f : |L| \rightarrow M$ is a map in which the image of every open star of L is an embedding in M, then every $\mathbf{a} \in M$ has same number of pre-images in $|L|$ under f.*

Proof: By Lemma 9 every $\mathbf{a} \in f(|L|)$ has a $\delta_{\mathbf{a}} > 0$ such that $\forall \mathbf{b} \in B(\mathbf{a}, \delta_{\mathbf{a}}),\ PI(\mathbf{b}) \geq PI(\mathbf{a})$, and by Lemma 10, $f(|L|) = M$. Further by Lemma 11, $A = \{\mathbf{b} \mid PI(\mathbf{b}) > PI(\mathbf{a}), \mathbf{b} \in B(\mathbf{a}, \delta_{\mathbf{a}})\}$ is closed in $B(\mathbf{a}, \delta_{\mathbf{a}})$. The set $B(\mathbf{a}, \delta_{\mathbf{a}}) - A \neq \emptyset$ as $\mathbf{a} \in B(\mathbf{a}, \delta_{\mathbf{a}})$ and $\mathbf{a} \notin A$. If $A \neq \emptyset$, then $B(\mathbf{a}, \delta_{\mathbf{a}}) \cap A \neq \emptyset$ and this implies that there is a point $\mathbf{c} \in B(\mathbf{a}, \delta_{\mathbf{a}}) \cap A$ that is a limit point of $B(\mathbf{a}, \delta_{\mathbf{a}}) - A$. There exists $\mathbf{t} \in B(\mathbf{c}, \delta_{\mathbf{c}}) \cap (B(\mathbf{a}, \delta_{\mathbf{a}}) - A)$ and hence $PI(\mathbf{t}) \geq PI(\mathbf{c})$ which is a contradiction as $PI(\mathbf{t}) = PI(\mathbf{a}) < PI(\mathbf{c})$. This shows that there does not exist $\mathbf{c}$ with the above criterion, and hence $A = \emptyset$. Therefore all points in $B(\mathbf{a}, \delta_{\mathbf{a}})$ have same number of pre-images. Further, due to Lemma 10, Lemma 9 can be reworded as follows. For every $\mathbf{a} \in M$ there exists a $\delta_{\mathbf{a}} > 0$ such that for all $\mathbf{b} \in B(\mathbf{a}, \delta_{\mathbf{a}})$, $PI(\mathbf{a}) = PI(\mathbf{b})$.

Consider all open sets $B(\mathbf{a}, \delta_{\mathbf{a}})$, for all $\mathbf{a} \in M$. The union of all these open sets, $\cup_{\mathbf{a} \in M} B(\mathbf{a}, \delta_{\mathbf{a}})$, covers M. Since M is compact, there exists a finite number of open sets whose union, say, $\cup_{1 \leq i \leq n} B(\mathbf{a}_i, \delta_{\mathbf{a}_i})$, $\mathbf{a}_i \in M$, covers M. Denote $B_i = B(\mathbf{a}_i, \delta_{\mathbf{a}_i})$. If $B_i \cap B_j \neq \emptyset$ then all $\mathbf{b} \in B_i \cup B_j$ have same number of pre-images. Define an equivalence relation between these open sets as follows. The open sets B_i and B_j are related to each other, denoted as $B_i \mathbf{R} B_j$, if and only if $B_i \cap B_j \neq \emptyset$ or there exists an l such that $B_i \mathbf{R} B_l$ and $B_l \mathbf{R} B_j$. This shows that all points belonging to open sets in one partition have same number of pre-images. Now it remains to be shown that there is only one partition of M under this equivalence relation. Suppose there are k partitions, say, $P_1, P_2, \ldots, P_k$ of M due to this equivalence relation. Then for $s \neq t$, the intersection of any two open sets $B_i \in P_s$ and $B_j \in P_t$, is empty. Let the union of all the open sets in one partition P_i be V_i. Then the set $V_i = \cup B_j,\ B_j \in P_i$ is open. Further, $\cup V_i = M$ and for $i \neq j,\ V_i \cap V_j = \emptyset$. This shows that M is not connected, which is a contradiction. Therefore, there is only one partition of M due to this equivalence relation, and hence all points in M have same number of pre-images. $\diamondsuit$

Definition 18 *The* Multicover Index *of a multicover $f : |L| \rightarrow M$, $MI(f)$ is the number of pre-images of at least one point $\mathbf{x} \in M$.*

The above definition is justified by the fact that all points in M have pre-images (by Lemma 10) and all these points have same number of pre-images (by Theorem 8).

