

**From Range Images to 3D Models**

by

Ravi Krishna Bala Venkata Sai Kolluri

B.E. (Birla Institute of Technology and Science, Pilani) 1998

M.S. (University of Texas, Austin) 2000

A dissertation submitted in partial satisfaction  
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Jonathan Shewchuk, Chair

Professor James O'Brien

Professor Nina Amenta

Professor Ming Gu

Fall 2005



The dissertation of Ravi Krishna Bala Venkata Sai Kolluri is approved:

---

Professor Jonathan Shewchuk, Chair

Date

---

Professor James O'Brien

Date

---

Professor Nina Amenta

Date

---

Professor Ming Gu

Date

University of California, Berkeley

Fall 2005

From Range Images to 3D Models

Copyright © 2005

by

Ravi Krishna Bala Venkata Sai Kolluri



## Abstract

From Range Images to 3D Models

by

Ravi Krishna Bala Venkata Sai Kolluri

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Jonathan Shewchuk, Chair

Surface reconstruction algorithms build digital models of real world objects from data recorded by a scanning device. Since scanning devices are not perfect, they introduce noise and outliers into the recorded data. From such noisy data, an effective reconstruction algorithm must produce models that reflect the geometry and the topology of the sampled surface. In this thesis we analyze surface reconstruction algorithms and describe a software system that we have developed for building three-dimensional models of real world objects.

Implicit methods for surface reconstruction are widely used in computer graphics as they are fast, easy to implement, and scale well to large point clouds. However, these implicit methods come with no provable guarantees on the reconstructed surface. We analyze an implicit surface reconstruction algorithm based on a data interpolation technique called *moving least squares* (MLS). We prove that under certain sampling conditions, the reconstructed surface is an accurate geometric and topological representation of the original surface. Our sampling requirements are adaptive and allow for noise in the input data set.

Delaunay-based surface reconstruction algorithms build the reconstructed surface as a set of triangles from the Delaunay tetrahedralization of the sample points. Many Delaunay-based reconstruction algorithms have been proposed with guarantees on the reconstructed

surface when the input point cloud satisfies certain sampling conditions. However, most point clouds obtained from scanning devices violate these sampling conditions. We present a Delaunay-based reconstruction algorithm called *eigenrust* that uses spectral partitioning to robustly deal with noise, outliers, and regions of undersampling in the input point cloud. We show empirical evidence that our implementation of *eigenrust* is substantially more robust than several closely related surface reconstruction programs.

We describe a software system that we have developed for reconstructing three-dimensional models from data recorded using range scanners. Range images from a scanning device are automatically aligned using point signatures called *harmonic shape contexts*. An implementation of the MLS algorithm defines a smooth surface approximating the scanned surface. Finally, an *eigenrust* implementation meshes the surface defined by the MLS algorithm.

---

Professor Jonathan Shewchuk, Chair

Date

## Acknowledgements

Sincere thanks to my adviser Jonathan Shewchuk. Jonathan always encouraged me to work on projects I was interested in, and provided critical feedback. His advice on research and writing were invaluable.

Thanks also to my adviser at U.T. Austin, Nina Amenta, for introducing me to research, and for supporting me in my move to Berkeley. Many of the ideas in this thesis came out of my work with Nina.

Thanks to James O'Brien as well. It was James who first suggested that I look into moving least squares methods. James and Jonathan complement each other; I benefited a lot from having both of them listen to my ideas, and from their inputs.

My work would not have been possible without the help of many of my colleagues at Berkeley. François Labelle helped me improve many of the proofs related to moving least squares. Thanks to Andrea Frome and Hao Zhang for their help with scan registration. Also thanks to Charless Fowlkes for help with spectral partitioning software.

I had a great time at Berkeley thanks to the Graphics group: Adam Bargteil, Okan Arikan, Chen Shen, Hayley Iben, Bryan Feldman, Jordan Smith, Adam Kirk, Pushkar Joshi, Lillian Chu, and Leslie Ikemoto. Thanks to the folks in the vision group too: Alex Berg, Deva Ramanan, Ashley Iden, Stella Yu, and Ryan White.

I want to thank my parents and my brother, Phani for their support. I consider myself lucky to have them as family. Finally, my deepest gratitude goes to my wife, Shanti. She had faith in me even when I had doubts, and always encouraged me to pursue my interests. I dedicate this thesis to her.

*To Shanti*

# Contents

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Creating 3D Models . . . . .	2
1.2 Related Work . . . . .	6
1.3 Our Contributions . . . . .	7
<b>2 Provably Good Moving Least Squares</b>	<b>9</b>
2.1 Related Work . . . . .	11
2.2 Sampling Preconditions . . . . .	17
2.3 Reconstructed Surface Definition . . . . .	18
2.4 Geometry of the Reconstructed Surface . . . . .	19
2.5 Topological Properties . . . . .	38
2.6 Discussion . . . . .	40
<b>3 Provably Better Moving Least Squares</b>	<b>50</b>
3.1 Guarantees . . . . .	51
3.2 Related Work . . . . .	51
3.3 Sampling Preconditions . . . . .	52
3.4 Formula for the Cut Function . . . . .	54
3.5 Geometry of the Reconstructed Surface . . . . .	58
3.6 Topology of the Reconstructed Surface . . . . .	91
<b>4 An Implementation of Moving Least Squares</b>	<b>99</b>
4.1 Implementation . . . . .	100
4.2 Results . . . . .	105
4.3 Discussion . . . . .	105

<b>5</b>	<b>Eigencrust</b>	<b>108</b>
5.1	Introduction . . . . .	108
5.2	Related work . . . . .	110
5.3	Spectral Surface Reconstruction . . . . .	112
5.4	Results . . . . .	131
5.5	Discussion . . . . .	134
<b>6</b>	<b>Registration</b>	<b>145</b>
6.1	Introduction . . . . .	145
6.2	Related Work . . . . .	146
6.3	Overview of Our Algorithm . . . . .	148
6.4	Shape Contexts as Signatures . . . . .	149
6.5	Coarse Registration, Step 1: Harmonic Shape Contexts . . . . .	152
6.6	Coarse Registration, Step 2: Computing Correspondences and Poses . . . . .	156
6.7	Coarse Registration, Step 3: Pose Improvement . . . . .	159
6.8	Coarse Registration, Step 4: Global Scan Registration . . . . .	160
6.9	Signature Comparison and Results . . . . .	160
	<b>Bibliography</b>	<b>167</b>

# List of Figures

1.1	Pipeline for creating three-dimensional models. . . . .	3
1.2	Registration of range scans. . . . .	4
1.3	Classification of surface reconstruction algorithms. . . . .	5
2.1	MLS reconstruction of a two-dimensional point cloud. . . . .	10
2.2	Sampling preconditions for the uniform MLS algorithm. . . . .	17
2.3	Partition of space into spherical shells. . . . .	19
2.4	The inside and outside offset surfaces of a two-dimensional curve. . . . .	21
2.5	Sample points near a surface point are between two planes. . . . .	22
2.6	Nearby sample points have good point functions. . . . .	26
2.7	Good sample points near the surface. . . . .	30
2.8	Uniform MLS: Proof of topological correctness. . . . .	38
2.9	Improved bounds near the sampled surface. . . . .	42
2.10	Covering argument near the sampled surface. . . . .	43
3.1	Adaptive sampling proportional to surface feature size. . . . .	50
3.2	Computing the function $\epsilon_{fs}$ efficiently. . . . .	56
3.3	Normalization factor in the adaptive MLS algorithm. . . . .	57
3.4	Adaptive offset curves. . . . .	60
3.5	Adaptive MLS: Setup for the geometry results. . . . .	61
3.6	Adaptive MLS: Nearby sample points have good point functions. . . . .	64
3.7	Lower bound for the weight function. . . . .	67
3.8	Definition of radial density functions. . . . .	69
3.9	Relation between the functions $s_{if}$ and $\epsilon_{fs}$ . . . . .	77
3.10	Sample points near a surface point are between two planes. . . . .	80
3.11	Good sample points near the surface. . . . .	81
3.12	Adaptive MLS: Proof of topological correctness. . . . .	91
3.13	Improved bounds for the adaptive MLS algorithm. . . . .	94
4.1	MLS reconstruction of a noisy point cloud. . . . .	100
4.2	Triangulating range images. . . . .	102

4.3	Limitations of the MLS algorithm. . . . .	104
4.4	Eigenrust reconstructions with and without MLS smoothing. . . . .	106
5.1	Eigenrust reconstruction of the angel data set . . . . .	109
5.2	A two-dimensional Delaunay triangulation. . . . .	113
5.3	Pole graph of a two-dimensional point cloud. . . . .	114
5.4	Sliver tetrahedra and poles of a Voronoi cell. . . . .	116
5.5	Edges in the pole graph. . . . .	118
5.6	Angle of intersection of circumscribing spheres. . . . .	119
5.7	Eigenrust reconstruction of the skeleton point cloud. . . . .	122
5.8	Eigenrust reconstruction of the Stanford bunny. . . . .	126
5.9	Eigenrust reconstruction of an undersampled point cloud. . . . .	127
5.10	Eigenrust reconstruction of a mechanical part that has sharp features. . . . .	129
5.11	Eigenrust reconstruction of the Stanford dragon model. . . . .	132
5.12	Reconstructions of the Utah Teapot. . . . .	135
5.13	Outlier experiment comparing different reconstruction algorithms. . . . .	136
5.14	Noise experiment comparing different reconstruction algorithms. . . . .	137
5.15	Eigenrust reconstruction of a sparse point cloud. . . . .	138
5.16	Reconstructions of a sparse point cloud with adaptive sample spacing. . . . .	139
5.17	Eigenrust reconstruction of the Stanford dragon model after MLS smoothing. . . . .	140
5.18	Eigenrust reconstructions of the Stanford bunny after MLS smoothing. . . . .	141
5.19	Eigenrust reconstructions of the armadillo model after MLS smoothing. . . . .	142
5.20	Eigenrust reconstructions of the armadillo model after MLS smoothing. . . . .	143
5.21	Eigenrust reconstructions of the happy buddha model after MLS smoothing. . . . .	144
6.1	The bins of a three-dimensional shape context. . . . .	151
6.2	Polar coordinates in three-dimensions. . . . .	153
6.3	Spherical harmonic basis functions. . . . .	155
6.4	A histogram of harmonic shape context distances. . . . .	161
6.5	Precision-recall curves for spin images. . . . .	162
6.6	Precision-recall curves for fourier shape contexts. . . . .	163
6.7	Precision-recall curves for harmonic shape contexts. . . . .	163
6.8	Comparison of three point signatures. . . . .	164
6.9	Coarse registration results. . . . .	166



# List of Tables

2.1	Comparison of surface reconstruction algorithms. . . . .	12
2.2	Surface reconstruction algorithms comparison-continued. . . . .	13
4.1	Running times for projecting sample points on to the zero set of the MLS cut function. . . . .	105
5.1	Eigencrust reconstruction times for the data sets in Table 4.1. . . . .	134
6.1	Running times of the coarse registration algorithm . . . . .	165

# Chapter 1

## Introduction

Three-dimensional geometry plays an important role in applications such as physical simulation, medicine, and preservation and restoration of archaeological artifacts. Industrial applications include reverse engineering of CAD models and product design. Three-dimensional models required by these applications can be built in two ways: they can either be designed using interactive modeling software, or they can be created by digitizing a physical model. Digitizing is often the more accurate and cost effective option for creating complex models, and for applications such as medical imaging that require an accurate model of an existing physical object. Surface reconstruction algorithms are used to build three-dimensional models of physical objects from data captured using a scanning device.

The output of scanning devices such as laser range scanners is a set of point samples that lie near the surface of a three-dimensional object. In this thesis, we address methods that reconstruct a digital representation of the scanned object from these point samples. In the first part of the thesis, we analyze a surface reconstruction algorithm based on a scattered data interpolation technique called *moving least squares*. We prove that under certain sampling conditions, the reconstructed surface is a good approximation to the sampled surface. In the second part we describe a system for reconstructing surfaces that builds smooth, watertight, outlier-free surfaces from noisy range data.

## 1.1 Creating 3D Models

Developments in the design of laser range scanners allow us to create highly accurate models with millions of points in three-dimensional space. Figure 1.1 illustrates the pipeline for creating three-dimensional models using acquisition devices such as range scanners. There are several steps in the pipeline.

### 1.1.1 Acquiring Models

Methods of acquiring shape information can be broadly classified into active and passive techniques. Active techniques include contact devices that probe the physical model by touching, and non-contact laser range scanners [30] that project light onto the physical object and record shape from the reflected light. Typically, the output of a scanning device is a range scan which gives depth information indicating the surface as seen from the view point of the scanner. Figure 1.2 shows two such range scans. Most modern range scanners have digital cameras that can capture texture information along with the geometry of the scanned object.

A lot of research in computer vision is focused on passive techniques for extracting shape. These techniques include shape from multiple images [71], shape from shading [86], and shape from texture [62]. While passive methods require only commonplace hardware, they are less accurate than active methods and the data obtained is often very noisy. The techniques described in this thesis can be applied to data derived from active as well as passive techniques.

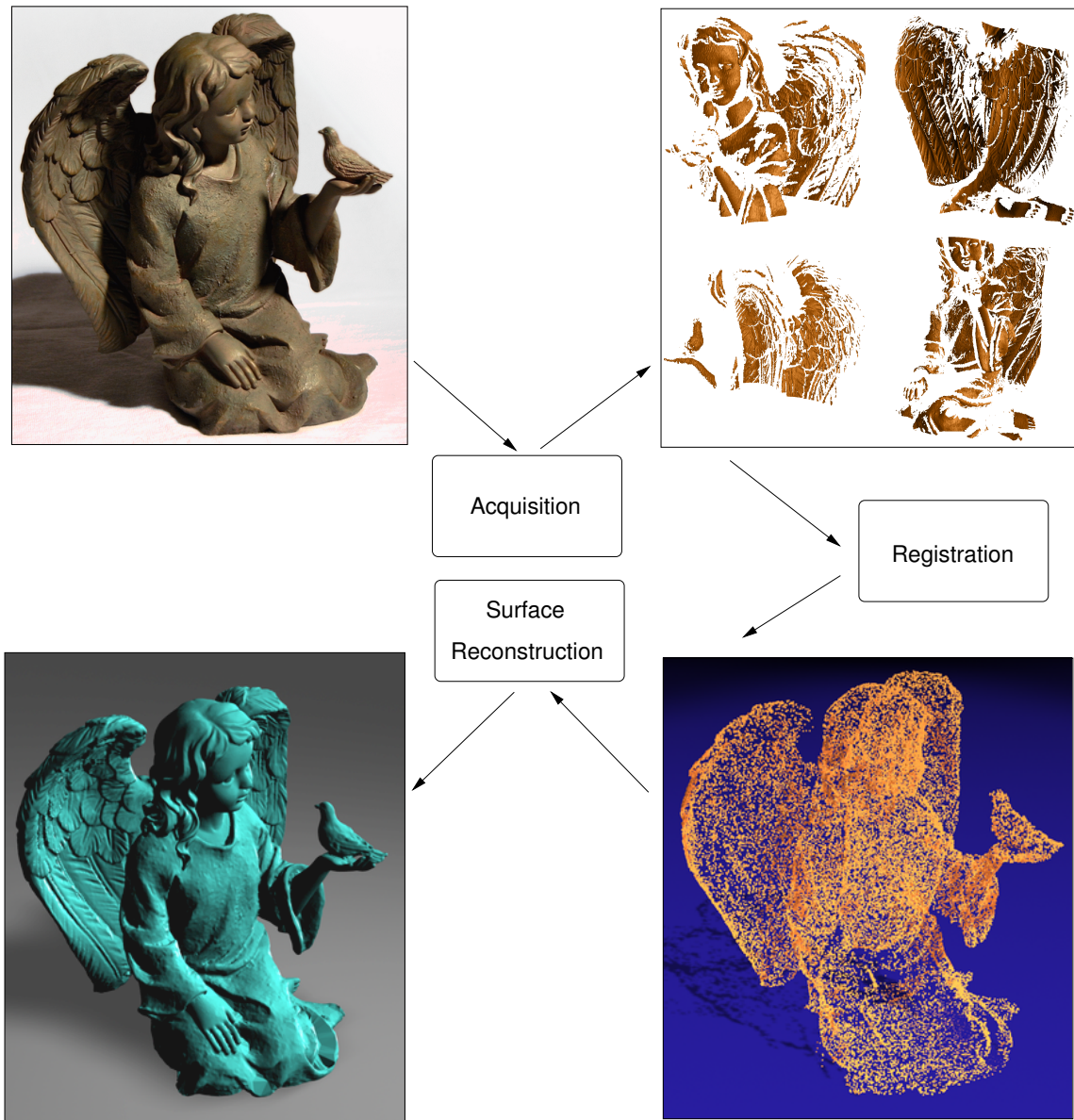


Figure 1.1: Pipeline for creating three-dimensional models using laser range scanners.

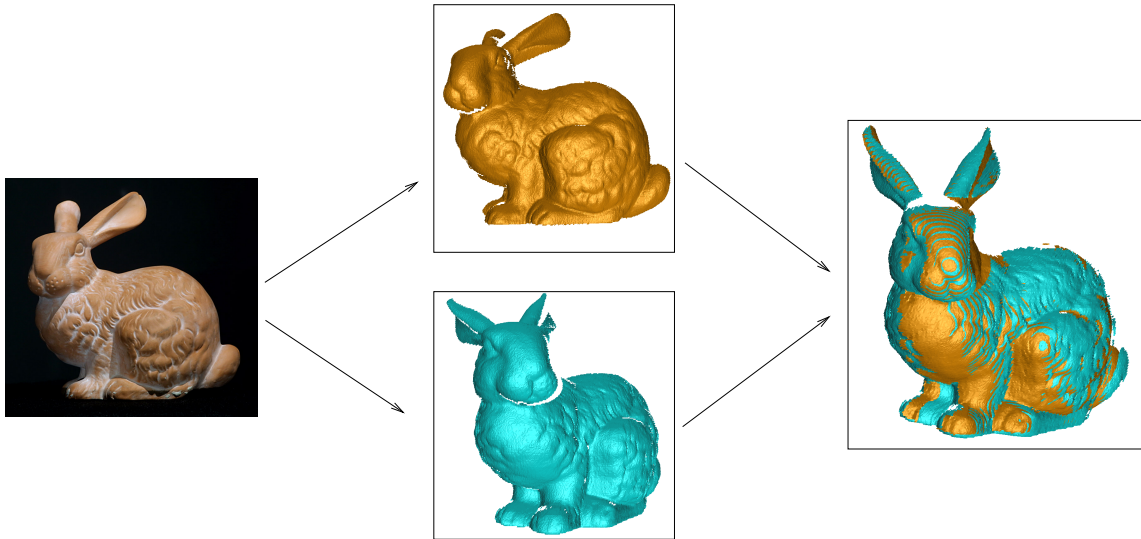


Figure 1.2: Left, the Stanford bunny model. Shown in the center are triangulations of two range scans of the bunny taken from different viewpoints. Multiple scans must be registered as shown on the right to build a single model of the bunny.

### 1.1.2 Registration

Range scanners cannot scan an entire object at once because the whole surface is not visible from a single point of view. Multiple scans, each taken from a different viewpoint, must be *registered*—aligned into a common coordinate system—to form a unified model that covers most or all of the object’s surface. Figure 1.2 shows the alignment of two scans of the Stanford bunny model.

The registration of multiple surfaces can be aided by external measuring devices like trackers attached to a moving scanner, or a calibrated turntable used in conjunction with a fixed scanner, but these methods are not always available or accurate, and registration must often be done by hand or by algorithm.

Automatic surface registration typically takes two computational stages. *Coarse registration* provides a rough positioning of the surfaces, and *fine registration* brings the scans into tight alignment. The most successful technique for coarse registration is based on computing *point signatures* that summarize the shape of the scan relative to a given point.

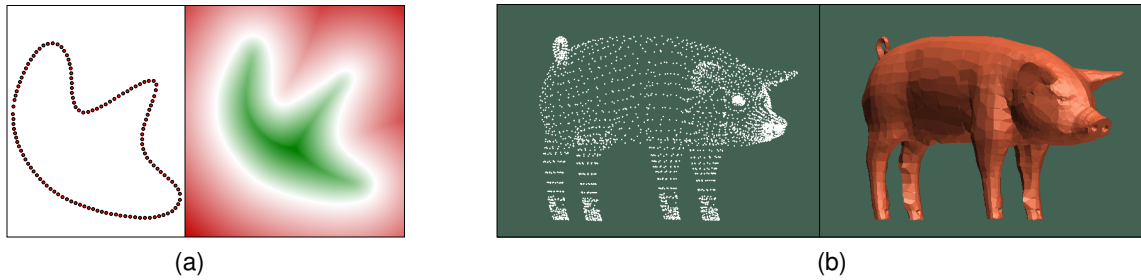


Figure 1.3: (a) Implicit surface reconstruction: The reconstructed surface is the zero set of a scalar function built from the sample points. (b) Mesh-based surface reconstruction: The reconstructed surface is a piecewise linear approximation to the sampled surface.

The computed signatures are used to find points with similar point signatures in different scans. Fine registration is typically done using a variant of Besl and McKay’s iterative closest point algorithm [16] [28].

### 1.1.3 Surface Reconstruction

Scanners invariably introduce at least two kinds of errors into the data they record: *measurement errors* (random or systematic) in the point coordinates, and, *outliers*, which are spurious points far from the true surface. Furthermore, objects often have regions that are not accessible to scanning and so remain undersampled or unsampled. Surface reconstruction algorithms must process such imperfect data and produce a smooth surface that accurately reproduces the topology and the geometry of the sampled surface. They must also scale to handle data with billions of points such as the data sets created in the Digital Michelangelo project [61].

Surface reconstruction algorithms can be broadly divided into implicit methods and mesh-based methods. Implicit methods build a three-dimensional scalar function  $I$  from the point samples. The reconstructed surface is the zero set of  $I$  given by  $\{x \in \mathbb{R}^3 : I(x) = 0\}$ . Figure 1.3(a) shows a two-dimensional example.

Mesh-based methods build a piecewise linear approximation of the sampled surface.

These include Delaunay-based surface reconstruction algorithms that recover the surface as a set of triangles in the Delaunay tetrahedralization of the sample points. Figure 1.3(b) shows an example of a surface reconstructed by a mesh-based surface reconstruction algorithm called Powercrust [6].

The two representations of the reconstructed surface have different applications. Implicit representations are useful in physical simulations, level set methods, and point-based modeling and rendering. They can also be rendered directly using ray marching methods. Surface meshes are important for applications such as rapid prototyping and volume mesh generation for finite element methods.

## 1.2 Related Work

We give an overview of previous work on surface reconstruction in this section. We discuss related work in more detail in later chapters.

Surface reconstruction from unorganized points was introduced to the graphics community by Hoppe, DeRose, Duchamp, McDonald, and Stuetzle [50]. They construct an implicit method based on an approximation of the signed distance function. Implicit surface reconstruction algorithms derived on the data interpolation technique called moving least squares (MLS) are now widely used as part of point-based methods for modeling, simulation, and rendering. MLS methods have been used for scattered data interpolation, and in meshless methods for simulation.

Although implicit methods are widely used in computer graphics, not much work has been done on analyzing the surfaces reconstructed by these methods. All work on provably good surface reconstruction has focused on Delaunay-based methods. The Delaunay-based *crust* algorithm of Amenta and Bern [3] was the first algorithm with theoretical guarantees on the reconstructed surface. The advantage of Delaunay algorithms is that they do not require sample normals and they easily adapt to changes in sample spacing.

The software system Scanalyze [76] is widely used for surface reconstruction. It contains implementations of user-assisted coarse registration, the iterative closest point algorithm for fine registration, and the space carving technique of Curless and Levoy [31] for surface reconstruction. Scanalyze was used to reconstruct models from the point clouds created in the Digital Michelangelo Project [61]. The Pietá project [15] used a mesh-constructing ball-pivoting algorithm [14] as part of a software system to build a model of Michelangelo's Florentine Pietá. Software systems for surface reconstruction have also been developed for creating models of urban environments using range scanners. Examples include the MIT city scanning project [81], and the work done by Früh and Zakhor [45] on fast three-dimensional city model generation.

### 1.3 Our Contributions

The main contributions of this thesis are the following:

- An analysis of an implicit surface reconstruction algorithm based on a data interpolation technique called *moving least squares*. We prove that under certain sampling conditions, the reconstructed surface is an accurate geometric and topological representation of the original surface. Our sampling requirements are adaptive and allow for noise in the input data set.
- A Delaunay-based reconstruction algorithm called *eigen crust* that uses spectral partitioning to robustly deal with noise, outliers, and regions of undersampling.
- A software system for automatically reconstructing three-dimensional models from range data that contains the following components.
  - An implementation of a registration algorithm automatically aligns range images using point signatures called *harmonic shape contexts*



- An implementation of the MLS surface reconstruction algorithm that defines a smooth surface approximating the scanned surface. This step reduces the scanner measurement noise and errors introduced during registration.
- An *eigenrust* implementation that meshes the zero set of the implicit surface defined by the MLS algorithm. The reconstructed mesh is outlier-free and watertight.

## Chapter 2

# Provably Good Moving Least Squares

In this chapter we analyze an implicit surface reconstruction algorithm based on an interpolation technique called *moving least squares* (or MLS). The input to the MLS algorithm is a set of sample points  $S$  close to the surface  $F$  of a smooth, closed, orientable three-dimensional object. For each sample point, an approximate surface normal is also given. The output is a surface passing near the sample points.

For each sample point  $s \in S$  we define a *point function*, that approximates the *signed distance function* of  $F$  in the local neighborhood of  $s$ . These point functions are then blended using Gaussian weight functions, yielding a three-dimensional function  $I$ , which we refer to as the cut function. The reconstructed surface  $U$  is given by  $\{x \in \mathbb{R}^3 : I(x) = 0\}$ . Figure 2.1 shows a two-dimensional example.

MLS methods have been used for interpolation of irregularly distributed function-value data, and for building meshless interpolants in computational mechanics. One of the earliest MLS algorithms is the metric interpolation technique of Shepard [79]. Lancaster and Salkauskas [58] give an excellent description of the geometric and differential properties of MLS interpolants. The MLS method analyzed in this chapter is not entirely new; it is a variation of the implicit MLS algorithm proposed by Chen, O'Brien, and Shewchuk [78]. We use the Gaussian function instead of the inverse distance as the weight function used to

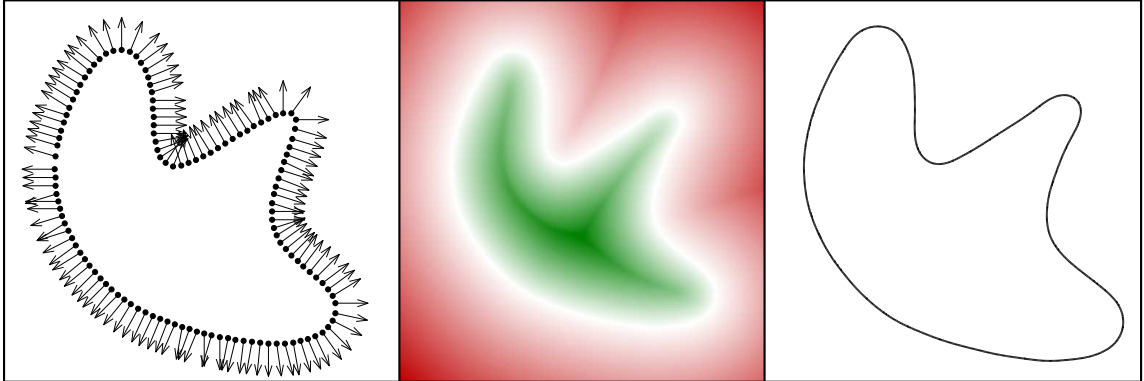


Figure 2.1: Left, a set of points with approximate outside normals. Center, the cut function built by our algorithm from the points. The zero set of the cut function, which is the reconstructed curve is shown on the right.

blend the point functions together. Our main contribution is to introduce theoretical guarantees for MLS algorithms. We prove that the cut function  $I$  is a good approximation of the signed distance function of the sampled surface  $F$ . We also show that the reconstructed surface is an accurate geometric and topological reconstruction of the sampled surface.

The Delaunay-based *crust* algorithm of Amenta and Bern [3] was the first surface reconstruction algorithm that guaranteed a correct reconstruction for sufficiently dense sample sets. The algorithm is guaranteed to work only if the input satisfies certain sampling requirements, which are defined in terms of *local feature size*. The local feature size of a point on the surface is the distance from that point to its closest point on the medial axis. Our algorithm has its own sampling requirements, described in Section 2.2, which are also based on the *local feature size* function. The MLS algorithm analyzed here requires uniform sampling in which the spacing between sample points is proportional to the smallest feature size of the sampled surface. In Chapter 3 we relax the uniformity requirements to accommodate adaptive sampling. The MLS algorithm reconstructs smooth surfaces from noisy point clouds, as the zero surface of the cut function is not constrained to interpolate the sample points. Our analysis can handle noisy data provided the noise is small compared to the local feature size of the sample points.

## 2.1 Related Work

### 2.1.1 Implicit Methods

Implicit methods reconstruct the surface as the zero set of a three-dimensional function. They are widely used in computer graphics as they are fast, easy to implement, and scale well to point clouds with millions of points. Hoppe, DeRose, Duchamp, McDonald, and Stuetzle [50] provide one of the earliest algorithms, which locally estimates the signed distance function induced by the “true” surface being sampled. Sample point normals, which are needed to estimate the signed distance function, are computed by a local least squares fit of the sample points. Curless and Levoy [31] developed an algorithm that is particularly effective for laser range data comprising billions of point samples, like the statue of David reconstructed by the Digital Michelangelo Project [61].

Smooth surfaces can also be built by fitting globally supported basis functions to a point cloud. Turk and O’Brien [83] show that a global smooth approximation can be obtained by fitting radial basis functions. This method builds a linear system from the locations of the sample points and from the normal at each sample point. The linear system is ill-conditioned, and grows linearly with the size of the input point cloud. Carr et al. [26] adapt the radial basis function-fitting algorithm to large data sets using multipole expansions. Implicit methods are popular as they are easy to implement, scale well to large data sets, and are robust against noise in the input. However, current implicit methods come with no guarantees on the correctness of the reconstructed surface.

### 2.1.2 Moving Least Squares

Moving least squares belongs to a class of meshless interpolation methods used in computational mechanics that also includes partition of unity methods, kernel methods, and smoothed particle hydrodynamics. See the survey paper by Belytschko et al. [13] for a

	Method	No normals required	Theoretical guarantees	Scalable
Hoppe et al. [50]	Implicit			✓✓
Curless, Levoy [30]	Implicit			✓✓✓
Turk, O'Brien [83]	Radial basis functions			
Carr et al. [26]	Radial basis functions			✓
Pauly et al. [69]	Projection MLS			✓✓✓
Alexa et al. [2]	Projection MLS			✓✓✓
Amenta, Kil [7]	Projection MLS			✓✓✓
Fleishman et al. [41]	Projection MLS			
Ohtake et al. [67]	Implicit MLS			✓✓✓
Shen et al. [78]	Implicit MLS			✓✓✓
Chapter 2	Implicit MLS		✓	✓✓✓
Chapter 3	Implicit MLS		✓	✓✓✓
Ball-Pivoting [14]	Mesh	✓		✓✓✓
Gopi, Krishnan [46]	Mesh			✓✓✓
Boissonnat [19]	Delaunay	✓		✓✓
$\alpha$ -shape[38]	Delaunay		✓	✓✓
Crust [4]	Delaunay	✓	✓	✓✓
Boissonnat, Cazals [20]	Delaunay Implicit	✓	✓	✓✓
Powercrust [6]	Delaunay	✓	✓	✓
Edelsbrunner [36]	Delaunay	✓		✓✓
Cocone [5]	Delaunay	✓	✓	✓✓
Tight Cocone [33]	Delaunay	✓	✓	✓✓
Eigenrust, Chapter 5	Delaunay	✓		✓✓
Robust Cocone [34]	Delaunay	✓	✓	✓✓

Table 2.1: Comparison of surface reconstruction algorithms. No normals required: Can the algorithm reconstruct with no normal information at the sample points? Theoretical guarantees: Does the algorithm have provable guarantees on the reconstructed surface? Scalable: Can the algorithm scale to large data sets? Algorithms with single checkmark can scale up to a million sample points, algorithms with two checkmarks can handle points clouds with ten million points, and algorithms with three checkmarks can scale up to a billion sample points.

	Noise practice	Noise theory	Adaptive sampling	Watertight reconstruction	Sharp corners
Hoppe et al. [50]					
Curless, Levoy [30]	✓				
Turk, O'Brien [83]	✓		✓		
Carr et al. [26]	✓		✓		
Pauly et al. [69]	✓				
Alexa et al. [2]	✓				
Amenta, Kil [7]	✓				
Fleishman et al. [41]	✓				✓
Ohtake et al. [67]	✓				
Shen et al. [78]	✓				
Chapter 2	✓	✓			
Chapter 3	✓	✓	✓		
Ball-Pivoting [14]					
Gopi, Krishnan [46]			✓		
Boissonnat [19]			✓	✓	
$\alpha$ -shape [38]					
Crust [4]			✓		
Boissonnat, Cazals [20]			✓	✓	
Powercrust [6]	✓		✓	✓	✓
Edelsbrunner [36]	✓		✓	✓	
Cocone [5]			✓		
Tight Cocone [33]	✓		✓	✓	
Eigencrust, Chapter 5	✓✓		✓	✓	
Robust Cocone [34]	✓	✓	✓	✓	

Table 2.2: Comparison of surface reconstruction algorithms. Noise practice: Can the algorithm reconstruct from noisy point clouds? Noise theory: Does the algorithm have provable guarantees for noisy point clouds? Adaptive sampling: Can the algorithm accommodate point clouds in which sampling is proportional to feature size? Watertight reconstruction: Is the reconstructed surface guaranteed to be free of holes? Sharp corners: Can the algorithm reconstruct a surface with sharp corners?

comparison between these different formulations. In computer graphics, Ohtake, Belyaev, Alexa, Turk, and Seidel [67] present a partition of unity method with a fast hierarchical evaluation scheme to compute surfaces from large point clouds. Our MLS construction is based on the implicit MLS algorithm given by Shen, O’Brien, and Shewchuk [78], which introduced the idea of associating functions, rather than just values, with each point to ensure that the gradient of the implicit function matches the gradient of the signed distance function near the sample points.

A different approach to moving least squares is the nonlinear projection method originally proposed by Levin [60]. A point set surface is defined as the set of stationary points of a projection operator. This surface definition was first used by Alexa, Behr, Cohen-Or, Fleishman, Levin, and Silva [2] for point-based modeling and rendering. Since then the surface definition has been used for progressive point-set surfaces [40] and in PointShop3D [70], a point-based modeling tool. Because Levin’s method is not guaranteed to converge, Amenta and Kil [7] give an explicit definition of point set surfaces as the set of local minima of an energy function along the directions given by a vector field. Adamson and Alexa [1] provide a simplified implicit surface definition for efficient ray tracing. They also describe sampling conditions that guarantee a manifold reconstruction. The implicit and the projection-based MLS algorithms require normals at each sample point. They scale well to large data sets and are robust against noise in the input point cloud. However, there are no provable guarantees on the reconstructed surface.

### 2.1.3 Computational Geometry

Following the Crust algorithm of Amenta and Bern [3], many Delaunay-based algorithms for surface reconstruction with provable guarantees have been proposed. Amenta, Choi, Dey, and Leekha [5] present the Cocone algorithm, which is much simpler than the Crust, and comes with a proof that the reconstructed surface is homeomorphic to the original sur-

face. The Powercrust algorithm of Amenta, Choi, and Kolluri [6] uses weighted Delaunay triangulations to avoid the manifold extraction step of the Crust and Cocone algorithms, and to recover sharp corners. Boissonnat and Cazals [20] build a smooth surface by blending together functions associated with sample points, using natural neighbor coordinates derived from the Voronoi diagram of the sample points. The Robust Cocone algorithm of Dey and Goswami [34] guarantees a correct reconstruction for noisy point data. Delaunay-based algorithms do not require sample point normals, and current Delaunay software can tetrahedralize hundreds of millions of points. However, surfaces reconstructed by Delaunay algorithms interpolate (a subset of) the sample points and a mesh smoothing step is often necessary when the input point cloud is noisy.

### **2.1.4 Signed Distance Functions**

Signed distance functions of surfaces are useful in their own right. Ray marching methods used to render implicit surfaces can use the signed distance function to quickly compute the intersection of a ray and the implicit surface. Mitra, Gelfand, Pottmann, and Guibas [64] use approximation of signed distance functions to align overlapping surfaces. The cut function constructed by our algorithm is a good approximation to the signed distance function.

### **2.1.5 Level Set Methods**

Level set methods that have been used in surface reconstruction [87], physical modeling of fluids, and in many other areas rely on signed distance functions to implicitly maintain moving surfaces. See the books by Sethian [77] and Osher and Fedkiw [68] for an introduction to level set methods. Level set methods require the signed distance function and its gradient to be very accurate near the sample points. We prove that the cut function constructed by the MLS algorithm satisfies this requirement.



### 2.1.6 Estimating Point Normals

The MLS surface definition analyzed here requires normals at sample points. In practice, the sample point normals are obtained by local least squares fitting of a plane. This least squares method was first suggested by Hoppe, DeRose, Duchamp, McDonald, and Stuetzle [50]. Mitra, Nguyen and Guibas [65] analyze the least squares method for normal estimation and present an algorithm for choosing an optimal neighborhood around each sample point. They show that under certain adaptive sampling conditions, the least squares fit will produce an accurate normal with high probability. Bremer and Hart [24] prove a similar result based on a weighted least squares fitting near each sample. Alternatively, Amenta and Bern [3] show that sample point normals can be approximated using the Voronoi diagram of the sample points. Dey, Li, and Sun [35] compare these two methods of estimating sample point normals, and show that the Voronoi method is more robust against undersampling and anisotropic sampling.

### 2.1.7 Contouring Algorithms

Sometimes it is desirable to generate a triangulation that approximates the implicit surface that is the zero set of a cut function. The marching cubes [63] algorithm is widely used in computer graphics for triangulating isocontours of implicit functions. Marching cubes performs poorly near sharp corners, and often the contoured surface has bad quality triangles. When the gradient of the cut function is available, the extended marching cubes algorithm [57] can be used to recover sharp corners. Dual contouring [55] is also effective for reconstructing meshes with sharp corners. There has been some recent work on contouring algorithms with theoretical guarantees. Boissonnat and Oudot [23] give a Delaunay-based contouring algorithm that guarantees good-quality triangles in the reconstructed surface. Boissonnat, Cohen-Steiner and Vegter [22] present a contouring algorithm with guarantees on the topology of the reconstructed triangulation.

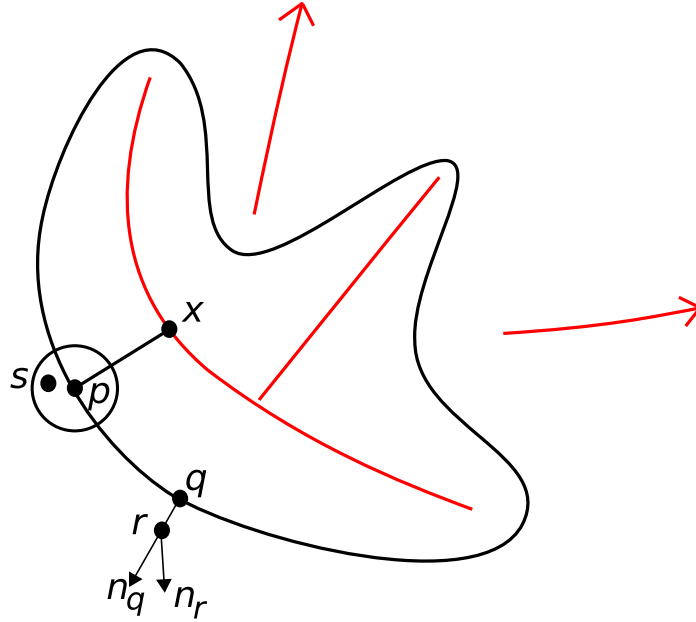


Figure 2.2: A closed curve along with its medial axis. The local feature size of  $p$  is the distance to the closest point  $x$  on the medial axis.

## 2.2 Sampling Preconditions

Our sampling preconditions are based on the *local feature size* (lfs) function proposed by Amenta and Bern [3]. For each point  $p \in F$ ,  $\text{lfs}(p)$  is defined as the distance from  $p$  to the nearest point of the medial axis of  $F$ . Our theoretical guarantees require sampling proportional to the smallest local feature size of  $F$ .

Assume that the sampled surface has been scaled such that the lfs of any point on  $F$  is at least 1. Any smooth surface can be scaled in this manner because for each point  $p$  on a smooth surface  $F$ ,  $\text{lfs}(p) > 0$ . We require that for each point  $p \in F$ , the distance from  $p$  to its closest sample point  $s$  is less than  $\epsilon$  as shown in Figure 2.2.

The amount of noise in the sample points should be small, compared to the sample point spacing. For each sample point  $r$ , the distance to its closest surface point  $q$  should be less than  $\epsilon^2$  as shown in Figure 2.2. Moreover, the angle between the normal  $\vec{n}_r$  of  $r$  and the normal  $\vec{n}_q$ , of  $q$  should be than  $\epsilon$ .

Arbitrary oversampling in one region of the surface can distort the value of the cut function in other parts of the surface. Hence we require that local changes in the sampling density are bounded. Let  $\alpha_p$  be the number of sample points inside a ball of radius  $\epsilon$  centered at a point  $p$ . If  $\alpha_p > 0$ , the number of sample points inside a ball at radius  $2\epsilon$  at  $p$  should be less than  $8\alpha$ . The results in this chapter hold true for values of  $\epsilon \leq 1/50$ .

### 2.3 Reconstructed Surface Definition

Let  $s_i \in S$  be the sample points in the input point cloud that lie near the surface  $F$  and let  $\vec{n}_i$  be the approximate outside normal of sample point  $s_i \in S$ . For each sample point  $s_i$  we define a point function  $P_i(x)$  whose value is the signed distance from  $x$  to the tangent plane at  $s_i$ ,  $P_i(x) = (x - s_i) \cdot \vec{n}_i$ . The cut function  $I$  is a weighted average of the point functions.

$$I(x) = \frac{1}{\sum_{s_j \in S} W_j(x)} \sum_{s_i \in S} W_i(x) ((x - s_i) \cdot \vec{n}_i). \quad (2.1)$$

For each sample point  $s_i$  define a oversampling factor  $a_i$  given by the number of sample points inside a ball of radius  $\epsilon$  around  $s_i$ . The oversampling factor  $a_i$  accounts for oversampling in the neighborhood of sample point  $s_i$ . We use Gaussian weight functions along with the oversampling factor,

$$W_i(x) = e^{-\|x-s_i\|^2/\epsilon^2}/a_i,$$

in computing the weighted average of the point functions. Here  $\epsilon$  is the parameter defined in our sampling requirements.