Lemma 12 *A function $f : |L| \rightarrow M$ is a homeomorphism if and only if f is a multicover with $MI(f) = 1$.*

Proof: ($\Rightarrow$) Since f is a homeomorphism it is an one-to-one and onto continuous function whose inverse is also continuous. Hence f is a map where the image of a star in $|L|$ is an embedding in M and every point in M has only one pre-image. Hence f is a multicover with $MI(f) = 1$.

($\Leftarrow$) By Lemma 10, f is onto. By Theorem 8, if $MI(f) = 1$, then every point in M has a unique pre-image and hence f is one-to-one. Since f is a map, f is continuous. $|L|$ is compact and M is a Hausdorff space. A continuous function from a compact space to a Hausdorff space is a homeomorphism [Munkres75]. Therefore, f is a homeomorphism. $\diamondsuit$

Lemma 13 *A geometric multicover $f : |L| \rightarrow M$ has no internal vertex if and only if $MI(f) = 1$.*

Proof: ($\Rightarrow$) Let f be a multicover with no internal vertex, and let $MI(f) > 1$. Since f is a geometric multicover, $f(\mathbf{p}) = \mathbf{p}$ for all sample points $\mathbf{p} \in S$. Let $f^{-1}(\mathbf{p}) = \mathbf{x},\ \mathbf{x} \neq \mathbf{p},\ \mathbf{x} \in |L|$ be one of the pre-images of **p**. There exists a $\mathbf{q} \in S$ such that $\mathbf{x} \in$ Star(**q**). This shows that there exists a triangle **qrs** in the star of **q** such that **x** is in the interior of this triangle or on the edges **qr** or **qs**, or **x** is the vertex **q**. (If **x** is on the edge **rs** or is the vertex **r** or **s**, then **x** belongs to the star of **r** and/or **s** and not of **q**). Let $\mathbf{x} = \mathbf{q}$, and by assumption earlier $\mathbf{x} \neq \mathbf{p}$ and hence $\mathbf{q} \neq \mathbf{p}$. Since $\mathbf{x} = \mathbf{q}$, $f(\mathbf{x}) = \mathbf{q}$ and $f(\mathbf{x}) \neq \mathbf{p}$, which is a contradiction. Therefore, $\mathbf{x} \neq \mathbf{q}$. As f has no internal vertex, **x** cannot lie on the interior of any triangle or edge. Hence there exists no $\mathbf{x} \neq \mathbf{p}$ such that $f^{-1}(\mathbf{p}) = \mathbf{x}$. Therefore $f^{-1}(\mathbf{p}) = \{\mathbf{p}\}$ which implies that $PI(\mathbf{p}) = 1$. By Theorem 8, this implies that for every $\mathbf{a} \in M, PI(\mathbf{a}) = 1$ and by the definition of multicover index, $MI(f) = 1$.

($\Leftarrow$)The function f is a geometric multicover implies that $f(\mathbf{p}) = \mathbf{p} \Rightarrow \mathbf{p} \in f^{-1}(\mathbf{p})$. As $MI(f) = 1$, **p** does not have any other pre-image other than **p** itself and hence **p** is not an internal vertex of any triangle. Since **p** is an arbitrary sample point, no sample point is an internal vertex under f. $\diamondsuit$

Theorem 9 *Let L be a finite simplicial complex, and $|L|$ and M be compact, connected manifolds. Further, let the vertex set of L, S, be a subset of M. (L,f) is a geometric triangulation of M if and only if $f : |L| \rightarrow M$ is a map with the following properties:*

i. the image of every open star of L is an embedding in M,

ii. $f(\mathbf{p}) = \mathbf{p},\ \forall \mathbf{p} \in S$, and

iii. f *has no internal vertex.*

Proof: ($\Rightarrow$) Since (L,f) is a geometric triangulation, property *(ii)* is satisfied, and f is a homeomorphism. Since f is a homeomorphism, by Lemma 12, f is a multicover with $MI(f) = 1$. Further, f is a geometric multicover due to property *(ii)*. Since f is a geometric multicover with $MI(f) = 1$, it satisfies property *(i)* (by the definition of multicover) and property *(iii)* (by Lemma 13).