The cut function is the best least squares fit to the point functions in the following sense: at point  $x$ , each sample point  $s_i$  votes for the value of the cut function to be  $P_i(x)$  with a weight  $W_i(x)$ . Setting the cut function  $I(x)$  to the weighted average of these point

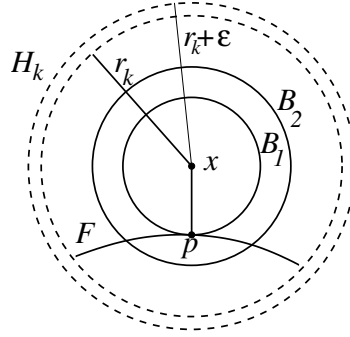


Figure 2.3: For a point  $x$ ,  $p$  is the closest point to  $x$  on the surface. The space outside the ball  $B_2(x)$  is divided into spherical shells of width  $\epsilon$ .  $H_k$  is the shell bounded by spheres of radius  $r_k$  and  $r_k + \epsilon$ .

functions minimizes the weighted least squares error in the estimate of  $I(x)$ .

$$I(x) = \min_{\beta} \sum_{s_i \in S} W_i(x) (P_i(x) - \beta)^2.$$

## 2.4 Geometry of the Reconstructed Surface

Consider a point  $x$  shown in Figure 2.3, whose closest point on the surface is  $p$ . Let  $\phi(x)$  be the signed distance function of  $F$ . Inside  $F$ ,  $\phi(x) < 0$ , and outside  $F$ ,  $\phi(x) > 0$ . The vector  $\vec{x}\vec{p}$  is parallel to the surface normal of  $p$  and  $\|\vec{x}\vec{p}\| = |\phi(x)|$ . Let  $B_1(x), B_2(x)$  be two balls centered at point  $x$  as shown in Figure 2.3. The radius of  $B_1(x)$  is  $|\phi(x)|$  and the radius of  $B_2(x)$  is  $|\phi(x)| + 4\epsilon$ .

We define an error function  $E(x)$  that measures the difference between the cut function computed by the MLS algorithm and the signed distance function  $\phi(x)$ . This error function

$E(x)$  is given by,

$$\begin{aligned}
 E(x) &= I(x) - \phi(x) \\
 &= \frac{1}{\sum_{s_j \in S} W_j(x)} \sum_{s_i \in S} W_i(x) (P_i(x) - \phi(x)) \\
 &\equiv \frac{1}{\sum_{s_j \in S} W_j(x)} \sum_{s_i \in S} W_i(x) \zeta_i(x). \tag{2.2}
 \end{aligned}$$

The function  $\zeta_i(x)$  defined in Equation 2.2 measures the error in the point function of sample point  $s_i$ .

Our analysis of the cut function has two main ideas. First, the exponential decay of the Gaussian weight functions means that the cut function is mostly determined by sample points inside  $B_2(x)$ . Second, the cut function converges to the signed distance function with  $\epsilon$  because the error functions associated with sample points inside  $B_2(x)$  converge to zero as  $\epsilon$  goes to zero.

To prove our results, it is convenient to separate the contributions of sample points inside  $B_2(x)$  and outside  $B_2(x)$  to the error function  $E(x)$ . Let  $\xi(x) = \sum_{s_i \in S} \zeta_i(x) W_i(x)$  be the weighted combination of sample point error functions, and let  $W(x) = \sum_{s_i \in S} W_i(x)$  be the sum of all weight functions at  $x$ . Let  $\xi_{\text{in}}(x)$  and  $\xi_{\text{out}}(x)$  be the contributions to  $\xi(x)$  by sample points inside and outside  $B_2(x)$ . Similarly, let  $W_{\text{in}}(x)$  be the sum of weights of all sample points inside  $B_2(x)$ , and let  $W_{\text{out}}(x)$  be the weight of all sample points outside  $B_2(x)$ .

Let  $F_{\text{out}}$  be the outside  $\epsilon$ -offset surface of  $F$  that is obtained by moving each point  $p$  on  $F$  along the normal at  $p$  by a distance  $\epsilon$ . Similarly, let  $F_{\text{in}}$  be the inside  $\epsilon$ -offset surface of  $F$  as shown in Figure 2.4. Dey and Goswami [34] (Lemma 3) prove that the offset surfaces have the same topology as  $F$  when  $\epsilon$  is small relative to the feature size of each point on the surface. The  $\epsilon$ -neighborhood is the region bounded by the inside and the outside offset surfaces. For any point  $x$  inside the  $\epsilon$ -neighborhood,  $|\phi(x)| < \epsilon$ . We study the properties of

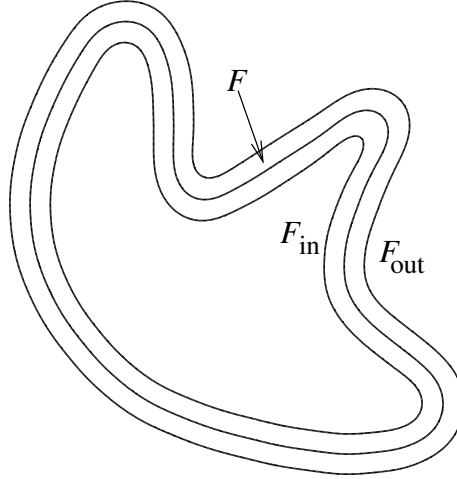


Figure 2.4: The inside and outside offset surfaces of a two-dimensional curve.

the cut function for points outside the  $\epsilon$ -neighborhood, in Section 2.4.1 and prove that the cut function is non-zero outside the  $\epsilon$ -neighborhood of  $F$  (Theorem 2.7). This means that the reconstructed surface is inside the  $\epsilon$ -neighborhood of  $F$ . To prove that the reconstructed surface is a manifold, we show that the cut function has a non-zero gradient at points inside the  $\epsilon$ -neighborhood. These geometric results will be used in Section 2.5 to show that  $U$  is isotopic to  $F$ .

We begin by proving some useful geometric properties of the surface and surface normals which we use later in the analysis of the cut function. Amenta and Bern [3] prove the following Lipschitz condition on the surface normal with respect to the function lfs. As we assume that for each point  $p \in F$ ,  $\text{lfs}(p) \geq 1$ , we can state the Lipschitz condition in terms of the distance between two points.

**Theorem 2.1.** *For points  $p, q$  on the surface  $F$  with  $d(p, q) \leq r$ , for any  $r < 1/3$ , the angle between the normals at  $p$  and  $q$  is at most  $r/(1 - 3r)$  radians.*

Consider the surface inside a small ball  $B$  centered at a point  $p \in F$  as shown in Figure 2.5. The following lemma proves that the surface inside  $B$  has to be close to the tangent plane at  $p$ .

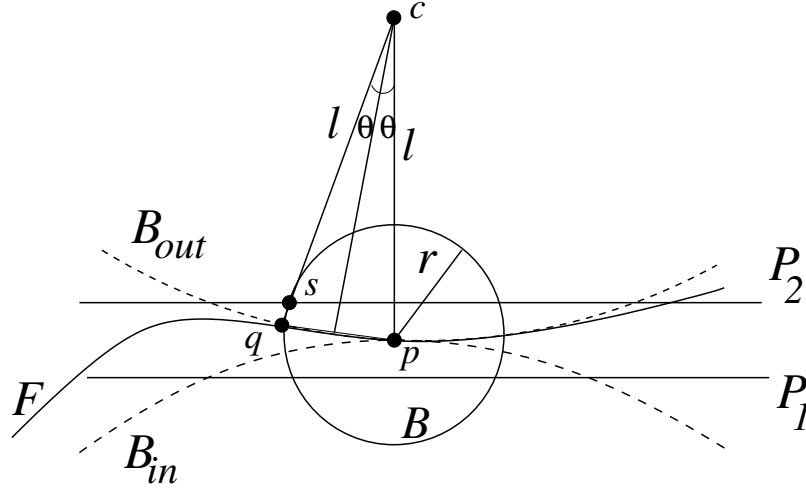


Figure 2.5: The surface inside a ball  $B$  of radius  $r$  has to be outside the medial balls  $B_{in}$  and  $B_{out}$ . As a result, all sample points in  $B$  are between two planes  $P_1$  and  $P_2$  that are at a distance of  $O(r^2 + \epsilon^2)$  from  $p$ .

**Lemma 2.2.** *For a point  $p \in F$ , let  $B$  be a ball of radius  $r < 1/4$  centered at  $p$ . The sample points inside  $B$  lie between two planes  $P_1, P_2$  parallel to the tangent plane at  $p$ . The distance from  $p$  to  $P_1, P_2$  is less than  $(r + \epsilon^2)^2/2 + \epsilon^2$ .*

**Proof.** Consider sample point  $s \in B$ , and let  $p$  be the point closest to  $s$  on  $F$ . By the sampling preconditions  $d(s, q) \leq \epsilon^2$ . Without loss of generality assume that  $s$  is above the tangent plane at  $p$  as shown in Figure 2.5. The distance from  $p$  to  $q$  can be written as

$$d(p, q) \leq d(p, s) + d(s, q) \leq r + \epsilon^2. \quad (2.3)$$

As point  $q$  is on the surface, it has to be outside the medial ball  $B_{out}$  of radius  $l \geq 1$ . Hence the distance from  $q$  to the tangent plane at  $p$  is less than  $l(1 - \cos 2\theta) = 2l \sin^2 \theta \leq d^2(p, q)/2$ . Let  $P_2$  be the plane passing through  $s$  and parallel to the tangent plane at  $p$ . As  $d(s, q) \leq \epsilon^2$ , the distance from  $p$  to the plane  $P_2$  is

$$d(p, P_2) \leq d^2(p, q)/2 + \epsilon^2 = (r + \epsilon^2)^2/2 + \epsilon^2,$$

by Equation 2.3. □

Partition the space outside  $B_2(x)$  into spherical shells of width  $\epsilon$  as shown in Figure 2.3. Let  $H_k$  be the region between balls of radius  $r_k$  and  $r_k + \epsilon$ . We sum over the spherical shells starting at  $B_2(x)$  to prove an upper bound on the error in the cut function due to sample points outside  $B_2(x)$ . In the following two lemmas we prove an upper bound on the weight of sample points inside these spherical shells. Recall that for each sample point  $s_i$ , the oversampling factor  $a_i$  is given by the number of sample points in a ball of radius  $\epsilon$  around  $s_i$ .

**Lemma 2.3.** *For a ball  $B$  of radius  $\frac{\epsilon}{2}$ ,  $\sum_{s_i \in B} \frac{1}{a_i} \leq 1$ .*

**Proof.** Without any loss of generality assume that  $B$  contains  $\alpha > 0$  sample points. Let  $s_i$  be a sample point inside  $B$ . As all sample points inside  $B$  are inside a ball of radius  $\epsilon$  centered at  $s_i$ ,  $a_i \geq \alpha$ . The contribution of all sample points inside  $B$  is given by,  $\sum_{s_i \in B} \frac{1}{a_i} \leq \alpha \frac{1}{\alpha} \leq 1$ . □

**Lemma 2.4.** *Let  $H_k$  be a spherical shell of width  $\epsilon$ , centered at point  $x$ . Let the radius of the smaller sphere surrounding  $H_k$  be  $r_k > 4\epsilon$ . Then,  $\sum_{s_i \in H_k} \frac{1}{a_i} < 300 \frac{r_k^2}{\epsilon^2}$ .*

**Proof.** Let  $C$  be the smallest number of spheres of radius  $\epsilon/2$  that cover  $H_k$ . Consider a covering of  $H_k$  with axis-parallel cubes of size  $\epsilon/\sqrt{3}$ . Any cube that intersects  $H_k$  is inside a slightly larger shell bounded by spheres of radius  $r_k + 2\epsilon$  and  $r_k - \epsilon$  centered at  $x$ . So the number of cubes that cover  $H_k$  is less than  $36\sqrt{3}\pi\epsilon(r_k^2 + r_k\epsilon + \epsilon^2)/\epsilon^3$ . Any cube in this grid is covered by a sphere of radius  $\epsilon/2$ . Applying Lemma 2.3 to each sphere,  $\sum_{s_i \in H_k} \frac{1}{a_i} \leq C \leq 36\sqrt{3}\pi(r_k^2 + r_k\epsilon + \epsilon^2)/\epsilon^2$ . As  $r_k > 4\epsilon$  we can simplify the upper bound on  $C$  to

$$\sum_{s_i \in H_k} \frac{1}{a_i} < 36\sqrt{3}\pi(r_k^2 + r_k(\frac{r_k}{4}) + (\frac{r_k}{4})^2)/\epsilon^2 < 300 \frac{r_k^2}{\epsilon^2}. \quad \square$$



### 2.4.1 Outside the $\epsilon$ -neighborhood

In this section we analyze the error function  $E(x)$  for  $x$  outside the  $\epsilon$ -neighborhood. We prove that the cut function is mostly determined by sample points inside  $B_2(x)$ , and that the cut function is non-zero outside the  $\epsilon$ -neighborhood.

The following Lemma shows that for any sample point  $s_i \in B_2(x)$ ,  $P_i(x)$  is close to  $\phi(x)$ . In order to state this result for points in the inside and outside offset regions, it is convenient to define  $\mu(x) = \frac{\phi(x)}{|\phi(x)|}$  to be the sign function of  $F$ . Since  $x$  is outside the  $\epsilon$ -neighborhood,  $|\phi(x)| > 0$ . For  $x$  outside  $F_{\text{out}}$ ,  $\mu(x) = 1$  and for  $x$  inside  $F_{\text{in}}$ ,  $\mu(x) = -1$ .

**Lemma 2.5.** *Let  $x$  be a point outside the  $\epsilon$ -neighborhood. Let  $\zeta_i(x)$  be the error in the point function of sample point  $s_i \in B_2(x)$  evaluated at  $x$ .*

- $\mu(x)\zeta_i(x) \leq 4\epsilon$ .
- $\mu(x)\zeta_i(x) \geq -(6\epsilon\mu(x)\phi(x) + 13\epsilon^2)$ .

**Proof.** Let  $p$  be a closest point to  $x$  on the surface. By the definition of the signed distance function,  $d(x, p) = \mu(x)\phi(x)$ . As  $s_i \in B_2(x)$

$$\begin{aligned} \mu(x)P_i(x) &\leq d(x, s_i) \\ &\leq \mu(x)\phi(x) + 4\epsilon. \\ \mu(x)\zeta_i(x) &\leq 4\epsilon. \end{aligned}$$

To prove the lower bound on  $\mu(x)\zeta_i(x)$  consider the point  $p'$  closest to  $s_i$  on the surface  $F$  as shown in Figure 2.6. Let  $B_m$  be the medial ball touching  $p'$  on the side of  $F$  opposite  $x$  and let  $l$  be the radius of  $B_m$ . Let  $\theta$  be the angle between  $xs_i$  and the normal at  $p'$ . We prove a lower bound on  $d(x, s_i) \cos \theta$  and use it in proving the desired lower bound on  $\mu(x)\zeta_i(x)$ .

## Chapter 2. Provably Good Moving Least Squares

---

The distance between  $x$  and the center  $q$  of  $B_m$  is

$$\begin{aligned} d^2(x, q) &= (d(x, s_i) \cos \theta + l + d(s_i, q))^2 + d^2(x, p') \sin^2 \theta \\ &\leq (d(x, s_i) \cos \theta + l + \epsilon^2)^2 + d^2(x, p') \sin^2 \theta, \end{aligned} \quad (2.4)$$

by the sampling preconditions.

The medial ball  $B_m$  cannot intersect  $B_1(x)$  which is contained in a medial ball on the opposite side of the surface. Therefore the sum of their radii should be less than the distance between their centers. By Equation 2.4 we have

$$(l + d(x, p))^2 \leq (d(x, s_i) \cos \theta + l + \epsilon^2)^2 + d^2(x, s_i) \sin^2 \theta. \quad (2.5)$$

$$d(x, s_i) \cos \theta \geq \frac{1}{2(l + \epsilon^2)} (2ld(x, p) - (2l + \epsilon^2)\epsilon^2 - 16\epsilon^2 - d^2(x, s) + d^2(x, p)). \quad (2.6)$$

Since  $s_i$  is inside  $B_2(x)$ ,  $d(x, s_i) \leq d(x, p) + 4\epsilon$ . From the local feature size assumption,  $l \geq 1$ . Substituting into Equation 2.6,

$$d(x, s_i) \cos \theta \geq d(x, p)(1 - 5\epsilon) - 9\epsilon^2. \quad (2.7)$$

The angle between the normal at  $s_i$  and the surface normal at  $q$  is at most  $\epsilon$  by the sampling preconditions. Therefore,  $\mu(x)P_i(x) \geq d(x, s_i) \cos(\theta + \epsilon)$ . Using standard trigonometric formulas, it is easy to show that,  $\cos(\theta + \epsilon) \geq \cos \theta - \epsilon$ . By Equation 2.7 we have

$$\begin{aligned} \mu(x)P_i(x) &\geq d(x, s_i) \cos \theta - \epsilon d(x, s_i) \\ &\geq d(x, p)(1 - 5\epsilon) - 9\epsilon^2 - \epsilon(d(x, p) + 4\epsilon) \\ &\geq d(x, p)(1 - 6\epsilon) - 13\epsilon^2. \end{aligned} \quad (2.8)$$

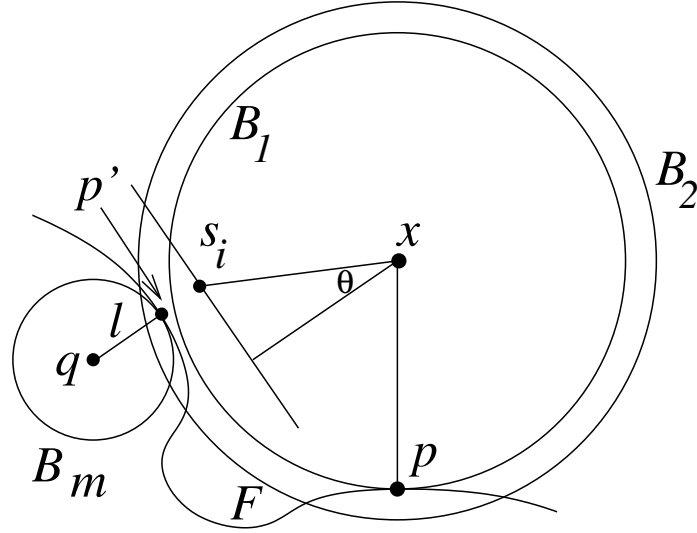


Figure 2.6: Sample Point  $s_i$  is inside  $B_2(x)$  and  $p'$  is the point closest to  $s$  on  $F$ .  $B_m$  is a medial ball touching  $p'$  on the side of  $F$  opposite  $x$ .

Since  $p$  is the point closest to  $x$  on  $F$ ,  $\mu(x)\phi(x) = d(x, p)$ . Substituting into Equation 2.8

$$\mu(x)\zeta_i(x) = \mu(x)P_i(x) - \mu(x)\phi(x) \geq -(6\epsilon\mu(x)\phi(x) + 13\epsilon^2). \quad \square$$

In the following lemma we prove two results showing that the points outside  $B_2(x)$  have little effect on the value of  $I(x)$ . We use a constant  $c_1 = 0.001$  to state these results.

**Lemma 2.6.** *Let  $x$  be a point outside the  $\epsilon$ -neighborhood. Let  $W_{\text{out}}(x)$  be the total weight of sample points outside  $B_2(x)$ . Then  $\frac{W_{\text{out}}(x)}{W(x)} < c_1$ , and  $|E_{\text{out}}(x)| < c_1\epsilon$ .*

**Proof.** Consider the division of space outside  $B_2(x)$  into spherical shells of width  $\epsilon$  starting with  $B_2(x)$  as shown in Figure 2.3. Let the radius of  $B_2(x)$  be  $r_0 = |\phi(x)| + 4\epsilon$ . The value of the Gaussian weight function of each sample point inside shell  $H_k$  at  $x$  is at most  $e^{-r_k^2/\epsilon^2}$ .

By Lemma 2.4 we have

$$W_{\text{out}}(x) = \sum_{k=0}^{\infty} \frac{W_i(x)}{a_i} \leq \frac{300}{\epsilon^2} \sum_{k=0}^{\infty} r_k^2 e^{-r_k^2/\epsilon^2} \leq \frac{300}{\epsilon^2} \sum_{k=0}^{\infty} r_k^2 e^{-(r_0 r_k)/\epsilon^2}. \quad (2.9)$$

Here  $r_k = r_0 + k\epsilon$  is the radius of the smaller sphere bounding  $H_k$ . The summation in Equation 2.9 is a geometric series with a common ratio  $e^{-r_0/\epsilon} < 0.01$  that has a closed form solution. An upper bound is given by,

$$W_{\text{out}}(x) \leq 450 \frac{r_0^2}{\epsilon^2} e^{-r_0^2/\epsilon^2}. \quad (2.10)$$

Let  $B_\epsilon$  be a ball of radius  $\epsilon$  centered at  $p$ . The weight of samples inside  $B_\epsilon$  is a lower bound for  $W(x)$ . From the sampling preconditions,  $B_\epsilon$  contains  $\alpha \geq 1$  sample points, and a ball of radius  $2\epsilon$  centered at  $p$  contains at most  $8\alpha$  sample points. Recall that the oversampling factor  $a_i$  associated with  $s_i$  is the number of sample points inside a ball of radius  $\epsilon$  around  $s_i$ . Therefore for  $s_i \in B_\epsilon$ ,  $\frac{1}{a_i} \geq \frac{1}{8\alpha}$ .

$$W(x) \geq \sum_{s_i \in B_\epsilon} \frac{1}{a_i} e^{-\left(|\phi(x)| + \epsilon\right)^2/\epsilon^2} \geq \frac{\alpha}{8\alpha} e^{-\left(|\phi(x)| + \epsilon\right)^2/\epsilon^2} = \frac{1}{8} e^{-\left(|\phi(x)| + \epsilon\right)^2/\epsilon^2}. \quad (2.11)$$

Combining results in Equation 2.10 and Equation 2.11 we have

$$\frac{W_{\text{out}}(x)}{W(x)} \leq 3600 \frac{r_0^2}{\epsilon^2} e^{-(r_0^2 - \left(|\phi(x)| + \epsilon\right)^2)/\epsilon^2} = 3600 \frac{\left(|\phi(x)| + 4\epsilon\right)^2}{\epsilon^2} e^{-3\epsilon(2|\phi(x)| + 5\epsilon)/\epsilon^2}. \quad (2.12)$$

## Chapter 2. Provably Good Moving Least Squares

---

For  $|\phi(x)| \geq \epsilon$ , the expression in Equation 2.12 is a monotonically decreasing function of  $|\phi(x)|$  whose value is maximum when  $|\phi(x)| = \epsilon$ .

$$\frac{W_{\text{out}}(x)}{W(x)} \leq 3600 \frac{25\epsilon^2}{\epsilon^2} e^{-21} < c_1.$$

To prove a similar bound on  $|E_{\text{out}}(x)|$  consider sample point  $s_i \in H_k$ . By definition,  $|P_i(x)| = (x - s_i) \cdot \vec{n}_i \leq \|x - s_i\|$ , and  $|\phi(x)| < \|x - s_i\|$ . Therefore,

$$|\zeta_i(x)| \leq |\phi(x)| + \|x - s_i\| < 2\|x - s_i\|.$$

Recall that  $\xi_{\text{out}}(x)$  is the sum of the error functions associated with sample points outside  $B_2(x)$ .

$$|\xi_{\text{out}}(x)| \leq \sum_{s_i \notin B_2(x)} |\zeta_i(x)| W_i(x) < 2 \sum_{s_i \notin B_2(x)} \|x - s_i\| W_i(x). \quad (2.13)$$

For  $\|x - s_i\| \geq r_0 \geq 5\epsilon$ , the value of  $\|x - s_i\| e^{-\|x - s_i\|^2/\epsilon^2}$  decreases as  $\|x - s_i\|$  increases. Hence for each  $s_i \in H_k$ ,  $|\zeta_i(x)| W_i(x) \leq 2r_k e^{-r_k^2/\epsilon^2} / a_i$ .

$$|\xi_{\text{out}}(x)| < \frac{600}{\epsilon^2} \sum_{k=0}^{\infty} r_k^3 e^{-r_k^2/\epsilon^2}.$$

Just like Equation 2.9, an upper bound of the summation in the above equation is given by,

$$|\xi_{\text{out}}(x)| < \frac{900}{\epsilon^2} r_0^3 e^{-r_0^2/\epsilon^2}. \quad (2.14)$$

Substituting the lower bound on  $W(x)$  from Equation 2.11 into Equation 2.14 we have

$$\frac{|\xi_{\text{out}}(x)|}{W(x)} < \frac{7200}{\epsilon^2} r_0^3 e^{-(r_0^2 - (|\phi(x)| + \epsilon)^2)/\epsilon^2} = \frac{7200}{\epsilon^2} (|\phi(x)| + 4\epsilon)^3 e^{-3\epsilon(2|\phi(x)| + 5\epsilon)/\epsilon^2}. \quad (2.15)$$

## Chapter 2. Provably Good Moving Least Squares

---

The value of  $E_{\text{out}}(x)$  is maximized when  $|\phi(x)| = 5\epsilon$ . By Equation 2.15 we have

$$|E_{\text{out}}(x)| = \frac{|\xi_{\text{out}}(x)|}{W(x)} < \frac{7200}{\epsilon^2} (125\epsilon^3) e^{-21} < c_1\epsilon. \quad \square$$

In the following theorem we prove that the cut function  $I(x)$  is non-zero outside the  $\epsilon$ -neighborhood.

**Theorem 2.7.** *For each point  $x$  outside  $F_{\text{out}}$ ,  $I(x) > 0$  and for each point  $y$  inside  $F_{\text{in}}$ ,  $I(y) < 0$ .*

**Proof.** Consider point  $x$  outside  $F_{\text{out}}$ . The cut function at  $x$

$$I(x) = \phi(x) + E_{\text{in}}(x) + E_{\text{out}}(x) \geq \phi(x) + E_{\text{in}}(x) - c_1\epsilon, \quad (2.16)$$

by Lemma 2.6. We have a lower bound on  $E_{\text{in}}(x)$  from Lemma 2.5.

$$\begin{aligned} E_{\text{in}}(x) &= \frac{1}{W(x)} \sum_{s_i \in B_2(x)} \zeta_i(x) W_i(x) \\ &\geq \min\{\zeta_i(x) | s_i \in B_2(x)\} \frac{W_{\text{in}}(x)}{W(x)} \\ &> -(6\epsilon\phi(x) + 13\epsilon^2). \end{aligned} \quad (2.17)$$

Substituting the result in Equation 2.17 into Equation 2.16 we have

$$I(x) > \phi(x)(1 - 6\epsilon) - 13\epsilon^2 - c_1\epsilon.$$

As  $x$  is outside  $F_{\text{out}}$ ,  $\phi(x) \geq \epsilon$ . It is easy to check that  $I(x) > 0$  when  $\epsilon \leq 1/50$ . A similar argument proves that the cut function is negative at any point  $y$  inside  $F_{\text{in}}$ .  $\square$

Theorem 2.7 proves that the cut function  $I$  does not have any spurious zero crossings far away from the sample points, and gives an upper bound of  $\epsilon$  on the Hausdorff distance

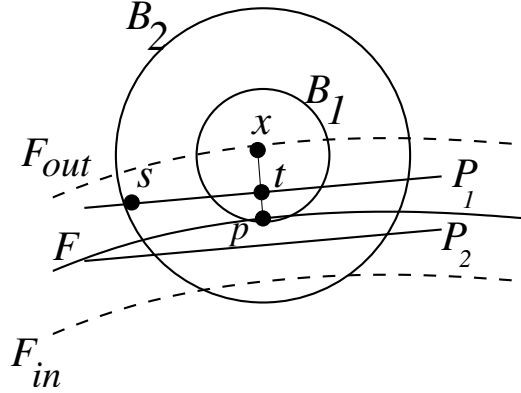


Figure 2.7: For  $x$  in the  $\epsilon$ -neighborhood, the sample points inside balls  $B_2(x)$  are contained in a ball of radius  $4\epsilon$  centered at  $p$ .

between  $F$  and  $U$ . In Section 2.4.2, we derive tighter bounds which show that the Hausdorff distance between  $U$  and  $F$  is  $O(\epsilon^2)$ .

Combining the results in Lemma 2.5 and Lemma 2.6, it is easy to show that the cut function converges to the signed distance function as  $\epsilon$  goes to zero.

$$\mu(x)I(x) \geq \mu(x)\phi(x)(1 - 6\epsilon) - 13\epsilon^2 - c_1\epsilon.$$

$$\mu(x)I(x) \leq \mu(x)\phi(x) + 4\epsilon + c_1\epsilon.$$

### 2.4.2 The $\epsilon$ -neighborhood

In this section we analyze the cut function inside the  $\epsilon$ -neighborhood which contains the reconstructed surface. Our prove that the reconstructed surface  $U$  is a smooth manifold that converges to the sampled surface as  $\epsilon$  goes to zero. The reconstructed surface  $U$  is smooth because the point functions and the weight functions used in computing the cut function are all smooth. To prove that  $U$  is a manifold. we show (Theorem 2.17) that the cut function has a non-zero gradient inside the  $\epsilon$ -neighborhood.

Inside the  $\epsilon$ -neighborhood we define  $B_2(x)$  as a ball centered at  $x$  whose radius  $r_0$  is

given by

$$r_0^2 = (|\phi(x)| + \epsilon)^2 + 12\epsilon^2. \quad (2.18)$$

In the following lemma we prove that sample points inside  $B_2(x)$  are contained in a small ball centered at the point  $p$  closest to  $x$  on  $F$ .

**Lemma 2.8.** *Let  $x$  be a point that is inside the  $\epsilon$ -neighborhood, and let  $p$  be the closest point to  $x$  in  $F$ . All sample points inside  $B_2(x)$  are contained in a ball of radius  $4\epsilon$  centered at  $p$ .*

**Proof.** Let  $t$  be the point at which  $xp$  intersects the plane  $P_1$  which is parallel to the tangent plane at  $p$  as shown in Figure 2.7. The distance between  $s$  and  $t$  can be written as

$$d^2(s, t) = d^2(s, p) - d^2(p, t) = d^2(x, s) - d^2(x, t). \quad (2.19)$$

As  $s$  is inside  $B_2(x)$   $d^2(x, s) \leq (|\phi(x)| + \epsilon)^2 + 12\epsilon^2$ . Substituting into Equation 2.19

$$\begin{aligned} d^2(s, p) - d^2(p, t) &\leq (|\phi(x)| + \epsilon)^2 + 12\epsilon^2 - (|\phi(x)| - d(p, t))^2 \\ d^2(s, p) &\leq 13\epsilon^2 + 2|\phi(x)|\epsilon + 2|\phi(x)|d(p, t) \\ &< 15\epsilon^2 + 2\epsilon d(p, t). \end{aligned} \quad (2.20)$$

Since  $x$  is inside the  $\epsilon$ -neighborhood, any sample point inside  $B_2(x)$  is clearly inside a ball of radius  $5\epsilon$  centered at  $p$ . By Lemma 2.2  $s$  is between two planes at a distance  $(5\epsilon + \epsilon^2)/2 + \epsilon^2$  from  $p$ . Hence  $d(p, t) \leq (5\epsilon + \epsilon^2)/2 + \epsilon^2 < 14\epsilon^2$ . Substituting into Equation 2.20  $d^2(s, p) < 15\epsilon^2 + 28\epsilon^3 < 16\epsilon^2$ .  $\square$

In the following lemma we show that for a point  $x$  whose closest point on the surface is  $p$ , the point functions of sample points near  $p$  have small error when evaluated at  $x$ .



**Lemma 2.9.** *Consider a point  $x$  whose closest point on the surface  $F$  is  $p$ . Let  $\vec{n}$  be the surface normal at  $p$  and let  $B$  be a ball of radius  $r \leq 1/4$  at  $p$ . For each sample point  $s_i \in B$ , the angle between the normal at  $s_i$  and  $\vec{n}$  is less than  $\Theta(r) = \frac{r+\epsilon^2}{1-3(r+\epsilon^2)} + \epsilon$ . For each sample point  $s_i \in B$ ,  $|\zeta_i(x)| \leq |\phi(x)|\Theta(r)^2/2 + r\Theta(r) + (r + \epsilon^2)^2/2 + \epsilon^2$ .*

**Proof.** Let  $p_i$  be the point closest to  $s_i$  on  $F$ . Then,  $d(p, p_i) \leq d(p, s_i) + d(s_i, p_i) \leq r + \epsilon^2$ . The angle between the normal at  $p_i$  and the surface normal  $\vec{n}$  at  $p$  is less than  $\frac{r+\epsilon^2}{1-3(r+\epsilon^2)}$  by Theorem 2.1.

Let  $\vec{n}_i$  be the normal associated with  $s_i$ . From the sampling preconditions the angle between the normal of  $p_i$  and  $\vec{n}_i$  is at most  $\epsilon$ . So the angle between  $\vec{n}_i$  and  $\vec{n}$  is given by  $\Theta(r) < \frac{r+\epsilon^2}{1-3(r+\epsilon^2)} + \epsilon$ . We can now write  $\vec{n}_i = \vec{n} + \vec{\rho}_i$ , where  $\|\vec{\rho}_i\| \leq \Theta(r)$ .

$$\begin{aligned} \zeta_i(x) &= \phi(x) - P_i(x) \\ &= \phi(x) - (x - s_i) \cdot \vec{n}_i \\ &= \phi(x) - (x - p) \cdot \vec{n}_i - (p - s_i) \cdot (\vec{n} + \vec{\rho}_i). \end{aligned} \tag{2.21}$$

Because  $p$  is the closest point to  $x$  on the surface,  $(x - p)$  is parallel to  $\vec{n}$  and  $\|x - p\| = |\phi(x)|$ .

$$|\phi(x) - (x - p) \cdot \vec{n}_i| \leq |\phi(x)|(1 - \cos \Theta(r)) \leq |\phi(x)|\Theta^2(r)/2.$$

Since sample point  $s_i$  is inside  $B$ ,

$$|(p - s_i) \cdot \vec{\rho}_i| \leq r\Theta(r).$$

From Lemma 2.2, the distance from each sample point inside  $B$  to the tangent plane at  $p$  is at most

$$|(p - s_i) \cdot \vec{n}| \leq (r + \epsilon^2)^2/2 + \epsilon^2.$$

Substituting the upper bounds on the individual terms into Equation 2.21

$$\begin{aligned} |\zeta_i(x)| &\leq |\phi(x) - (x - p) \cdot \vec{n}_i| + |(p - s_i) \cdot \vec{\rho}_i| + |(p - s_i) \cdot \vec{n}| \\ &\leq |\phi(x)|\Theta^2(r)/2 + r\Theta(r) + (r + \epsilon^2)^2/2 + \epsilon^2. \end{aligned}$$

□

All sample points inside  $B_2(x)$  are contained in a small ball around the point closest to  $x$  on  $F$  by Lemma 2.8. So the result in Lemma 2.9 gives us an upper bound on the error functions of all sample points inside  $B_2(x)$ .

**Lemma 2.10.** *For a point  $x$  inside the  $\epsilon$ -neighborhood,  $|E_{\text{in}}| \leq 30\epsilon^2$ .*

**Proof.** Consider the error in the cut function due to samples inside  $B_2(x)$ .

$$|E_{\text{in}}(x)| = \frac{|\xi_{\text{in}}(x)|}{W(x)} \leq \sum_{s_i \in B_2(x)} \frac{|\zeta_i(x)|W_i(x)}{W(x)} \leq \max\{|\zeta_i(x)| \mid s_i \in B_2(x)\}.$$

From Lemma 2.8, we know that each sample point  $s_i \in B_2(x)$  is inside a ball of radius  $4\epsilon$  around  $p$  the point closest to  $x$  on  $F$ . By Lemma 2.9 we have

$$|\zeta_i(x)| \leq |\phi(x)|\Theta^2(4\epsilon)/2 + 4\epsilon\Theta(4\epsilon) + (4\epsilon + \epsilon^2)/2 + \epsilon^2 < 30\epsilon^2. \quad (2.22)$$

□

In the following lemma we show that sample points outside  $B_2(x)$  have little effect on the value of the cut function at  $x$ . The proof is similar to the proof of Lemma 2.6 and is given in the appendix.

**Lemma 2.11.** *For a point  $x$  inside the  $\epsilon$ -neighborhood,  $|E_{\text{out}}(x)| < 4\epsilon^2$ .*

**Proof.** Given in the appendix.

□

## Chapter 2. Provably Good Moving Least Squares

---

We will now prove a bound on the error in the cut-function inside the  $\epsilon$ -neighborhood which shows that the Hausdorff distance between  $F$  and the reconstructed surface  $U$  is  $O(\epsilon^2)$ .

**Theorem 2.12.** *For a point  $x$  inside the  $\epsilon$ -neighborhood,  $|E(x)| < 34\epsilon^2$ .*

**Proof.** Adding the bounds on  $|E_{\text{in}}(x)|$  and  $|E_{\text{out}}(x)|$  in Lemma 2.10 and Lemma 2.11 gives the desired result.

$$|E(x)| \leq |E_{\text{in}}(x)| + |E_{\text{out}}(x)| < 30\epsilon^2 + 4\epsilon^2 = 34\epsilon^2. \quad \square$$

**Theorem 2.13.** *For a point  $x \in U$ , let  $p$  be the closest point in  $F$ . Then  $d(x, p) < 34\epsilon^2$ .*

**Proof.** Since  $x \in U$ ,  $I(x) = 0$  and the result in Theorem 2.7 tells us that  $x$  is inside the  $\epsilon$ -neighborhood. Hence from Theorem 2.12,  $|E(x)| < 34\epsilon^2$ . So,  $d(x, p) = |\phi(x)| \leq |I(x)| + |E(x)| < 34\epsilon^2$ .  $\square$

**Theorem 2.14.** *For a point  $p \in F$ , let  $q$  be the closest point in  $U$ . Then,  $d(p, q) \leq 34\epsilon^2$ .*

**Proof.** If  $I(p) = 0$  we are done; assume without loss of generality that  $I(p) < 0$ . Let  $t$  be the point on the outside normal of  $p$  at a distance of  $34\epsilon^2$  from  $p$ .

From Theorem 2.12,  $|I(t)| \geq |\phi(t)| - |E(x)| > d(p, t) - 34\epsilon^2 = 0$ . As the cut function  $I$  is continuous, there is a point  $s$  on  $pt$  at which  $I(s) = 0$  and  $d(p, s) < 34\epsilon^2$ . Since  $q$  is the closest point to  $p$  in  $U$ ,  $d(p, q) \leq d(p, s) < 34\epsilon^2$ .  $\square$

Consider the gradient of the cut function  $\nabla I(x) = \nabla \phi(x) + \nabla E(x)$ . Let  $p$  be the point closest to  $x$  on the surface  $F$ , and let  $\vec{n}$  be the surface normal at  $p$ . The gradient of the signed distance function is given by  $\nabla \phi(x) = \vec{n}$ . The gradient of the error function can be written as

$$\nabla E(x) = \sum_{s_i \in S} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} + \sum_{s_i, s_j \in S} \frac{2W_i(x)W_j(x)\zeta_i(x)(s_i - s_j)}{\epsilon^2 W^2(x)}.$$

We separate the contributions of sample points inside and outside  $B_2(x)$  to the gradient of the error function.

$$\nabla E_{\text{in}}(x) = \sum_{s_i \in B_2(x)} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} + \sum_{s_i, s_j \in B_2(x)} \frac{2W_i(x)W_j(x)\zeta_i(x)(s_i - s_j)}{\epsilon^2 W^2(x)}, \quad (2.23)$$

and,

$$\nabla E_{\text{out}}(x) = \sum_{s_i \notin B_2(x)} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} + \sum_{s_i \notin B_2(x) \vee s_j \notin B_2(x)} \frac{2W_i(x)W_j(x)\zeta_i(x)(s_i - s_j)}{\epsilon^2 W^2(x)}. \quad (2.24)$$

**Lemma 2.15.** *Let  $x$  be a point in the  $\epsilon$ -neighborhood of  $F$  and let  $p$  be the point on  $F$  closest to  $x$ . Let  $\vec{n}$  be the normal of  $p$ . Then  $\|\nabla E_{\text{in}}(x)\| < 486\epsilon$  and  $|\vec{n} \cdot \nabla E_{\text{in}}(x)| < 1158\epsilon^2$ .*

**Proof.** By Equation 2.23 we have

$$\begin{aligned} \|\nabla E_{\text{in}}(x)\| &\leq \sum_{s_i \in S} \frac{\|\vec{n} - \vec{n}_i\|W_i(x)}{W(x)} + \sum_{s_i, s_j \in B_2(x)} \frac{2W_i(x)W_j(x)|\zeta_i(x)|\|s_i - s_j\|}{\epsilon^2 W^2(x)} \\ &\leq \max_{s_i \in B_2(x)} \{\|\vec{n} - \vec{n}_i\|\} + \frac{2}{\epsilon^2} \max_{s_i, s_j \in B_2(x)} \{|\zeta_i(x)|\|s_i - s_j\|\}. \end{aligned} \quad (2.25)$$

By Lemma 2.9  $\|\vec{n} - \vec{n}_i\| \leq \Theta(4\epsilon) < 6\epsilon$ . By Equation 2.22 for each sample point  $s_i \in B_2(x)$ ,  $|\zeta_i(x)| < 30\epsilon^2$ . Since sample points  $s_i, s_j$  are inside a ball  $B_2(x)$  whose radius  $r_0 \leq 4\epsilon$ ,  $\|s_i - s_j\| \leq 8\epsilon$ . Substituting into Equation 2.25

$$\|\nabla E_{\text{in}}(x)\| < 6\epsilon + \frac{2}{\epsilon^2} \times 30\epsilon^2 \times 8\epsilon < 486\epsilon.$$

Consider  $\nabla E_{\text{in}}(x)$  projected onto the normal vector  $\vec{n}$ .

$$\begin{aligned} \vec{n} \cdot \nabla E_{\text{in}}(x) &\leq \max_{s_i \in B_2(x)} \{|\vec{n} \cdot (\vec{n} - \vec{n}_i)|\} \\ &\quad + \frac{2}{\epsilon^2} \max_{s_i, s_j \in B_2(x)} \{|\zeta_i(x)\vec{n} \cdot (s_i - s_j)|\} \end{aligned} \quad (2.26)$$

## Chapter 2. Provably Good Moving Least Squares

---

Since the angle between  $\vec{n}$  and  $\vec{n}_i$  is less than  $\Theta(4\epsilon) \leq 6\epsilon$ ,  $\vec{n} \cdot (\vec{n} - \vec{n}_i) = 1 - \cos 6\epsilon \leq 18\epsilon^2$ . By Lemma 2.2, the distance from all sample points inside  $B_2(x)$  to the tangent plane at  $p$  is at most  $(4\epsilon + \epsilon^2)^2/2 + \epsilon^2$ . Hence  $|\vec{n} \cdot (s_i - s_j)| \leq (4\epsilon + \epsilon^2)^2 + 2\epsilon^2 < 19\epsilon^2$ . Substituting into Equation 2.26

$$\vec{n} \cdot \nabla E_{\text{in}}(x) < 18\epsilon^2 + 1140\epsilon^2 < 1158\epsilon^2. \quad \square$$

In the following lemma we show that the sample points outside  $B_2(x)$  do not affect the gradient of the cut function. The proof is similar to the proof of Lemma 2.11 and is given in the appendix.

**Lemma 2.16.** *For each point  $x$  inside the  $\epsilon$ -neighborhood,  $\|\nabla E_{\text{out}}(x)\| < 146\epsilon$  and  $|\vec{n} \cdot \nabla E_{\text{out}}(x)| < 496\epsilon^2$ .*

**Proof.** Given in the appendix. □

**Theorem 2.17.** *For a point  $x$  inside the  $\epsilon$ -neighborhood, let  $\vec{n}$  be the surface normal of  $p$ , the point closest to  $x$  on  $F$ . Then,  $\vec{n} \cdot \nabla I(x) > 0$ .*

**Proof.** From the definition of the error function,

$$\vec{n} \cdot \nabla I(x) = \vec{n} \cdot \nabla \phi(x) + \vec{n} \cdot \nabla E(x) \geq 1 - |\vec{n} \cdot \nabla E_{\text{in}}(x)| - |\vec{n} \cdot \nabla E_{\text{out}}(x)|. \quad (2.27)$$

By Lemma 2.15  $|\vec{n} \cdot \nabla E_{\text{in}}(x)| < 1158\epsilon^2$ , and by Lemma 2.16  $|\vec{n} \cdot \nabla E_{\text{out}}(x)| < 496\epsilon^2$ . Substituting into Equation 2.27

$$\vec{n} \cdot \nabla I(x) > 1 - 1158\epsilon^2 - 496\epsilon^2 = 1 - 1654\epsilon^2 > 0. \quad \square$$

The result in Theorem 2.17 also proves that the gradient can never be zero inside the  $\epsilon$ -neighborhood. The zero set of  $I$  is inside the  $\epsilon$ -neighborhood of  $F$  by Theorem 2.7. Therefore from the implicit function theorem [17], zero is a *regular* value of  $I$  and the zero set  $U$  is a compact, two-dimensional manifold.

## Chapter 2. Provably Good Moving Least Squares

---

The normal of the reconstructed surface at a point  $u \in U$  is given by  $\vec{n}_u = \frac{\nabla I(u)}{\|\nabla I(u)\|}$ . In the following theorem we prove an upper bound on the angle between  $\vec{n}_u$  and the normal  $\vec{n}$  of the point  $p$  closest to  $u$  in  $F$ . We show that  $\vec{n}_u$  converges to  $\vec{n}$  as  $\epsilon$  goes to zero.

**Theorem 2.18.** *Let  $u$  be a point on the reconstructed surface  $U$  whose closest point on  $F$  is  $p$ . Let  $\vec{n}_u$  be the normal of  $U$  at  $u$  and let  $\vec{n}$  be the normal of  $F$  at  $p$ . An upper bound on the angle  $\theta$  between  $\vec{n}_u$  and  $\vec{n}$  is given by*

$$\cos \theta > \frac{1 - 1654\epsilon^2}{1 + 632\epsilon}.$$

**Proof.** The angle between  $\vec{n}$  and  $\vec{n}_u$  is given by

$$\cos \theta = \frac{\vec{n} \cdot \nabla I(u)}{\|\nabla I(u)\|}. \quad (2.28)$$

By Theorem 2.17 we have

$$\vec{n} \cdot \nabla I(u) \geq 1 - 1654\epsilon^2.$$

Consider the following upper bound on the norm of the gradient of the cut function,

$$\|\nabla I(u)\| \leq \|\nabla \phi(u)\| + \|\nabla E(u)\| \leq 1 + \|\nabla E_{\text{in}}(x)\| + \|\nabla E_{\text{out}}(u)\|.$$

By Lemma 2.15  $\|\nabla E_{\text{in}}(x)\| < 486\epsilon$ , and by Lemma 2.16  $\|\nabla E_{\text{out}}(x)\| < 146\epsilon$ . Substituting into Equation 2.28

$$\cos \theta > \frac{1 - 1654\epsilon^2}{1 + 632\epsilon}. \quad \square$$

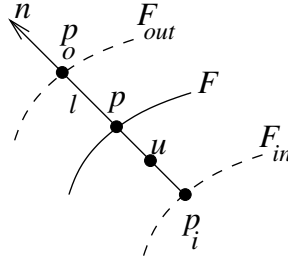


Figure 2.8: Points  $r, t$  are the closest points to  $p$  on the offset surfaces. The line segment  $p_i p_o$  intersects the zero set  $U$  at a unique point  $u$ .

## 2.5 Topological Properties

In this section we show that the reconstructed surface  $U$  has the same topology as the sampled surface  $F$ . Our first topological result is that  $U$  is homeomorphic to  $F$ . As surfaces  $F$  and  $U$  are compact, a one-to-one, onto, and continuous function from  $U$  to  $F$  defines a homeomorphism.

**Definition:** Let  $\Gamma : \mathbb{R}^3 \rightarrow F$  map each point  $q \in \mathbb{R}^3$  to the closest point of  $F$ .

**Theorem 2.19.** *The restriction of  $\Gamma$  to  $U$  is a homeomorphism from  $U$  to  $F$ .*

**Proof.** The discontinuities of  $\Gamma$  are the points on the medial axis of  $F$ . As  $U$  is constrained to be inside the  $\epsilon$ -neighborhood of  $F$ , the restriction of  $\Gamma$  to  $U$  is continuous.

Now we show that  $\Gamma$  is one-to-one. Let  $p$  be a point on  $F$  and let  $\vec{n}$  be the normal at  $p$  as shown in Figure 2.8. Consider the line segment  $l$  parallel to  $\vec{n}$  that intersects  $F_{out}$  and  $F_{in}$  at  $p_o$  and  $p_i$  respectively. At each point  $y \in p_o p_i$ ,  $\nabla I(y) \cdot \vec{n} > 0$  by Theorem 2.17. So the function  $I(x)$  is monotonically decreasing from  $r$  to  $t$  and there is a unique point  $u$  on  $p_o p_i$  where  $I(u) = 0$ . Assume there is another point  $v \in U$  for which  $\Gamma(v) = x$ . The point  $v$  has to be outside the segment  $p_o p_i$  which means that the distance from  $v$  to its closest point on  $F$  is greater than  $\epsilon$ . This contradicts Theorem 2.13.

Finally we need to show that  $\Gamma$  is onto. As  $\Gamma$  maps closed components of  $U$  onto closed

components of  $F$  in a continuous manner,  $\Gamma(U)$  should consist of a set of closed connected components. Consider the point  $p$  in Figure 2.8. Assume that  $q = \Gamma(u)$  is not in the same component of  $F$  as  $p$ . Since  $q$  is closest to  $u$ ,  $d(u, q) \leq d(u, p) \leq \epsilon$ . Let  $B_u$  be the ball of radius  $\epsilon$  centered at  $u$  that intersects two components of  $F$ , one containing point  $p$  and one containing point  $q$ . Boissonnat and Cazals [21] (Proposition 12) show that any ball whose intersection with  $F$  is not a topological disc, contains a point of the medial axis of  $F$ . Therefore point  $p$  is inside the ball  $B_u$  that contains a point of the medial axis. By the definition of the function  $\text{lfs}$ ,  $\text{lfs}(p) \leq 2\epsilon$ . Recall that our sampling preconditions require  $\epsilon \leq 1/50$ . Therefore  $\text{lfs}(p) \leq 2\epsilon \leq 1/25$ . This violates our assumption that  $\text{lfs}(p) \geq 1$ . Hence the function  $\Gamma$  maps  $U$  onto every closed component of  $F$ .  $\square$

Amenta, Peters, and Russell [9] have argued that a guarantee of homeomorphism is insufficient for applications in graphics and simulations. They propose using ambient isotopy to show topological equivalence. An isotopy from  $U$  to  $F$  intuitively means that  $U$  can be continuously deformed into  $F$  without any change in topology.

**Definition:** An isotopy between two compact orientable surfaces in  $\mathbb{R}^3$  is a continuous map  $\Psi : U \times [0, 1] \rightarrow \mathbb{R}^3$  such that  $\Psi(\cdot, 0)$  is the identity of  $U$ ,  $\Psi(\cdot, 1) = F$ , and for each  $t \in [0, 1]$ ,  $\Psi(\cdot, t)$  is homeomorphic to  $U$ .

**Definition:** An ambient isotopy between two compact orientable surfaces  $U$  and  $F$  is a continuous map  $\Lambda : \mathbb{R}^3 \times [0, 1] \rightarrow \mathbb{R}^3$ , such that  $\Lambda(\cdot, 0)$  is the identity of  $\mathbb{R}^3$ ,  $\Lambda(U, 1) = F$ , and for each  $t \in [0, 1]$ ,  $\Lambda(\cdot, t)$  is a homeomorphism of  $\mathbb{R}^3$ .

**Theorem 2.20.** *The zero surface  $U$  is isotopic to the sampled surface  $F$ .*

**Proof.** We will define an ambient isotopy  $\Lambda$  whose restriction to  $U$  will be an isotopy to  $F$ . Outside the  $\epsilon$ -neighborhood, the ambient isotopy is the identity  $\Lambda(x, t) = x$  for  $t \in [0, 1]$ .



From the proof of Theorem 2.19, a line segment  $l$  normal to a point  $p \in F$  intersects  $U$  only at one point  $u$  inside the  $\epsilon$ -neighborhood. Let  $p_i$  and  $p_o$  be the end points of  $l$  on the inside and outside  $\epsilon$ -offset surfaces respectively as shown in Figure 2.8. We define the ambient isotopy at  $u$  to be  $\Lambda(u, t) = tp + (1 - t)u$ . The line segment  $p_i u$  is linearly mapped to  $p_i \Lambda(u, t)$ . Similarly, the line segment  $u p_o$  is mapped to  $\Lambda(u, t) p_o$ .  $\square$

Recently, Chazal and Cohen-Steiner [27] proved a condition for isotopic approximation that gives an alternate proof for Theorem 2.20. They show that if two surfaces  $A$  and  $B$  are homeomorphic, if  $A$  is contained in a topological thickening  $M$  of  $B$ , and if  $A$  separates the sides of  $M$ , then  $A$  is isotopic to  $B$ . From Theorem 2.19, we know that  $U$  is homeomorphic to  $F$ . The  $\epsilon$ -neighborhood is a topological thickening of the sampled surface  $F$ , and by Theorem 2.7 the zero set  $U$  of the cut function separates the two sides of the  $\epsilon$ -neighborhood. Therefore  $U$  is isotopic to  $F$ .

## 2.6 Discussion

Recall that the width of the Gaussian functions used in our algorithm depends on the smallest local feature size. As a result, our sampling requirements and the noise preconditions are determined by the smallest local feature size of a point on  $F$ . Ideally, we would like to handle sampling proportional to the local feature size. When the width of the Gaussian weight functions is fixed, spacing between sample points in areas of the surface with large local feature size might be much larger than the width of the Gaussians. As a result, the reconstructed surface will be noisy and might have the wrong topology. One solution is to make the width of the Gaussian weight functions proportional to the spacing between sample points. We present an adaptive version of the MLS algorithm based on this idea in Chapter 3.

One disadvantage of our algorithm is that it requires sample point normals. However, approximate sample point normals can be easily obtained for laser range data by triangulating the range images. Each sample point normal can be oriented using the location of the range scanner. When oriented normals are unavailable, the absolute distance to the tangent plane at each sample point can be used instead of the signed distance as a point function to define a new function  $I_u(x)$ . The zero set of this function is hard to analyze as its gradient is not smooth near the sample points. However, the results in this chapter can be easily extended to show that the  $\epsilon$ -level set of  $I_u(x)$  consists of two components on each side of the surface, each homeomorphic to  $F$ .

The zero set of the cut function  $I$  only passes near the sample points, but we can construct a surface that interpolates the sample points with weight functions such as  $W_s(x) = \frac{e^{-\|x-s\|^2}}{\|x-s\|^2}$ , that are infinite at the sample points. We can prove that the zero set is restricted to the  $\epsilon$ -neighborhood when this weight function is used, but, we could not prove results about the gradient approximations.

Our analysis of the MLS algorithm works in any dimension. The Lipschitz condition in Theorem 2.1, and our results which show that sample points outside  $B_2(x)$  have little effect on the cut function at point  $x$  can be easily extended to higher dimensions. The value of  $\epsilon$  required for the theoretical guarantees would decrease with increasing dimensionality.

It would be interesting to prove similar results for different weight functions and point functions. Gaussian weight functions that have infinite support yield a cut function that is defined everywhere. Our results show that we only pay a small penalty for the infinite support; the cut function at point  $x$  is mostly determined by sample points inside  $B_2(x)$ . It might be easier to analyze other commonly used weight functions such as cubic B-splines that have compact support. Of course, the cut function and its analysis would be valid only in areas covered by the support of the weight functions.

Here we have used point functions that are a first-order approximation to the neighborhood of each sample. For a sample point  $s$  that is near sharp features, the point function

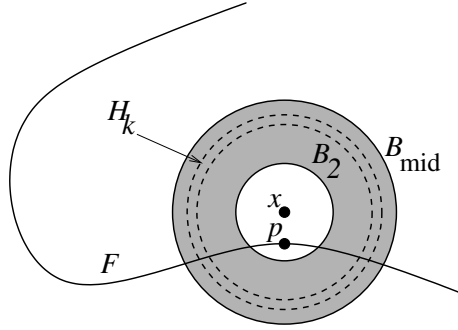


Figure 2.9: For point  $x$  inside the  $\epsilon$ -neighborhood, all sample points inside  $B_{\text{mid}}(x)$  are near  $p$  the point closest to  $x$  on  $F$ . As a result, the error in the point functions of sample points inside  $B_{\text{mid}}(x)$ , evaluated at  $x$ , is small.

of  $s$  is a poor representation of the local neighborhood of  $s$ . Higher order approximations such as quadric surface patches might be more robust to sharp features.

## Appendix

Consider a ball  $B_{\text{mid}}(x)$  of radius  $r_{\text{mid}} = 0.125$  centered at point  $x$  as shown in Figure 2.9. Note that  $r_{\text{mid}}$  is a fixed constant and does not depend on  $\epsilon$ . All sample points inside  $B_{\text{mid}}(x)$  are near the point  $p$  closest to  $x$  on  $F$ . Recall that for each sample point  $s_i$ , the oversampling factor  $a_i$  is the number of sample points inside a ball of radius  $\epsilon$  around  $s_i$ . In the following lemma we prove a bound on the oversampling factors of sample points in a shell  $H_k$  when  $H_k$  is contained inside  $B_{\text{mid}}(x)$ .

**Lemma 2.21.** *For a point  $x$  inside the  $\epsilon$ -neighborhood, let  $H_k$  be a spherical shell centered at  $x$  that is outside  $B_2(x)$ , and inside  $B_{\text{mid}}(x)$ . Let the radii of the two spheres bounding  $H_k$  be  $r_k$  and  $r_k + \epsilon$ . Then  $\sum_{s_i \in H_k} \frac{1}{a_i} < 4 \frac{r_k^3}{\epsilon^3}$ .*

**Proof.** Let  $p$  be the point closest to  $x$  on the surface. as shown in Figure 2.10(a). As  $x$  is inside the  $\epsilon$ -neighborhood, each sample point  $s \in H_k$  is inside a ball  $B$  of radius  $r_k + 2\epsilon \leq r_{\text{mid}} + 2\epsilon < 1/4$  centered at  $p$ . All sample points inside  $B$  are squeezed between two parallel planes  $P_1, P_2$  that are  $(r_k + 2\epsilon + \epsilon^2)^2 + 2\epsilon^2$  apart, by Lemma 2.2.

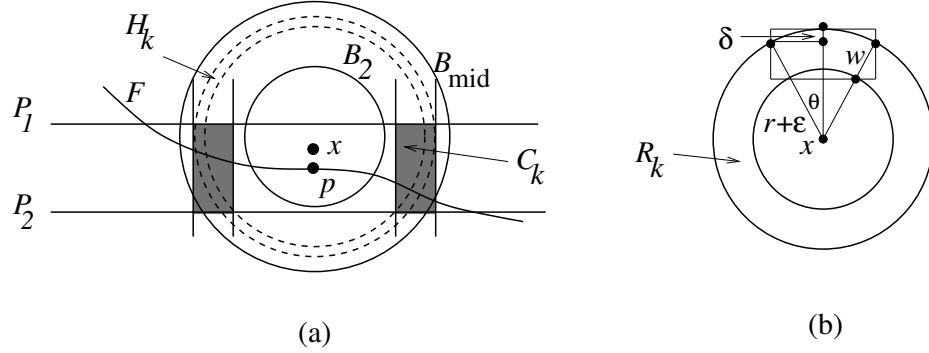


Figure 2.10: (a) Point  $x$  is inside the  $\epsilon$ -neighborhood and  $p$  is the point closest to  $x$  on  $F$ .  $H_k$  is a spherical shell of width  $\epsilon$  contained inside  $B_{\text{mid}}$ . (b) The base of  $C_k$  is a ring  $R_k$  whose width is  $w$ . The ring  $R_k$  is divided into sectors such that the arc length of the outside circle within each sector is less than  $\epsilon/\sqrt{3}$ .

The intersection of  $H_k$  and the space between  $P_1, P_2$  is a cylindrical annulus  $C_k$  whose cross section is the shaded region shown in Figure 2.10(a). The radius of the outer circle of  $C_k$  is  $r + \epsilon$ . To compute the inner radius, consider the distance from  $x$  to  $P_1, P_2$  given by

$$l = \max\{d(x, P_1), d(x, P_2)\} \leq (r_k + 2\epsilon + \epsilon^2)^2/2 + \epsilon^2 + \epsilon.$$

The radius of the smaller circle enclosing  $C_k$  can be written as  $r_i = \sqrt{r_k^2 - l^2}$ . Let  $w = r_k - r_i$  be the width of the annulus.

Let  $R_k$  be the base of the cylindrical annulus  $C_k$  as shown in Figure 2.10(b). Divide the ring  $R_k$  into sectors such that the length of the outer arc inside each sector is  $\epsilon/\sqrt{3}$ . Each sector is contained in a rectangle of length at most  $\epsilon/\sqrt{3}$  as shown in Figure 2.10(b). The height of the rectangle is  $h \leq \delta + w \leq \frac{\epsilon^2}{24(r+\epsilon)} + w$ . Consider a covering of the rectangle with squares of size  $\epsilon/\sqrt{3}$ . Since the squares can extend out of the rectangle, we can cover each rectangle with

$$t_1 = \frac{\sqrt{3}}{\epsilon} \times (h + \epsilon/\sqrt{3}) \leq \frac{\sqrt{3}}{\epsilon} \left( \frac{\epsilon^2}{24(r+\epsilon)} + w + \epsilon/\sqrt{3} \right).$$

squares of size  $\epsilon/\sqrt{3}$ . As each rectangle covers a sector of length  $\epsilon/\sqrt{3}$ , ring  $R_k$  can be covered with,

$$t_2 = t_1 \times \left\lceil \frac{2\sqrt{3}\pi}{\epsilon}(r_k + \epsilon) \right\rceil \leq t_1 \times \left( \frac{2\sqrt{3}\pi}{\epsilon}(r_k + \epsilon) + 1 \right)$$

squares.

We can now fill the cylindrical annulus  $C_k$  by stacking cubes of size  $\epsilon/\sqrt{3}$  on top of these squares. The number of cubes required to cover  $C_k$  is

$$t_1 \times t_2 \times \frac{\sqrt{3}}{\epsilon} ((r_k + 2\epsilon + \epsilon^2)^2 + 2\epsilon^2 + \epsilon/\sqrt{3}).$$

Each cube in this grid is covered by a sphere of radius  $\epsilon/2$ . For  $x$  inside the  $\epsilon$ -neighborhood,  $\epsilon \leq r_0/\sqrt{12} \leq r_k/\sqrt{12}$  by Equation 2.18. Applying Lemma 2.3 to each sphere, and simplifying

$$\sum_{s_i \in H_2} \frac{1}{a_i} < 6 \frac{r_k^3}{\epsilon^3}. \quad \square$$

**Proof of Lemma 2.11.** We compute the desired upper bound by summing over the contributions of sample points inside shells  $H_k$  starting at  $B_2(x)$ . Recall that  $\xi_{\text{out}}(x)$  is the sum of the error functions associated with sample points outside  $B_2(x)$ . Let  $\xi_1(x)$  and  $\xi_2(x)$  be the contributions to  $\xi_{\text{out}}(x)$  by sample points inside  $B_{\text{mid}}(x)$  and outside  $B_{\text{mid}}(x)$  respectively. An upper bound for  $|\xi_1(x)|$  is given by  $|\xi_1(x)| \leq \sum_{s_i \in B_{\text{mid}}(x)} |\zeta_i(x)| W_i(x)$ . Consider shell  $H_k$  inside  $B_{\text{mid}}$ . The radius of the smaller sphere bounding  $H_k$  is  $r_k \leq r_{\text{mid}}$ . Let  $p$  be the point closest to  $x$  on  $F$ . The distance from  $p$  to each sample point  $s$  inside  $H_k$  is given by

$$d(p, s) \leq d(p, x) + d(x, s) \leq r_k + 2\epsilon \leq r_{\text{mid}} + 2\epsilon < 1/4.$$

Recall that we proved an upper bound on the error in the point functions of sample points

## Chapter 2. Provably Good Moving Least Squares

---

near  $p$  in Lemma 2.9. By the inequality  $\sqrt{12}\epsilon \leq r_0 \leq r_k \leq r_{\text{mid}}$ ,

$$\Theta(r_k + 2\epsilon) \leq \frac{r_k + 2\epsilon + \epsilon^2}{1 - 3(r_k + 2\epsilon + \epsilon^2)} \leq \frac{r_k(1 + 2/\sqrt{12} + \epsilon/\sqrt{12})}{1 - 3(r_{\text{mid}} + 2/50 + 1/2500)} \approx 3.06r_k, \quad (2.29)$$

and,

$$\begin{aligned} |\zeta_i(x)| &\leq |\phi(x)|\Theta^2(r_k + 2\epsilon)/2 + (r_k + 2\epsilon)\Theta(r_k + 2\epsilon) + (r_k + 2\epsilon + \epsilon^2)^2/2 + \epsilon^2 \\ &\approx 6.25r_k^2 < 7r_k^2. \end{aligned} \quad (2.30)$$

Substituting the upper bound on the weights of sample points in shells  $H_k$  contained in  $B_{\text{mid}}(x)$  from Lemma 2.21 we have

$$|\xi_1(x)| \leq \frac{42}{\epsilon^3} \sum_{k=0}^{r_k \leq r_{\text{mid}}} r_k^5 e^{-r_k^2/\epsilon^2} < \frac{42}{\epsilon^3} \sum_{k=0}^{\infty} r_k^5 e^{-r_0 r_k/\epsilon^2} \leq \frac{63}{\epsilon^3} r_0^5 e^{-r_0^2/\epsilon^2}. \quad (2.31)$$

By the lower bound on  $W(x)$  in Equation 2.11

$$\frac{|\xi_1(x)|}{W(x)} < \frac{504}{\epsilon^3} r_0^5 e^{-(r_0^2 - (|\phi(x)| + \epsilon)^2)/\epsilon^2} < \frac{504}{\epsilon^3} r_0^5 e^{-12}. \quad (2.32)$$

For sample points outside  $B_{\text{mid}}(x)$ , we revert to the bound obtained in Lemma 2.6 on the error function associated with each sample. By Equation 2.13 we have

$$|\xi_2(x)| \leq \sum_{s_i \notin B_{\text{mid}}(x)} |\zeta_i(x)| W_i(x) \leq 2 \sum_{s_i \notin B_{\text{mid}}(x)} \|x - s_i\| W_i(x). \quad (2.33)$$

The expression for  $|\xi_2(x)|$  in Equation 2.33 is the same as the expression for  $|\xi_{\text{out}}(x)|$  in Equation 2.13 except that the summation begins at  $r_{\text{mid}}$  instead of  $r_{\text{out}}$ . By Equation 2.15 we have

$$\frac{|\xi_2(x)|}{W(x)} \leq \frac{7200}{\epsilon^2} r_{\text{mid}}^3 e^{-(r_{\text{mid}}^2 - (|\phi(x)| + \epsilon)^2)/\epsilon^2} \leq \frac{7200e^4}{\epsilon^2} r_{\text{mid}}^3 e^{-r_{\text{mid}}^2/\epsilon^2}. \quad (2.34)$$

## Chapter 2. Provably Good Moving Least Squares

---

Note that the expression in Equation 2.34 is  $o(\epsilon^2)$  because of the exponential term. For each point  $x$  is inside the  $\epsilon$ -neighborhood,  $r_0 \leq \sqrt{(\epsilon + \epsilon)^2 + 12\epsilon^2} = 4\epsilon$ . Adding the upper bounds in Equation 2.32 and Equation 2.34 we have

$$|E_{\text{out}}(x)| \leq \frac{504}{\epsilon^3}(4\epsilon)^5 e^{-12} + \frac{7200e^4}{\epsilon^2} r_{\text{mid}}^3 e^{-r_{\text{mid}}^2/\epsilon^2} < 4\epsilon^2. \quad \square$$

**Proof of Lemma 2.16.** Recall that  $x$  is a point inside the  $\epsilon$ -neighborhood, and  $p$  is the point closest to  $x$  on  $F$ . Let  $\vec{n}$  be the normal at  $p$ . The contribution of sample points outside  $B_2(x)$  to  $\nabla E(x)$  is given by,

$$\nabla E_{\text{out}}(x) = \sum_{s_i \notin B_2(x)} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} + \sum_{s_i \notin B_2(x) \vee s_j \notin B_2(x)} \frac{2W_i(x)W_j(x)\zeta_i(x)(s_i - s_j)}{\epsilon^2 W^2(x)}. \quad (2.35)$$

The first term in Equation 2.35 can be written as a summation over sample points inside spherical shell  $H_k$  outside  $B_2(x)$ .

$$\frac{1}{W(x)} \sum_{k=0}^{\infty} \sum_{s_i \in H_k} (\vec{n} - \vec{n}_i)W_i(x).$$

Assume without loss of generality that the indices of sample points are in the increasing order of distance to  $x$ . We can write the second term in Equation 2.35 as

$$T = \sum_{s_i \notin B_2(x), i > j} \frac{2W_i(x)W_j(x)(\zeta_i(x) - \zeta_j(x))(s_i - s_j)}{\epsilon^2 W^2(x)}. \quad (2.36)$$

For each spherical shell  $H_k$  define a set  $S_k$  containing pairs of sample points given by

$$S_k = \{(s_i, s_j) \mid i > j, s_i \in H_k\}.$$

Intuitively,  $s_i$  is inside the shell  $H_k$  and  $s_j$  is inside the larger sphere that surrounds  $H_k$ .

The summation in Equation 2.36 can be written as,

$$T = \frac{2}{\epsilon^2 W^2(x)} \sum_{k=0}^{k=\infty} \sum_{(s_i, s_j) \in S_k} W_i(x) W_j(x) (\zeta_i(x) - \zeta_j(x)) (s_i - s_j). \quad (2.37)$$

An upper bound on the norm of  $T$  is given by

$$\begin{aligned} \|T\| &\leq \frac{2}{\epsilon^2 W^2(x)} \sum_{k=0}^{k=\infty} \sum_{(s_i, s_j) \in S_k} W_i(x) W_j(x) |\zeta_i(x) - \zeta_j(x)| \|s_i - s_j\| \\ &< \frac{2}{\epsilon^2 W(x)} \sum_{k=0}^{k=\infty} \sum_{s_i \in S_k} W_i(x) \max_{(s_i, s_j) \in S_k} \{|\zeta_i(x) - \zeta_j(x)| \|s_i - s_j\|\}. \end{aligned}$$

So the expression for  $\|\nabla E_{\text{out}}(x)\|$  can be written as,

$$\begin{aligned} \|\nabla E_{\text{out}}(x)\| &\leq \frac{1}{W(x)} \sum_{k=0}^{k=\infty} \sum_{s_i \in H_k} W_i(x) (\|\vec{n} - \vec{n}_i\| \\ &\quad + \frac{2}{\epsilon^2} \max\{|\zeta_i(x) - \zeta_j(x)| \|s_i - s_j\| \mid (s_i, s_j) \in S_k\}) \quad (2.38) \end{aligned}$$

We split the summation in the above equation into  $\|\nabla E_1(x)\|$  containing the contributions of all sample points in shells  $H_k \in B_{\text{mid}}(x)$ , and  $\|\nabla E_2(x)\|$  containing the contributions of sample points in shells  $H_k \notin B_{\text{mid}}(x)$ .

As  $x$  is inside the  $\epsilon$ -neighborhood, for each sample point  $s_i$  inside shell  $H_k$ ,  $d(p, s_i) \leq r + 2\epsilon$ . For each sample point  $s_i$  that is inside  $H_k \in B_{\text{mid}}(x)$ ,  $\|n - n_i\| \leq \Theta(r_k + 2\epsilon) < 4r_k$  by Equation 2.29, and  $|\zeta_i(x) - \zeta_j(x)| \leq |\zeta_i(x)| + |\zeta_j(x)| \approx 12.5r_k^2$  by Equation 2.30. Since  $s_i, s_j$  are inside a ball of radius  $r_k + 2\epsilon$  at  $p$ ,

$$|\zeta_i(x) - \zeta_j(x)| \times \|s_i - s_j\| \leq 12.5r_k^2 \times 2(r_k + 2\epsilon) \leq 12.5r_k^2 \times 2(r_k + 2r_k/\sqrt{12}) < 40r_k^3.$$



## Chapter 2. Provably Good Moving Least Squares

---

Substituting the above upper bound into Equation 2.38,  $\|\nabla E_1(x)\|$  can be written as

$$\|\nabla E_1(x)\| < \frac{1}{W(x)} \sum_{k=0}^{r_k \leq r_{\text{mid}}} \sum_{s_i \in H_k} \left(4r_k + \frac{80r_k^3}{\epsilon^2}\right) \frac{e^{-r_k^2/\epsilon^2}}{a_i}.$$

By the bound on the weights of sample points inside each shell  $H_k \in B_{\text{mid}}(x)$  proved in Lemma 2.21 we have

$$\begin{aligned} \|\nabla E_1(x)\| &\leq \frac{6}{W(x)\epsilon^3} \sum_{k=0}^{r_k \leq r_{\text{mid}}} \left(4r_k + \frac{80r_k^3}{\epsilon^2}\right) r_k^3 e^{-r_k^2/\epsilon^2} \\ &\leq \frac{9}{W(x)\epsilon^3} \left(4r_0 + \frac{80r_0^3}{\epsilon^2}\right) r_0^3 e^{-r_0^2/\epsilon^2} \end{aligned}$$

Substituting the lower bound for  $W(x)$  from Equation 2.11

$$\|\nabla E_1(x)\| \leq \frac{72}{\epsilon^3} \left(4r_0 + \frac{80r_0^3}{\epsilon^2}\right) r_0^3 e^{-(r_0^2 - (|\phi(x)| + \epsilon^2)^2)} = \frac{72}{\epsilon^3} \left(4r_0 + \frac{80r_0^3}{\epsilon^2}\right) r_0^3 e^{-12}.$$

For shells  $H_k$  such that  $r_k \geq r_{\text{mid}}$ ,  $\|n - n_i\| \leq 2$ , and  $|\zeta_i(x) - \zeta_j(x)| \leq |\zeta_i(x)| + |\zeta_j(x)| \leq 4(r_k + \epsilon)$  by Equation 2.13. Since,  $s_i, s_j$  are inside  $H_k$ ,

$$|\zeta_i(x) - \zeta_j(x)| \times \|s_i - s_j\| \leq 4(r_k + \epsilon) \times 2(r_k + \epsilon) < 14r_k^2.$$

By the upper bound on the weight of sample points inside  $H_k$  from Lemma 2.4 we have

$$\|\nabla E_2(x)\| \leq \frac{300}{W(x)\epsilon^2} \sum_{r_k=r_{\text{mid}}}^{\infty} \left(2 + \frac{28r_k^2}{\epsilon^2}\right) r_k^2 e^{-r_k^2/\epsilon^2} \leq \frac{3600e^4}{\epsilon^2} \left(2 + \frac{28r_{\text{mid}}^2}{\epsilon^2}\right) r_{\text{mid}}^2 e^{-r_{\text{mid}}^2/\epsilon^2}.$$

Recall that  $r_0 \leq 4\epsilon$  for  $x$  inside the  $\epsilon$ -neighborhood. An upper bound on  $\|\nabla E_{\text{out}}(x)\|$  is

## Chapter 2. Provably Good Moving Least Squares

---

given by,

$$\begin{aligned}
 \|\nabla E_{\text{out}}(x)\| &\leq \|\nabla E_1\| + \|\nabla E_2(x)\| \\
 &\leq \frac{72}{\epsilon^3}(4r_0 + \frac{80r_0^3}{\epsilon^2})r_0^3e^{-12} + \frac{3600e^4}{\epsilon^2}(2 + \frac{28r_{\text{mid}}^2}{\epsilon^2})r_{\text{mid}}^2e^{-r_{\text{mid}}^2/\epsilon^2} \\
 &< 146\epsilon.
 \end{aligned}$$

To derive an upper bound on  $|\vec{n} \cdot \nabla E_{\text{out}}(x)|$ , consider a sample point  $s_i$  inside shell  $H_k \in B_{\text{mid}}(x)$ . By Equation 2.29 we have

$$\vec{n} \cdot (\vec{n} - \vec{n}_i) = 1 - \cos \Theta(r_k + 2\epsilon) \leq \Theta^2(r_k + 2\epsilon)/2 < 5r^2.$$

Sample point  $s_i$  is between two planes  $P_1, P_2$  orthogonal to  $\vec{n}$  that are at a distance  $(r_k + 2\epsilon + \epsilon^2)^2 + 2\epsilon^2$  apart, by Lemma 2.2. Therefore

$$|(\zeta_i(x) - \zeta_j(x)) \times \vec{n} \cdot (s_i - s_j)| \leq 12.5r_k^2 \times ((r_k + 2\epsilon + \epsilon^2)^2 + 2\epsilon^2) < 34r_k^4,$$

by Equation 2.30. It is easy to show that

$$|\vec{n} \cdot \nabla E_1(x)| \leq \frac{72}{\epsilon^3}(5r_0^2 + \frac{68r_0^4}{\epsilon^2})r_0^3e^{-12},$$

and,

$$|\vec{n} \cdot \nabla E_2(x)| \leq \frac{3600e^4}{\epsilon^2}(2 + \frac{28r_{\text{mid}}^2}{\epsilon^2})r_{\text{mid}}^2e^{-r_{\text{mid}}^2/\epsilon^2}.$$

Adding the upper bounds on  $|\vec{n} \cdot \nabla E_1(x)|$ , and  $|\vec{n} \cdot \nabla E_2(x)|$  we have

$$|\vec{n} \cdot \nabla E_{\text{out}}(x)| < \frac{72}{\epsilon^3}(5r_0^2 + \frac{68r_0^4}{\epsilon^2})r_0^3e^{-12} + \frac{3600e^4}{\epsilon^2}(2 + \frac{28r_{\text{mid}}^2}{\epsilon^2})r_{\text{mid}}^2e^{-r_{\text{mid}}^2/\epsilon^2} < 496\epsilon^2. \quad \square$$

## Chapter 3

# Provably Better Moving Least Squares

In this chapter we generalize the MLS algorithm of Chapter 2 to handle adaptively sampled point clouds. Consider the curve shown in Figure 3.1. Because of a portion of the curve that has a small feature size, the sampling should be dense everywhere under the uniform sampling conditions of Chapter 2, as shown in Figure 3.1(b). Moreover, the noise in the sample points is required to be much smaller than the smallest feature size of the surface. Ideally, we would like the algorithm to accommodate adaptive spacing in the input point cloud as well as data noise proportional to the local feature size. Figure 3.1(c) shows a two-dimensional example.

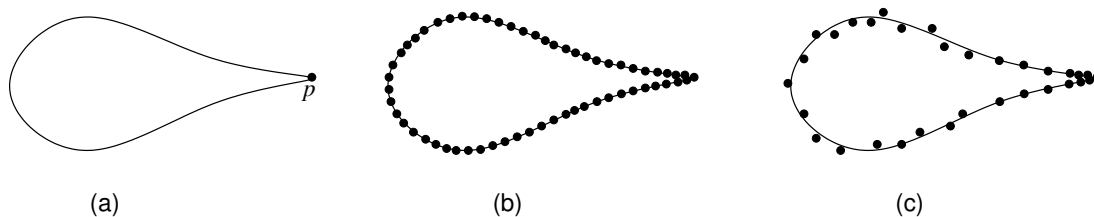


Figure 3.1: (a) Point  $p$  near the tip has the smallest local feature size of all the points on the curve. (b) Uniform sampling means that the spacing between sample points and the noise are, at most, proportional to  $l_{fs}(p)$ . (c) Adaptive sampling, proportional to the local feature size of surface points. The noise in the sample points can also vary with the local feature size.

Before we describe the adaptive MLS algorithm, it is useful to understand why the uniform MLS algorithm of Chapter 2 breaks down under adaptive sampling conditions. The width of Gaussian functions in the definition of the uniform MLS surface depends on the smallest feature size of the sampled surface. When the spacing between the sample points is adaptive, this width is much smaller than the sample point spacing near areas of the surface with large feature size. The Gaussian weight functions no longer smoothly blend the point functions of nearby sample points. As a result, both the gradient and the topology of the reconstructed surface are unreliable.

The main idea of the adaptive MLS algorithm is to vary the widths of the Gaussian weight functions to match the local sample point spacing. These widths are computed by estimating the local feature size near each sample point.

### 3.1 Guarantees

Our theoretical guarantees are similar to the results in Chapter 2. We prove that the cut function defined by the adaptive MLS algorithm is non-zero outside a small neighborhood  $N$  of the sampled surface. The width of the neighborhood  $N$  is proportional to the local feature size of the sampled surface. As the spacing between sample points in the input point cloud goes to zero, the Hausdorff distance between the reconstructed surface and the sampled surface goes to zero and the reconstructed surface normals converge to the true surface normals. The reconstructed surface is guaranteed to be a smooth manifold that has the same topology as the sampled surface.

### 3.2 Related Work

For adaptively sampled point cloud data, Delaunay-based surface reconstruction algorithms such as the *crust* proposed by Amenta and Bern [3] are the gold standard as they can easily

adapt to changes in sample point spacing. However, MLS-based algorithms are better than Delaunay-based algorithms at handling noisy data.

Varying the widths of the Gaussian weight functions that define MLS surfaces was first proposed by Pauly, Gross, and Kobbelt [69]. They estimate the feature size of the surface near each sample point  $s$  as the radius of the smallest ball at  $s$  that contains  $k$  ( $\approx 20$ ) sample points. The widths of the Gaussian weight functions are computed by interpolating the feature size estimates with radial basis functions.

Dey and Sun [32] analyze an adaptive MLS surface and prove certain reconstruction guarantees. The width of the Gaussian weight function associated with sample point  $s$  at a given point  $x$  is a function of the feature size of  $s$  and the feature size of the point closest to  $x$  on the sampled surface. Unlike our results which prove that the entire cut function converges to the signed distance function, their analysis of the MLS surface is limited to a small neighborhood of the sample points.

### 3.3 Sampling Preconditions

The sampling preconditions required to prove theoretical guarantees for the adaptive MLS algorithm are a generalization of the sampling preconditions defined in Chapter 2, and are based on the *local feature size* (lfs) function proposed by Amenta and Bern [3]. Recall that for point  $p \in F$ ,  $\text{lfs}(p)$  is the distance from  $p$  to the medial axis of  $F$ . We require sample point spacing near  $p$  to be small compared to  $\text{lfs}(p)$ .

Given a set of sample points on the surface  $F$ , computing the local feature size near each sample point is impossible because the medial axis of  $F$  is unknown. The best we can hope for is an approximation to the function  $\text{lfs}$ . The adaptive MLS algorithm builds the weight functions from an approximation to the function  $\text{lfs}$  given at each sample point. We call this approximation the *sample induced feature size* (or *sif*).

The function  $\text{sif}$  is a 1-Lipschitz function whose value at each sample point is given as

input to the adaptive MLS algorithm. This function  $\text{sif}$  should satisfy the following two conditions. First, the set of sample points  $S$  should be dense with respect to the function  $\text{sif}$ . For each point  $p \in F$ , the distance from  $p$  to its closest sample point  $s$  should be less than  $\epsilon \text{sif}(p)$ . Second, the noise in each sample point  $s$  should be small compared to the the function  $\text{sif}$  near  $s$ . The distance from each sample point  $s$  to its closest surface point  $p \in F$  is less than  $\epsilon^2 \text{sif}(p)$ . Moreover, the angle between the normal  $\vec{n}_s$  at  $s$  and the true normal  $\vec{n}_p$  at  $p$  should be less than  $\epsilon$ . The results in this chapter hold true for values of  $\epsilon \leq 1/150$ .

The two conditions force the value of the function  $\text{sif}$  at each sample point  $s$  to be larger than a certain value that depends on the sample spacing near  $s$ . However, the function  $\text{sif}$  cannot be arbitrarily large. To prove our theoretical guarantees, we require the function  $\text{sif}$  to be less than the function  $\text{lfs}$  on the surface. For each point  $p \in F$ ,

$$\text{sif}(p) \leq \text{lfs}(p). \tag{3.1}$$

Unlike the function  $\text{lfs}$ , the function  $\text{sif}$  is defined everywhere because the sample points might not lie on  $F$ . We assume that the values of the function  $\text{sif}$  are known at each sample point. Finally, we need one condition on the rate at which the sample spacing inside  $S$  changes. Arbitrary oversampling in one region of the surface can distort the value of the cut function in other parts of the surface. Our MLS algorithm estimates oversampling near each sample point  $s$  from the number of sample points that are inside a ball of radius  $\epsilon \text{sif}(s)$  around  $s$ . This estimation only works if changes in sample point spacing near  $s$  are bounded. Let  $\alpha_i, \alpha_j$  be the number of sample points inside two balls of radii  $r_i \leq r_j \leq 3\epsilon \text{sif}(s)$  at a sample point  $s$  including  $s$ . Then the ratio of the number of sample points inside the two balls is bounded by the ratio of their volumes,  $\alpha_j/\alpha_i \leq r_j^3/r_i^3$ . In Section 3.4.2 we discuss our method for estimating oversampling in the input point cloud.