($\Leftarrow$)Due to properties *(i)* and *(ii)*, f is a geometric multicover and due to property *(iii)* and by the Lemma 13, $MI(f) = 1$. By Lemma 12, if $MI(f) = 1$ then f is a homeomorphism. As the homeomorphism f satisfies property *(ii)*, (L,f) is a geometric triangulation of M. $\diamondsuit$

7.3 Geometric Triangulation and Localized Delaunay Triangulation

In this section, I prove that the Localized Delaunay Triangulation is homeomorphic to the underlying surface. I define a few terms used in this section and state an assumption before proving this result.

Definition 19 *"Projection" of a point* **b** *means the following operation. A point* **b** *is projected orthogonally to the plane fixed at an origin, say* **a**. *Let the projection of* **b** *be* $\mathbf{b}'$. *The vector* $\mathbf{ab}'$ *is scaled to the length of the vector* **ab**.

The length is measured using the Euclidean metric.

Assumption 2 *The projections of a triangulation of* M *around a vertex* $\mathbf{a} \in S$ *and the neighborhood of* **a** *in* M *to the tangent plane* $T_{\mathbf{a}}$ *are one-to-one and continuous.*

Definition 20 *Let* **abc** *be a triangle, and* $\mathbf{d} \in S - \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ *be in a candidate neighbor of* **a**. *Consider the projection of* **abc** *and* **d** *to* $T_{\mathbf{a}}$. *If* **d** *projects to the interior of the projection of* **abc** *then* **d** *is called an* interior sample of the triangle **abc**.

Theorem 10 *The Localized Delaunay Triangulation,* L*, is homeomorphic to the underlying surface* M.

Proof: The Localized Delaunay Triangulation algorithm chooses the candidate point around the reference point using the sampling conditions presented in the previous chapter. These sampling conditions ensure that the triangulation around a reference point using these candidate points is homeomorphic to a disk. Further, the Delaunay triangulation on the tangent plane of the reference

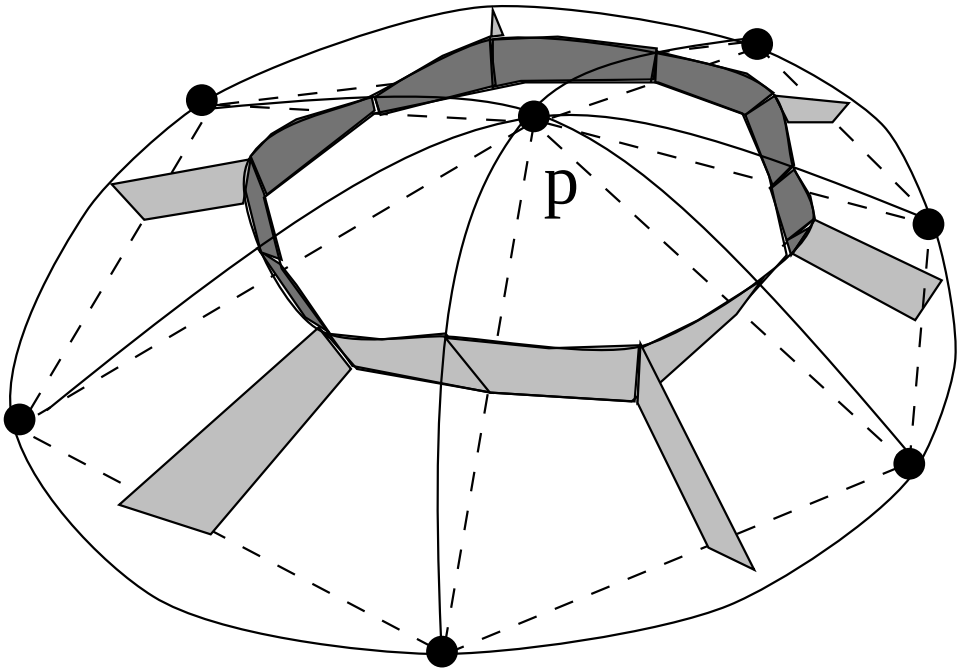

Figure 7.2: *The line segments connecting the centroids of the triangle with the midpoints of the edges define a region around a sample. The planes are constructed perpendicular to the triangles, and passing through these line segments. These planes intersect the surface above and define another region on the surface around the sample point.*

point ensures that the projection of the star of the reference point to its tangent plane is one-to-one and there is no interior vertex in the projection of this star.

Choose the mid points of the edges and the centroids of the triangles incident on each reference point. Connect these centroids and mid points around each sample point, **p**, *in order* using straight lines. This results in a closed polygon on the triangulation enclosing a region containing **p** (Figure 7.2). Let this region be $R_{\mathbf{p}}$. Note that there is no sample point in $R_{\mathbf{p}}$ other than **p**, and $\cup_{\mathbf{p}\in S} R_{\mathbf{p}}$ covers the triangulation. Further, the intersection of the interiors of any two regions $R_{\mathbf{p}}$ and $R_{\mathbf{q}}$, $\mathbf{p} \neq \mathbf{q}$ is empty.

Consider the boundary of the region $R_{\mathbf{p}}$. This is made up of two edges on each triangle incident on **p** (Figure 7.2). For each edge, construct a plane passing through it and perpendicular to the triangle in which the edge lies. Since the projection of the triangulation to the tangent plane is one-to-one, these planes intersect the tangent plane of **p** and this intersection yields a simple polygon on the tangent plane. Further, since the projection of the neighborhood of **p** in surface M to the tangent plane of **p** is one-to-one, these planes intersect M and form a simple closed curve on M. Let the region defined by this closed curve be $P_{\mathbf{p}}$. Clearly $P_{\mathbf{p}}$ contains **p**. There is a one-to-one correspondence between the edges of $R_{\mathbf{p}}$ on the triangulation and curve segments of $P_{\mathbf{p}}$, and the order of edges of $R_{\mathbf{p}}$

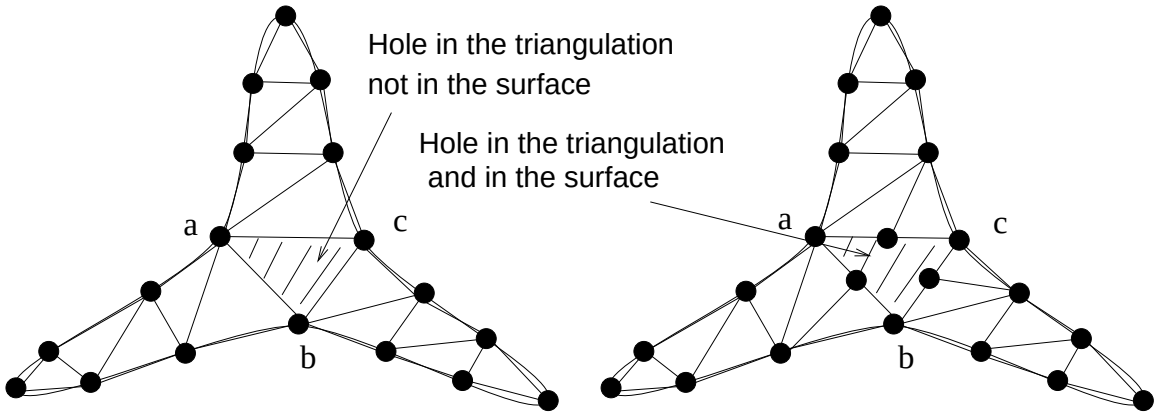

Figure 7.3: *(a) A boundary in the middle of the object is allowed by the properties of the triangulation even though there is no boundary in the actual model. This is avoided by considering a triangle* **abc** *if there exist edges* **ab**, **bc**, *and* **ca**. *(b) If there were an actual hole* **abc** *in the model, then according the sampling conditions, there needs to be a sample point representing each edge of this triangle. Final triangulation with these vertices on edges is also shown, which retains that hole.*

around **p** is preserved in $P_{\mathbf{p}}$. Further, there is no sample point other than **p** in $P_{\mathbf{p}}$. By construction, the intersection of the interiors of any two regions $P_{\mathbf{p}}$ and $P_{\mathbf{q}}$, $\mathbf{p} \neq \mathbf{q}$, is empty.

Construct a map $f : |L| \rightarrow M$ such that $f(R_{\mathbf{p}}) = P_{\mathbf{p}}$ and $f(\mathbf{p}) = \mathbf{p}$ $\forall \mathbf{p} \in S$. The map f embeds Star(**p**), $\forall \mathbf{p} \in S$. By construction, f does not have any interior vertex and maps every sample point to itself. By Theorem 9, f is a homeomorphism between $|L|$ and M, and hence the Localized Delaunay Triangulation, L, is a geometric triangulation of M. $\diamondsuit$

7.3.1 Triangulation for Surfaces with Boundaries

The results presented in the previous chapter hold for triangulation of manifolds. We have to extend these results to manifolds with boundaries also. By the sampling conditions presented in the previous chapter, the cover regions of the boundary sample points do not intersect if they are not visible from each other. Hence, in general, there will be only one triangle incident on an edge connecting two boundary points, leading to missing triangles in the disk around a boundary sample. Using this feature, we can identify all the boundary points. There might be cases when an edge connecting two boundary points has two triangles incident on it, when the boundary points are connected using an internal edge (Edge **ac** in Figure 6.3). Let us analyze this case.