### 3.3.1 Sample Induced Feature Size

The MLS algorithm needs the the value of the function  $\text{sif}$  only at each sample point  $s \in S$ . In the following lemma we show that  $\text{sif}(s)$  is a good estimate of the sample induced feature size of surface points near  $s$ .

**Lemma 3.1.** *Let  $s$  be a sample point, and let  $p$  be the point closes to  $s$  on  $F$ . Then,*

$$(1 - \epsilon^2)\text{sif}(p) \leq \text{sif}(s) \leq (1 + \epsilon^2)\text{sif}(p).$$

**Proof.** Since the function  $\text{sif}$  is 1-Lipschitz,

$$\text{sif}(s) \leq \text{sif}(p) + d(s, p) \leq (1 + \epsilon^2)\text{sif}(p),$$

by the sampling preconditions. Similarly

$$\text{sif}(s) \geq \text{sif}(p) - d(s, q) \geq (1 - \epsilon^2)\text{sif}(p). \quad \square$$

Since we require the function  $\text{sif}$  to be a lower bound of the function  $\text{lfs}$  on the surface  $F$  (Equation 3.1)

$$\text{sif}(s) \leq (1 + \epsilon^2)\text{sif}(p) \leq (1 + \epsilon^2)\text{lfs}(p), \quad (3.2)$$

by Lemma 3.1.

## 3.4 Formula for the Cut Function

Let the input  $S$  be a set of sample points that lie near the surface  $F$ , and let  $\vec{n}_i$  be an approximate outside normal at sample point  $s_i \in S$ . For each sample point  $s_i$  we define a point function  $P_i(x)$  whose value is the signed distance from  $x$  to the tangent plane at  $s_i$ ,

### Chapter 3. Provably Better Moving Least Squares

---

$P_i(x) = (x - s_i) \cdot \vec{n}_i$ . The cut function  $I$  is a weighted average of the point functions.

$$I(x) = \frac{1}{\sum_{s_j \in S} W_j(x)} \sum_{s_i \in S} W_i(x) ((x - s_i) \cdot \vec{n}_i).$$

For each sample point  $s_i$  let  $a_i$  be the number of sample points inside a ball of radius  $\epsilon \text{sif}(s_i)$  centered at  $s_i$ , including  $s_i$  itself. Define a normalization factor associated with  $s_i$  as  $\epsilon^3 \text{sif}^3(s_i) / a_i$ . We use Gaussian weight functions along with the normalization factor,

$$W_i(x) = \frac{\epsilon^3 \text{sif}^3(s_i)}{a_i} e^{-\|x - s_i\|^2 / \sigma^2(x)},$$

to weight the average of the point functions.

The function  $\sigma(x)$  sets the width of the Gaussian weight functions. The value of  $\sigma(s)$  at each sample point  $s$  should be set to the sample point spacing near  $s$  ( $\approx \epsilon \text{sif}(s)$  from the sampling preconditions) to smoothly interpolate between point functions. Since sudden changes in the width of the Gaussian weight functions affect the gradient of the cut function, the function  $\sigma(x)$  should satisfy a Lipschitz condition. To compute such a function we define a new function that we call the *extended feature size* (or *efs*). This function *efs* is an extension of the *sif* values at each sample point to all points. The value of  $\text{efs}(x)$  is given by

$$\text{efs}(x) = \min_{s \in S} \{d(x, s) + \text{sif}(s)\} - d(x, s_n). \quad (3.3)$$

Here  $d(x, s_n)$  is the distance between  $x$  and the sample point  $s_n$  closest to  $x$ . The width of the Gaussian weight functions  $\sigma(x)$  is defined as

$$\sigma(x) = \epsilon \text{efs}(x).$$



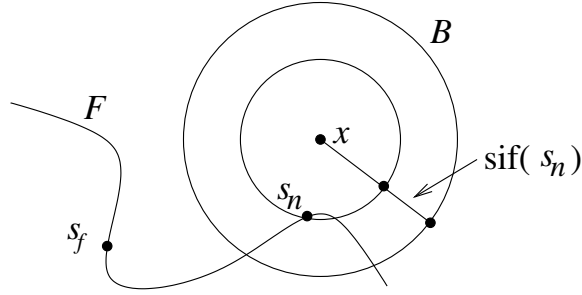


Figure 3.2: Sample points that influence the value of  $\text{efs}(x)$  are inside ball  $B$ . Sample point  $s_f$  has no effect on the value of  $\text{efs}(x)$  as it lies outside  $B$ .

### 3.4.1 Computing the Function $\text{efs}$

The function  $\text{efs}$  given by Equation 3.3 can be naively computed by iterating over all the sample points in the input point cloud. To derive a more efficient algorithm, consider a sample point  $s_f$  at a distance  $d(x, s_f) > d(x, s_n) + \text{sif}(s_n)$  from  $x$  as shown in Figure 3.2.

$$\begin{aligned} d(x, s_f) + \text{sif}(s_f) - d(x, s_n) &> d(x, s_n) + \text{sif}(s_n) + \text{sif}(s_f) - d(x, s_n) \\ &= \text{sif}(s_f) + \text{sif}(s_n). \end{aligned} \quad (3.4)$$

Consider the following upper bound on  $\text{efs}(x)$ .

$$\begin{aligned} \text{efs}(x) &\leq d(x, s_n) + \text{sif}(s_n) - d(x, s_n) \\ &\leq \text{sif}(s_n) \\ &< \text{sif}(s_f) + \text{sif}(s_n) \\ &< d(x, s_f) + \text{sif}(s_f) - d(x, s_n), \end{aligned}$$

by Equation 3.4. So the sample point that achieves the minimum in Equation 3.3 is guaranteed to be inside a ball  $B$  of radius  $d(x, s_n) + \text{sif}(s_n)$  around  $x$  as shown in Figure 3.2. The nearest sample point  $s_n$ , and the sample points inside  $B$  can be efficiently estimated using kd-trees [43].

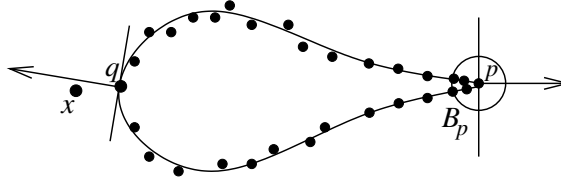


Figure 3.3: The sample points near  $q$  have large feature size and vote for the cut function to have a positive value at  $x$  whereas the sample points inside  $B_p$  have a small feature size and vote for the cut function to have a negative value at point  $x$ . Under adaptive sampling, the number of sample points inside  $B_p$  can be arbitrarily greater than the number of sample points near  $q$ .

### 3.4.2 The Normalization Factor

The curve shown in Figure 3.3 illustrates the necessity of the normalization factor used in our definition of the MLS surface. Consider the value of the cut function at the point  $x$ , which is clearly outside the sampled surface  $F$ . Let  $S$  be a point cloud in which the sampling is proportional to the feature size. Since the feature size is small near  $p$ , there can be an arbitrarily large number of sample points in a small ball  $B_p$  around  $p$ , and each sample point inside  $B_p$  votes for the value of the cut function  $I$  at  $x$  to be negative. The sample points that vote for a positive value at  $x$  are near point  $q$  which has a large feature size. Hence, regardless of the width of the Gaussian weight functions at point  $x$ , the sample points inside  $B_p$  can force the value of the cut function at  $x$  to be negative if the normalization factor is not used. For each sample point  $s_i \in B_p$ , the numerator  $\epsilon^3 \text{sif}^3(s_i)$  in the normalization factor of each sample point  $s_i \in B_p$  compensates for the large number of sample points inside  $B_p$ . With the normalization factor, we prove that sample points inside  $B_p$  have little effect on the value of the cut function at point  $x$ . The denominator in the normalization factor (which also appears in the definition of the uniform MLS surface) accounts for oversampling in the input point cloud.

Note that the denominator  $a_i$  in the normalization factor of  $s_i$  is the number of sample points in a small ball  $B$  of radius  $\epsilon \text{sif}(s_i)$  around  $s_i$ . This means that we expect the number of sample points inside  $B$  to be a good estimate of sample spacing near  $s_i$ . Adding

a large number of sample points outside  $B$ , and arbitrarily close to  $B$ , would make the normalization factor a poor estimate of the sample point spacing near  $s_i$ . This motivates the local uniformity requirements in our sampling preconditions which limit the change in sample point spacing inside a larger ball of radius  $3\epsilon \text{sif}(s_i)$  around  $s_i$ . The local uniformity preconditions guarantee that the normalization factor  $a_i$  is a good estimate of the sample point spacing near  $s_i$ .

### 3.5 Geometry of the Reconstructed Surface

In this section we study the geometric properties of the cut function. The main result is that the distance between the reconstructed surface  $U$  and  $F$  goes to zero with  $\epsilon$ . We also prove that the normals of the reconstructed surface converge to the normals of  $F$  with  $\epsilon$ .

The following three results are related to the Lipschitz conditions on the surface normals, and the feature sizes of the surface points. Recall that Amenta and Bern [3] proved the following Lipschitz condition on the surface normal with respect to the function  $\text{lfs}$ . We replace the function  $\text{lfs}$  with the function  $\text{sif}$  as it is a lower bound on  $\text{lfs}$  for points on the surface  $F$ .

**Theorem 3.2 (Amenta and Bern [3]).** *For points  $p, q$  on the surface  $F$  with  $d(p, q) \leq \alpha \min\{\text{sif}(p), \text{sif}(q)\}$ , for any  $\alpha < 1/3$ , the angle between the normals at  $p$  and  $q$  is at most  $\alpha/(1 - 3\alpha)$  radians.*

The following lemma shows a useful relationship between the local feature sizes of two surface points, and the  $\text{sif}$  values associated with two sample points.

**Lemma 3.3.** *For points  $p, q$  on the surface  $F$  with  $d(p, q) \leq \alpha \text{lfs}(p) + \beta \text{lfs}(q)$ , for  $\beta < 1$ ,  $(1 - \beta)\text{lfs}(q) \leq (1 + \alpha)\text{lfs}(p)$ . For sample points  $s, t \in S$  with  $d(s, t) \leq \alpha \text{sif}(s) + \beta \text{sif}(t)$ , for  $\beta < 1$ ,  $(1 - \beta)\text{sif}(s) \leq (1 + \alpha)\text{sif}(t)$ .*

**Proof.** Since the function  $\text{lfs}$  is 1-Lipschitz,

$$\begin{aligned} \text{lfs}(q) &\leq \text{lfs}(p) + d(p, q) \\ &\leq \text{lfs}(p) + \alpha \text{lfs}(p) + \beta \text{lfs}(q) \\ (1 - \beta)\text{lfs}(q) &\leq (1 + \alpha)\text{lfs}(p). \end{aligned}$$

Since the function  $\text{sif}$  is 1-Lipschitz like the function  $\text{lfs}$ , the same argument shows that  $(1 - \beta)\text{sif}(s) \leq (1 + \alpha)\text{sif}(t)$  for sample points  $s, t$ .  $\square$

In the following lemma we prove that the function  $\text{efs}$  is 2-Lipschitz.

**Lemma 3.4.** *The function  $\text{efs}$  is 2-Lipschitz.*

**Proof.** For two points  $x$  and  $y$ , and let  $s_x, s_y$  be the sample points closest to  $x, y$  respectively. By the definition of the function  $\text{efs}$  in Equation 3.3 we have

$$\begin{aligned} \text{efs}(x) + 2d(x, y) &\leq \min_{s \in S} \{d(x, s) + \text{sif}(s)\} - d(x, s_x) + 2d(x, y) \\ &\leq \min_{s \in S} \{d(x, s) + d(x, y) + \text{sif}(s)\} - d(x, s_x) + d(x, y) \\ &\leq \min_{s \in S} \{d(y, s) + \text{sif}(s)\} - (d(x, s_x) - d(x, y)) \\ &\leq \min_{s \in S} \{d(y, s) + \text{sif}(s)\} - d(y, s_x). \end{aligned} \tag{3.5}$$

Since  $s_y$  is sample point closest to  $y$ ,  $d(y, s_x) \geq d(y, s_y)$ . By Equation 3.5 we have

$$\text{efs}(x) + 2d(x, y) \leq \min_{s \in S} \{d(y, s) + \text{sif}(s)\} - d(y, s_y) = \text{efs}(y). \quad \square$$

Let  $F_{\text{out}}$  be the outside  $3\epsilon$ -offset surface of  $F$  that is obtained by moving each point  $p$  on  $F$  along the normal at  $p$  by a distance  $3\epsilon \text{sif}(p)$ . Similarly, let  $F_{\text{in}}$  be the inside  $3\epsilon$ -offset surface of  $F$ . Figure 3.4(a) shows a two-dimensional example. Dey and Goswami [34] (Lemma 3) prove that the  $3\epsilon$ -offset surfaces have the same topology as  $F$ .

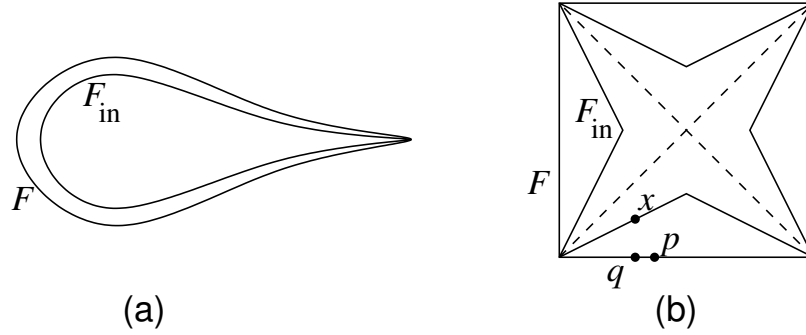


Figure 3.4: (a) An inside offset curve  $F_{\text{in}}$  whose distance to  $F$  varies with the feature size of points on  $F$ . (b) The dashed lines form the medial axis of the square. For a point  $p \in F$ , its closest point  $x$  on the offset curve  $F_{\text{in}}$  might not lie along the normal at  $p$ .

The  $3\epsilon$ -neighborhood is the region bounded by the inside and the outside offset surfaces. Note that unlike the neighborhood defined for the uniform case in Chapter 2, the neighborhood defined here adapts to the feature size. It is large near areas of large feature size and small when the surface feature size is small. The offset surface is always continuous, but might not be smooth as shown in Figure 3.4(b). The following result proves a lower bound on the distance from a point  $p \in F$  to the offset surfaces.

**Lemma 3.5.** *Let  $p$  be a point on the surface  $F$  and let  $x$  be the point on the  $\alpha$ -offset surface closest to  $p$ . For  $\alpha \leq 1/2$ ,  $d(x, p) \geq \alpha(1 - 2\alpha)\text{sif}(p)$ .*

**Proof.** Assume  $d(x, p) < \alpha \text{sif}(p)$ . Let  $q$  be the point on  $F$  closest to  $x$  as shown in Figure 3.4(b); then  $d(x, q) \leq d(x, p) < \alpha \text{sif}(p)$ . The distance from  $p$  to  $q$  can be written as  $d(p, q) \leq d(x, p) + d(x, q) \leq 2\alpha \text{sif}(p)$ . By Lemma 3.3 we have

$$\text{sif}(q) \geq (1 - 2\alpha)\text{sif}(p). \quad (3.6)$$

Since  $q$  is the point closest to  $x$  on the surface,  $xq$  is parallel to the normal of  $q$ . From the

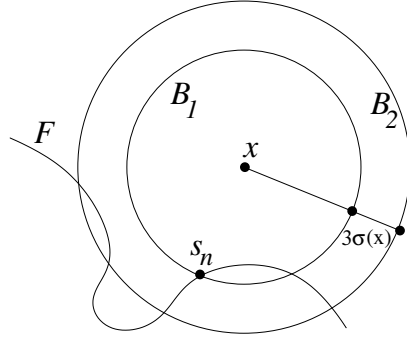


Figure 3.5: Point  $x$  is outside the  $3\epsilon$ -neighborhood and  $s_n$  is the sample point closest to  $x$ .

definition of the  $\alpha$ -offset surface,  $d(x, q) = \alpha \text{sif}(q)$ . By Equation 3.6 we have

$$d(x, p) \geq d(x, q) = \alpha \text{sif}(q) \geq \alpha(1 - 2\alpha)\text{sif}(p). \quad \square$$

Consider a point  $x$  whose closest sample point in  $S$  is  $s_n$ . Let  $B_1(x), B_2(x)$  be two balls centered at point  $x$  as shown in Figure 3.5. The radius of  $B_1(x)$  is  $d(x, s_n)$ ; a  $B_2(x)$  is a slightly larger ball whose radius is  $r_0 = d(x, s_n) + 3\epsilon \text{efs}(x)$ . Just like the analysis of the uniform MLS algorithm, our analysis of the adaptive MLS algorithm has two main ideas. First, the exponential decay of the Gaussian weight functions means that the cut function is mostly determined by sample points inside  $B_2(x)$ . Second, the cut function converges to the signed distance function  $\phi(x)$ . as  $\epsilon$  goes to zero, because the point functions associated with sample points inside  $B_2(x)$  converge to  $\phi(x)$ .

In Section 3.5.1, we study the properties of the cut function for points outside the  $3\epsilon$ -neighborhood, and prove that the cut function is non-zero outside the  $3\epsilon$ -neighborhood of  $F$  (Theorem 3.11). This means that the reconstructed surface  $U$  is restricted to the  $3\epsilon$ -neighborhood of  $F$ . To prove that the surface  $U$  is a manifold, we analyze the gradient of the cut function for points inside the  $3\epsilon$ -neighborhood in Section 3.5.2. These geometric results are used in Section 3.6 to show that  $U$  is isotopic to  $F$ .

### 3.5.1 Outside the $3\epsilon$ -neighborhood

Consider a point  $x$  outside the  $3\epsilon$ -neighborhood. Let  $p$  be the point closest to  $x$  on  $F$ . Let  $s_n$  be the sample point closest to  $x$ , and let  $p_n$  be the point on  $F$  closest to  $s_n$ . Since  $x$  is outside the  $3\epsilon$ -neighborhood,  $d(x, p_n) \geq 3\epsilon(1 - 6\epsilon)\text{efs}(p_n)$  from Lemma 3.5.

From the sampling preconditions  $d(s_n, p_n) \leq \epsilon^2\text{efs}(p_n)$ , and  $\text{efs}(p_n) \geq \text{efs}(s_n)/(1 + \epsilon^2)$ . Therefore

$$d(x, s_n) \geq d(x, p_n) - d(s_n, p_n) \geq (3\epsilon - 19\epsilon^2)\text{efs}(p_n) \geq \frac{3 - 19\epsilon}{1 + \epsilon^2}\epsilon \text{efs}(s_n).$$

The definition of the function  $\text{efs}$  guarantees that  $\text{efs}(x) \leq \text{efs}(s_n)$ , which gives the following lower bound on  $d(x, s_n)$  when point  $x$  outside the  $3\epsilon$ -neighborhood.

$$d(x, s_n) \geq \frac{3 - 19\epsilon}{1 + \epsilon^2}\epsilon \text{efs}(x). \quad (3.7)$$

Recall that  $\phi(x)$  is the signed distance from  $x$  to  $F$ . To prove our geometric results it is convenient to define two functions. For  $x$  outside the  $3\epsilon$ -neighborhood let  $\mu(x) = \phi(x)/|\phi(x)|$  be the sign function of  $F$ . When  $x$  is outside  $F_{\text{out}}$ ,  $\mu(x) = 1$ , and when  $x$  is inside  $F_{\text{in}}$   $\mu(x) = -1$ . Let  $E(x)$  be an error function that measures the difference between the cut function computed by the MLS algorithm and the signed distance to the nearest sample. This function  $E(x)$  is given by

$$\begin{aligned} E(x) &= I(x) - \mu(x)d(x, s_n) \\ &= \sum_{s_i \in S} \frac{W_i(x)}{\sum_{s_j \in S} W_j(x)} (P_i(x) - \mu(x)d(x, s_n)) \\ &\equiv \sum_{s_i \in S} \frac{W_i(x)}{\sum_{s_j \in S} W_j(x)} \zeta_i(x). \end{aligned} \quad (3.8)$$

The function  $\zeta_i(x)$  defined in Equation 3.8 measures the error in the point function of

sample point  $s_i$ . As the  $\epsilon$  parameter in our sampling preconditions decreases, the sampling is more dense, and has less noise. So the signed distance to the closest sample point  $\mu(x)d(x, s_n)$  will converge to the signed distance to  $F$  as  $\epsilon$  goes to zero. The error function  $E(x)$  indirectly measures the difference between  $I(x)$  and the signed distance function  $\phi(x)$ . Recall that we used a similar error function to analyze the uniform MLS algorithm in Chapter 2.

Define function  $\xi(x) = \sum_{s_i \in S} \zeta_i(x)W_i(x)$  to be the weighted combination of sample point error functions, and let the weight function  $W(x) = \sum_{s_i \in S} W_i(x)$  be the sum of all weight functions at  $x$ . Consider the contributions of sample points inside  $B_2(x)$  and outside  $B_2(x)$  to these two functions separately. Let  $\xi_{\text{in}}(x)$  and  $\xi_{\text{out}}(x)$  be the contributions to  $\xi(x)$  by sample points inside and outside  $B_2(x)$ . Similarly, let  $W_{\text{in}}(x)$  be the sum of weights of all sample points inside  $B_2(x)$ , and let  $W_{\text{out}}(x)$  be the sum of weights of all sample points outside  $B_2(x)$ .

The following lemma proves an upper bound and a lower bound on the error in the point functions of sample points inside  $B_2(x)$ . The result is similar to the result in Lemma 2.5 in Chapter 2. The main difference is that there is no lower bound on the feature size of sample points inside  $B_2(x)$ .

**Lemma 3.6.** *Let  $x$  be a point outside the  $3\epsilon$ -neighborhood and let  $s_n$  be the sample point closest to  $x$ . Let  $s_i$  be a sample point inside  $B_2(x)$  whose closest point on the surface is  $q$ . Then,*

$$\mu(x)\zeta_i(x) \leq 3\epsilon \text{efs}(x),$$

and,

$$\mu(x)\zeta_i(x) \geq -6\epsilon d(x, s_n) - \epsilon^2 \text{sif}(q) - 8\epsilon^2 \text{efs}(x).$$

**Proof.** Proving the upper bound is easy; since  $s_i$  is inside  $B_2(x)$ ,  $\mu(x)P_i(x) \leq d(x, s_i) \leq d(x, s_n) + 3\epsilon \text{efs}(x)$  which means  $\mu(x)\zeta_i(x) = \mu(x)(P_i(x) - \mu(x)d(x, s_i)) \leq 3\epsilon \text{efs}(x)$ .

To prove the lower bound consider point  $q$  which is closest to sample point  $s_i$  on the



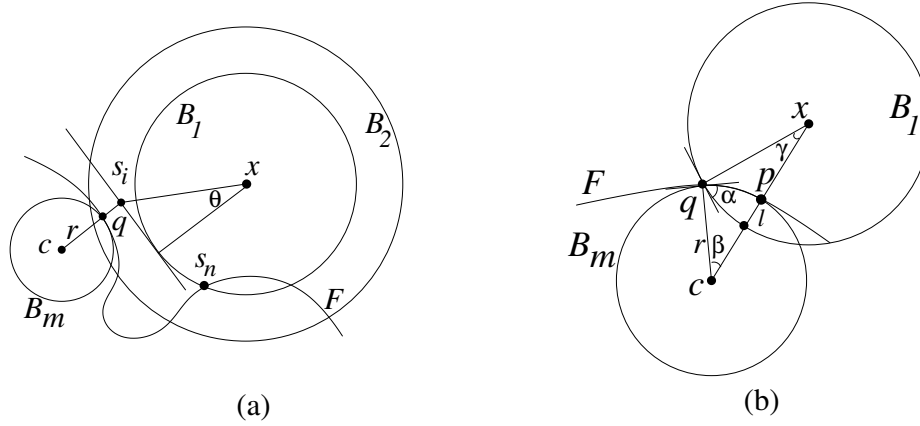


Figure 3.6: (a) Sample point  $s_i$  is inside  $B_2(x)$  and  $q$  is the point closest to  $s_i$  on  $F$ .  $B_m$  is a ball of radius  $\text{lfs}(q)$  touching  $q$  on the side of  $F$  opposite  $x$ . (b) The medial ball  $B_m$  centered at point  $c$  can only intersect  $B_1(x)$  at an angle  $\alpha$  given by  $4 \sin^2(\alpha/2) = \epsilon$ .

surface  $F$  as shown in Figure 3.6(a). Let  $\theta$  be the angle between  $xs_i$  and the normal  $\text{vec}n_q$  at  $q$ . The vector  $qs_i$  is parallel to  $\vec{n}_q$  because  $q$  is the point closest to  $s_i$  on  $F$ . By the sampling preconditions the angle between  $\vec{n}_q$  and the sample point normal  $\vec{n}_i$  at  $s_i$  is at most  $\epsilon$ . Therefore the point function associated with  $s_i$ ,  $P_i(x) = (x - s_i) \cdot \vec{n}_i \geq d(x, s_i) \cos(\theta + \epsilon)$ . We first show a lower bound on  $d(x, s_i) \cos \theta$  and then prove the desired lower bound on  $\mu(x)\zeta_i(x)$ .

Let  $B_m$  be a medial ball of radius  $r \geq \text{lfs}(q) \geq \text{sif}(q)$  touching  $q$  on the side of  $F$  opposite  $x$ . For two intersecting spheres, the angle of intersection is defined as the angle between the tangent planes of the two spheres at any point of intersection.

To prove a contradiction, assume that  $B_m$  intersects  $B_1(x)$  at an angle larger than  $\alpha$  given by  $4 \sin^2(\alpha/2) = \epsilon$ , as shown in Figure 3.6(b). Since points  $c$  and  $x$  are on opposite sides of the surface, the line segment  $cx$  intersects  $F$  at a point  $p$ . Without any loss of generality, we can assume that  $p$  is on  $B_m$ . Observe that  $B_1(x)$  does not contain any sample points because  $s_n$  is the sample point closest to  $x$ . The medial ball  $B_m$  does not contain any surface points, but there might sample points inside  $B_m$ . Therefore the distance from  $p$  to

its closest sample point is greater than the width  $l$  of the lune formed when  $B_m$  and  $B_1(x)$  intersect. Consider the angles  $\beta, \gamma$  shown in Figure 3.6(b). By the definition of the angle of intersection,  $\beta + \gamma = \alpha$ . So the width of the lune

$$l = 2r(1 - \cos \beta) \geq 2r(1 - \cos \alpha) = 4r \sin^2(\alpha/2).$$

Since  $c$  is a point on the medial axis,  $\text{sif}(p) \leq \text{lfs}(p) \leq d(p, c) = r$ . When  $4 \sin^2(\alpha/2) > \epsilon$ , the distance from  $p$  to its closest sample point is greater than  $\epsilon r \geq \epsilon \text{sif}(p)$ , which violates our sampling preconditions. Therefore

$$\sin^2(\alpha/2) \leq \epsilon/4. \quad (3.9)$$

The angle of intersection  $\alpha$  of two intersecting spheres  $B_m, B_1(x)$  of radii  $r, d(x, s_n)$  is given by

$$\cos \alpha = \frac{d^2(c, x) - r^2 - d^2(x, s_n)}{2rd(x, s_n)}. \quad (3.10)$$

The distance  $d(c, x)$  can be written as

$$d^2(c, x) = (d(x, s_i) \cos \theta + d(s_i, q) + r)^2 + d^2(x, s_i) \sin^2 \theta.$$

By our sampling preconditions,  $d(s_i, q) \leq \epsilon^2 \text{sif}(q)$ . Substituting into Equation 3.10 we have

$$\begin{aligned} r^2 + d^2(x, s_n) + 2rd(x, s_n) \cos \alpha &\leq (d(x, s_i) \cos \theta + \epsilon^2 \text{sif}(q) + r)^2 + d^2(x, s_i) \sin^2 \theta. \\ d(x, s_i) \cos \theta &\geq \frac{rd(x, s_n) \cos \alpha}{\epsilon^2 \text{sif}(q) + r} - \frac{d^2(x, s_i) - d^2(x, s_n)}{2(\epsilon^2 \text{sif}(q) + r)} \\ &\quad - \frac{\epsilon^2 \text{sif}(q)(\epsilon^2 \text{sif}(q) + 2r)}{2(\epsilon^2 \text{sif}(q) + r)}. \end{aligned} \quad (3.11)$$

Since  $s_i$  is inside  $B_2(x)$   $d(x, s_i) \leq d(x, s_n) + 3\epsilon \text{efs}(x)$ , and since  $B_m$  is medial ball touching

### Chapter 3. Provably Better Moving Least Squares

---

$q, r \geq \text{lfs}(q) \geq \text{sif}(q)$ . Using standard trigonometric formulas, it is easy to show  $\cos(\theta + \epsilon) \geq \cos \theta - \epsilon$ . Substituting into Equation 3.11 we have

$$\begin{aligned} \mu(x)P_i(x) &\geq d(x, s_i)(\cos \theta - \epsilon) \\ &\geq \frac{d(x, s_n) \cos \alpha}{1 + \epsilon^2} - \frac{3\epsilon \text{efs}(x)(2d(x, s_n) + 3\epsilon \text{efs}(x))}{2(\epsilon^2 \text{sif}(q) + r)} \\ &\quad - \epsilon^2 \text{sif}(q) - \epsilon(d(x, s_n) + 3\epsilon \text{efs}(x)). \end{aligned} \quad (3.12)$$

To prove a lower bound on  $\text{sif}(q)$  in terms of  $\text{efs}(x)$  consider the definition of the function  $\text{efs}$ .

$$\text{efs}(x) \leq \text{sif}(s_i) + d(x, s_i) - d(x, s_n). \quad (3.13)$$

Since sample point  $s_i$  is inside  $B_2(x)$ ,  $d(x, s_i) - d(x, s_n) \leq 3\epsilon \text{efs}(x)$ . Substituting into Equation 3.13

$$\text{sif}(s_i) \geq (1 - 3\epsilon)\text{efs}(x). \quad (3.14)$$

Recall that in Lemma 3.1 we showed that  $\text{sif}(s_i)$  is a good approximation to  $\text{sif}(q)$ .

$$\text{sif}(q) \geq \frac{\text{sif}(s_i)}{1 + \epsilon^2} \geq \frac{(1 - 3\epsilon)\text{efs}(x)}{1 + \epsilon^2},$$

by Equation 3.14. Substituting into Equation 3.12

$$\begin{aligned} \mu(x)\zeta_i(x) &= \mu(x)(P_i(x) - \mu(x)d(x, s_n)) \\ &\geq \frac{-d(x, s_n)(1 - \cos \alpha + 5\epsilon)}{1 + \epsilon^2} - \epsilon^2 \text{sif}(q) - 8\epsilon^2 \text{efs}(x) \\ &= \frac{-d(x, s_n)(\sin^2(\alpha/2) + 5\epsilon)}{1 + \epsilon^2} - \epsilon^2 \text{sif}(q) - 8\epsilon^2 \text{efs}(x). \end{aligned}$$

Substituting the upper bound on  $\sin^2(\alpha/2)$  from Equation 3.9 and simplifying using  $\epsilon \leq 1/150$  we have

$$\mu(x)\zeta_i(x) > -6\epsilon d(x, s_n) - \epsilon^2 \text{sif}(q) - 8\epsilon^2 \text{efs}(x). \quad \square$$

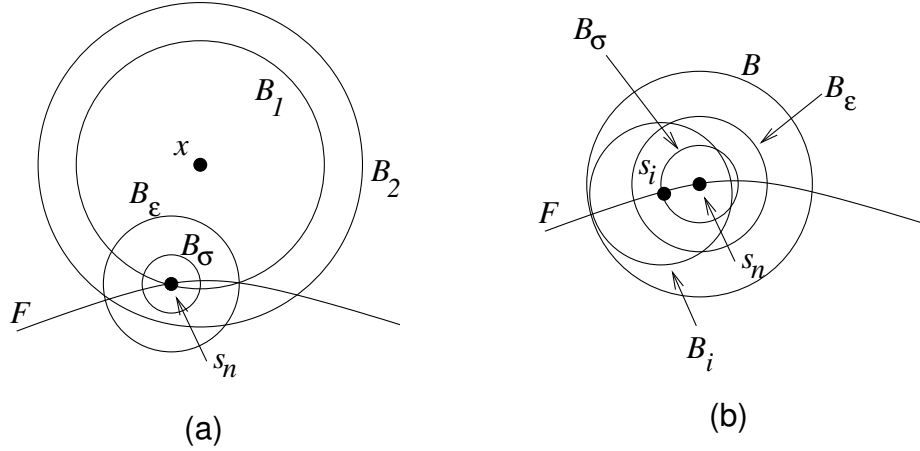


Figure 3.7: For point  $x$ , the closest sample point is  $s_n$ , and  $B_2(x)$  is a ball of radius  $d(x, s_n) + 3\epsilon \text{efs}(x)$  at  $x$ . The ball  $B_\sigma$  of radius  $\epsilon \text{efs}(x)$  at point  $x$  is completely contained inside  $B_2(x)$ .

Since the signed distance to the nearest sample point  $\mu(x)d(x, s_n)$  converges to  $\phi(x)$  as  $\epsilon$  goes to zero, the result in Lemma 3.6 shows that for each sample point  $s_i \in B_2(x)$ ,  $P_i(x)$  converges to  $\phi(x)$  as  $\epsilon$  approaches zero. Unlike sample points inside  $B_2(x)$ , the error functions of sample points outside  $B_2(x)$  can be arbitrarily large. In the remainder of this section, we show that the error in the sample points outside  $B_2(x)$  does not adversely affect the cut function.

In Lemma 3.7 we derive a lower bound on the weight of sample points inside  $B_2(x)$ . Lemma 3.7 is the only result which depends on the local uniformity requirements in our sampling preconditions.

**Lemma 3.7.** *Let  $x$  be a point outside the  $3\epsilon$ -neighborhood, and let  $W_{\text{in}}(x)$  be the sum of the weights of all sample points inside  $B_2(x)$ . Then*

$$W_{\text{in}}(x) \geq \frac{1}{10} \sigma^3(x) e^{-(d(x, s_n) + \sigma(x))^2 / \sigma^2(x)}.$$

**Proof.** Let  $B_\sigma$  be a ball of radius  $\sigma(x) = \epsilon \text{efs}(x)$  centered at the sample point  $s_n$  closest to  $x$  as shown in Figure 3.7(a). Let  $\alpha$  be the number of sample points inside  $B_\sigma$ , and

### Chapter 3. Provably Better Moving Least Squares

---

let  $\beta$  be the number of sample points inside a ball  $B_\epsilon$  of radius  $\epsilon \text{sif}(s_n)$  at  $s_n$ . Since  $\text{efs}(x) \leq \text{sif}(s_n)$ , the ball  $B_\sigma$  is contained inside  $B_\epsilon$ .

Consider a sample point  $s_i$  inside  $B_\sigma$  as shown in Figure 3.7. Recall that  $a_i$  is the number of sample points inside a ball  $B_i$  of radius  $\epsilon \text{sif}(s_i)$  around  $s_i$ . Since  $d(s_n, s_i) \leq \epsilon \text{efs}(x) \leq \epsilon \text{sif}(s_n)$ ,  $\text{sif}(s_i) \leq (1 + \epsilon)\text{sif}(s_n)$  by Lemma 3.3. Therefore all sample points inside  $B_i$  are inside a ball  $B$  of radius  $\epsilon(2 + \epsilon)\text{sif}(s_n)$  around  $s_n$  as shown in Figure 3.7(b). Let  $\gamma$  be the number of samples inside  $B$ . Recall that in our sampling preconditions, we required the sampling near  $s_n$  to be uniform. So the ratio of the number of sample points inside  $B$  to the number of sample points inside  $B_\epsilon$  is less than the ratio of their volumes.

$$\frac{a_i}{\beta} \leq \frac{\gamma}{\beta} \leq \frac{\epsilon^3(2 + \epsilon)^3 \text{sif}^3(s_n)}{\epsilon^3 \text{sif}^3(s_n)} = (2 + \epsilon)^3. \quad (3.15)$$

As the weight of all sample points inside  $B_\sigma$  is a lower bound for  $W_{\text{in}}(x)$  we have

$$W_{\text{in}}(x) \geq \sum_{s_i \in B_\sigma} \frac{\epsilon^3 \text{sif}^3(s_i)}{a_i} e^{-(d(x, s_n) + \sigma(x))^2 / \sigma^2(x)}. \quad (3.16)$$

Since  $d(s_i, s_n) \leq \epsilon \text{sif}(s_n)$ ,  $\text{sif}(s_i) \geq (1 - \epsilon)\text{sif}(s_n)$  by Lemma 3.3. Substituting into Equation 3.16 we have

$$\begin{aligned} W_{\text{in}}(x) &\geq \epsilon^3(1 - \epsilon)^3 \text{sif}^3(s_n) e^{-(d(x, s_n) + \sigma(x))^2 / \sigma^2(x)} \sum_{s_i \in B_\sigma} \frac{1}{a_i} \\ &\geq \epsilon^3(1 - \epsilon)^3 \text{sif}^3(s_n) e^{-(d(x, s_n) + \sigma(x))^2 / \sigma^2(x)} \sum_{s_i \in B_\sigma} \frac{1}{(2 + \epsilon)^3 \beta}, \end{aligned}$$

by Equation 3.15. Recall that  $B_\sigma$  contains  $\alpha$  sample points. Therefore

$$W_{\text{in}}(x) \geq \frac{\alpha \epsilon^3(1 - \epsilon)^3 \text{sif}^3(s_n)}{(2 + \epsilon)^3 \beta} e^{-(d(x, s_n) + \sigma(x))^2 / \sigma^2(x)}. \quad (3.17)$$

Again from the uniformity conditions in our sampling requirements, the ratio of the number

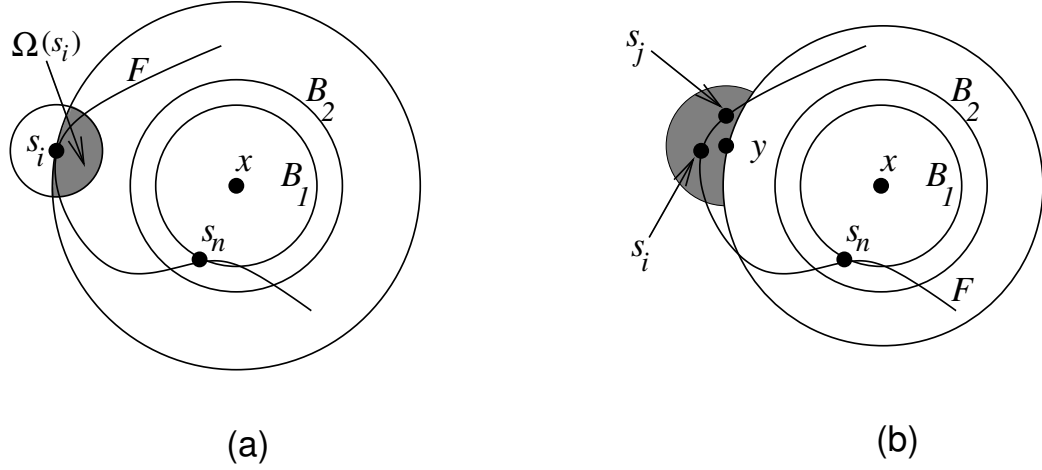


Figure 3.8: (a) Sample point  $s_i$  contributes to the density functions inside the shaded region  $\Omega(s_i)$ . Each point inside  $\Omega(s_i)$  is closer than  $s_i$  to the point  $x$ . (b) Each sample point that contributes to the density function at  $y$  is contained in ball around  $y$  and is farther from  $x$  than  $y$ .

of sample points inside  $B_\epsilon$  to the number of sample points inside a smaller ball  $B_\sigma$  is less than the ratio of their volumes,  $\frac{\beta}{\alpha} < \frac{\text{sif}^3(s_n)}{\text{efs}^3(x)}$ . Substituting into Equation 3.17, and simplifying using the inequality  $\epsilon \leq 1/150$  we have

$$W(x) > W_{\text{in}}(x) > \frac{\epsilon^3(x)\text{efs}^3(x)}{10} e^{-(d(x,s_n)+\sigma(x))^2/\sigma^2(x)} = \frac{\sigma^3(x)}{10} e^{-(d(x,s_n)+\sigma(x))^2/\sigma^2(x)}.$$

□

To prove an upper bound on  $W_{\text{out}}(x)$  we define radial density functions  $\lambda_k(x, r)$  such that

$$\int_{\mathbb{R}^3} \lambda_k(x, r) dr \geq \sum_{s_i \notin B_2(x)} d^k(x, s_i) W_i(x),$$

for  $k = 0, 1, \dots, 4$ . The integral of  $\lambda_0(x, r)$  is an upper bound on  $W_{\text{out}}(x)$ . Functions that involve higher values of  $k$  appear later in the analysis of the cut function and its gradient. With each sample point  $s_i$  we associate a region of space  $\Omega(s_i)$  inside which  $s_i$  will contribute to the density functions. This region  $\Omega(x, s_i)$  is defined as the lune formed by intersecting a ball of radius  $2\epsilon \text{sif}(s_i)/5$  centered at  $s_i$ , and a ball of radius  $d(x, s_i)$  centered

at  $x$  as shown in Figure 3.8(a). In the following lemma we prove that  $x$  is outside  $\Omega(s_i)$ .

**Lemma 3.8.** *Let  $s_i$  be a sample point outside  $B_2(x)$ . Then  $d(x, s_i) > 2\epsilon \text{sif}(s_i)/5$ .*

**Proof.** To prove a contradiction assume that  $d(x, s_i) \leq 2\epsilon \text{sif}(s_i)/5$ . By the definition of the function  $\text{efs}$ ,  $\text{efs}(s_i) = \text{sif}(s_i)$  at each sample point  $s_i$ . Since the function  $\text{efs}$  is 2-Lipschitz

$$\begin{aligned} \text{sif}(s_i) &= \text{efs}(s_i) \\ &\leq \text{efs}(x) + 4\epsilon \text{sif}(s_i)/5. \\ \text{sif}(s_i) &\leq \frac{\text{efs}(x)}{1 - \frac{4\epsilon}{5}}. \end{aligned} \tag{3.18}$$

Therefore the distance from  $x$  to  $s_i$  is given by

$$d(x, s_i) \leq \frac{2\epsilon}{5} \text{sif}(s_i) \leq \frac{4\epsilon}{5(1 - \frac{4\epsilon}{5})} \text{efs}(x).$$

For values of  $\epsilon \leq 1/150$ ,  $\frac{4\epsilon}{5(1 - \frac{4\epsilon}{5})} \text{efs}(x) < 3\epsilon \text{efs}(x)$ . This is clearly a contradiction because for  $s_i$  outside  $B_2(x)$ ,  $d(x, s_i) \geq r_0 \geq 3\epsilon \text{efs}(x)$ .  $\square$

For sample point  $s_i$  define a density function  $\gamma_k^i$  whose value inside  $\Omega(x, s_i)$  is given by dividing of  $d^k(x, s_i)e^{-d^2(x, s_i)/\sigma^2(x)}$  by the volume of  $\Omega(x, s_i)$ . It is easy to show that the volume of  $\Omega(x, s_i)$  is greater than  $\epsilon^3 \text{sif}^3(s_i)/12$  since  $x$  is outside  $\Omega(x, s_i)$  by Lemma 3.8. Therefore  $\gamma_k^i(x, y) \leq 12d^k(x, s_i)e^{-d(x, s_i)^2/\sigma^2(x)}/a_i$ . For  $y \notin \Omega(s_i)$ ,  $\gamma_k^i(x, y) = 0$ .

From the definition of  $\gamma_k^i$ ,  $\int_{\mathbb{R}^3} \gamma_k^i(x, y)dy \geq d^k(x, y)W_i(x)$ . The density function  $\gamma_k$  that accounts for the contributions of all sample points outside  $B_2(x)$  is given by

$$\gamma_k(x, y) = \sum_{s_i} \gamma_k^i(x, y). \tag{3.19}$$

In the following lemma we prove an upper bound on the value of the density functions  $\gamma_k$ .

**Lemma 3.9.** *Let  $r_0$  is the radius of  $B_2(x)$ . For a point  $y \notin B_2(x)$*

$$\gamma_k(x, y) \leq 12d^k(x, y)e^{-d^2(x, y)/\sigma^2(x)},$$

*and for  $y \in B_2(x)$ ,*

$$\gamma_k(x, y) \leq 12r_0^k e^{-r_0^2/\sigma^2(x)}.$$

**Proof.** Let  $y$  be outside  $B_2(x)$  and let  $S_y$  be the set of sample points  $s_i$  such that  $y \in \Omega(x, s_i)$ . Assume that the set  $S_y$  it not empty. By the definition of  $\Omega(x, s_i)$ , each sample point  $s_i \in S_y$  is farther from  $x$  than  $y$ , and is contained in a small ball around  $y$  a shown in Figure 3.8(b). By Equation 3.19 the density function at point  $y$  is given by

$$\gamma_k(x, y) \leq \sum_{s_i \in S_y} \frac{12}{a_i} d^k(x, s_i) e^{-d^2(x, s_i)/\sigma^2(x)}. \quad (3.20)$$

Since  $s_i$  is outside  $B_2(x)$ ,  $d(x, s_i) \geq d(x, s_n) + 3\sigma(x) > 3\sigma(x)$ . For values of  $r > 3\sigma(x)$ , the function  $r^k e^{-r^2/\sigma^2(x)}$  ( $0 \leq k \leq 4$ ) is a monotonically decreasing function of  $r$ . Therefore  $d^k(x, s_i) e^{-d^2(x, s_i)/\sigma^2(x)} \leq d^k(x, y) e^{-d^2(x, y)/\sigma^2(x)}$ . Substituting into Equation 3.20 we have

$$\gamma_k(x, y) \leq \sum_{s_i \in S_y} \frac{12}{a_i} d^k(x, y) e^{-d^2(x, y)/\sigma^2(x)}.$$

Consider sample points  $s_i, s_j \in S_y$  as shown in Figure 3.8(b). By the definition of the regions  $\Omega(x, s_i)$  and  $\Omega(x, s_j)$ ,  $d(s_i, y) \leq 2\epsilon \text{sif}(s_i)/5$ , and  $d(s_j, y) \leq 2\epsilon \text{sif}(s_j)/5$ . Therefore



## Chapter 3. Provably Better Moving Least Squares

---

the distance between  $s_i$  and  $s_j$  can be written as

$$\begin{aligned} d(s_i, s_j) &\leq d(s_i, y) + d(y, s_j) \\ &\leq \frac{2}{5}\epsilon \text{sif}(s_i) + \frac{2}{5}\epsilon \text{sif}(s_j). \end{aligned} \quad (3.21)$$

By Lemma 3.3 we have  $\text{sif}(s_j) \leq (1 + 2\epsilon/5)\text{sif}(s_i)/(1 - 2\epsilon/5)$ . Substituting into Equation 3.21 we have

$$d(s_i, s_j) \leq \left( \frac{2}{5} + \frac{2(1 + 2\epsilon/5)}{5(1 - 2\epsilon/5)} \right) \epsilon \text{sif}(s_i) < \epsilon \text{sif}(s_i).$$

Hence every sample point  $s_j \in S_y$  is inside a ball of radius  $\epsilon \text{sif}(s_i)$  centered at  $s_i$ . Recall that  $a_i$  is the number of sample points in a ball of radius  $\epsilon \text{sif}(s_i)$  around  $s_i$ . which means  $a_i \geq |S_y|$  for each  $s_i \in S_y$ . The density function  $\gamma_k(x, y)$  can be written as

$$\gamma_k(x, y) \leq d^k(x, y)e^{-d^2(x, y)/\sigma^2(x)} \sum_{s_i \in S_y} \frac{12}{|S_y|} \leq 12d^k(x, y)e^{-d^2(x, y)/\sigma^2(x)}. \quad (3.22)$$

Consider point  $y \in B_2(x)$ . The sample points that contribute to the density functions  $\lambda_k$  are all outside  $B_2(x)$ , which means,  $d(x, s_i) \geq r_0$  for all  $s_i \in S_y$ . The proof for sample points outside  $B_2(x)$  can be easily adapted to show that  $\gamma_k(x, y) \leq 12r_0^k e^{-r_0^2/\sigma^2(x)}$ .  $\square$

For a point  $x$ , define radial density function  $\lambda_k(x, r)$  as

$$\lambda_k(x, r) = 12r^k e^{-r^2/\sigma^2(x)}. \quad (3.23)$$

From the result in Lemma 3.9,

$$\int_0^\infty 4\pi r^2 \lambda_k(x, r) dr \geq \int_{\mathbb{R}^3} \gamma_k(x, y) dy \geq \sum_{s_i \notin B_2(x)} \frac{d^k(x, s_i)}{a_i} e^{-d^2(x, s_i)/\sigma^2(x)}. \quad (3.24)$$

In the following lemma we prove that  $W_{\text{out}}(x) \ll W(x)$ , and that the error in the cut function is small.

**Lemma 3.10.** *Let  $x$  be a point outside the  $3\epsilon$ -neighborhood. Let  $W_{\text{out}}(x)$  be the sum of weights of sample points outside  $B_2(x)$ , and  $E_{\text{out}}(x)$  be the error in the cut function due to sample points outside  $B_2(x)$ . For a small constant  $c_1 = 0.002$ ,  $\frac{W_{\text{out}}(x)}{W(x)} < c_1$ , and  $|E_{\text{out}}(x)| < c_1\epsilon$ .*

**Proof.** Integrating the density function  $\lambda_0$  over  $\mathbb{R}^3$  gives an upper bound on  $W_{\text{out}}(x)$ . By the definition of  $\lambda_0$  in Equation 3.23 we have

$$\begin{aligned} W_{\text{out}}(x) &\leq \int_{r=0}^{r=\infty} 4\pi r^2 \lambda_0(x, r) dr \\ &= \int_{r=0}^{r=r_0} 4\pi r^2 \lambda_0(x, r) dr + \int_{r=r_0}^{r=\infty} 4\pi r^2 \lambda_0(x, r) dr \\ &\leq 48\pi \left( \frac{r_0^3}{3} e^{-r_0^2/\sigma^2(x)} + \int_{r_0}^{\infty} r^2 e^{-r^2/\sigma^2(x)} dr \right) \\ &< 48\pi \left( \frac{r_0^3}{3} e^{-r_0^2/\sigma^2(x)} + \int_{r_0}^{\infty} r^2 e^{-r_0 r/\sigma^2(x)} dr \right). \end{aligned}$$

Substituting the lower bound on  $W(x)$  in Lemma 3.7 into the above integral

$$\frac{W_{\text{out}}(x)}{W(x)} \leq \frac{480\pi}{\sigma^3(x)} \left( \frac{r_0^3}{3} + r_0\sigma^2(x) + \frac{2}{r_0}\sigma^4(x) + \frac{2}{r_0^3}\sigma^6(x) \right) e^{-2(2r_0-2\sigma(x))/\sigma(x)}.$$

From Equation 3.7  $d(x, s_n) \geq (3 - 19\epsilon)\epsilon \text{fs}(x)/(1 + \epsilon^2) > 2.5\sigma(x)$ , which means  $r_0 > 5.5\sigma(x)$ . For  $r_0 > 5.5\sigma(x)$ , the ratio  $W_{\text{out}}(x)/W(x)$  is a decreasing function of  $r$  that is maximized when  $r_0 = 5.5\sigma(x)$ .

$$\begin{aligned} \frac{W_{\text{out}}(x)}{W(x)} &\leq \frac{480\pi}{\sigma^3(x)} \left( \frac{(5.5\sigma(x))^3}{3} + (5.5\sigma(x))\sigma^2(x) + \frac{2\sigma^4(x)}{5.5\sigma(x)} + \frac{2\sigma^6(x)}{(5.5\sigma(x))^3} \right) e^{-18} \\ &< c_1. \end{aligned} \tag{3.25}$$

To prove the desired upper bound on the error function, consider the contribution of sample

points outside  $B_2(x)$  to the error function  $E_{\text{out}}(x)$ . Recall that  $\xi_{\text{out}}(x)$  is the sum of the error functions of sample points outside  $B_2(x)$ . For sample point  $s_i$ ,  $|P_i(x)| \leq d(x, s)$ , and  $d(x, s_n) \leq d(x, s_i)$ . Therefore the magnitude of the error function associated with sample  $s_i$ ,  $|\zeta_i(x)| \leq 2d(x, s_i)$ . The contribution of sample point  $s_i$  to  $\xi_{\text{out}}(x)$  is less than  $2d(x, s_i)W_i(x)$ . By Equation 3.24, the integral of the radial density function  $\lambda_1$  over  $\mathbb{R}^3$  is an upper bound on  $|\xi_{\text{out}}(x)|$ .

$$\begin{aligned} |\xi_{\text{out}}(x)| &\leq 2 \int_{r=0}^{r=\infty} 4\pi r^2 \lambda_1(x, r) dr \\ &\leq 96\pi \left( \frac{r_0^3}{3} + r_0^2 \sigma^2(x) + 3\sigma^4(x) + \frac{6}{r_0^2} \sigma^6(x) + \frac{6}{r_0^4} \sigma^8(x) \right) e^{-r_0^2/\sigma^2(x)}. \end{aligned}$$

Substituting the lower bound on  $W(x)$  from Lemma 3.7

$$\begin{aligned} |E_{\text{out}}(x)| &= \frac{|\xi_{\text{out}}(x)|}{W(x)} \\ &\leq \frac{960\pi}{\sigma^3(x)} \left( \frac{r_0^4}{3} + r_0^2 \sigma^2(x) + 3\sigma^4(x) + \frac{6}{r_0^2} \sigma^6(x) \right. \\ &\quad \left. + \frac{6}{r_0^4} \sigma^8(x) \right) e^{-2(2r_0 - 2\sigma(x))/\sigma(x)}. \end{aligned} \tag{3.26}$$

The error  $|E_{\text{out}}(x)|$  is maximized when  $r_0 = 5.5\sigma(x)$ . Substituting  $r_0 = 5.5\sigma(x)$  into the above inequality it is easy to verify that  $|E_{\text{out}}(x)| \leq c_1\epsilon$ .  $\square$

We now have all the tools required to prove the main geometric result for points outside the  $3\epsilon$ -neighborhood: the cut function  $I(x)$  is non-zero outside the  $3\epsilon$ -neighborhood.

**Theorem 3.11.** *For each point  $x$  outside  $F_{\text{out}}$ ,  $I(x) > 0$  and for each point  $y$  inside  $F_{\text{in}}$ ,  $I(y) < 0$ .*

**Proof.** Consider point  $x$  outside  $F_{\text{out}}$ . The value of the cut function at point  $x$  is

$$I(x) = d(x, s_n) + E_{\text{in}}(x) + E_{\text{out}}(x) > d(x, s_n) + E_{\text{in}}(x) - c_1\epsilon, \tag{3.27}$$

by Lemma 3.10. Lemma 3.6 gives us a bound on  $E_{\text{in}}(x)$ .

$$\begin{aligned}
 E_{\text{in}}(x) &= \frac{1}{W(x)} \sum_{s_i \in B_2(x)} |\zeta_i(x)| W_i(x) & (3.28) \\
 &\geq \min\{P_i(x) | s_i \in B_2(x)\} \frac{\sum_{s_i \in B_2(x)} W_i(x)}{W(x)} \\
 &> \min\{-6\epsilon d(x, s_n) - \epsilon^2 \text{sif}(q) - 8\epsilon^2 \text{efs}(x) | s_i \in B_2(x)\}. & (3.29)
 \end{aligned}$$

Recall that  $q$  is the point on  $F$  closest to  $s_i$ . To prove an upper bound on  $\epsilon^2 \text{sif}(q)$  in terms of  $\sigma(x)$  consider the distance from point  $x$  outside the  $3\epsilon$ -neighborhood to point  $q$  on the surface.

$$d(x, q) \geq 3\epsilon(1 - 6\epsilon) \text{sif}(q), \quad (3.30)$$

by Lemma 3.5. Since  $q$  is at a distance less than  $\epsilon^2 \text{sif}(q)$  from sample point  $s_i$  inside  $B_2(x)$ ,  $d(x, q) \leq d(x, s_i) + \epsilon^2 \text{sif}(q) \leq d(x, s_n) + 3\sigma(x) + \epsilon^2 \text{sif}(q)$ . Substituting into Equation 3.30 we have

$$\begin{aligned}
 d(x, s_n) + 3\epsilon \text{efs}(x) + \epsilon^2 \text{sif}(q) &\geq 3\epsilon(1 - 6\epsilon) \text{sif}(q). \\
 \epsilon^2 \text{sif}(q) &\leq \frac{\epsilon}{3 - 19\epsilon} (d(x, s_n) + 3\sigma(x)).
 \end{aligned}$$

Substituting the above upper bound on  $\epsilon^2 \text{sif}(q)$  into Equation 3.29 we have

$$E_{\text{in}}(x) > - \left( d(x, s_n) \left( 6\epsilon + \frac{\epsilon}{3 - 19\epsilon} \right) - \frac{3\epsilon^2}{1 - 19\epsilon} - 8\epsilon^2 \right) \text{efs}(x).$$

Substituting the lower bound on  $d(x, s_n)$  from Equation 3.7, and the above lower bound

into Equation 3.27

$$\begin{aligned}
 I(x) &> d(x, s_n) - |E_{\text{in}}(x)| - c_1 \epsilon \text{efs}(x) \\
 &\geq \left( \frac{3 - 19\epsilon}{1 + \epsilon^2} \left( 1 - 6\epsilon - \frac{\epsilon}{3 - 19\epsilon} \right) - \frac{3\epsilon}{1 - 19\epsilon} - 8\epsilon - c_1 \right) \epsilon \text{efs}(x) \\
 &> 0,
 \end{aligned}$$

for  $\epsilon \leq 1/150$ . A similar argument proves that the cut function is negative at any point  $y$  inside  $F_{\text{in}}$ .  $\square$

Theorem 3.11 proves that the cut function  $I$  does not have any spurious zero crossings far away from the sample points, and shows that the Hausdorff distance between  $F$  and  $U$  converges to zero with  $\epsilon$ . In Section 3.5.2 we derive tighter bounds and show that the rate of convergence of the Hausdorff distance between  $U$  and  $F$  is  $O(\epsilon^2)$ .

### 3.5.2 The $3\epsilon$ -neighborhood

In this section we analyze the cut function inside the  $3\epsilon$ -neighborhood. The main result is that the reconstructed surface  $U$  is a manifold that converges to the sampled surface as  $\epsilon$  goes to zero. We study the of the gradient of the cut function to prove that  $U$  is a manifold. The gradient analysis also shows that the normals of the reconstructed surface converge to the normals of  $F$  as  $\epsilon$  goes to zero.

For  $x$  inside the  $3\epsilon$ -neighborhood, all sample points that have a large influence on the value of the cut function at  $x$  have approximately the same feature size. These sample points are also sampled uniformly. As a result, all the proofs in this section are an easy extension of the proofs in Section 2.4.2 in Chapter 2. The main difference is in proving that samples far away from  $x$  have little effect on the value of the cut function at  $x$ .

Inside the  $3\epsilon$ -neighborhood we modify the error function  $E(x)$  and define it to be the difference between the cut function and the signed distance function  $\phi(x)$ . Similarly, the

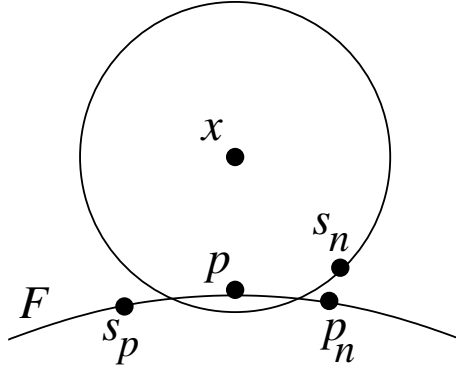


Figure 3.9: Point  $x$  is inside the  $3\epsilon$ -neighborhood and  $p$  is the point closest to  $x$  on the surface  $F$ . The sample point closest to  $x$  is  $s_n$  and  $p_n$  is the point on  $F$  closest to  $s_n$ .

error function  $\zeta_i$  associated with sample point  $s_i$  measures the difference between  $P_i(x)$  and  $\phi(x)$ .

$$E(x) = I(x) - \phi(x) = \frac{1}{W(x)} \sum_{s_i \in S} (P_i(x) - \phi(x))W_i(x) \equiv \frac{1}{W(x)} \sum_{s_i \in S} \zeta_i(x)W_i(x).$$

Consider a point  $x$  inside the  $3\epsilon$ -neighborhood whose closest point on the surface is  $p$ . The following lemma proves an upper bound and a lower bound on  $\text{efs}(x)$  in terms of  $\text{sif}(p)$ .

**Lemma 3.12.** *Let  $x$  be a point inside the  $3\epsilon$ -neighborhood whose closest point on the surface is  $p$ . Then*

$$(1 - 16\epsilon)\text{sif}(p) \leq \text{efs}(x) \leq (1 + 8\epsilon)\text{sif}(p).$$

**Proof.** Let  $s_n$  be the sample point closest to  $x$  and let  $p_n$  be the surface point closest to  $s_n$  as shown in Figure 3.9. By the sampling preconditions  $\text{efs}(s_n) \leq (1 + \epsilon^2)\text{sif}(p_n)$ , and by the definition of the function  $\text{efs}$ ,  $\text{efs}(x) \leq \text{efs}(s_n)$ . Therefore

$$\text{efs}(x) \leq \text{efs}(s_n) \leq (1 + \epsilon^2)\text{sif}(p_n). \quad (3.31)$$

By the sampling preconditions, there is a sample point  $s_p$  within a distance  $\epsilon \text{sif}(p)$  from  $p$ .

## Chapter 3. Provably Better Moving Least Squares

---

Since  $s_n$  is the sample point closest to  $x$

$$d(x, s_n) \leq d(x, p) + \epsilon \operatorname{sif}(p) \leq 4\epsilon \operatorname{sif}(p). \quad (3.32)$$

The distance between  $p, p_n$  is given by

$$\begin{aligned} d(p, p_n) &\leq d(p, x) + d(x, s_n) + d(s_n, p_n) \\ &\leq 3\epsilon \operatorname{sif}(p) + 4\epsilon \operatorname{sif}(p) + \epsilon^2 \operatorname{sif}(p_n) \\ &\leq 7\epsilon \operatorname{sif}(p) + \epsilon^2 \operatorname{sif}(p_n). \end{aligned}$$

Applying Lemma 3.3

$$\frac{1 - 7\epsilon}{1 + \epsilon^2} \operatorname{sif}(p) \leq \operatorname{sif}(p_n) \leq \frac{1 + 7\epsilon}{1 - \epsilon^2} \operatorname{sif}(p). \quad (3.33)$$

Substituting the upper bound on  $\operatorname{sif}(p_n)$  into Equation 3.31 we have

$$\operatorname{efs}(x) \leq \frac{(1 + 7\epsilon)(1 + \epsilon^2)}{1 - \epsilon^2} \operatorname{sif}(p) < (1 + 8\epsilon) \operatorname{sif}(p).$$

To prove the lower bound on  $\operatorname{efs}(x)$  recall that the function  $\operatorname{efs}$  is 2-Lipschitz.

$$\operatorname{efs}(x) \geq \operatorname{efs}(s_n) - 2d(x, s_n) \geq \operatorname{efs}(s_n) - 8\epsilon \operatorname{sif}(p), \quad (3.34)$$

by Equation 3.32, and from the sampling requirements,  $\operatorname{efs}(s_n) \geq (1 - \epsilon^2) \operatorname{sif}(p_n)$ . Substituting the lower bound on  $\operatorname{sif}(p_n)$  in Equation 3.33 into Equation 3.34 we have

$$\operatorname{efs}(x) \geq \frac{(1 - \epsilon^2)(1 - 7\epsilon)}{1 + \epsilon^2} - 8\epsilon \operatorname{sif}(p) > (1 - 16\epsilon) \operatorname{sif}(p). \quad \square$$

For  $x$  inside the  $3\epsilon$ -neighborhood, we define  $B_2(x)$  as a ball of radius

$$r_0^2 = ((d(x, s_n) + \sigma(x))^2 + 17\sigma^2(x), \quad (3.35)$$

centered at  $x$ . We modify the definition mainly to improve the upper bound on  $\epsilon$  in our sampling preconditions. Consider point  $p$  closest to  $x$  on the surface. From the sampling requirements, there is always a sample point within a distance of  $\epsilon \text{sif}(p)$  from  $p$ . Therefore,

$$\begin{aligned} r_0^2 &= ((d(x, s_n) + \sigma(x))^2 + 17\sigma^2(x) \\ &\leq (4\epsilon \text{sif}(p) + \epsilon \text{efs}(x))^2 + 17\epsilon^2(x) \text{efs}^2(x) \\ &\leq \left( \frac{4\epsilon}{1-16\epsilon} \text{efs}(x) + \epsilon \text{efs}(x) \right)^2 + 17\epsilon^2(x) \text{efs}^2(x) \\ r_0 &< 6.83 \epsilon \text{efs}(x), \end{aligned} \quad (3.36)$$

and,

$$\begin{aligned} r_0^2 &\leq (4\epsilon \text{sif}(p) + \epsilon \text{efs}(x))^2 + 17\epsilon^2(x) \text{efs}^2(x) \\ &\leq (4\epsilon \text{sif}(p) + \epsilon(1+8\epsilon) \text{sif}(p))^2 + 17\epsilon^2(x)(1+8\epsilon)^2 \text{sif}^2(p) \\ r_0 &< 6.67 \epsilon \text{sif}(p). \end{aligned} \quad (3.37)$$

For  $x$  inside  $3\epsilon$ -neighborhood, all sample points inside  $B_2(x)$  are near the point  $p$  closest to  $x$  on  $F$ . In the following lemma we prove that the sample points near  $p$  are constrained to lie between two planes close to the tangent plane of  $p$ . This result is later used to show that the error in the sample points near  $p$  is small.

**Lemma 3.13.** *For a point  $p \in F$ , let  $B$  be a ball of radius  $\alpha \text{sif}(p)$  centered at  $p$  for  $\alpha \leq 1/4$ . The sample points inside  $B$  lie between two planes  $P_1, P_2$  parallel to the tangent plane at  $p$ . The distance from  $p$  to  $P_1, P_2$  is less than  $D(\alpha) = \left( \frac{(\alpha+\epsilon^2)^2}{2(1-\epsilon^2)^2} + \frac{\epsilon^2(1+\alpha)}{1-\epsilon^2} \right) \text{sif}(p)$ .*

**Proof.** Consider sample point  $s \in B$  whose closest point on  $F$  is  $q$ . From the sampling



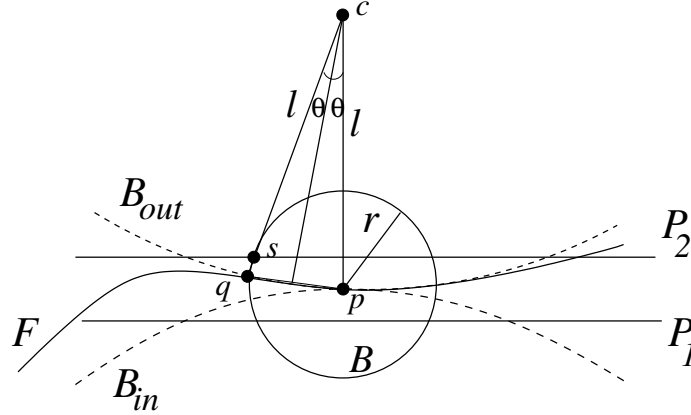


Figure 3.10: The surface inside a ball  $B$  of radius  $r$  has to be outside the medial balls  $B_{in}$  and  $B_{out}$ . As a result, all sample points in  $B$  are between two planes  $P_1$  and  $P_2$ .

requirements  $d(s, q) \leq \epsilon^2 \text{sif}(q)$ . Without loss of generality assume that  $s$  is above the tangent plane at  $p$  as shown in Figure 3.10.

$$d(p, q) \leq d(p, s) + d(s, q) \leq \alpha \text{sif}(p) + \epsilon^2 \text{sif}(q). \quad (3.38)$$

Let  $c$  be the center of the medial ball  $B_{out}$ , and let the angle between  $cq$  and  $cp$  be  $2\theta$ . As point  $q$  is on the surface, it has to be on or outside the medial ball  $B_{out}$  whose radius is  $l \geq \text{lfs}(p) \geq \text{sif}(p)$ . Hence the distance from  $p$  to the plane passing through  $q$  and parallel to the tangent plane at  $P$  is at most  $l(1 - \cos 2\theta) = 2l \sin^2 \theta \leq \frac{d^2(p, q)}{2l}$ . The distance from  $p$  to the tangent plane  $P_2$  passing through  $s$  is given by

$$D(\alpha) \leq \frac{d^2(p, q)}{2l} + \epsilon^2 \text{sif}(q) \leq \frac{(\alpha \text{sif}(p) + \epsilon^2 \text{sif}(q))^2}{2\text{sif}(p)} + \epsilon^2 \text{sif}(q), \quad (3.39)$$

by Equation 3.38.

The upper bound in Equation 3.38 shows that  $d(p, q)$  is small relative to  $\text{sif}(p)$  and  $\text{sif}(q)$ . Hence we can apply the result in Lemma 3.3 to get an upper bound on  $\text{sif}(q)$ ,

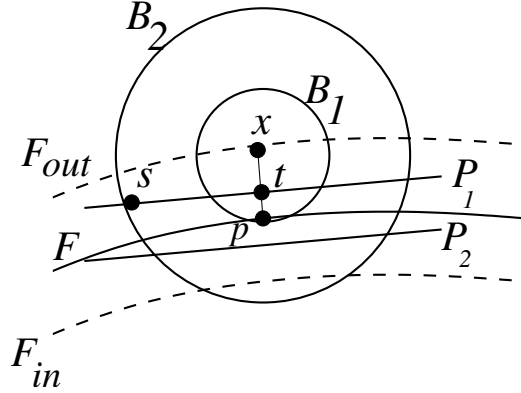


Figure 3.11: For  $x$  in the  $3\epsilon$ -neighborhood, the sample points inside balls  $B_2(x)$  are contained in a ball of radius  $6\epsilon \text{sif}(p)$  centered at  $p$ .

$\text{sif}(q) \leq \frac{1+\alpha}{1-\epsilon^2} \text{sif}(p)$ . Substituting into Equation 3.39 we have

$$\begin{aligned} D(\alpha) &\leq \frac{1}{2\text{sif}(p)} \left( \alpha + \frac{\epsilon^2(1+\alpha)}{1-\epsilon^2} \right)^2 \text{sif}^2(p) + \frac{\epsilon^2(1+\alpha)}{1-\epsilon^2} \text{sif}(p) \\ &= \frac{(\alpha + \epsilon^2)^2}{2(1-\epsilon^2)^2} \text{sif}(p) + \frac{\epsilon^2(1+\alpha)}{1-\epsilon^2} \text{sif}(p) \end{aligned}$$

□

The following lemma proves that the sample points inside  $B_2(x)$  are contained in a small ball centered at  $p$  as shown in Figure 3.11.

**Lemma 3.14.** *Let  $x$  be a point that is inside the  $3\epsilon$ -neighborhood as shown in Figure 3.11. Let  $p$  be the closest point to  $x$  on  $F$ . All sample points inside  $B_2(x)$  are contained in a ball of radius  $6\epsilon \text{sif}(p)$  centered at  $p$ .*

**Proof.** Without loss of generality assume that  $x$  is on or outside  $F$ , and let  $s$  be a sample point inside  $B_2(x)$ . Let  $t$  be the point at which  $xp$  intersects the plane  $P_1$  that contains  $s$  and is parallel to the tangent plane at  $p$ .

$$d^2(s, t) = d^2(s, p) - d^2(p, t) = d^2(x, s) - d^2(x, t). \quad (3.40)$$

### Chapter 3. Provably Better Moving Least Squares

---

As  $s$  is inside  $B_2(x)$ ,  $d^2(x, s) \leq (d(x, s_n) + \sigma(x))^2 + 17\sigma^2(x)$  by Equation 3.35. The distance from  $x$  to its closest sample point  $d(x, s_n) \leq \phi(x) + \epsilon \text{sif}(p)$  by the sampling preconditions. Substituting into Equation 3.40

$$\begin{aligned} d^2(s, p) - d^2(p, t) &\leq (|\phi(x)| + \epsilon \text{sif}(p) + \sigma(x))^2 + 17\sigma^2(x) - (|\phi(x)| - d(p, t))^2 \\ d^2(s, p) &\leq 2|\phi(x)|(\epsilon \text{sif}(p) + \sigma(x)) \\ &\quad + (\epsilon \text{sif}(p) + \sigma(x))^2 + 17\epsilon^2 \text{efs}^2(x) + 2|\phi(x)|d(p, t). \end{aligned} \quad (3.41)$$

The expression on the right hand side is maximized when  $|\phi(x)|$  is maximized. Since  $x$  is inside  $3\epsilon$ -neighborhood  $|\phi(x)| \leq 3\epsilon \text{sif}(p)$ . The distance between  $x$  and any sample point inside  $B_2(x)$  is less than

$$d(x, p) + d(p, s) < 3\epsilon \text{sif}(p) + 6.67\epsilon \text{sif}(p) < 9.67\epsilon \text{sif}(p),$$

by Equation 3.37). Sample point  $s$  is between two planes at a distance  $D(9.67\epsilon)$  from  $p$  by Lemma 3.13. Therefore  $d(p, t) \leq D(9.67\epsilon)$ . Substituting into Equation 3.41, and simplifying using the upper bound on  $\sigma(x) = \epsilon \text{efs}(x)$  in Lemma 3.12,

$$\begin{aligned} d^2(s, p) &< 6\epsilon \text{sif}(p)(\epsilon \text{sif}(p) + \epsilon \text{efs}(x)) + (\epsilon \text{sif}(p) + \sigma(x))^2 \\ &\quad + 17\sigma^2(x) + 6\epsilon \text{sif}(p)D(9.67\epsilon) \\ &< 36\epsilon^2 \text{sif}^2(p). \end{aligned}$$

□

In Lemma 3.14 we showed that all sample points inside  $B_2(x)$  are near point  $p$  closest to  $x$  on  $F$ . In the following lemma we show that the point functions of these sample points have a small error when evaluated at  $x$ .

**Lemma 3.15.** *Consider a point  $x$  whose closest point on the surface  $F$  is  $p$ . Let  $\vec{n}$  be the surface normal at  $p$ . For  $\alpha \leq 0.25$  let  $B$  be a ball of radius  $\alpha \text{sif}(p)$  at  $p$ . For each sample point  $s_i \in B$ , the angle between the normal at  $s_i$  and  $\vec{n}$  is less than  $\Theta(\alpha) = \frac{\alpha + \epsilon^2}{1 - 4\alpha - 4\epsilon^2} + \epsilon$ . For each sample point  $s_i \in B$ ,  $|\zeta_i(x)| \leq |\phi(x)|\Theta(\alpha)^2/2 + \alpha \text{sif}(p)\Theta(\alpha) + D(\alpha)$ .*

**Proof.** Let  $p_i$  be the point closest to  $s_i$  on  $F$ . Then,

$$d(p, p_i) \leq d(p, s_i) + d(s_i, p_i) \leq \alpha \text{efs}(p) + \epsilon^2 \text{sif}(p_i).$$

Applying the result in Lemma 3.3  $d(p, p_i) \leq \frac{\alpha + \epsilon^2}{1 - \epsilon^2} \text{sif}(p)$ , and  $d(p, p_i) \leq \frac{\alpha + \epsilon^2}{1 - \alpha} \text{sif}(p_i)$ . Combining these two results  $d(p, p_i) \leq \frac{\alpha + \epsilon^2}{1 - \alpha - \epsilon^2} \min\{\text{sif}(p), \text{sif}(p_i)\}$ . By the Lipschitz condition on surface normals in Theorem 3.2, the angle between the normal at  $p_i$  and the normal at  $p$  is less than

$$\frac{\frac{\alpha + \epsilon^2}{1 - \epsilon^2 - \alpha}}{1 - 3\frac{\alpha + \epsilon^2}{1 - \epsilon^2 - \alpha}} = \frac{\alpha + \epsilon^2}{1 - 4\alpha - 4\epsilon^2}. \quad (3.42)$$

Let  $\vec{n}_i$  be the normal associated with  $s_i$ . From the sampling preconditions we know that the angle between the normal of  $p_i$  and  $\vec{n}_i$  is at most  $\epsilon$ . By Equation 3.42 the angle between  $\vec{n}_i$  and  $\vec{n}$  is

$$\Theta(\alpha) < \frac{\alpha + \epsilon^2}{1 - 4\alpha - 4\epsilon^2} + \epsilon.$$

We can therefore write the normal of sample point  $s_i$  as  $\vec{n}_i = \vec{n} + \vec{\rho}_i$ , where  $\|\vec{\rho}_i\| \leq \Theta(\alpha)$ . The error function of sample point  $s_i$  when evaluated at  $x$  is given by

$$\begin{aligned} \zeta_i(x) &= \phi(x) - P_i(x) \\ &= \phi(x) - (x - s_i) \cdot \vec{n}_i \\ &= \phi(x) - (x - p) \cdot \vec{n}_i - (p - s_i) \cdot (\vec{n} + \vec{\rho}_i). \end{aligned}$$

Because  $p$  is the closest point to  $x$  on the surface,  $(x - p)$  is parallel to  $\vec{n}$  and  $\|x - p\| =$

$|\phi(x)|$ .

$$\begin{aligned}
 |\phi(x) - (x - p) \cdot \vec{n}_i| &\leq |\phi(x)|(1 - \cos \Theta(\alpha)) \\
 &= 2|\phi(x)| \sin^2(\Theta(\alpha)/2) \\
 &\leq |\phi(x)|\Theta^2(\alpha)/2.
 \end{aligned} \tag{3.43}$$

Since sample point  $s_i$  is inside  $B$ ,

$$|(p - s_i) \cdot \vec{\rho}_i| \leq \alpha \text{fs}(p) \Theta(\alpha). \tag{3.44}$$

From Lemma 3.13, the distance from sample point  $s_i$  inside  $B$  to the tangent plane at  $p$  is at most  $D(\alpha)$ . Therefore

$$|(p - s_i) \cdot \vec{n}| \leq D(\alpha). \tag{3.45}$$

Adding the upper bounds on the individual terms in Equations 3.43, 3.44, and 3.45 gives the desired result.  $\square$

In the following lemma we prove a result similar to Lemma 3.6 for points inside the  $3\epsilon$ -neighborhood. We show that the error functions of all sample points inside  $B_2(x)$  have a small value at  $x$ .

**Lemma 3.16.** *For a point  $x$  inside the  $3\epsilon$ -neighborhood,  $|E_{\text{in}}| \leq 69\epsilon^2 \text{sif}(p)$ .*

**Proof.** Combining the results in Lemma 3.14 and Lemma 3.15 gives the desired bound.

$$|E_{\text{in}}(x)| \leq \sum_{s_i \in B_2(x)} \frac{|\zeta_i(x)| W_i(x)}{W(x)} \leq \max\{|\zeta_i(x)| \mid s_i \in B_2(x)\}.$$

By Lemma 3.14, each sample point  $s_i \in B_2(x)$  is inside a ball of radius  $6\epsilon \text{sif}(p)$  around  $p$ ,

the point closest to  $x$  on  $F$ . By Lemma 3.15 we have

$$|\zeta_i(x)| \leq 3\epsilon \operatorname{sif}(p)\Theta^2(6\epsilon)/2 + 6\epsilon \operatorname{sif}(p)\Theta(6\epsilon) + D(6\epsilon) < 69\epsilon^2 \operatorname{sif}(p). \quad (3.46)$$

□

In the following lemma we prove an upper bound on the contribution of samples outside  $B_2(x)$  to the error function. The proof is similar to the proof of Lemma 2.6, and is given in the appendix.

**Lemma 3.17.** *For a point  $x$  inside the  $3\epsilon$ -neighborhood,  $|E_{\text{out}}(x)| < 3\epsilon^2 \operatorname{sif}(p)$ .*

**Proof.** Given in the appendix. □

In the following three theorems we prove that the reconstructed surface is inside the  $72\epsilon^2$ -neighborhood of the sampled surface  $F$ .

**Theorem 3.18.** *For a point  $x$  inside the  $3\epsilon$ -neighborhood whose closest point on  $F$  is  $p$ ,  $|E(x)| < 72\epsilon^2 \operatorname{sif}(p)$ .*

**Proof.** Adding the bounds on  $|E_{\text{in}}(x)|$  and  $|E_{\text{out}}(x)|$  in Lemma 3.16 and Lemma 3.17 gives the desired result.

$$|E(x)| \leq |E_{\text{in}}(x)| + |E_{\text{out}}(x)| < 69\epsilon^2 \operatorname{sif}(p) + 3\epsilon^2 \operatorname{sif}(p) = 72\epsilon^2 \operatorname{sif}(p). \quad \square$$

**Theorem 3.19.** *For a point  $x \in U$ , let  $p$  be the closest point in  $F$ . Then  $d(x, p) < 72\epsilon^2 \operatorname{sif}(p)$ .*

**Proof.** For  $x \in U$   $I(x) = 0$ . Hence  $x$  is inside the  $3\epsilon$ -neighborhood by Theorem 2.7.x. Applying the result in Theorem 3.18 we have  $|E(x)| < 72\epsilon^2 \operatorname{sif}(p)$ . Therefore  $d(x, p) = |\phi(x)| \leq |I(x)| + |E(x)| < 72\epsilon^2 \operatorname{sif}(p)$ . □

**Theorem 3.20.** *For a point  $p \in F$ , let  $q$  be the closest point in  $U$ . Then,  $d(p, q) \leq 72\epsilon^2 \text{sif}(p)$ .*

**Proof.** If  $I(p) = 0$  we are done; assume without loss of generality that  $I(p) < 0$ . Let  $t$  be the point on the outside normal of  $p$  at a distance of  $72\epsilon^2 \text{sif}(p)$  from  $p$ . From Theorem 3.18 we have

$$|I(t)| \geq |\phi(t)| - |E(x)| > d(p, t) - 72\epsilon^2 \text{sif}(p) = 0.$$

As the cut function  $I$  is continuous, there is a point  $s$  on  $pt$  at which  $I(s) = 0$  and  $d(p, s) < 72\epsilon^2 \text{sif}(p)$ . Since  $q$  is the point closest to  $p$  on  $U$ ,  $d(p, q) \leq d(p, s) < 72\epsilon^2 \text{sif}(p)$ .  $\square$

The above two results prove that the distance between  $U$  and  $F$  is small. In the remainder of this section, we analyze the gradient of the cut function to show that the normals of the reconstructed surface  $U$  are close to the normals of the sampled surface  $F$ . The gradient of the cut function  $\nabla I(x)$  can be written as

$$\nabla I(x) = \nabla \phi(x) + \nabla E(x).$$

Let  $p$  be the point closest to  $x$  on  $F$ , and let  $\vec{n}$  be the surface normal at  $p$ . The gradient of the signed distance function at  $x$  is given by  $\nabla \phi(x) = \vec{n}$ . The expression for the gradient of the error function can be simplified to

$$\begin{aligned} \nabla E(x) = & \sum_{s_i \in S} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} + \sum_{s_i, s_j \in S} \frac{2W_i(x)W_j(x)\zeta_i(x)(s_i - s_j)}{\sigma^2(x)W^2(x)} \\ & + \sum_{s_i, s_j \in S} \frac{2W_i(x)W_j(x)\zeta_i(x)(\|x - s_j\|^2 - \|x - s_i\|^2)\nabla \text{efs}(x)}{\sigma^2(x)W^2(x)\text{efs}(x)} \end{aligned} \quad (3.47)$$

We separate the contributions of sample points inside and outside  $B_2(x)$  to the gradient of

the error function as

$$\begin{aligned} \nabla E_{\text{in}}(x) = & \sum_{s_i \in B_2(x)} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} + \sum_{s_i, s_j \in B_2(x)} \frac{2W_i(x)W_j(x)\zeta_i(x)(s_i - s_j)}{\sigma^2(x)W^2(x)} \\ & + \sum_{s_i, s_j \in B_2(x)} \frac{2W_i(x)W_j(x)\zeta_i(x)(\|x - s_j\|^2 - \|x - s_i\|^2)\nabla \text{efs}(x)}{\sigma^2(x)W^2(x)\text{efs}(x)}, \end{aligned} \quad (3.48)$$

and,

$$\begin{aligned} \nabla E_{\text{out}}(x) = & \sum_{s_i \notin B_2(x)} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} + \sum_{s_i \notin B_2(x) \vee s_j \notin B_2(x)} \frac{2W_i(x)W_j(x)\zeta_i(x)(s_i - s_j)}{\sigma^2(x)W^2(x)} \\ & + \sum_{s_i \notin B_2(x) \vee s_j \notin B_2(x)} \frac{2W_i(x)W_j(x)\zeta_i(x)(\|x - s_j\|^2 - \|x - s_i\|^2)\nabla \text{efs}(x)}{\sigma^2(x)W^2(x)\text{efs}(x)}. \end{aligned} \quad (3.49)$$

In the following lemma we prove that  $\nabla E_{\text{in}}(x)$  has a small norm.