Consider a hypothetical case as shown in the left of Figure 7.3. Assume that there is no hole **abc** in the actual model. During triangulation, this hole is allowed as the edges **ab**, **bc**, and **ca** connect boundary points, and there can be a missing triangle along these edges. To avoid this situation, we

assume the presence of a triangle **abc** if there exist edges **ab**, **bc**, and **ca** irrespective of whether they are boundary samples or not. This may lead to a conclusion that we cannot have a triangular hole as in Figure 7.3. But according to the sampling condition, if there was a triangular hole, there need to be samples on each edge due to the following reason. The sampling region of any point on the edge does not include the vertices of the triangle as angle between the tangent vectors along the edge and the vertex is more than the tolerance. The sampling under this condition, and its corresponding triangulation is shown in the right of Figure 7.3.

Thus, the boundary points and the triangulation of boundary can be identified consistently, and is supported by appropriate sampling conditions presented in the previous chapter. Once the boundary samples, and the edges connecting them are identified, the results of the previous section can be directly extended to surfaces with boundaries.

7.4 Cover Regions, Nerve, and the Triangulation

In this section I will discuss about other related work on triangulations that can be used for proving the homeomorphism between the triangulation and the surface. I will also discuss about a few gaps that have to be filled for their applicability to our problem.

In the previous chapter we defined the cover regions of the sample points and the ellipsoids $C^R(\mathbf{p})$ around a sample point $\mathbf{p}$ based on distance function $Q_{\mathbf{p}}$. Further in Section 6.7 we showed the importance of estimating the intersection of cover regions for the final triangulation. We also developed theory to approximately find the intersection of ellipsoids. In this section, I use these intersection patterns to construct a simplicial complex K, which is also called the *nerve* of the ellipsoidal intersections.

The complex K has the set of samples S as its vertex set. An edge is constructed between two samples if their ellipsoids intersect, and a triangle is formed if the ellipsoids of three sample points intersect in a common region. Any higher dimensional simplex is removed as they are not of any interest to us with respect to surface reconstruction. The resulting complex is called the *nerve* induced by the intersection pattern of the ellipsoids.

We also assumed in the previous chapter that when two ellipsoids $C^R(\mathbf{p})$ and $C^R(\mathbf{q})$ intersect their cover regions $C(\mathbf{p})$ and $C(\mathbf{q})$ also intersect. So the nerve can be assumed to be induced by the intersection pattern of the cover regions in $\mathbf{R}^3$.

Once K is constructed, the next step is to find the geometric triangulation L of M that is a subset of K. Before finding a triangulation, we have to talk about the existence of a triangulation. Edelsbrunner and Shah [Edelsbrunner97b] showed that if the (cover) regions intersect in a pseudo-disk configuration, where the boundaries of two regions intersect either at zero or two points, there exists a triangulation of the underlying space in the nerve. In our case, I conjecture that the cover regions on a smooth surface intersect in a pseudo-disk configuration, and the above result holds good on surfaces.

Assuming that the triangulation L exists in K, we have to extract L from K. But this problem, in its general setting, is proved to be NP-Complete.

Theorem 11 *Lloyd [Lloyd77] has proved that the following triangulation extraction problem is NP-complete. The* triangulation existence problem *is: Let V be the set of vertices. Let P be the set of all straight line segments between the vertices in V. Given V, and the set of edges $E \subset P$, does there exist a subset $E_s \subset E$, such that E_s is a triangulation of V?*

Lemma 14 *The triangulation problem "Let V be the set of vertices. Let K be the set of all triangles between the vertices in V. Given V, and the set of triangles $T \subset K$, does there exist a subset $L \subset T$ such that L is a triangulation of V?" is NP-complete.*

Proof: Problem as stated in Theorem 11 can be reduced to the problem specified in this lemma. Given the set of edges, the input set is converted to the set of *all possible* triangles formed by the given set of edges in $O(n^3)$ using an exhaustive search algorithm. Then, once the set of triangles that triangulate V is found, its edge set defines the solution to the original problem. Further, given a set of triangles, it can be verified in polynomial time whether this set is a solution to the given problem or not. Hence, given the vertex set V, and the set of triangles T, the problem of finding a subset $L \subset T$ such that L is a triangulation of V, is NP-Complete. $\diamondsuit$

But this result has been proved in a general set of edges (triangles). It still remains open to see whether the problem of finding the triangulation from a special set of edges with properties, for example if the edges are from a nerve complex, is NP-complete.