**Lemma 3.21.** *Let  $x$  be a point in the  $3\epsilon$ -neighborhood of  $F$  and let  $p$  be the point on  $F$  closest to  $x$ . Let  $\vec{n}$  be the normal of  $p$ . Then  $\|\nabla E_{\text{in}}(x)\| < 2096\epsilon$ , and  $|\vec{n} \cdot \nabla E_{\text{in}}(x)| < 19250\epsilon^2$ .*

**Proof.** By Equation 3.48

$$\begin{aligned} \|\nabla E_{\text{in}}(x)\| \leq & \max_{s_i \in B_2(x)} \{\|\vec{n} - \vec{n}_i\|\} \\ & + \frac{2}{\sigma^2(x)} \max_{s_i, s_j \in B_2(x)} \{|\zeta_i(x)|\|s_i - s_j\|\} \\ & + \frac{2\|\nabla \text{efs}(x)\|}{\sigma^2(x)\text{efs}(x)} \max_{s_i, s_j \in B_2(x)} \{|\zeta_i(x)|(\|x - s_j\|^2 - \|x - s_i\|^2)\}. \end{aligned} \quad (3.50)$$

Consider sample point  $s_i$  inside  $B_2(x)$ . By Lemma 3.14  $d(s_i, p) \leq 6\epsilon \text{ sif}(p)$ , and by Lemma 3.15 the angle between  $\vec{n}$ , and the normal  $\vec{n}_i$  of  $s_i$  is less than  $\Theta(6\epsilon)$ . Therefore

$$\|\vec{n} - \vec{n}_i\| \leq \Theta(6\epsilon). \quad (3.51)$$



### Chapter 3. Provably Better Moving Least Squares

---

Consider the error function  $\zeta_i$  associated with sample point  $s_i$ . By Equation 3.46  $|\zeta_i(x)| < 69\epsilon^2 \text{sif}(p)$ . Substituting the upper bound on  $\text{sif}(p)$  Lemma 3.12 we have

$$|\zeta_i(x)| < 69\epsilon^2 \text{sif}(p) < \frac{69\epsilon^2}{(1-16\epsilon)} \text{efs}(x) < 77\epsilon^2 \text{efs}(x),$$

Since sample points are inside a ball  $B_2(x)$  whose radius is  $r_0$ ,  $\|s_i - s_j\| \leq 2r_0$ , and  $(\|x - s_j\|^2 - \|x - s_i\|^2) \leq r_0^2$ . By Equation 3.36  $r_0 \leq 6.83\epsilon \text{efs}(x)$ . Since the function  $\text{sif}$  is 2-Lipschitz,  $\|\nabla \text{efs}(x)\| \leq 2$ . Substituting into Equation 3.50 we have

$$\|\nabla E_{\text{in}}(x)\| < 2096\epsilon.$$

Consider  $\nabla E_{\text{in}}(x)$  projected onto the normal vector  $\vec{n}$ . By Equation 3.48 we have

$$\begin{aligned} |\vec{n} \cdot \nabla E_{\text{in}}(x)| &\leq \max_{s_i \in B_2(x)} \{|\vec{n} \cdot (\vec{n} - \vec{n}_i)|\} \\ &\quad + \frac{2}{\sigma^2(x)} \max_{s_i, s_j \in B_2(x)} \{|\zeta_i(x)| \vec{n} \cdot (s_i - s_j)\} \\ &\quad \frac{2|\vec{n} \cdot \nabla \text{efs}(x)|}{\sigma^2(x) \text{efs}(x)} \max_{s_i, s_j \in B_2(x)} \{\zeta_i(x) (\|x - s_j\|^2 - \|x - s_i\|^2)\}. \end{aligned} \quad (3.52)$$

Since the angle between  $\vec{n}$  and  $\vec{n}_i$  is less than  $\Theta(6\epsilon)$ ,

$$\vec{n} \cdot (\vec{n} - \vec{n}_i) = 1 - \cos \Theta(6\epsilon) = \sin^2(\Theta(6\epsilon)) \leq \Theta^2(6\epsilon)/2.$$

The distance from each sample point  $s_i$  inside  $B_2(x)$  to the tangent plane at  $p$  is at most  $D(6\epsilon)$  by From Lemma 3.13. Hence  $|\vec{n} \cdot (s_i - s_j)| \leq 2D(6\epsilon)$ . Substituting these upper bounds into Equation 3.52 we have

$$\vec{n} \cdot \nabla E_{\text{in}}(x) < 19250\epsilon^2. \quad \square$$

## Chapter 3. Provably Better Moving Least Squares

---

In the following lemma we show that samples outside  $B_2(x)$  have little effect on  $\nabla I(x)$  by proving that  $\nabla E_{\text{out}}(x)$  has a small norm.

**Lemma 3.22.** *For each point  $x$  inside the  $3\epsilon$ -neighborhood,  $\|\nabla E_{\text{out}}(x)\| < 137\epsilon$ , and  $|\vec{n} \cdot \nabla E_{\text{out}}(x)| < 2958\epsilon^2$ .*

**Proof.** Given in the appendix. □

In the following lemma we prove that the gradient of the cut function is non-zero inside the  $3\epsilon$ -neighborhood. This result is later used in Section 3.6 to prove that  $U$  is topologically equivalent to  $F$ .

**Theorem 3.23.** *For a point  $x$  inside the  $3\epsilon$ -neighborhood, let  $p$  be the point closest to  $x$  on the surface  $F$ . Let  $\vec{n}$  be the surface normal of  $p$ . Then,  $\vec{n} \cdot \nabla I(x) > 0$ .*

**Proof.** By the definition of the error function

$$\vec{n} \cdot \nabla I(x) = \vec{n} \cdot \nabla \phi(x) + \vec{n} \cdot \nabla E(x) \geq 1 - |\vec{n} \cdot \nabla E_{\text{in}}(x)| - |\vec{n} \cdot \nabla E_{\text{out}}(x)|. \quad (3.53)$$

By Lemma 3.21 we have  $|\vec{n} \cdot \nabla E_{\text{in}}(x)| < 18752\epsilon^2$ , and by Lemma 3.22 we have  $|\vec{n} \cdot \nabla E_{\text{out}}(x)| < 2620\epsilon^2$ . Substituting into Equation 3.53

$$\vec{n} \cdot \nabla I(x) > 1 - 19250\epsilon^2 - 2958\epsilon^2 = 1 - 22208\epsilon^2 > 0,$$

for  $\epsilon \leq 1/150$ . □

The result in Theorem 3.23 also proves that the gradient can never be zero inside the  $3\epsilon$ -neighborhood. The zero set of  $I$  is inside the  $3\epsilon$ -neighborhood of  $F$  by Theorem 2.7. Hence by the implicit function theorem [17], zero is a *regular* value of  $I$  and the zero set  $U$  is a compact, two-dimensional manifold.

The gradient of the cut function at a point  $u \in U$  determines the normal of the reconstructed surface at  $u$ ,  $\vec{n}_u = \frac{\nabla I(u)}{\|\nabla I(u)\|}$ . In the following lemma we prove that  $\vec{n}_u$  converges to  $\vec{n}$  with  $\epsilon$ .

**Theorem 3.24.** *Let  $u$  be a point on the reconstructed surface  $U$  whose closest point on  $F$  is  $p$ . Let  $\vec{n}_u$  be the normal of  $U$  at point  $u$  and let  $\vec{n}$  be the normal of  $F$  at point  $p$ . An upper bound on the angle  $\theta$  between  $\vec{n}_u$  and  $\vec{n}$  is given by*

$$\cos \theta > \frac{1 - 22208\epsilon^2}{1 + 2233\epsilon}.$$

**Proof.** The angle between  $\vec{n}_u$  and  $\vec{n}$  is given by

$$\cos \theta = \frac{\vec{n} \cdot \nabla I(u)}{\|\nabla I(u)\|}. \quad (3.54)$$

From Theorem 3.23,

$$\vec{n} \cdot \nabla I(u) \geq 1 - 22208\epsilon^2.$$

Consider the following upper bound for  $\|\nabla I(x)\|$ ,

$$\|\nabla I(u)\| \leq \|\nabla \phi(u)\| + \|\nabla E(u)\| \leq 1 + \|\nabla E_{\text{in}}(x)\| + \|\nabla E_{\text{out}}(u)\|.$$

From Lemma 3.21  $\|\nabla E_{\text{in}}(x)\| < 2096\epsilon$ , and from Lemma 3.22  $\|\nabla E_{\text{out}}(x)\| < 137\epsilon$ .

Substituting into Equation 3.54

$$\cos \theta > \frac{1 - 22208\epsilon^2}{1 + 2233\epsilon}. \quad \square$$

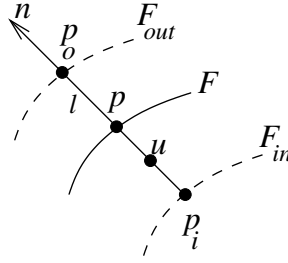


Figure 3.12: Points  $r, t$  are the closest points to  $p$  on the offset surfaces. The line segment  $p_i p_o$  intersects the zero set  $U$  at a unique point  $u$ .

### 3.6 Topology of the Reconstructed Surface

We now use the geometric results in Section 3.5 to show that the reconstructed surface  $U$  has the same topology as the sampled surface  $F$ . The proof of correct topology is identical to the proof in Section 2.5.

**Definition:** Let  $\Gamma : \mathbb{R}^3 \rightarrow F$  map each point  $q \in \mathbb{R}^3$  to the closest point of  $F$ .

**Theorem 3.25.** *The restriction of  $\Gamma$  to  $U$  is a homeomorphism from  $U$  to  $F$ .*

**Proof.** The discontinuities of  $\Gamma$  are the points on the medial axis of  $F$ . As  $U$  is constrained to be inside the  $3\epsilon$ -neighborhood of  $F$ , the restriction of  $\Gamma$  to  $U$  is continuous.

Now we show that  $\Gamma$  is one-to-one. Let  $p$  be a point on  $F$  and let  $\vec{n}$  be the normal at  $p$  as shown in Figure 3.12. Without any loss of generality, assume  $\text{sif}(p) = 1$ . Consider the line segment  $l$  parallel to  $\vec{n}$  that intersects  $F_{\text{out}}$  and  $F_{\text{in}}$  at  $p_o$  and  $p_i$  respectively. At each point  $y \in p_o p_i$ ,  $\nabla I(y) \cdot \vec{n} > 0$  from Theorem 3.23. So the function  $I(x)$  is monotonically decreasing from  $r$  to  $t$  and there is a unique point  $u$  on  $p_o p_i$  where  $I(u) = 0$ . Assume there is another point  $v \in U$  for which  $\Gamma(v) = x$ . The point  $v$  has to be outside the segment  $p_o p_i$  and the distance from  $v$  to its closest point on  $F$  is greater than  $3\epsilon$ . This contradicts Theorem 3.19.

Finally we need to show that  $\Gamma$  is onto. As  $\Gamma$  maps closed components of  $U$  onto closed

components of  $F$  in a continuous manner,  $\Gamma(U)$  should consist of a set of closed connected components. Consider the point  $p$  in Figure 3.12. Assume that  $q = \Gamma(u)$  is not in the same component of  $F$  as  $p$ . Let  $B_u$  be the ball of radius  $3\epsilon$  centered at  $u$  that intersects two components of  $F$ , one containing point  $p$  and one containing point  $q$ . Boissonnat and Cazals [21] (Proposition 12) show that any ball whose intersection with  $F$  is not a topological disc contains a point of the medial axis of  $F$ . Since point  $p$  is inside the ball  $B_u$  that contains a point of the medial axis,  $\text{sif}(p) \leq 6\epsilon$ . Recall that our sampling conditions require  $\epsilon \leq 1/150$ . Hence,  $\text{sif}(p) \leq 6\epsilon \leq 1/25$ . This violates our assumption that  $\text{sif}(p) = 1$ .  $\square$

We will now prove a stronger topological result which shows that the reconstructed surface  $U$  can be continuously deformed into the sampled surface  $F$  without any change in topology. Recall the definitions of isotopy and ambient isotopy from Chapter 2.

**Definition:** An isotopy between two compact orientable surfaces in  $\mathbb{R}^3$  is a continuous map  $\Psi : U \times [0, 1] \rightarrow \mathbb{R}^3$  such that  $\Psi(\cdot, 0)$  is the identity of  $U$ ,  $\Psi(\cdot, 1) = F$ , and for each  $t \in [0, 1]$ ,  $\Psi(\cdot, t)$  is homeomorphic to  $U$ .

**Definition:** An ambient isotopy between two compact orientable surfaces  $U$  and  $F$  is a continuous map  $\Psi : \mathbb{R}^3 \times [0, 1] \rightarrow \mathbb{R}^3$ , such that  $\Psi(\cdot, 0)$  is the identity of  $\mathbb{R}^3$ ,  $\Psi(U, 1) = F$ , and for each  $t \in [0, 1]$ ,  $\Psi(\cdot, t)$  is a homeomorphism of  $\mathbb{R}^3$ .

**Theorem 3.26.** *The zero surface  $U$  is isotopic to the sampled surface  $F$ .*

**Proof.** We will define an ambient isotopy  $\Psi$  whose restriction to  $U$  will be an isotopy to  $F$ . Outside the  $3\epsilon$ -neighborhood, the ambient isotopy is the identity  $\Psi(x, t) = x$  for  $t \in [0, 1]$ . From the proof of Theorem 3.25 we know that a line segment  $l$  normal to a point  $p \in F$  intersects  $U$  only at one point  $u$  inside the  $3\epsilon$ -neighborhood. Let  $p_i$  and  $p_o$  be the end points

of  $l$  on the inside and outside  $3\epsilon$ -offset surfaces respectively as shown in Figure 3.12. We define the ambient isotopy to be  $\Psi(u, t) = tp + (1 - t)u$ . The line segment  $p_i u$  is linearly mapped to  $p_i \Psi(u, t)$ . Similarly, the line segment  $u p_o$  is mapped to  $\Psi(u, t) p_o$ .  $\square$

## Appendix

Recall that for points inside the  $3\epsilon$ -neighborhood the radius  $r_0$  of the ball  $B_2(x)$  is given by

$$r_0^2 = (d(x, s_n) + \epsilon \text{efs}(x))^2 + 17\epsilon^2 \text{efs}^2(x).$$

With this new definition of  $B_2(x)$ , it is easy to prove that Lemma 3.8 is true for  $x$  inside the  $3\epsilon$ -neighborhood. Therefore the density functions defined in Lemma 3.9 are valid inside the  $3\epsilon$ -neighborhood.

**Observation 1.** *Let  $x$  be a point inside the  $3\epsilon$ -neighborhood, and let  $B(x)$  be a ball of radius  $r_{\min} \geq r_0$  around  $x$ . Let  $\lambda_k$  ( $0 \leq k \leq 4$ ) be radial density functions given by  $\lambda_k(x, r) = 12r_{\min}^k e^{-r_{\min}^2/\sigma^2(x)}$  for  $r \leq r_{\min}$ , and  $\lambda_k(x, r) = 12r^k e^{-r^2/\sigma^2(x)}$  for  $r \geq r_{\min}$ .*

*Then*

$$\int_0^\infty 4\pi r^2 \lambda_k(x, r) \geq \sum_{s_i \notin B} \frac{d^k(x, s_i)}{a_i} e^{-d^2(x, s_i)/\sigma^2(x)}.$$

Let  $\alpha_{\text{mid}} = 0.06$  be a small constant. Since  $\epsilon \leq 1/150$ ,  $\alpha_{\text{mid}} \geq 9\epsilon$ . Consider a ball  $B_{\text{mid}}(x)$  of radius  $\alpha_{\text{mid}} \text{efs}(x)$  centered at point  $x$  as shown in Figure 3.13. All sample points inside  $B_{\text{mid}}(x)$  are near  $p$ , the point closest to  $x$  on  $F$ . This definition of  $B_{\text{mid}}(x)$  is used in the following two results in which we prove that samples outside  $B_2(x)$  have little effect on the cut function for  $x$  inside the  $3\epsilon$ -neighborhood.

**Proof of Lemma 3.17.** Recall that  $\xi_{\text{out}}(x)$  that is the sum of the error in all sample points outside  $B_2(x)$ . Since  $\alpha_{\text{mid}} \geq 9\epsilon$ ,  $B_{\text{mid}}$  is clearly larger than  $B_2(x)$  as shown in Fig-

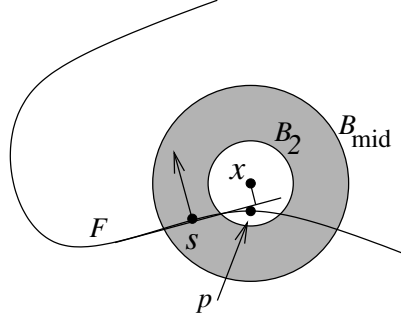


Figure 3.13: The normal of each sample point  $s \in B_{\text{mid}}$  is close to the surface normal at  $p$ . As a result, the point function of  $s$  when evaluated at  $x$  is close to  $\phi(x)$ .

Figure 3.13. Let  $\xi_1(x)$  and  $\xi_2(x)$  be the contribution of sample points inside  $B_{\text{mid}}(x)$  and outside  $B_{\text{mid}}(x)$  to  $\xi_{\text{out}}(x)$  respectively. We compute the desired upper bound by integrating over density functions defined in Observation 1.

Consider a sample point  $s_i$  in the spherical shell between  $B_2(x)$  and  $B_{\text{mid}}(x)$  at a distance  $\alpha \text{efs}(x) \leq \alpha_{\text{mid}} \text{efs}(x)$  from  $x$ . By the upper bound on  $\text{efs}(x)$  in Lemma 3.12

$$d(p, s_i) \leq \alpha \text{efs}(x) + 3\epsilon \text{sif}(p) < (\alpha' + 3\epsilon)\text{sif}(p),$$

where  $\alpha' = (1 + 8\epsilon)\alpha/(1 - \epsilon^2)$ . Since  $s_i$  is outside  $B_2(x)$ ,  $\alpha$  is clearly larger than  $4\epsilon$ . Recall that we defined a function  $\Theta$  in Lemma 3.15 such that  $\Theta(\alpha)$  is an upper bound on the angle between the normal of  $p$  and the approximate normal of a sample point that is at a distance of  $\alpha \text{sif}(p)$  from  $p$ . We can simplify the expression for the angle between the normal of  $s_i$  and the normal  $\vec{n}$  of  $p$  to

$$\Theta(\alpha' + 3\epsilon) = \frac{\alpha' + 3\epsilon + \epsilon^2}{1 - 4\alpha' - 12\epsilon - 4\epsilon^2} < \frac{\alpha' + 3\alpha/4 + (1/150)\alpha/4}{1 - 4\alpha'_{\text{mid}} - 12(1/150) - 5(1/150)^2} \approx 2.85\alpha. \quad (3.55)$$

Recall that in Lemma 3.13 we proved that the distance from  $s_i$  to the tangent plane at  $p$  is given by  $D(\alpha' + 3\epsilon) \approx 1.62\alpha^2\text{sif}(p)$ .

By the upper bound on the error function associated with  $s_i$  derived in Lemma 3.15 we

have

$$\begin{aligned}
 \zeta_i(x) &\leq 3\epsilon\Theta^2(\alpha' + 3\epsilon)/2 + \Theta(\alpha' + 3\epsilon)(\alpha' + 3\epsilon)\text{sif}(p) + D(\alpha' + 3\epsilon) \\
 &< 7\alpha^2\text{sif}(p) \\
 &< 8\alpha^2\text{efs}(x).
 \end{aligned} \tag{3.56}$$

Consider sample point  $s_i$  inside  $B_{\text{mid}}(x)$  at a distance  $\alpha \text{efs}(x)$  from  $x$ . The contribution of sample point  $s_i$  to the function  $\xi_1(x)$  is given by

$$8\alpha^2\text{efs}(x)W_i(x) = 8d^2(x, s_i)W_i(x)/\text{efs}(x).$$

The integral of the density function  $\lambda_2$  defined in Observation 1 is useful in proving an upper bound for such a function.

$$\begin{aligned}
 |\xi_1(x)| &< \frac{8}{\text{efs}(x)} \int_{r=0}^{\infty} 4\pi r^2 \lambda_2(x, r) dr \\
 &< \frac{384\pi}{\text{efs}(x)} \left( r_0^5 e^{-r_0^2/\sigma^2(x)} + \int_{r=r_0}^{\infty} r^4 e^{-r_0 r/\sigma^2(x)} dr \right) \\
 &\leq \frac{384\pi}{\text{efs}(x)} \left( \frac{r_0^5}{3} + r_0^3 \sigma^2(x) + 4r_0 \sigma^4(x) + \frac{12\sigma^6(x)}{r_0} \right. \\
 &\quad \left. + \frac{24\sigma^8(x)}{r_0^3} + 24\sigma^{10}(x)r_0^5 \right) e^{-r_0^2/\sigma^2(x)}.
 \end{aligned}$$

For  $x$  inside the  $3\epsilon$ -neighborhood  $r_0 \leq 6.83\epsilon\text{efs}(x)$  by Equation 3.36. Substituting the lower bound on  $W(x)$  from Lemma 3.7

$$\begin{aligned}
 |E_1(x)| &\leq \frac{3840\pi e^{-17}}{\text{efs}(x)\sigma^3(x)} \left( \frac{r_0^5}{3} + r_0^3 \sigma^2(x) + 4r_0 \sigma^4(x) + \frac{12\sigma^6(x)}{r_0} \right. \\
 &\quad \left. + \frac{24\sigma^8(x)}{r_0^3} + \frac{24\sigma^{10}(x)}{r_0^5} \right) \\
 &< 2.46\epsilon^2\text{efs}(x).
 \end{aligned} \tag{3.57}$$



## Chapter 3. Provably Better Moving Least Squares

---

For sample points outside  $B_{\text{mid}}(x)$ , we revert to the bound obtained in Lemma 3.10 on the error function associated with each sample. For  $s_i \notin B_{\text{mid}}(x)$  at a distance  $r$  from  $x$ ,  $\zeta_i(x) \leq 2r$ . The expression for  $|\xi_2(x)|$  is the same as the expression for  $|\xi_{\text{out}}(x)|$  in the proof of Lemma 3.10 except that the sample points are outside  $r_{\text{mid}}$  instead of  $r_0$ . By Equation 3.26 we have

$$|E_2(x)| \leq \frac{960\pi}{\text{efs}(x)\sigma^3(x)} \left( \frac{r_{\text{mid}}^4}{3} + r_{\text{mid}}^2\sigma^2(x) + 3\sigma^4(x) + \frac{6}{r_{\text{mid}}^2}\sigma^6(x) + \frac{6}{r_{\text{mid}}^4}\sigma^8(x) \right) e^{-(r_{\text{mid}}^2 - (d(x, s_n) + \sigma(x))^2)/\sigma^2(x)}. \quad (3.58)$$

When  $\epsilon = 1/150$ ,  $|E_2(x)| \leq 10^{-15}\epsilon^2\text{efs}(x)$ , and because of the exponential term in its expression,  $|E_2(x)|$  remains less than  $10^{-14}\epsilon^2\text{efs}(x)$  for  $\epsilon < 1/150$ . By Equation 3.57 and Equation 3.58 we have

$$E_{\text{out}}(x) \leq |E_1(x)| + |E_2(x)| < 2.42\epsilon^2\text{efs}(p) < 3\epsilon^2\text{sif}(p),$$

by Lemma 3.12. □

**Proof of Lemma 3.22.** Recall that  $x$  is a point inside the  $3\epsilon$ -neighborhood, and  $p$  is the point closest to  $x$  on  $F$ . Assume without loss of generality that the indices of sample points are in the increasing order of distance to  $x$ . Let  $\vec{n}$  be the surface normal at  $p$ . The contribution of sample points outside  $B_2(x)$  to  $\nabla E(x)$  is given by

$$\begin{aligned} \nabla E_{\text{out}}(x) &= \sum_{s_i \notin B_2(x)} \frac{(\vec{n} - \vec{n}_i)W_i(x)}{W(x)} \\ &+ \sum_{s_i \notin B_2(x), i > j} \frac{2W_i(x)W_j(x)}{\sigma^2(x)W^2(x)} ((\zeta_i(x) - \zeta_j(x))(s_i - s_j)) \\ &+ \frac{1}{\text{efs}(x)} (\zeta_i(x) - \zeta_j(x)) (\|x - s_j\|^2 - \|x - s_i\|^2 \cdot \nabla \text{efs}(x)). \end{aligned} \quad (3.59)$$

### Chapter 3. Provably Better Moving Least Squares

---

Let  $\nabla E_1(x)$  and  $\nabla E_2(x)$  be the contributions of sample points inside and outside  $B_{\text{mid}}(x)$  to the gradient of the error function.

Consider a sample point  $s_i$  inside  $B_{\text{mid}}(x)$  at a distance  $r_i = \alpha \text{efs}(x)$  from  $x$ . From Equation 3.55,  $\|n - n_i\| \approx 2.84\alpha < 3\alpha$ , and from Equation 3.56,  $|\zeta_i(x)|, |\zeta_j(x)| \leq 8\alpha^2 \text{efs}(x)$ . So the expression for  $\|\nabla E_1(x)\|$  can be simplified to

$$\begin{aligned} \|\nabla E_1(x)\| &\leq \sum_{s_i \in B_{\text{mid}}(x) - B_2(x)} \frac{3r_i W_i(x)}{\text{efs}(x)W(x)} + \\ &\quad \sum_{s_i \in B_{\text{mid}}(x) - B_2(x), i > j} \frac{2W_i(x)W_j(x)}{\sigma^2(x)W^2(x)} \left( \frac{(16r_i^2)(2r_i)}{\text{efs}(x)} + \frac{(16r_i^2)(r_i^2)\|\nabla \text{efs}(x)\|}{\text{sf}^2(x)} \right) \\ &\leq \frac{1}{\text{efs}(x)W(x)} \sum_{s_i \in B_{\text{mid}}(x) - B_2(x)} \left( 3r_i + \frac{64r_i^3}{\sigma^2(x)} + \frac{64r_i^4}{\sigma^2(x)\text{efs}(x)} \right) W_i(x). \end{aligned}$$

The density functions  $\lambda_k$  are useful in proving an upper bound of the above summation.

$$\begin{aligned} \|\nabla E_1(x)\| &\leq \frac{1}{\text{efs}(x)W(x)} \left( 48\pi \left( \frac{1}{3} \left( 3r_0^4 + \frac{64r_0^6}{\sigma^2(x)} + \frac{64r_0^7}{\sigma^2(x)\text{efs}(x)} \right) e^{-r_0^2/\sigma^2(x)} \right. \right. \\ &\quad \left. \left. + \int_{r_0}^{\infty} 3r^3 + \frac{64r^5}{\sigma^2(x)} + \frac{64r^6}{\sigma^2(x)\text{efs}(x)} e^{-r_0 r/\sigma^2(x)} dr \right) \right). \end{aligned}$$

Evaluating the integral and substituting the lower bound on  $W(x)$  from Lemma 3.7, we have

$$\|\nabla E_1(x)\| \approx 131.5\epsilon.$$

To compute an upper bound on  $\|\nabla E_2(x)\|$  note that  $\|\vec{n} - \vec{n}_i\| \leq 2$ , and  $|\zeta_i|, |\zeta_j(x)| \leq 2\|x - s_i\|$  outside  $B_2(x)$ . So the expression for  $\|E_2(x)\|$  can be simplified to

$$\|\nabla E_2(x)\| \leq \frac{1}{W(x)\text{efs}(x)} \int_{r_{\text{mid}}}^{\infty} \left( 2 + \frac{16r^2}{\epsilon^2} + \frac{16r^3}{\epsilon^2} \right) dr.$$

Simplifying the integral, it is easy to show

$$\|\nabla E_2(x)\| < 10^{-7}\epsilon$$

when  $\epsilon \leq 1/150$ . Adding the upper bounds on  $\|E_1(x)\|$  and  $\|E_2(x)\|$ ,  $\|\nabla E(x)\| \leq 132\epsilon$ .

Consider the gradient of the error function projected onto the normal of  $p$ . Let  $s_i$  be a sample point inside  $B_{\text{mid}}(x)$  at a distance  $r_i = \alpha \text{efs}(x)$  from  $x$ . From Equation 3.55,  $\vec{n} \cdot (\vec{n} - \vec{n}_i) \leq \Theta^2(\alpha' + 3\epsilon)/2 < 5\alpha^2$ , and  $\vec{n} \cdot (s_i - s_j) \leq 2D(\alpha' + 3\epsilon) \leq 4\alpha^2 \text{efs}(x)$ . So the error due to all sample points inside  $B_{\text{mid}}(x)$  is given by

$$\begin{aligned} |\vec{n} \cdot E_1(x)| &= \frac{1}{\text{sif}^2(x)W(x)} \left( 48\pi \left( \frac{1}{3} \left( 5r_0^5 + \frac{128r_0^7}{\sigma^2(x)} + \frac{64r_0^7}{\sigma^2(x)} \right) e^{-r_0^2/\sigma^2(x)} \right. \right. \\ &\quad \left. \left. + \int_{r_0}^{\infty} 5r^4 + \frac{128r^6}{\sigma^2(x)} + \frac{64r^6}{\sigma^2(x)} e^{-r_0r/\sigma^2(x)} dr \right) \right). \end{aligned}$$

Evaluating the integral and substituting the lower bound on  $W(x)$  from Lemma 3.7 we have

$$|\vec{n} \cdot \nabla E_1(x)| \leq 2619.9\epsilon^2.$$

Since  $\|n\| \leq 1$ ,  $|\vec{n} \cdot \nabla E_2(x)| \leq \|\nabla E_2(x)\| < 10^{-5}\epsilon^2$  for  $\epsilon \leq 1/150$ . Adding the upper bounds on  $|\vec{n} \cdot \nabla E_1(x)|$  and  $|\vec{n} \cdot \nabla E_2(x)|$ , we have  $|\vec{n} \cdot \nabla E(x)| \leq 2620\epsilon$ .  $\square$

## Chapter 4

# An Implementation of Moving Least Squares

In this chapter we move from theory to practice and describe our implementation of the MLS algorithm analyzed in Chapter 2. The input to the MLS implementation is a set of sample points obtained from a scanning device or from passive techniques. We assume that the range images have been aligned into a single coordinate system by the automatic registration algorithm described in Chapter 6.

Sample points in the range images might not exactly lie on the sampled surface after registration. Instead, they typically form a thick cloud of points near the sampled surface. Figure 4.1 shows a two-dimensional point cloud with added noise, and the zero set of the cut function defined by the MLS algorithm. Notice that even when the point cloud is noisy, the reconstructed implicit surface is smooth. However, the cut function  $L$  might contain spurious zero crossings due to outliers and areas of undersampling in the range images. An isosurface triangulation algorithm such as Marching Cubes [63] cannot reconstruct the right surface from cut functions containing zero crossings far from the sampled surface.

Our approach is to project the sample points onto the zero set of the cut function, and



Figure 4.1: (a) A two-dimensional data set with added noise. (b) Cut function defined by the uniform MLS algorithm. (c) Zero set of the cut function.

reconstruct a surface mesh from the projected sample points using the Eigencrust algorithm described in Chapter 5. The Eigencrust algorithm is robust against outliers and undersampling. Projecting the sample points onto the zero set of the cut function reduces scanner measurement noise, and errors introduced due to poor alignment of scans.

## 4.1 Implementation

Let  $S$  be the set of sample points in the aligned range images and for each sample point  $s_i \in S$  let  $\vec{n}_i$  be the approximate outside normal of the sampled surface at  $s_i$ . We smooth the noisy point cloud by projecting each sample point onto the zero set  $F$  of the cut function  $I$  defined by all the sample points in  $S$ . In our implementation we use the uniform MLS algorithm analyzed in Chapter 2. Recall the definition of the uniform-MLS surface given by Equation 2.1.

$$I(x) = \sum_{s_i \in S} \frac{W_i(x)}{\sum_{s_j \in S} W_j(x)} ((x - s_i) \cdot \vec{n}_i).$$

The weight functions are Gaussian functions,  $W_i(x) = e^{-\|x-s_i\|^2/\epsilon^2}/a_i$ , with a normalization factor  $a_i$  which is set to the number of sample points inside a ball of radius  $\epsilon$  centered at  $s_i$ , including  $s_i$  itself.

### 4.1.1 Estimating Normals and Parameters

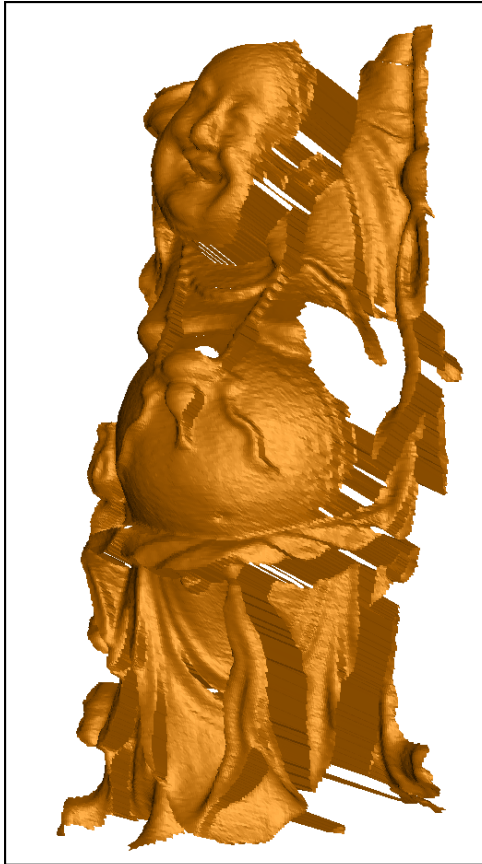
Range scanners sample points on a square grid (as viewed from the scanner), which forms a natural triangulation. We treat the triangulation as a piecewise linear surface embedded in three-dimensional space. This triangulation might contain long triangles where the scanner crosses a silhouette of the object being scanned as shown in Figure 4.2(a). We compute a *grid spacing*  $\ell$  equal to the median length of the diagonal edges of the triangulated grid, and we discard any triangle whose greatest edge length exceeds  $4\ell$ . Figure 4.2(b) shows the triangulation of after triangles with long edges are removed.

The normal at each sample point  $s \in S$  is given by an average of the normals of all faces incident on  $s$ . These normals are oriented along the outside direction using the location of range scanner. We discard sample points that have no incident triangles. We set the  $\epsilon$  parameter of the Gaussian weight functions to  $4\ell$  so that the Gaussian functions associated with adjacent sample points in the range images overlap significantly.

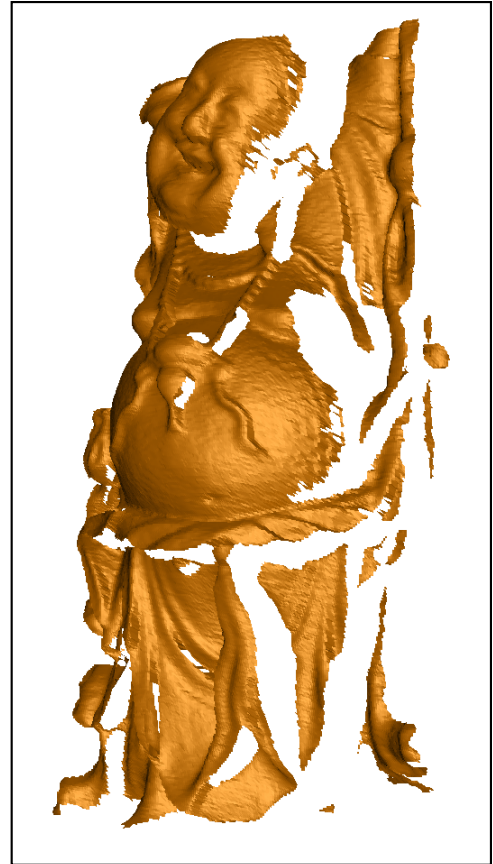
### 4.1.2 Projection

A simple Newton iteration procedure projects the sample points onto the zero set of the cut function. Let  $p_i$  be the position of the sample point  $s$  after  $i^{th}$  iteration. At each iteration we compute the value and the gradient of the cut function at  $p_i$ , and project  $p_i$  to the point  $p_{i+1}$  given by

$$p_{i+1} = p_i - \frac{I(p_i)}{\|\nabla I(x)\|} \nabla I(x).$$



(a)



(b)

Figure 4.2: (a) Triangulation of a range image in the Stanford Happy Buddha data set. (b) The triangulation after removing triangles with long edges.

The iterative procedure stops when the absolute value of the cut function at the projected point is below a threshold. If the iterative procedure does not converge after a user specified number of iterations (10 in our implementation) we leave the sample point to its original position.

Let  $B(x)$  be a ball of radius  $d(x, s_n) + 5\epsilon$  centered at  $x$ , where  $s_n$  is the nearest sample point to  $x$ . In Lemma 2.6 in Chapter 2 we proved that sample points outside  $B(x)$  have little effect on the value of the cut function at  $x$ . Let  $W_{\text{in}}(x)$  be the sum of the weights of sample points inside  $B(x)$ . We ignore sample points outside  $B(x)$  when evaluating the cut function. An approximation  $\tilde{I}(x)$  to the cut function is given by

$$\tilde{I}(x) = \frac{\sum_{s_i \in B(x)} P_i(x) W_i(x)}{W_{\text{in}}(x)}.$$

The projection step also requires the gradient of the cut function. In our analysis of the cut function given in Chapter 2, we wrote the expression for  $\nabla I(x)$  as a double summation over the sample points (Equation 2.4.2). We can rewrite the expression for the gradient so that the double summation is not required,

$$\nabla I(x) = \frac{\sum_{s_i \in S} (\vec{n}_i - P_i(x)(x - s_i)/\epsilon^2)}{W(x)} + I(x) \frac{\sum_{s_i \in S} (x - s_i)/\epsilon^2}{W(x)}.$$

Just as with the approximation of the cut function value, we approximate the gradient by ignoring sample points outside  $B(x)$ :

$$\nabla I(x) \approx \frac{\sum_{s_i \in B(x)} (\vec{n}_i - P_i(x)(x - s_i)/\epsilon^2)}{W_{\text{in}}(x)} + \tilde{I}(x) \frac{\sum_{s_i \in B(x)} (x - s_i)/\epsilon^2}{W_{\text{in}}(x)}.$$

Since the number of sample points inside  $B(x)$  is much smaller than the number of sample points in  $S$  for most point clouds, the approximate gradient is much faster to compute than the exact gradient. We use kd-trees [43] to efficiently find the sample point closest to  $x$  and the set of sample points inside  $B(x)$ .



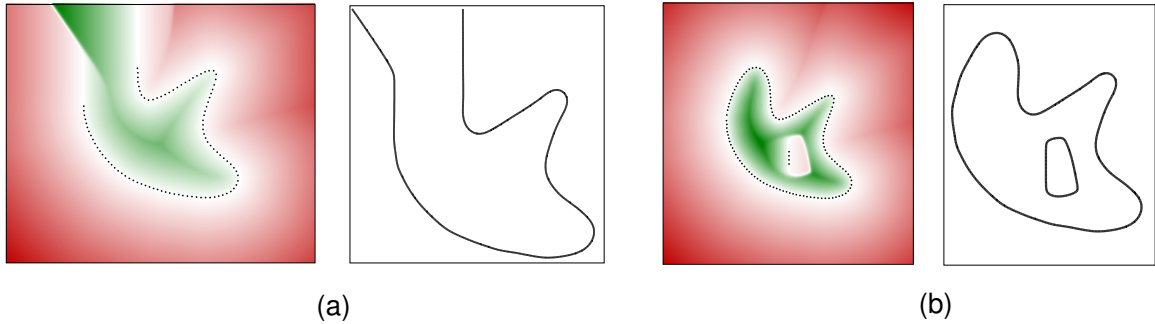


Figure 4.3: (a) The cut function defined by the MLS algorithm for a point cloud with undersampling. (b) Adding outliers to the point cloud introduces a zero set far away from the real sample points.

### 4.1.3 Limitations

Although the MLS algorithm effectively removes scanner and registration noise, it cannot handle undersampling and outliers in the input point cloud. Figure 4.3 shows a two-dimensional example. In Figure 4.3(a), removing some of the sample points causes the zero set of the cut function to appear far away from the sample points. This is a problem for point-based modeling algorithms that rely on projecting random points onto the zero set of the cut function. Similarly, the MLS algorithm cannot differentiate between sample points that lie on the true surface and outliers that are far away from the sampled surface as it relies on purely local information. Outliers introduce spurious zero crossings into the cut function as shown in Figure 4.3(b).

We deal with undersampling and outliers when we mesh the zero set using the *Eigen-crust* algorithm, which we describe in Chapter 5. While projecting sample points, we have to be careful not to move sample points that are near the true surface onto zero set components far away from the surface. We move the sample point to its projection on the zero set only if the projected point is less than a given threshold away ( $10\epsilon$  in our implementation) from its original position.

Model	Points	Iterations	Average number of iterations	Time (Seconds)
Bunny	362272	748563	2.07	42.34
Dragon	1769513	4850420	2.74	435.34
Armadillo	1706919	2735165	1.60	234.12
Buddha	2643108	6626532	2.50	559.164
Angel	9449840	35106984	3.71	2808.95

Table 4.1: Running times for projecting sample points on to the zero set of the MLS cut function.