As verification of a solution for an NP-complete problem can be done in polynomial time, I presented a direct method, the Localized Delaunay Triangulation, to construct a triangulation in Chapter 4, and earlier in this chapter, verified that this triangulation is a geometric triangulation of M.

7.5 Conclusion

I presented the definition of a geometric triangulation and multicovering. The definition of multicovering is based on a local property that the mapping of a star of a sample is an embedding. Based on this local definition, I proved a few global results on the sufficiency conditions for a multicovering function to be a homeomorphism. I also established the relationship between multicovering and geometric triangulation.

Above results were effectively used in constructing a homeomorphism f with the required properties, between the Localized Delaunay Triangulation, L, and the surface M, thus leading to the conclusion that L is a geometric triangulation of M. I also justified the approach of directly constructing the triangulation, rather than extracting a triangulation from the nerve due to the intersection of cover regions, using the existing results like [Edelsbrunner97b] and [Lloyd77].

CHAPTER 8

CONCLUSION

The main contribution of this dissertation is the analysis of the problem of sampling and reconstruction of surfaces with boundaries. This analysis led to the classification of surface reconstruction algorithms and to the realization that the conditions on minimum required sampling density are not sufficient for correct reconstruction of surfaces with boundaries. Till now the variation in the sampling density was used to maximize the number of samples to capture the complex geometric features where ever needed, and minimize the number of samples in featureless regions. Hence the variation in sampling density is seen as an optimization of the sampling and reconstruction method rather than a necessary requirement. In this dissertation, a novel way of using the variation in sampling density to encode the boundary was presented. This sampling approach captures both the geometric and topological features of the surface and requires no additional information other than sample points for the reconstruction process. Based on this analysis and results, an efficient algorithm called the Localized Delaunay Triangulation algorithm was also introduced.

Many fundamental and interesting results in topology were presented in this dissertation. This includes the definition of geometric triangulation, global properties of multicovering functions using just the local definition of multicovering, and relationship between geometric triangulation and multicovering. Using these results, it was proved that the Localized Delaunay Triangulation is homeomorphic to the sampled surface.

8.1 Open Problems

This dissertation also opened up a few interesting problems in geometry and topology. I list a few of the problems here.

1. The example of a surface with boundary in Section 5.3, prove that the Voronoi vertices that lie close the connecting line will be removed by the pole algorithm from consideration. [Probably use perturbation of sample points along the boundary].

2. What are the conditions in which the intersection of cover regions form a pseudo-disk configuration?

3. There exists a triangulation in the nerve of a pseudo-disk configuration on a plane [Edelsbrunner97a]. Does there exist a triangulation when the pseudo-disk configuration is on a 2-manifold?

4. Is there any polynomial time algorithm to extract a triangulation from the nerve?

8.2 Future Work

This dissertation has opened up the field of sampling and reconstruction of surfaces with boundaries for further research. At the same time, this has laid the theoretical foundation and introduced a framework for various algorithms to be developed. The idea of using variation in the sampling density as an encoding mechanism can be used in information encoding on geometric objects for visualization and analysis.

The sample points obtained by the present geometric sampling devices are probabilistic estimates of the positions of the points, and hence noisy. Interpolatory surface reconstruction algorithms (where the surface passes through the samples) fail to reconstruct the correct surface, and the refuge is to rely on approximating surface reconstruction algorithms (where the surface is guided by the samples). Since approximating algorithms assume that the samples are probabilistic estimates, conceptually, the "information content" of each sample is also reduced to a fraction of its value. Amenta *et al.*[Amenta01, Amenta] have proposed the Power Crust algorithm to handle samples of manifolds (with no boundaries) with noise. This is the best combination of interpolatory and approximating algorithms designed till now. In my opinion, this algorithm can be seen as dynamically assigning varying "probabilistic information content" to each sample to decide whether the surface should *pass* through the sample or *guided* by the sample.

There are a few problems due to noise when surfaces with boundaries are considered. The sampling density is based on the positions of points. As the positions are inaccurate due to noise, the estimates of sampling density will not be accurate. If the approach of encoding the presence of

boundary using the variation in sampling density is used, the sampling and reconstruction should take into account the inaccurate sampling density estimates also. Another problem is when the size of the boundary is comparable to the noise level. Hence the field of sampling and reconstruction of surfaces with boundaries in the presence of noise is open for future research. I believe that Class *D* algorithms (refer to Section 5.5) will be useful in addressing this problem.

Boundaries are features of a model, and more information has to be provided to preserve these features. Noise is a "feature" of the sampling process, and more information has to be provided to nullify this "feature". The following question is worth investigating. Can a feature of a model be substituted by a feature of the sampling process using the techniques discussed in this dissertation? In other words, can new algorithms be designed for surfaces with no boundaries but with noise in the samples, using the framework given in this dissertation?

BIBLIOGRAPHY

[Algorri96] Maria-Elena Algorri and Francis Schmitt. Surface reconstruction from unstructured 3D points. *Computer Graphics Forum*, 15(1):47–60, 1996.

[Amenta] N. Amenta, Sunghee Choi, and Ravi Kolluri. The power crust, unions of balls, and the medial axis transform. *International Journal of Computational Geometry and its Applications*.

[Amenta98a] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60:125–135, 1998.

[Amenta98b] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of ACM Siggraph*, pages 415–421, 1998.

[Amenta99] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry*, 22:481–504, 1999.

[Amenta00] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proc. 16th ACM Symposium on Computational Geometry*, pages 213–222, 2000.

[Amenta01] N. Amenta, Sunghee Choi, and Ravi Kolluri. The power crust. In *Solid Modeling*, pages 249–260, 2001.

[Attali97] D. Attali. r-regular shape reconstruction from unorganized points. In *ACM Symposium on Computational Geometry*, pages 248–253, 1997.

[B.89] Jackson Leland B. *Digital Filters and Signal Processing*. Kluwer Academic Publishers, 1989.

[Bajaj95] C. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of ACM Siggraph*, pages 109–118, 1995.

[Bernardini97] F. Bernardini and C. Bajaj. Sampling and reconstructing manifolds using alpha-shapes. In *Proc. of Ninth Canadian Conference on Computational Geometry*, pages 193–198, 1997.

[Bernardini99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4), 1999.

[Bittar95] E. Bittar, N. Tsingos, and M. P. Gascuel. Automatic reconstruction from unstructured data: Combining a medial axis and implicit surfaces. *Computer Graphics Forum, Proceedings of Eurographics*, 14(3):457–468, 1995.

[Boissonnat84] J. D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, 1984.

[Crossno99] Patricia J. Crossno and Edward S. Angel. Spiraling edge: Fast surface reconstruction from partially organized sample points. *IEEE Visualization '99*, pages 317–324, October 1999. ISBN 0-7803-5897-X. Held in San Francisco, California.

[Culver00] Tim Culver. *Computing the Medial Axis of a Polyhedron Reliably and Efficiently*. PhD Dissertation, Department of Computer Science, University of North Carolina at Chapel Hill, 2000.

[Curless96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of ACM Siggraph*, pages 303–312, 1996.

[Dey00] Tamal K. Dey, Kurt Mehlhorn, and Edgar A. Ramos. Curve reconstruction: Connecting dots with good reason. *Computational Geometry: Theory and Applications*, 15(4):229–244, 2000.

[Edelsbrunner93] H. Edelsbrunner. The union of balls and its dual shape. In *ACM Symposium on Computational Geometry*, pages 218–231, 1993.

[Edelsbrunner94] H. Edelsbrunner and E. Mucke. Three dimensional alpha shapes. *Transactions on Graphics*, 13(1):43–72, 1994.

[Edelsbrunner97a] H. Edelsbrunner and E. A. Ramos. Inclusion-exclusion complexes for pseudodisk collections. *Discrete and Computational Geometry*, 17:287–306, 1997.

[Edelsbrunner97b] H. Edelsbrunner and Nimish R. Shah. Triangulating topological spaces. *International Journal of Computational Geometry and Applications*, 7(4):365–378, 1997.

[Edelsbrunner01] H. Edelsbrunner. *Geometry and Topology for Grid Generation*. Cambridge University Press, 2001.