## 4.2 Results

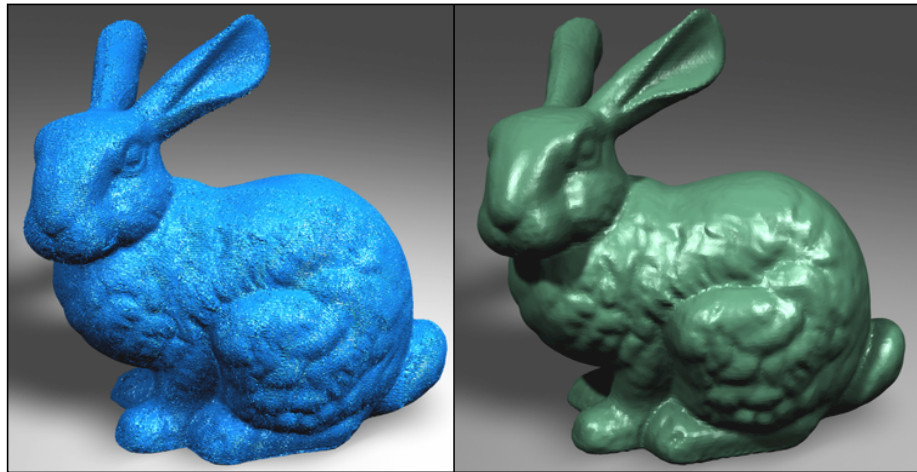
Figure 4.4(a) shows the Eigencrust reconstruction of the bunny model with and without MLS smoothing. Notice that the surface reconstructed without MLS smoothing is bumpy because of noise and poor alignment of scans. After MLS smoothing the reconstructed surface is smooth. However, MLS smoothing does not remove outliers in the input, as Figure 4.4(b) shows.

Table 4.1 shows the running time of MLS smoothing on a few data sets. The projection procedure converges very quickly, typically in 2–4 steps. The quick convergence confirms our gradient results in Chapter 2, where we showed that the gradient of the cut function approximates the sample point normals inside a small neighborhood of the sampled surface. The Eigencrust reconstructions of all the point clouds in Table 4.1 are given at the end of Chapter 5.

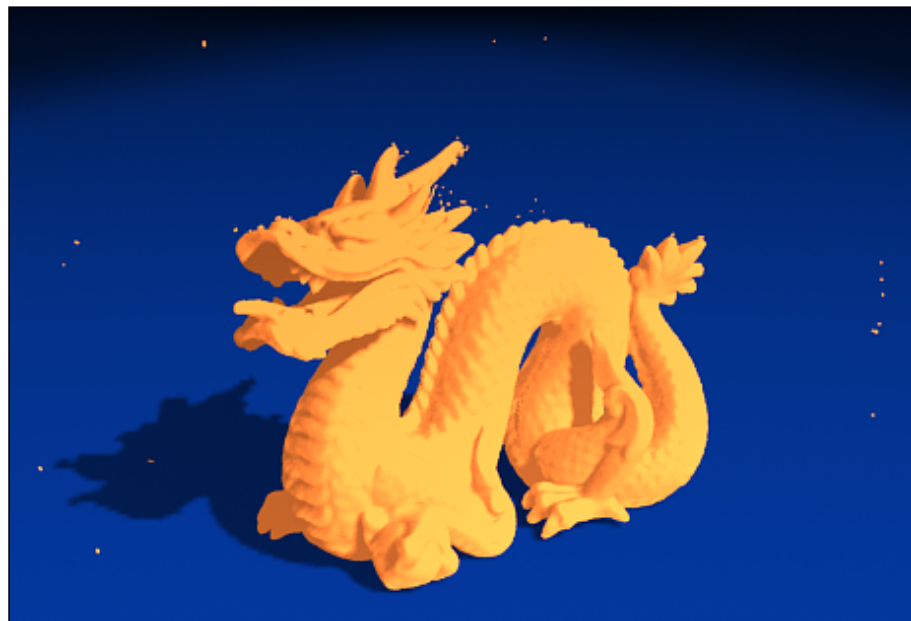
## 4.3 Discussion

Our implementation of the uniform MLS algorithm is effective for building a smooth consensus surface when the range data is obtained from a single range scanner. In cases when data from multiple sources is being combined, the adaptive MLS algorithm is necessary.

We estimate the  $\epsilon$  parameter from the sample point spacing in the input point cloud.



(a)



(b)

Figure 4.4: (a) Eigencrust reconstructions of the bunny model without (left) and with (right) MLS smoothing. (b) Sample points in the dragon model after MLS smoothing.

## Chapter 4. An Implementation of Moving Least Squares

---

This method only works when the magnitude of the noise in the point cloud is comparable to the sample point spacing. When noise in the point cloud is much larger than the sample point spacing, the width of the Gaussian weight functions should be determined by the noise characteristics of the scanner. Noise in the sample points can either be estimated from confidence values associated with sample points by some range scanners, or from the error in the alignment of scans during fine registration.

# Chapter 5

## Eigencrust

### 5.1 Introduction

There are two kinds of errors that cannot be handled by the MLS algorithm: outliers, which are spurious points far from the true surface, and, unsampled regions that are not accessible to scanning. A data set that suffers from both of these errors is shown in Figure 5.1. When these problems are severe, data arise for which no algorithm can construct an accurate, consistent, watertight model of an object's surface solely by examining local regions of a point cloud independently. A successful algorithm must take a global view. We propose a Delaunay-based reconstruction algorithm to handle outliers and undersampled regions. The general technique we use to produce a surface from a point cloud is well known: compute the Delaunay tetrahedralization of the points, then label each tetrahedron **inside** or **outside**. (Recent advances in Delaunay software make it possible to tetrahedralize sets of tens of millions of points.) The output is a triangulated surface, composed of every triangular face where an **inside** tetrahedron meets an **outside** tetrahedron. This procedure guarantees that the output surface is *watertight*—it bounds a volume, and there is no route from the inside to the outside of the volume that does not pass through the surface.

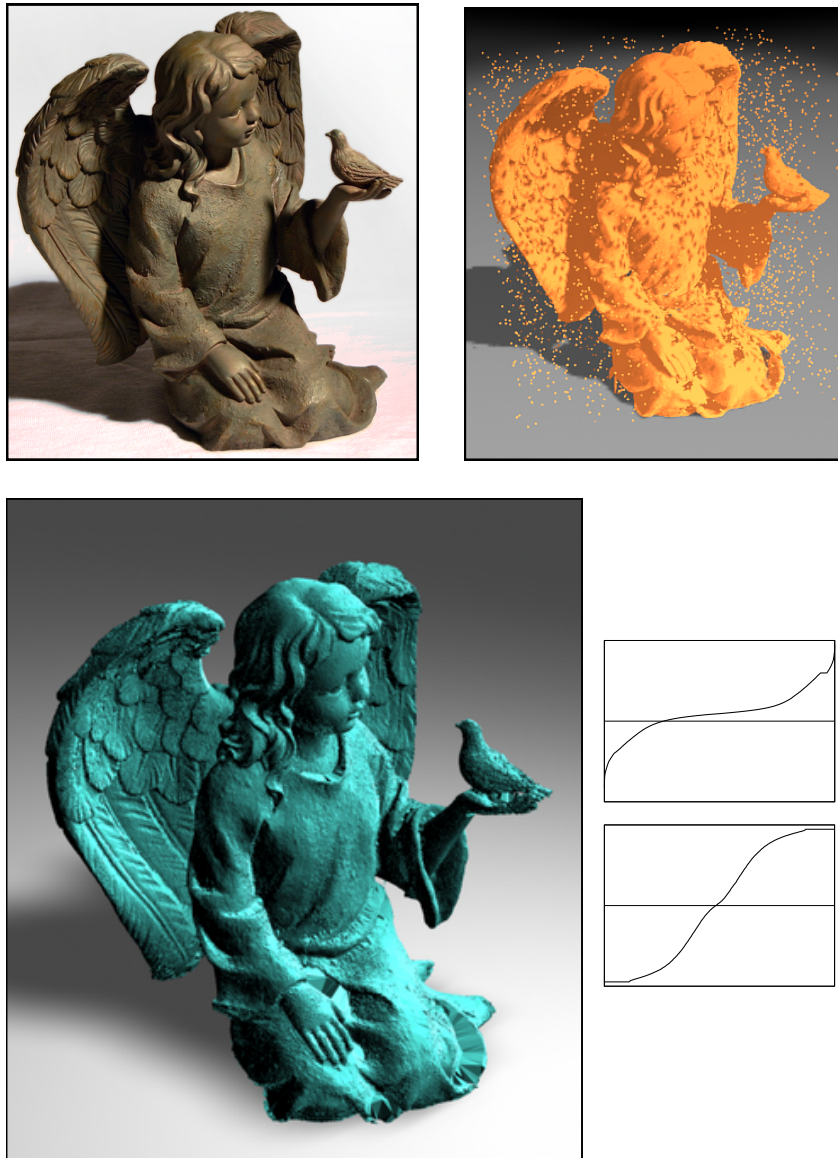


Figure 5.1: A watertight manifold surface triangulation reconstructed by our algorithm; a photograph of the source object; the point cloud input to the algorithm, with 4,000 artificial random outliers; and the sorted components of the two eigenvectors used for the reconstruction. 2,008,414 points; 12,926,063 tetrahedra; 3,605,096 output triangles; genus 14; 437 minutes reconstruction time, including 13.5 minutes to tetrahedralize the point cloud, and 157 minutes and 265 minutes to compute the first and second eigenvectors, respectively.

Our innovation is to introduce the techniques of spectral partitioning and normalized cuts into surface reconstruction. These techniques are used heavily for tasks such as image segmentation and parallel sparse matrix arithmetic, where partitioning decisions based on a global view of an image or a matrix can outperform local optimization algorithms. Although the global optimization step makes our algorithm slower than some competitors, it reconstructs many models that other methods cannot.

We create a graph that represents the tetrahedra. A spectral partitioner slices it into two subgraphs, an `inside` subgraph and an `outside` subgraph. Because the spectral partitioner has a global view of the point set, it is effective at identifying the triangular faces that are most likely to lie at the interface between an object and the space around it. Although the global optimization step makes our algorithm slower than some competitors, it reconstructs many models that other methods cannot.

## 5.2 Related work

### 5.2.1 Delaunay-based surface reconstruction

There has been much work on reconstructing surfaces from point clouds using the Delaunay tetrahedralization of the sample points. The idea of labeling each Delaunay tetrahedron `inside` or `outside`, then extracting a surface using the labels, appears in an early paper of Boissonat [18].

The  $\alpha$ -shape proposed by Edelsbrunner, Kirkpatrick, and Seidel [37] is a parameterized construction that associates a subset of the Delaunay tetrahedralization with an unorganized point set. A simplex is in the  $\alpha$ -shape if it has an circumsphere of radius at most  $\alpha$  that does not contain any sample points. Edelsbrunner and Mücke [38] use the  $\alpha$ -shape spectrum of the sample points for surface reconstruction. Bajaj, Bernardini, and Xu [11] use weighted  $\alpha$ -shapes as the first step in a reconstruction pipeline.

The *crust* algorithm of Amenta and Bern [3] was the first surface reconstruction algorithm with theoretical guarantees. The cocone algorithm of Amenta, Choi, Dey, and Leekha [5] simplified the crust algorithm and also proved that the reconstructed surface is homeomorphic to the sampled surface. The idea of labeling Delaunay tetrahedra also appears in the Tight Cocone algorithm of Dey and Goswami [33] and in the Powercrust algorithm of Amenta, Choi, and Kolluri [6] (with cells of a power diagram replacing tetrahedra). Both Tight Cocone and Powercrust avoid the manifold extraction step that is required by the crust and cocone algorithms.

A recent advance is the Robust Cocone algorithm of Dey and Goswami [34], a Delaunay reconstruction algorithm that is provably robust against small coordinate errors. Their algorithm is not robust against undersampling or outliers, and in fact is easily defeated by undersampling. We observe that a variant of our spectral algorithm could be used to help the Robust Cocone algorithm to label tetrahedra when the sample set is not dense enough for the Robust Cocone's original labeling algorithm to succeed.

Different branches of algorithms have different advantages. The main advantages of the Delaunay algorithms are the ease with which they exactly interpolate the sample points (except the samples judged to be erroneous outliers); the effortlessness with which they obtain watertight surfaces; the ease with which they adapt the density of the triangles to match the density of the points (unlike marching cubes), for models whose point density varies greatly from region to region; and the theoretical apparatus that makes it possible to prove the correctness of some reconstruction algorithms on well-sampled smooth surfaces.

### 5.2.2 Spectral Partitioning

Spectral methods for partitioning graphs were introduced by Hall [47] and Fiedler [39] and popularized by Pothen, Simon, and Liou [72]. They are used for tasks such as image segmentation, circuit layout, document clustering, and sparse matrix arithmetic on parallel



computers. The goal of graph partitioning is to cut a graph into two subgraphs, each roughly half the size of the original graph, so that the number of cut edges is small, or so that the total weight of the cut edges is small (if each edge is assigned a numerical weight). There are many ways to formulate the graph partitioning problem, which differ in how they trade off the weight of the cut against the balance between the two subgraphs. Most formulations are NP-hard, so practical partitioning algorithms (including spectral methods) are heuristics that try to find an approximate solution.

One of the most effective formulations of spectral partitioning is the *normalized cuts* criterion of Shi and Malik [80], which is particularly effective at trading off subgraph balance against cut weight. We make a simple modification to the Shi–Malik algorithm (closely related to a technique of Yu and Shi [85]) that greatly improves our surface reconstruction algorithm’s speed and the quality of the surfaces it produces. Ng, Jordan, and Weiss [66] analyze the normalized cut algorithm and also present a spectral clustering procedure to divide a graph into  $k(\geq 2)$  components. When the weight of the inter-cluster edges is much smaller compared to the weight of edges inside each cluster, they prove that that spectral clustering algorithm will recover the correct clusters.

## 5.3 Spectral Surface Reconstruction

### 5.3.1 Delaunay triangulation

The Delaunay triangulation is a fundamental geometric construction that has been well studied in computational geometry. See Fortune [42] for a nice survey of the properties of the Delaunay triangulation and its geometric dual, the Voronoi diagram. Although Delaunay triangulations and Voronoi diagrams are defined in any dimension, we are mainly interested in three-dimensional Delaunay triangulations. In three-dimensions, geometric duality maps each Delaunay tetrahedron to a Voronoi vertex and each Delaunay face to a

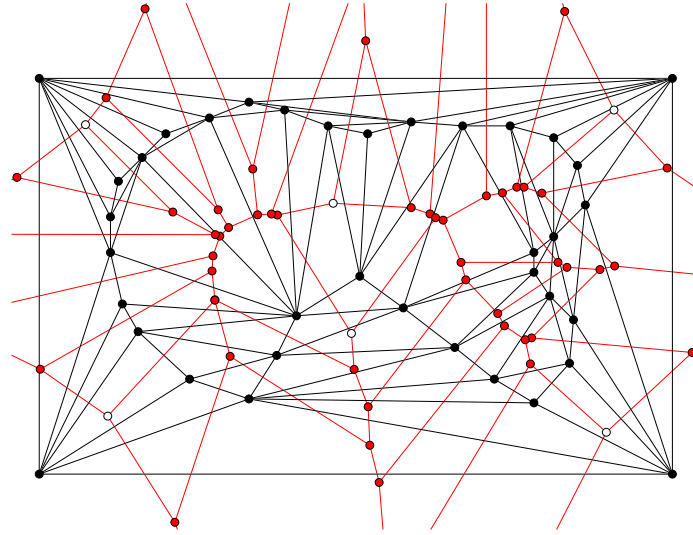


Figure 5.2: The Delaunay triangulation (black) and Voronoi diagram (red) of a set of points sampled from a closed curve.

Voronoi edge.

Our algorithm begins with a set  $S$  of sample points in space. Let  $S^+$  be the set  $S$  augmented with eight bounding box vertices, the corners of a large cube that encloses the sample points. (The width of the cube should be much greater than the diameter of  $S$ , so that no sample point lies near any side of the cube). Let  $T$  be the Delaunay tetrahedralization of  $S^+$ . Figure 5.2 shows a two-dimensional example. Let  $Q$  be the Voronoi diagram of  $S^+$ . For each tetrahedron  $t$  in the tetrahedralization  $T$ , there is a dual vertex  $v$  of the Voronoi diagram  $Q$ , and  $v$  is the center of the sphere that circumscribes  $t$ . By construction, the circumscribing sphere of  $t$  does not contain any of the sample points in  $S$ .

### 5.3.2 Overview of the reconstruction algorithm

The goal is to label each tetrahedron—or equivalently, each Voronoi vertex—inside or outside. Our algorithm labels the Voronoi vertices in two stages. Each stage forms a graph and partitions it.

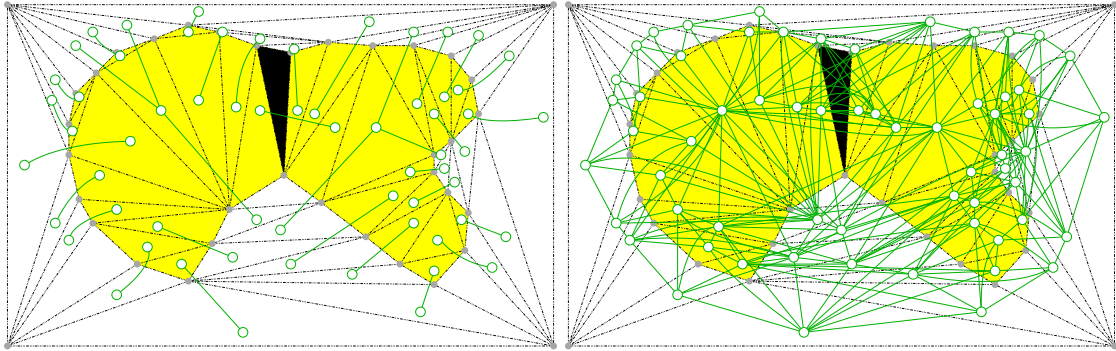


Figure 5.3: Left: the negatively weighted edges of the pole graph  $G$  (green), before the bounding box triangles are collapsed into a single supernode. Yellow triangles are the duals of poles labeled inside by the first stage of spectral partitioning. The black triangle does not dualize to a pole; it is labeled inside by the second partitioning stage. Right: the positively weighted edges of  $G$ .

In the first stage, our algorithm labels a subset of the Voronoi vertices called the *poles*, following Amenta and Bern [3]. We form a graph  $G$ , called the *pole graph*, whose nodes represent the poles. See Figure 5.3 for a two-dimensional example. The edges of  $G$  are assigned numerical weights that reflect the likelihood that certain pairs of poles are on the same side of the unknown surface that we wish to reconstruct.

The graph  $G$  is represented by a *pole matrix*  $L$ . We partition the poles of  $G$  by finding the eigenvector  $x$  that corresponds to the smallest eigenvalue of a generalized eigensystem  $Lx = \lambda Dx$ , and using that eigenvector to cut the graph into two pieces, the *inside* and *outside* subgraphs. Thus we label each pole *inside* or *outside*.

In the second stage, we form another graph  $H$  whose nodes represent the Voronoi vertices that are *not* poles, and partition  $H$  to label all the Voronoi vertices (equivalently, the tetrahedra) that were not labeled in the first stage. The goal of the second stage is different from the goal of the first: the non-poles are somewhat ambiguous—most of them could arguably be either *inside* or *outside*—so the partitioner tries to assign them labels that will yield a relatively smooth surface with low genus.

Now all the Voronoi vertices have labels, so all the tetrahedra of  $T$  have labels. The

algorithm outputs a surface triangulation consisting of every triangular face of  $T$  where an **inside** tetrahedron meets an **outside** tetrahedron. If the points in  $S$  are sampled densely enough from a simple closed surface, then the triangulation approximates the surface well.

If all the tetrahedra adjoining a sample point are labeled **outside** (or all **inside**), the point does not appear in the reconstructed surface triangulation. In Section 5.4, we see that this effect provides our algorithm with effective and automatic outlier removal. No other effort to identify outliers is required.

Why are the Voronoi vertices labeled in two separate stages? Because the non-poles are ambiguous, they tend to “glue” the inside and outside tetrahedra together. If they are included in the first partitioning stage, the graph partitioner is much less successful at choosing the right labels, and runs more slowly too. A two-stage procedure produces notably better and faster results.

We have chosen the graphs’ edge weights (by trial and error) so the algorithm tries to emulate the provably correct Cocone algorithm of Amenta et al. [5] when there is neither noise nor outliers, and the sampling requirements of the Cocone algorithm are met. Our algorithm usually returns significantly different results only under conditions where the Cocone algorithm has no guarantee of success.

Our algorithm includes three optional steps. After the first partitioning stage, we can identify some tetrahedra that may be mislabeled due to noise, and remove their labels (so new labels are assigned during the second stage). This step improves the resilience of our algorithm to measurement errors. After the second partitioning stage, we can convert the surface to a manifold (if it is not one already) by relabeling some **inside** tetrahedra **outside**. After the final surface recovery step, we can smooth the surface to make it more useful for rendering and simulation.

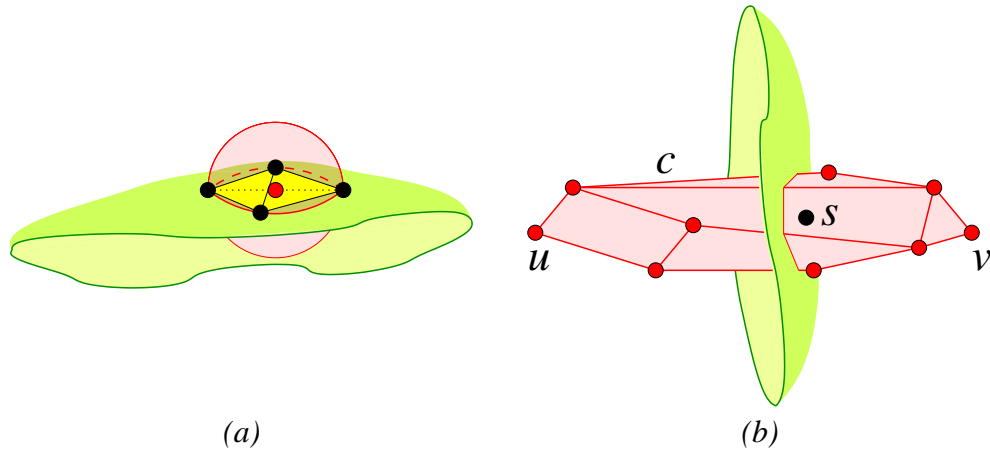


Figure 5.4: (a) No matter how finely a surface is sampled, tetrahedra (yellow) can appear whose circumscribing spheres are centered on or near the surface (green) being recovered. (b) The Voronoi cell (pink) of a sample point  $s$ . The poles of  $s$ —the Voronoi vertices  $u$  and  $v$ —typically lie on opposite sides of the surface being recovered, especially if the cell is long and thin.

### 5.3.3 The Pole Graph

Imagine that we form a graph whose nodes represent the vertices of the Voronoi diagram  $Q$  (and their dual tetrahedra in  $T$ ), and whose edges are the edges of  $Q$  (omitting the edges that are infinite rays). Suppose we then assign appropriate weights to the edges, and partition the graph into inside and outside subgraphs.

Unfortunately, this choice leads to poor results. The main difficulty is that the Delaunay tetrahedralization  $T$  invariably includes flat tetrahedra that lie in the surface we are trying to recover, as Figure 5.4(a) illustrates. These tetrahedra are not eliminated by sampling a surface extremely finely; they are a natural occurrence in Delaunay tetrahedralizations. Many of them could be labeled inside or outside equally well. They cause trouble because they can form strong links with both the tetrahedra inside an object and the tetrahedra outside an object, and thus prevent a graph partitioner from finding an effective cut between the inside and outside tetrahedra.

To solve this problem, Amenta and Bern [3] identify special Voronoi vertices called

*poles*. Poles are Voronoi vertices that are likely to lie near the medial axis of the surface being recovered. The Voronoi vertices whose duals are the troublesome flat tetrahedra are rarely poles, as the problem tetrahedra lie near the object surface, not near the medial axis.

Each sample point  $s$  in  $S$  can have two poles. Let  $c$  be the Voronoi cell of  $s$  in  $Q$  (i.e. the region of space composed of all points that are as close or closer to  $s$  than to any other sample point in  $S^+$ ). See Figure 5.4(b) for an example. The Voronoi cell  $c$  is a convex polyhedron whose vertices are Voronoi vertices. It is easy to compute the vertices of  $c$ , because they are the centers of the circumscribing spheres of the tetrahedra of  $T$  that have  $s$  for a vertex.

Let  $u$  be the vertex of  $c$  furthest from  $s$ ;  $u$  is considered a pole of  $s$ . Let  $v$  be the vertex of  $c$  furthest from  $s$  for which the angle  $\angle usv$  exceeds  $90^\circ$ ;  $v$  is also considered a pole of  $s$ . Figure 5.4(b) illustrates the two poles of a typical sample point. The eight bounding box vertices in  $S^+$  are not considered to have poles. Let  $V$  be the set of all the poles of all the samples in  $S$ .

Amenta and Bern show that in the absence of noise, the tetrahedra that are the duals of the poles are likely to extend well into the interior or exterior of the object whose surface is being recovered. The tetrahedra whose duals are not poles often lie entirely near the surface, as Figure 5.4(a) shows, so it is ambiguous whether they are inside or outside the object.

Our algorithm identifies the set  $V$  of all poles, then constructs a sparse *pole graph*  $G = (V, E)$ . The set  $E$  of edges is defined as follows. For each sample  $s$  with poles  $u$  and  $v$ ,  $(u, v)$  is an edge in  $E$  as shown in Figure 5.5. For each pair of samples  $p, s$  such that  $(p, s)$  is an edge of the Delaunay tetrahedralization  $T$ , let  $u$  and  $v$  be the poles of  $s$ , and let  $r$  and  $t$  be the poles of  $p$ ; then the edges  $(r, u)$ ,  $(r, v)$ ,  $(t, u)$ , and  $(t, v)$  are all edges of  $E$  as shown in Figure 5.5. Every pole is the dual of a tetrahedron, so tetrahedra that adjoin each other are often linked together in  $G$ , whereas tetrahedra that are not close to each other are not linked.

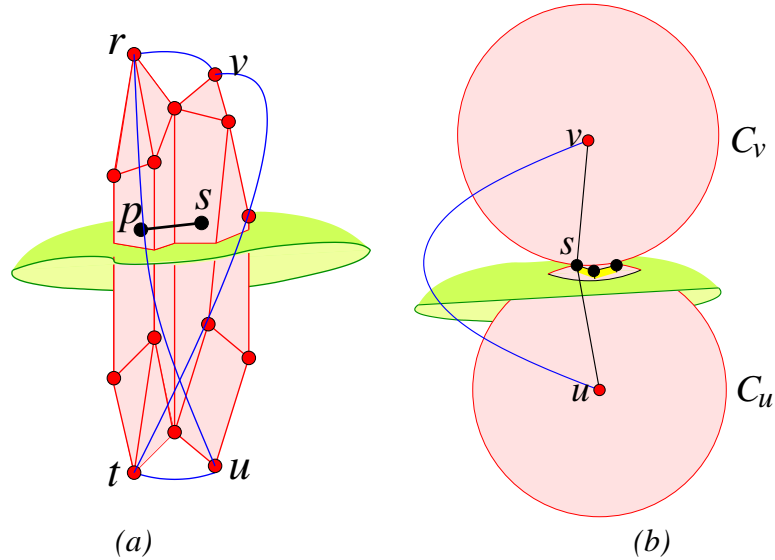


Figure 5.5: (a) For each pair of samples  $p, s$  that share an edge in the Delaunay triangulation, we add edges with positive weights between the poles of  $s$  and the poles of  $p$  to the pole graph. (b) The poles of sample  $s$ —the Voronoi vertices  $u$  and  $v$ —typically lie on opposite sides of the surface being recovered. We therefore add a negative edge between them in the pole graph.

We assign edge weights in a heuristic manner based on several observations of Amenta and Kolluri [8]. If  $S$  is sampled sufficiently densely from a smooth surface, the Voronoi cells are long and thin, and the longest dimension of each cell is oriented roughly perpendicular to the surface, as Figure 5.4(b) depicts. Of course, point sets that arise in practice are often not sampled densely enough, but if a sample  $s$  has a long, thin Voronoi cell  $c$ , the likelihood is high that its poles  $u$  and  $v$  are on opposite sides of the surface. Therefore, we assign the edge  $(u, v)$  a negative weight, to indicate that if one of  $u$  or  $v$  is labeled inside, the other should probably be labeled outside.

Let  $t_u$  and  $t_v$  be the tetrahedra in  $T$  whose duals are  $u$  and  $v$ . Let  $C_u$  and  $C_v$  be the circumscribing spheres of  $t_u$  and  $t_v$ . The spheres  $C_u$  and  $C_v$  intersect at an angle  $\phi$ . Amenta and Kolluri show that if  $\phi$  is small, as illustrated in Figure 5.6(a), then  $c$  is quite long and thin, and the likelihood is high that  $t_u$  and  $t_v$  lie on opposite sides of the surface. If  $\phi$  is close

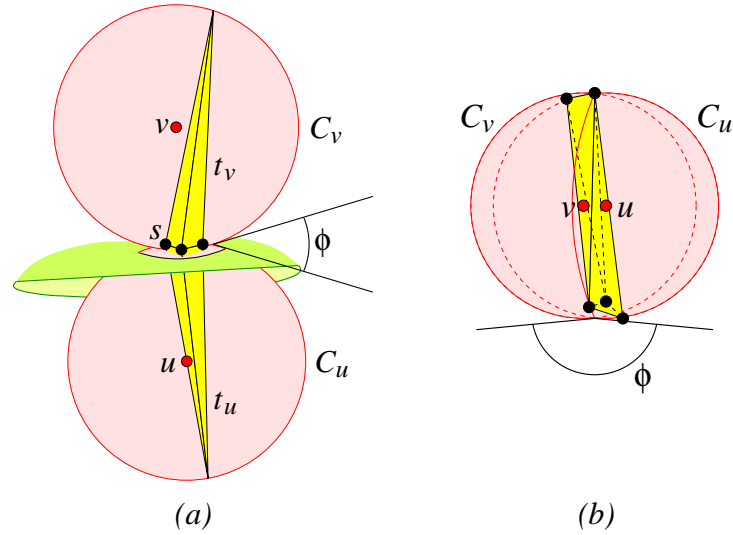


Figure 5.6: (a) Small angles of intersection between circumscribing spheres may indicate that two tetrahedra are on opposite sides of the surface being recovered. (b) Large angles of intersection usually indicate that two tetrahedra are on the same side of the surface.

to  $180^\circ$ , then  $c$  is relatively round, and it is unsafe to conclude that  $t_u$  and  $t_v$  lie on opposite sides. We assign  $(u, v)$  a weight of  $w_{u,v} = -e^{4+4\cos\phi}$ , so that  $w_{u,v}$  is most negative when  $\phi$  is closest to zero. For two spheres that intersect at an angle  $\phi$ , we can compute  $\cos\phi$  using the formula,

$$\cos\phi = \frac{1}{2r_u r_v} (d^2(u, v) - r_u^2 - r_v^2).$$

Here,  $d(u, v)$  is the distance between the points  $u, v$  and  $r_u, r_v$  are the radii of the two spheres  $C_u, C_v$  respectively.

We assign positive weights to the other edges in  $E$ . These weights are the glue that hold proximal tetrahedra together and ensure that  $G$  is likely to be cut only near the original surface, where the glue is weakest. Let  $(u, v)$  be an edge of  $E$  that is not assigned a negative weight—thus, there is a Delaunay edge  $(s, p)$  for which  $u$  is a pole of  $s$  and  $v$  is a pole of  $p$ , but there is no sample point  $t$  whose poles are  $u$  and  $v$ . Again, let  $t_u$  and  $t_v$  be the tetrahedra that are dual to  $u$  and  $v$ , and let  $C_u$  and  $C_v$  be their circumscribing spheres, which intersect



at an angle  $\phi$ . We assign  $(u, v)$  a weight of  $w_{u,v} = e^{4-4\cos\phi}$ . Amenta and Kolluri show that if  $\phi$  is close to  $180^\circ$ , as illustrated in Figure 5.6(b), then  $u$  and  $v$  are likely to lie on the same side of the surface, so we use a large, positive edge weight. If  $\phi$  is close to  $0^\circ$ , we choose a small edge weight, so that  $u$  and  $v$  are not strongly glued together. It may occur that the spheres  $C_u$  and  $C_v$  do not intersect at all, in which case we remove the edge  $(u, v)$  from  $E$ .

We could partition the graph  $G$  directly, but we know *a priori* that certain tetrahedra must be labeled **outside**, and it is advantageous to fix their labels prior to the partitioning step. Let  $O$  be the set of poles whose dual tetrahedra are known to be outside the object being reconstructed. We take advantage of this information by forming a new graph  $G'$  that is similar to  $G$ , but the poles in  $O$  are collapsed into a single *supernode*  $z$ . If  $u$  and  $u'$  are poles in  $O$ , and  $(u, u')$  is an edge of  $G$ , the edge is eliminated (not present in  $G'$ ). If  $v$  is a pole in  $G$  that is not in  $O$ , then in the new graph  $G'$ , the edge  $(z, v)$  has weight  $w_{z,v} = \sum_{u \in O} w_{u,v}$ . Collapsing **outside** poles into a single supernode makes the spectral partitioner faster and more accurate.

What poles does  $O$  contain? There are several types of tetrahedra that can be labeled **outside** prior to the partitioning step.

- Any tetrahedron with a vertex of the cubical bounding box must be **outside**.
- If the point samples were acquired by a laser range finder, the tetrahedra that lie between the laser source and any sample point it recorded must be **outside**. Of course, there may be measurement errors in the positions of the sample points and the laser source, so we recommend only labeling those tetrahedra that the laser penetrated more deeply than some tolerance depth, multiple times.
- For particularly difficult reconstructions, a user may visually identify specific points in space that are outside the object. The tetrahedra containing these points are labeled **outside**. Collapsing just one such tetrahedron into the **outside** supernode can change

the labeling of many other tetrahedra, so this is occasionally a practical option. (No example in this paper takes advantage of this possibility.)

Optionally, the algorithm may create an **inside** supernode as well, with a large negative weight connecting the **outside** and **inside** supernodes. This is particularly useful for reconstructing one-sided building facades or other sets of sample points that do not represent closed volumes. For this purpose, the tetrahedra adjoining the front face of the bounding box are labeled **outside**, and the tetrahedra adjoining the back face are **inside**.

### 5.3.4 Spectral Partitioning

From the modified pole graph  $G'$ , we construct a *pole matrix*  $L$ . ( $L$  is often called the *Laplacian matrix*, but our use of negative weights makes that name a misnomer.)  $L$  is sparse and symmetric, and has one row and one column for each node of the graph  $G'$ . For each edge  $(v_i, v_j)$  of  $G'$  with weight  $w_{v_i, v_j}$ , the pole matrix  $L$  has the components  $L_{ij} = L_{ji} = -w_{v_i, v_j}$ . (Positive, “attractive” weights become negative matrix components, and negative, “repulsive” weights become positive matrix components.) The diagonal components of  $L$  are the row sums  $L_{ii} = \sum_{j \neq i} |L_{ij}|$ . The remaining components of  $L$  (the off-diagonal components not represented by an edge of  $G'$ ) are zero. If  $G'$  is connected (which is always true in our application) and includes at least one edge with a negative weight,  $L$  is guaranteed to be positive definite.

The spectral analysis of a Laplacian matrix or pole matrix can be intuitively understood by analogy to the vibrational behavior of a system of masses and springs. Imagine that each node of  $G'$  represents a mass located in space, and that each edge represents a spring connecting two masses. Positive edge weights imply attractive forces, and negative edge weights imply repulsive forces. The eigensystem of  $L$  represents the transverse modes of vibration of the mass-and-spring system. The lowest-frequency modes give clues as to where the graph can be cut most effectively: the **inside** masses are usually found vibrating

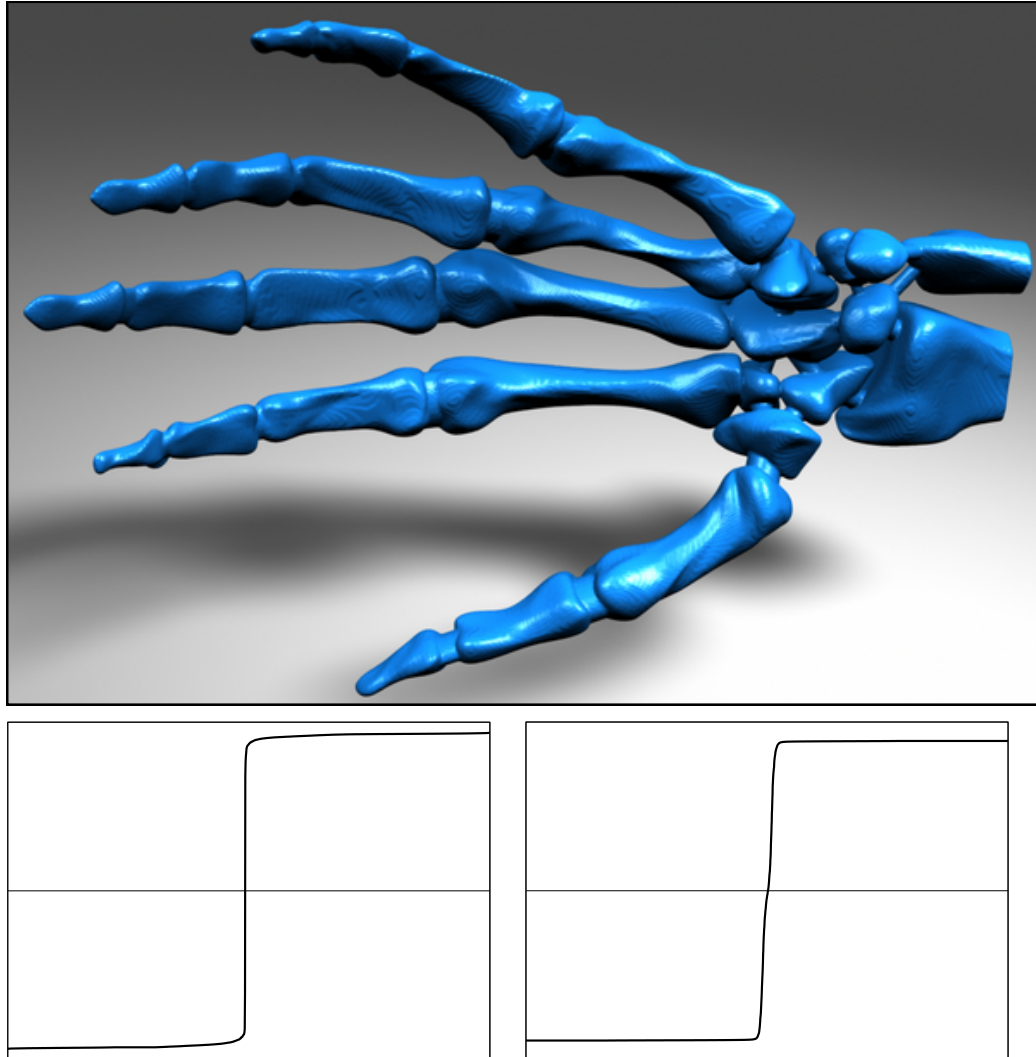


Figure 5.7: Watertight skeleton surface, and the sorted components of the eigenvectors computed during the two partitioning stages. The points are densely sampled from a smooth surface, so the eigenvectors are polarized. 327,323 input points; 2,334,597 tetrahedra; 654,596 output triangles; genus zero; 12.3 minutes reconstruction time, including 2.8 minutes for the tetrahedralization, and 5.1 minutes and 4.2 minutes for the eigenvectors.

out of phase with the **outside** masses.

We take advantage of this observation by finding the eigenvector  $x$  associated with the smallest eigenvalue  $\lambda$  of the generalized eigensystem  $Lx = \lambda Dx$ , where  $D$  is a diagonal matrix whose diagonal is identical to that of  $L$ . Because  $L$  is sparse, we compute the eigenvector  $x$  using the iterative Lanczos algorithm [59; 72]. Each component of the eigenvector  $x$  corresponds to one column of  $L$ , and therefore to one node of  $G'$ , to one pole of  $Q$ , and to one tetrahedron of  $T$ . (The exception is the component of  $x$  that corresponds to the supernode  $z$ .)

Figure 5.7 shows a reconstruction of a skeletal hand and the eigenvectors computed during the two partitioning stages. The components of the eigenvectors are sorted in increasing order. When our method is applied to smooth, well-sampled surfaces, we find that the eigenvector  $x$  is relatively polarized: most of its components are clearly negative or clearly positive, with few components near zero. However, noisy models produce more ambiguous labels—see Figures 5.1, 5.11, and 5.13. One of the components of  $x$  corresponds to the **outside** supernode  $z$ . Suppose this component is positive; then the nodes of  $G'$  whose components are positive are labeled **outside**, and the nodes whose components are negative are labeled **inside**. (If the component corresponding to  $z$  is negative, reverse this labeling.)

This procedure differs from the usual formulation of normalized cuts [80] in one critical way. The standard normalized cuts algorithm does not use negative weights, so its Laplacian matrix  $L$  is positive indefinite—it has one eigenvalue of zero, with an associated eigenvector whose components are all 1. Therefore, the standard formulation uses the eigenvector associated with the second-smallest eigenvalue (called the *Fiedler vector*) to dictate the partition. Our pole graph has negative weights, our pole matrix is positive definite, and we use the eigenvector associated with the smallest eigenvalue to dictate the partition. Because the negative weights encode information about tetrahedra that are likely to be on opposite sides of a surface, we find that this formulation reconstructs better sur-

faces, and permits us to calculate the eigenvector much more quickly than normalized cuts in their standard form.

Spectral partitioning solves a real valued approximation to the normalized-cut problem, which is NP-complete [80]. It is interesting to note that the first eigenvector  $x$  of the pole matrix minimizes the following summation:

$$\frac{1}{\sum_i x^2(i)D(i)} \left( \sum_{w_{v_i v_j} > 0} (x(i) - x(j))^2 w_{v_i v_j} + \sum_{w_{v_i v_j} < 0} (x(i) + x(j))^2 w_{v_i v_j} \right). \quad (5.1)$$

Consider an ideal pole graph in which all the **inside** poles (and **outside** poles) form a connected subgraph of  $G$ , there is no positive weight edge between an **inside** pole and an **outside** pole, and there is at least one negative weight edge between an **inside** pole and an **outside** pole. In the first eigenvector  $x$  of this ideal pole graph, the components of the **inside** poles and the **outside** poles will have the same value with opposite sign. The eigenvector of the skeletal hand shown in Figure 5.7 illustrates the clean separation of **inside** poles and **outside** poles. When there are no negative weight edges, all the components in  $x$  will have the same value.

The Lanczos algorithm is an iterative solver which typically takes about  $O(\sqrt{n})$  iterations to converge, where  $n$  is the number of nodes in  $G'$ . ( $L$  is an  $n \times n$  matrix.) The running time also depends on the distribution of eigenvalues of the generalized eigensystem, in a manner that is not simple to characterize and is not related to the condition number. The most expensive operation in a Lanczos iteration is matrix-vector multiplication, which takes  $O(n)$  time because  $L$  is sparse.