[Faugeras93] Oliver Faugeras. *Three-Dimensional Computer Vision – A Geometric Viewpoint*. The MIT Press, Cambridge, Massachusetts, 1993.

[Fua91] P. Fua and P. T. Sander. From points to surfaces. In B. C. Vemuri, editor, *Geometric Methods in Computer Vision*, pages 286–296, 1991.

[Fua92a] P. Fua and P. T. Sander. Reconstructing surfaces from unstructured 3d points. In *Proceedings of Image Understanding Workshop*, pages 615–625, 1992.

[Fua92b] P. Fua and P. T. Sander. Segmenting unstructured 3d points into surfaces. In G. Sandini, editor, *ECCV '92, Proceedings of Second European Conference on Computer Vision*, pages 676–680, 1992.

[Golub89] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins Press, Baltimore, 1989.

[Gopi00] M. Gopi, Shankar Krishnan, and Claudio Silva. Surface Reconstruction using Lower Dimensional Localized Delaunay Triangulation. *Eurographics*, 19(3):467–478, 2000.

[Heckel98] B. Heckel, A. C. Uva, and B. Hamann. Clustering-based generation of hierarchical surface models. In C. M. Wittenbrink and A. Varshney, editors, *IEEE Visualization '98: Late Breaking Hot Topics Proceedings*, pages 41–44, 1998.

[Hook86] T. Van Hook. Real-time shaded NC milling display. In *Proceedings of ACM Siggraph*, pages 15–20, 1986.

[Hoppe92] H. Hoppe, T. Derose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized point clouds. In *Proceedings of ACM Siggraph*, pages 71–78, 1992.

[Hoppe93] H. Hoppe, T. Derose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of ACM Siggraph*, pages 21–26, 1993.

[Isselhard97] Frank Isselhard, Guido Brunnett, and Thomas Schreiber. Polyhedral reconstruction of 3d objects by tetrahedra removal. Technical report, Report No. 288/97,Fachbereich Informatik, University of Kaiserslautern, Germany, Feb. 1997.

[Koenderink89] J. Koenderink. *Solid Shape*. MIT Press, 1989.

[Leibon00] Greg Leibon and David Letscher. Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. In *ACM Symposium on Computational Geometry*, pages 341–349, 2000.

[Lloyd77] E.L. Lloyd. On triangulations of a set of points in the plane. In *18th IEEE Symposium on Foundations of Computer Science*, pages 228–240, 1977.

[Mencl95] R. Mencl. A graph-based approach to surface reconstruction. *Computer Graphics Forum, Proceedings of Eurographics*, 14(3):445–456, 1995.

[Mencl98a] R. Mencl and H. Muller. Graph-based surface reconstruction using structures in scattered point sets. *Proceedings of CGI '98*, 1998.

[Mencl98b] R. Mencl and H. Muller. Interpolation and approximation of surfaces from three-dimensional scattered data points. *State of the Art Reports, Eurographics '98*, pages 51–67, 1998.

[Muller93] H. Muller and A. Klingert. Surface interpolation from cross sections. In H. Hagen, H. Muller, and G. M. Nielson, editors, *Focus on Scientific Visualization*, pages 139–189, 1993.

[Munkres75] James R. Munkres. *Topology: A First Course*. Prentice-Hall Inc., 1975.

[O'Neill97] B. O'Neill. *Elementary Differential Geometry, 2nd Edition*. Academic Press, London, UK, 1997.

[Roth97] G. Roth and E. Wibowoo. An efficient volumetric method for building closed triangular meshes from 3-d image and point data. In *Proc. of Graphics Interface*, pages 173–180, 1997.

[S.84] Jayant N. S. and Noll Peter. *Digital Coding of Waveforms, Principles and Applications to Speech and Video*. Prentice-Hall Inc., 1984.

[Szeliski92] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *Proceedings of ACM Siggraph*, pages 185–194, 1992.

[Teichmann98] M. Teichmann and M. Capps. Surface reconstruction with anisotropic density-scaled alpha shapes. In *Proceedings of IEEE Visualization*, pages 67–72, 1998.

[Terzopoulos88] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36:91–123, 1988.

[Veltkamp95] R. C. Veltkamp. Boundaries through scattered points of unknown density. *Graphical Models and Image Processing*, 57(6):441–452, 1995.

[Weller97] Frank Weller. Stability of Voronoi neighborhood under perturbations of the sites. In *Proc. of Ninth Canadian Conference on Computational Geometry*, 1997.