For noisy models, the  $O(n\sqrt{n})$  running time is justified. Because spectral partitioning searches for a cut that is “good” from a global point of view, it elegantly patches regions that are undersampled or not sampled at all (see Figure 5.9). Measurement errors may muddy up the edge weights, but a good deal of noise must accumulate globally before

the reconstruction is harmed (see Section 5.4). Many other algorithms are faster, and they are preferable for clean models. But most surface reconstruction algorithms are fooled by outliers and noise, and many leave holes in the reconstructed surface or make serious errors in deciding how to patch the holes.

### 5.3.5 Correcting Questionable Poles

An optional step, strongly recommended for noisy models, removes labels whose accuracy is questionable, so that some poles will be relabeled during the second step. Most poles lie near the medial axis of the original object, and dualize to tetrahedra that extend deeply into the object's interior. However, random measurement errors in the sample point coordinates can create spurious poles that are closer to the surface than the medial axis. Fortunately, a spurious pole is usually easy to recognize: it dualizes to a small tetrahedron that is entirely near the object surface.

Laser range finders typically sample points on a square grid. Using the coordinates of those samples, we compute a grid spacing  $\ell$  equal to the median length of the diagonals of the grid squares. Any labeled tetrahedron whose longest edge is less than  $4\ell$  is suspicious, so we remove its label. The grid resolution is typically small compared to the object's features, so poles near the medial axis are unaffected.

We find that this step consistently leads to more accurate labeling of noisy models. It is unnecessary for smooth, noise-free models.

### 5.3.6 Labeling the Remaining Tetrahedra

The first partitioning stage labels each tetrahedron whose dual Voronoi vertex is a pole, and labels some of the other tetrahedra too (such as those touching the bounding box). Many tetrahedra with more ambiguous identities remain unlabeled. To label them, we construct and partition a second graph  $H$ . The goal of the second partitioning stage is to label the

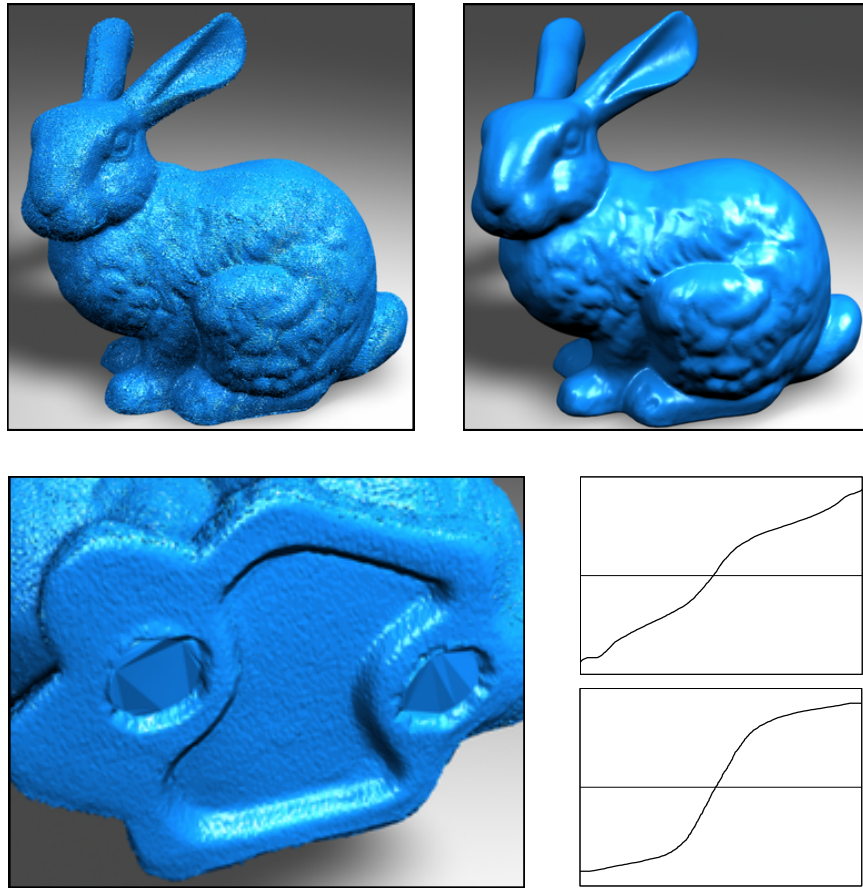


Figure 5.8: Left: the spectral reconstruction of the bunny (with no added noise) patches two unsampled holes in the bottom of the bunny. 362,272 input points; 2,283,480 tetrahedra; 679,360 output triangles; genus zero; 19.1 minutes reconstruction time, of which 17.5 minutes is spent computing the eigenvectors.

ambiguous tetrahedra in a manner that produces a relatively smooth surface of low genus.

The graph  $H$  has two supernodes, representing all the tetrahedra that were labeled inside and outside, respectively, during the first stage.  $H$  also has one node for each unlabeled tetrahedron. If two unlabeled tetrahedra share a triangular face, they are connected by an edge of  $H$ . If an unlabeled tetrahedron shares a face with a labeled tetrahedron, the former is connected by an edge to one of the supernodes.

We have tried a variety of ways of assigning weights to the edges of  $H$ . We obtained

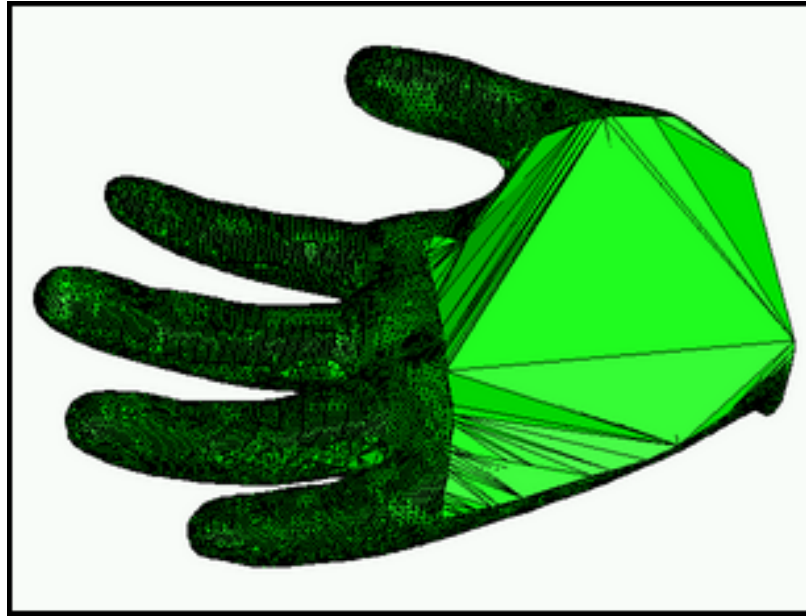


Figure 5.9: Spectral reconstruction of a point cloud with a large unsampled region, and its first partitioning eigenvector. 23,767 input points; 47,536 output triangles; 80 seconds reconstruction time.

our best results (surfaces with the fewest handles) by choosing each weight to be the aspect ratio of the corresponding triangular face, defined as the face’s longest edge length divided by its shortest edge length. These weights encourage the use of “nicely shaped” triangles in the final surface, and discourage the appearance of “skinny” triangles (whose large edge weights resist cutting).

$H$  has just one negative edge weight: an edge connecting the **inside** and **outside** supernodes, whose weight is the negation of the sum of all the other edge weights adjoining the supernodes. The negative edge ensures that the supernodes are assigned opposite signs in the eigenvector.

We partition  $H$  as described in Section 5.3.4. A tetrahedron is labeled **inside** if the corresponding value in the eigenvector has the same sign as the **inside** supernode, and vice versa.

As an alternative to this stage, we can label power cells of the poles instead of labeling



Delaunay tetrahedra—in other words, we replace the Powercrust’s pole labeling algorithm with our spectral pole labeling algorithm. In the Powercrust algorithm, *every* power cell is the dual of a pole, so there is nothing left to label after the first partitioning stage.

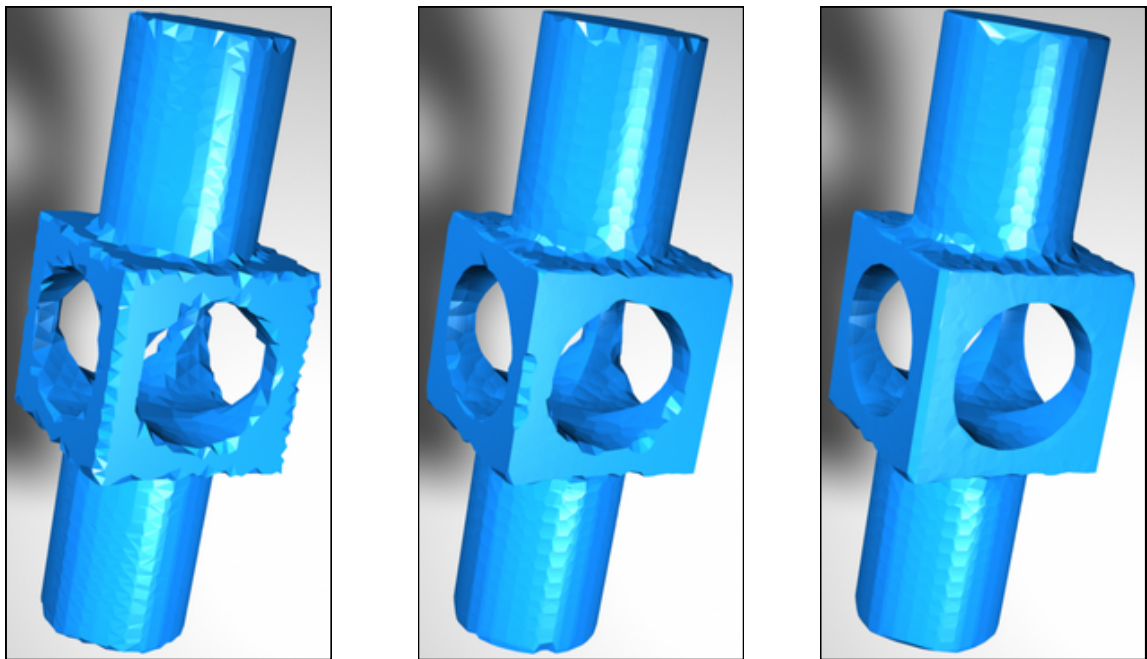
Figure 5.10 shows that spectral tetrahedron labeling is poor at capturing sharp corners, and the Powercrust algorithm is much better, but the hybrid algorithm is even more effective. Spectral partitioning labels power cells better than the original Powercrust. The hybrid spectral Powercrust algorithm shares the Powercrust’s advantage of recovering sharp corners well, but it also shares its disadvantage of increasing the number of vertices many-fold. The number of vertices in each model is 4,100, 52,078, and 51,069, respectively.

### 5.3.7 Constructing Manifolds

An optional step searches for local topological irregularities that prevent the reconstructed surface from being a manifold, and makes the surface a manifold by relabeling selected tetrahedra from *inside* to *outside*.

These irregularities come in two types. First, consider any edge  $e$  of the Delaunay tetrahedralization  $T$ . The tetrahedra that have  $e$  for an edge form a ring around  $e$ . If the reconstructed surface is a manifold, there are three possibilities: the tetrahedra in the ring are all *outside*, they are all *inside*, or the ring can be divided into a contiguous strand of *inside* tetrahedra and a contiguous strand of *outside* tetrahedra. If the ring of tetrahedra around  $e$  do not follow any of these patterns—if there are two or more contiguous strands of *inside* tetrahedra in the ring—then we fix the irregularities by relabeling some of the tetrahedra from *inside* to *outside* so that only one contiguous strand of *inside* tetrahedra survives. The surviving strand is chosen so that it contains the *inside* tetrahedron that was assigned the largest absolute eigenvector component during the second partitioning stage.

The second type of irregularity involves any point sample  $s$  in  $S$ . If the reconstructed surface is a manifold, then the tetrahedra that have  $s$  for a vertex are either all *outside*,



Spectral

Powercrust

Hybrid

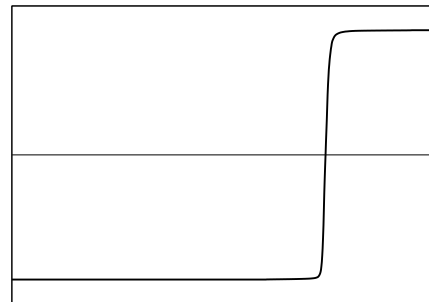
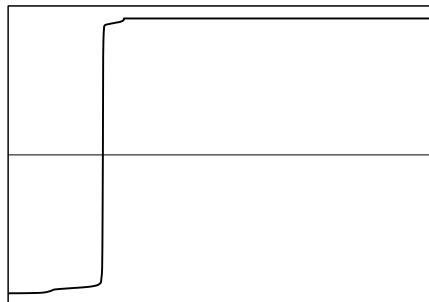


Figure 5.10: Reconstructions of a mechanical part by three algorithms. The hybrid algorithm uses only the first eigenvector, whereas the spectral tetrahedron labeling algorithm uses both.

all **inside**, or divided into one face-connected block of **outside** tetrahedra and one face-connected block of **inside** tetrahedra. A topological irregularity at  $s$  may take the form of two **inside** tetrahedra that have  $s$  for a vertex, but are not connected to each other through a path of face-connected **inside** tetrahedra all having  $s$  for a vertex. In this case, the **inside** tetrahedra adjoining  $s$  can be divided into two or more face-connected components. Only one of these **inside** components survives; we relabel the others **outside**. The surviving component is the one that contains a pole of  $s$ . (In the unlikely case that there are two such components, choose one arbitrarily.)

It is also possible to have two or more face-connected components of **outside** tetrahedra (and just one component of **inside** tetrahedra). Let  $W$  and  $X$  be two of them. We compute the shortest face-connected path from  $W$  to  $X$ , where the length of a path is defined to be the sum of the absolute eigenvector components of the **inside** tetrahedra on the path. The tetrahedra on the shortest path are relabeled **outside**.

These operations are repeated until no irregularity remains. The final surface is guaranteed to be a manifold. One can imagine that for a pathological model this procedure might whittle down the object to a few tetrahedra, but in practice it rarely takes an unjustifiably large bite out of an object.

### 5.3.8 Smoothing

Triangulated surfaces extracted from noisy models are bumpy. The final optional step is to use standard Laplacian smoothing [49] to remove the artifacts created by measurement errors in laser range finding, and to make the model more amenable to rendering and simulation. Laplacian smoothing visits each vertex in the triangulation in turn, and moves it to the centroid of its neighboring vertices. We performed five iterations of smoothing on the smoothed dragon and bunny in Figures 5.11 and 5.8.

## 5.4 Results

Our implementation uses our own Delaunay tetrahedralization software, and TRLAN<sup>1</sup>, an implementation of the Lanczos algorithm by Kesheng Wu and Horst Simon of the National Energy Research Scientific Computing Center.

Figure 5.11 illustrates the performance of the spectral algorithm on the Stanford dragon. The raw data exhibit random measurement errors and include natural outliers. Spectral reconstruction yields a watertight manifold surface, and removes all the outliers.

The model is too large for the Powercrust implementation, so we have applied the Powercrust labeling algorithm to the Delaunay tetrahedra. (We use our algorithm discussed in Section 5.3.6 to label the tetrahedra that are not poles, just like with our spectral algorithm.) The Powercrust labeling algorithm mislabels tetrahedra because of bad local decisions that the spectral algorithm averts.

Figure 5.13 shows how several algorithms degrade as randomly generated outliers are added to the input data. Even 1,200 random outliers have no influence on the spectral reconstruction except to affect how a hole at the base of the hand is patched. The Tight Cocone and Powercrust algorithms correctly reconstruct clean models, but they are incapacitated by relatively few outliers.

The fourth row of Figure 5.13 shows the degradation of several algorithms as increasing amounts of random Gaussian noise are added to the point coordinates of the Stanford bunny, which already includes measurement errors. The expression under each reconstruction is the variance of the Gaussian distribution used to produce additional noise, expressed in terms of the grid spacing  $\ell$  defined in Section 5.3.5. The spectral reconstruction remains a genus zero manifold when the added noise has variance  $2\ell$ , but begins to disintegrate as the measurement errors become notably larger than the resolution of the range data. With added noise of variance  $\ell$ , the Powercrust algorithm succeeds, but with variance  $2\ell$  the

---

<sup>1</sup> <http://www.nersc.gov/research/SIMON/trlan.html>

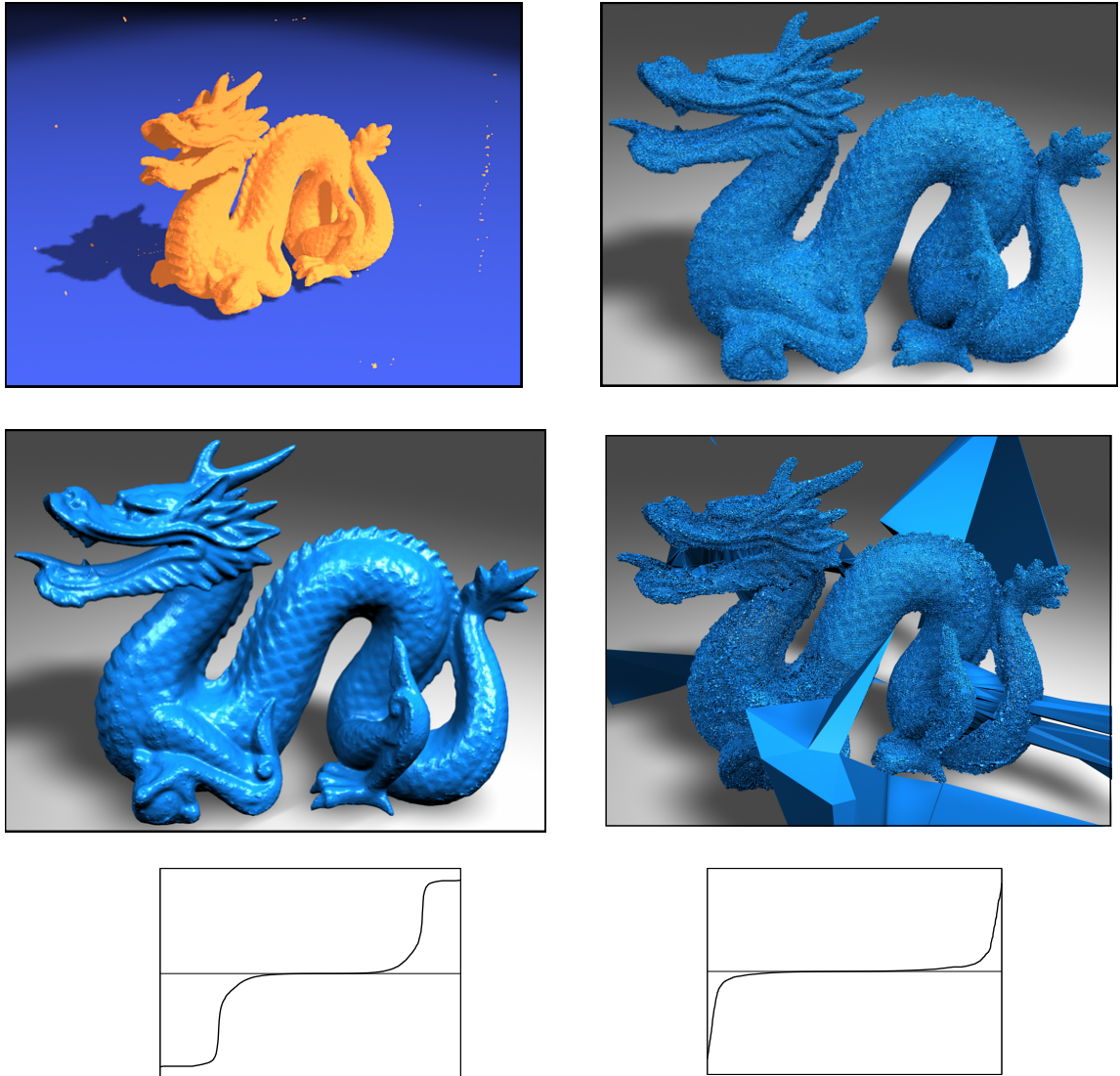


Figure 5.11: Reconstruction of the Stanford dragon model from raw data. The point cloud (upper left) has many outliers, which are automatically omitted from the spectrally recovered surface (upper right). The eigenvectors are less polarized than the eigenvectors in Figure 5.7, reflecting the labeling ambiguities due to measurement errors. The surface produced using Powercrust's labeling (middle right) fails to handle the outliers correctly. 1,769,513 input points; 11,660,147 tetrahedra; 2,599,114 surface triangles; genus 1; 197 minutes reconstruction time.

structure is full of holes. (It is a watertight surface, but what it bounds is Swiss cheese.) The Tight Cocone algorithm can only cope with noise of less than  $0.8\ell$  variance.

Figure 5.12 illustrates the three algorithms on a set of points densely sampled from the smooth splines of the Utah Teapot. The difficulties here are more subtle. The teapot's spout penetrates the body deeply enough to create an ambiguity that neither the Powercrust nor Tight Cocone algorithms solve correctly. The spectral reconstruction algorithm treats the interior spout sample points as outliers, and correctly omits them from the surface.

The eigencrust algorithm is effective on sparse point clouds. Figure 5.15 shows the eigencrust reconstruction of the schale data set which contains some thin regions that are sparsely sampled. Figure 5.16 shows how different algorithms degrade as sampling becomes more and more sparse. The first column is a uniformly sparse point cloud. In this case the Spectral and the powercrust algorithms correctly reconstruct the knot. In the second and third columns we linearly increase the sampling density from left to right. As shown on the right, both Powercrust and Tight cocone reconstructions begin to degrade before the spectral reconstruction.

### 5.4.1 Effect of MLS smoothing

The bunny model in Figure 5.8, and the dragon model in Figure 5.11 were reconstructed directly from the laser range data. Notice that the measurement noise and the errors in alignment make the eigenvectors ambiguous. Smoothing the point sampled by the MLS algorithm described in Chapter 2 reduces this ambiguity in the eigenvectors as shown in Figure 5.17. Computing the eigenvectors is faster, and the reconstructed models have less topological noise. Figures 5.18, 5.19, 5.20 and 5.21 show spectral reconstructions of the models listed in Table 4.1 after MLS smoothing.

Model	Points	Tetrahedra	Delaunay (Seconds)	Eigenvector (Seconds)	genus
Bunny	362272	2711705	53	487	0
Dragon	1769513	13159466	478	5263	1
Armadillo	1541522	11454212	235	2735	6
Buddha	1849576	12571346	306	5257	7
Angel	1889131	13346501	338	4625	9

Table 5.1: Eigencrust reconstruction times for the data sets in Table 4.1.

## 5.5 Discussion

Our modification to the normalized cut algorithm that adds negative weight edges into the pole graph works well only when the pole graph has two components that repel each other with negative weight edges. The spectral clustering algorithm proposed by Ng, Jordan, and Weiss [66] to partition a graph into  $k(\geq 2)$  components does not handle negative weight edges.

The spectral algorithm is not infallible. It occasionally creates unwanted handles—fourteen on the angel. The global eigenvector computation is slow. Tetrahedron-labeling algorithms do not reconstruct sharp corners well—observe how the teapot adjoins its spout. This problem can be overcome by labeling power cells rather than tetrahedra, at the cost of much larger model complexity. Either way, however, spectral surface reconstruction is remarkably robust against noise, outliers, and undersampling.

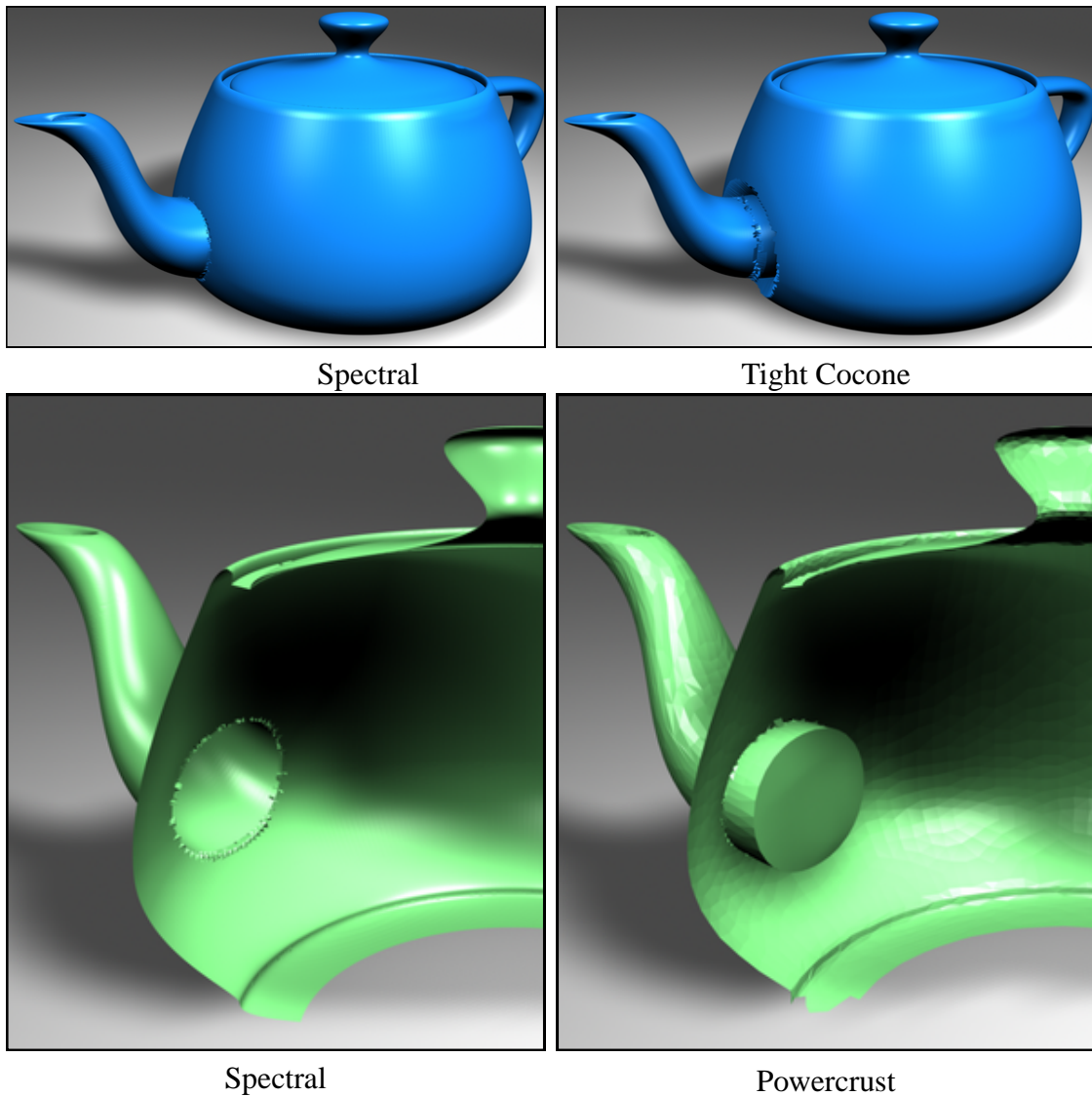


Figure 5.12: Reconstructions from 253,859 points sampled on the Utah Teapot. The spout's splines penetrate into the body of the teapot, causing difficulties for both the Powercrust and Tight Cocone algorithms. A cutaway view shows that Powercrust mislabels as outside a cluster of cells where the spout enters the body. The spectral algorithm correctly identifies the same poles as inside.



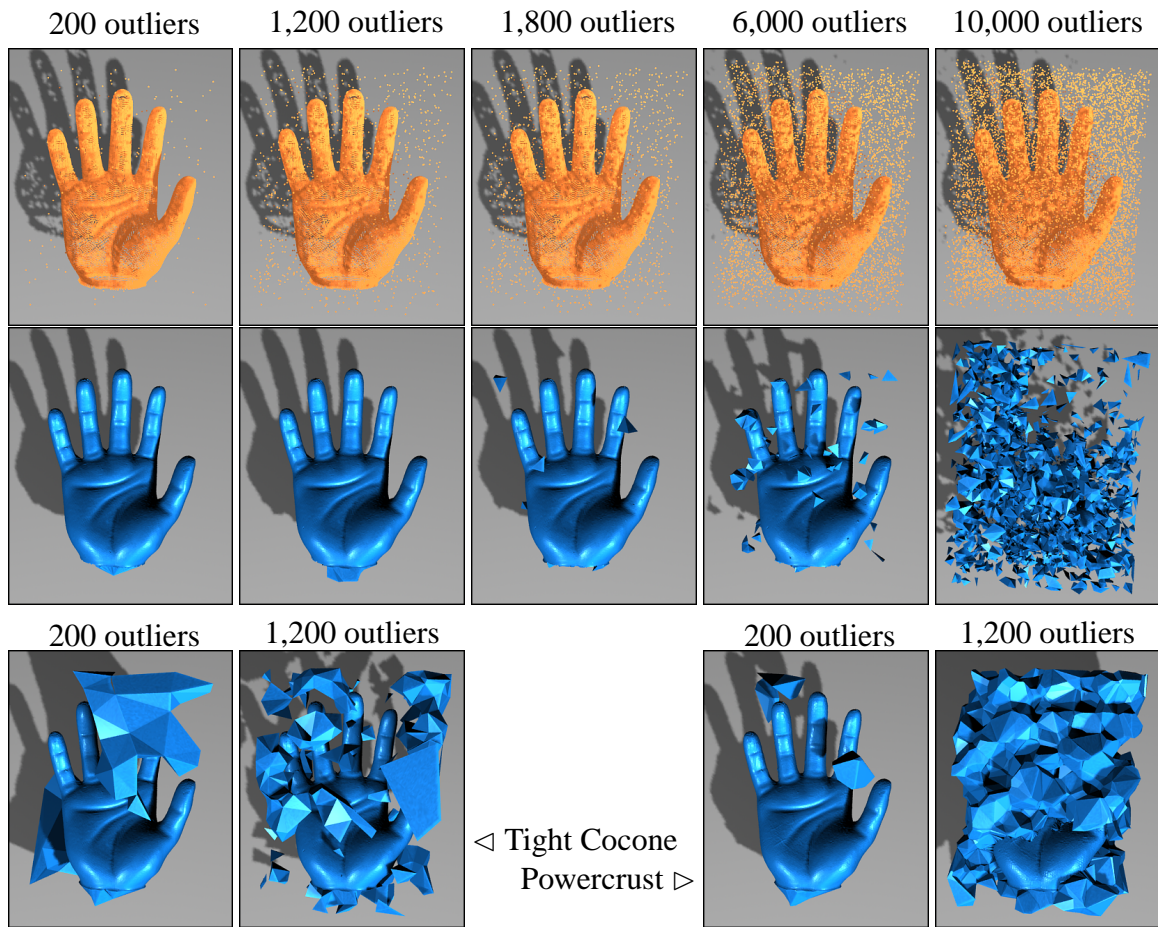


Figure 5.13: Top row: 25,626 noise-free points plus randomly generated outliers. Second row: spectral reconstructions of this model maintain their integrity to 1,200 outliers, then begin to degrade. Third row: other Delaunay-based algorithms degrade much earlier.

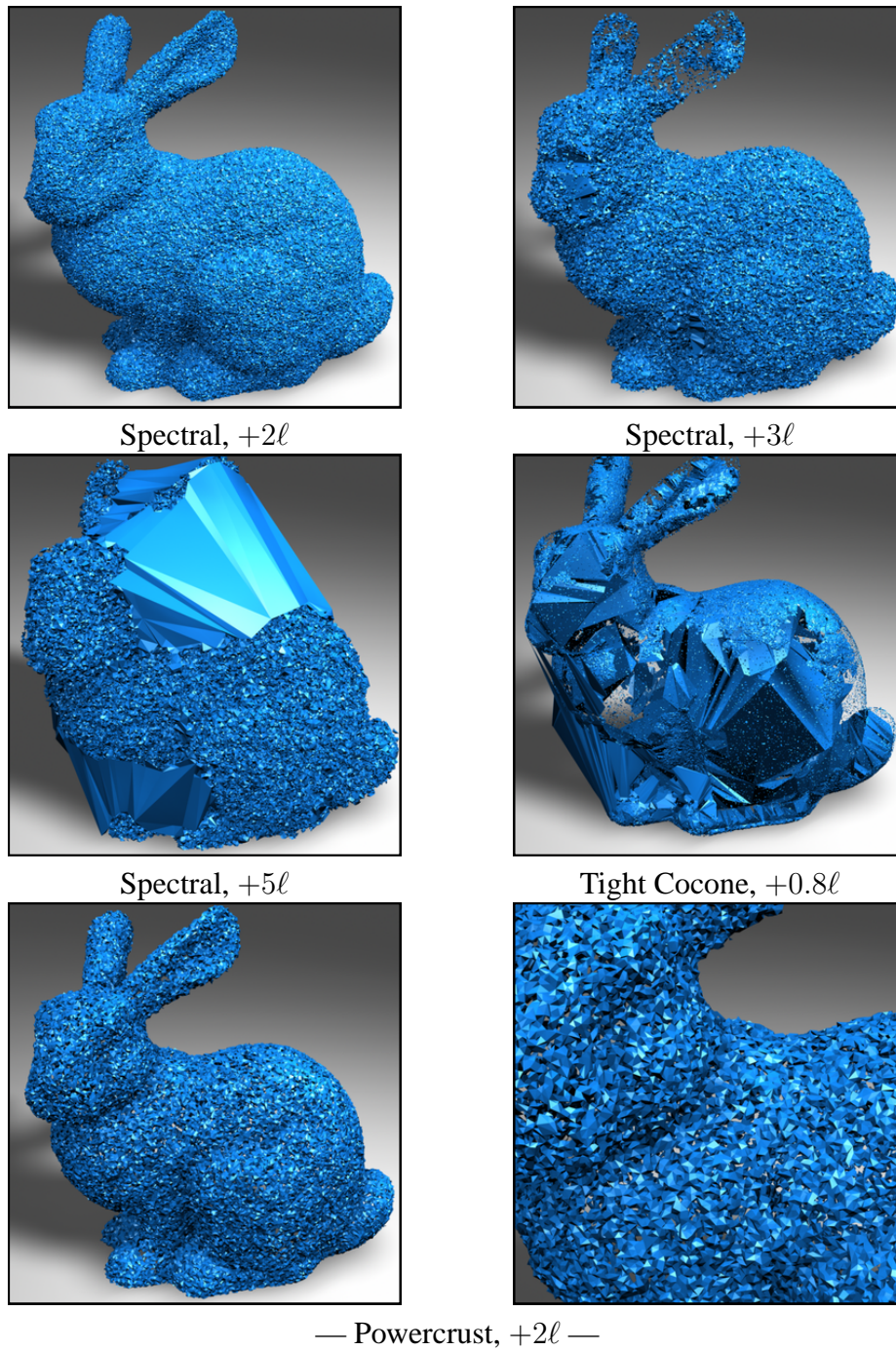


Figure 5.14: Stanford bunny reconstructions from raw data (with no outliers), some with added random Gaussian noise.

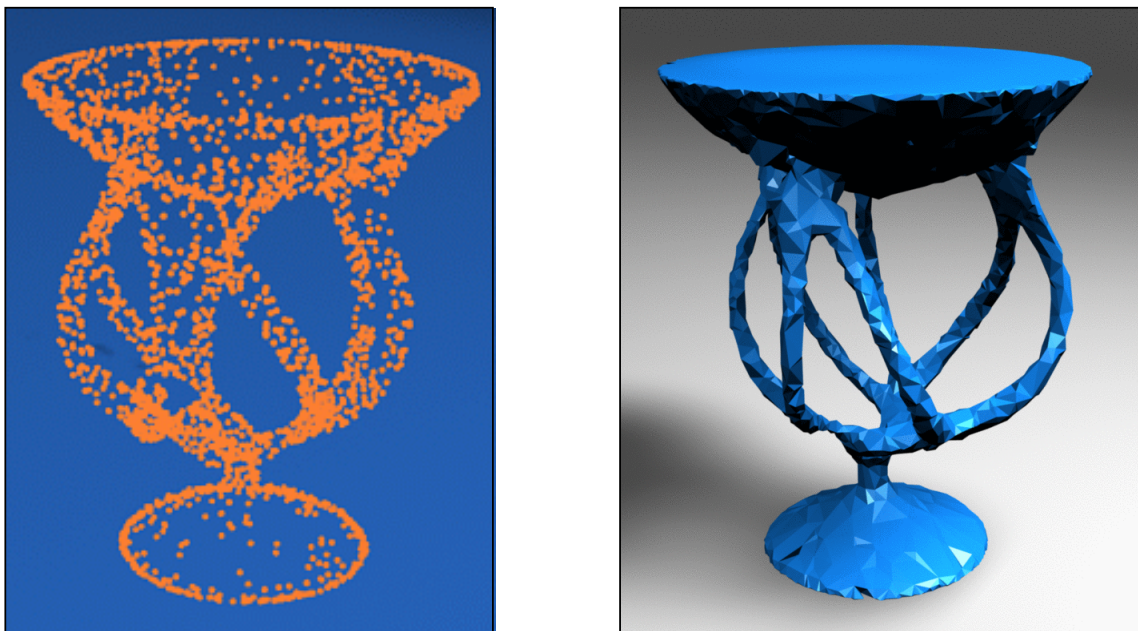


Figure 5.15: Eigencrust reconstruction of a sparse point cloud; 2714 points; 48727 tetrahedra; 5 seconds reconstruction time.



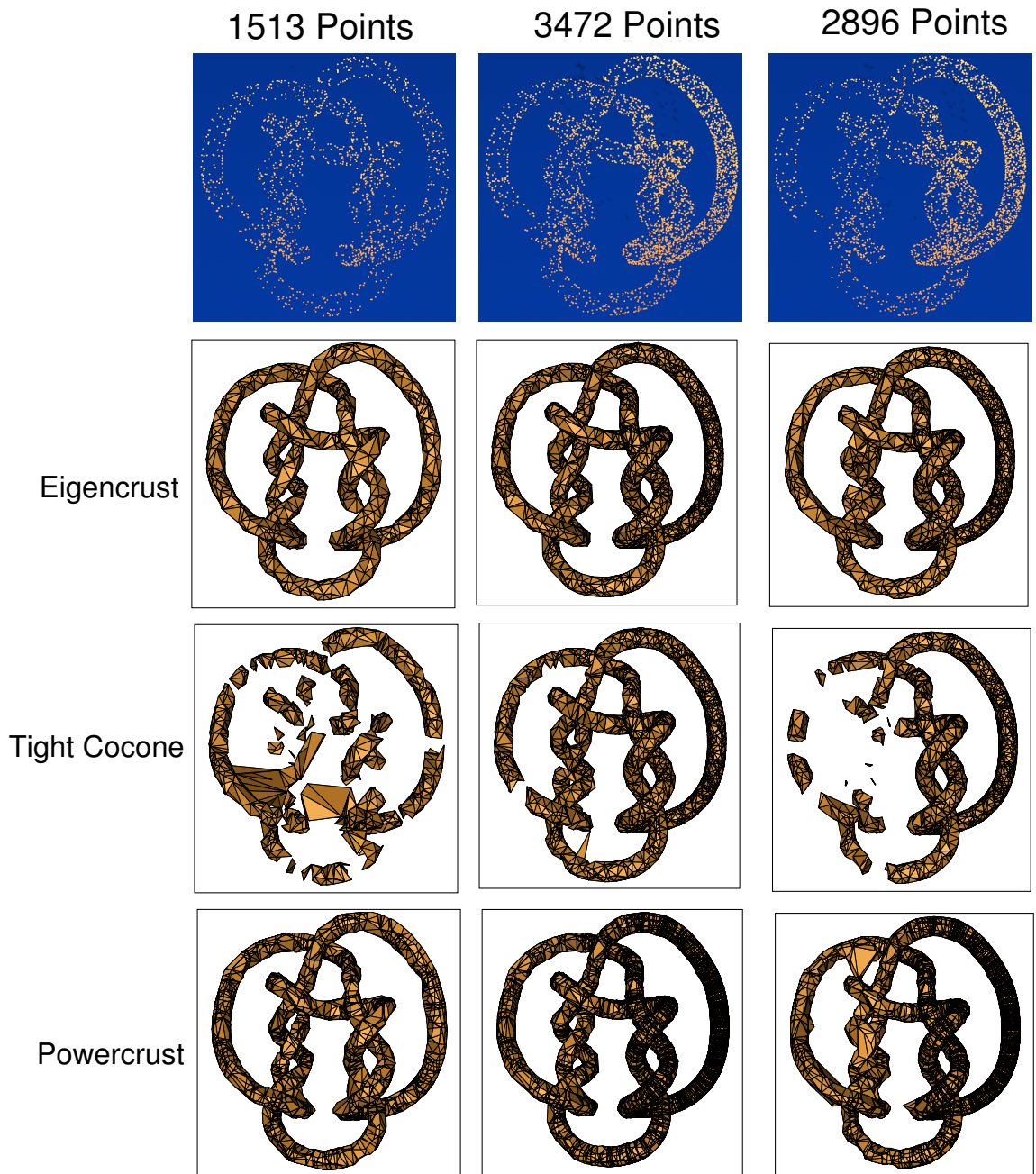


Figure 5.16: Reconstructions of a sparse point cloud with adaptive sample spacing.

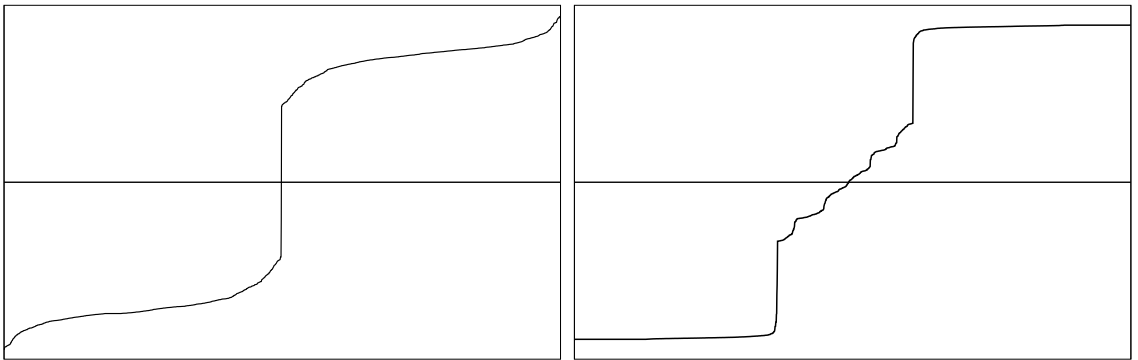
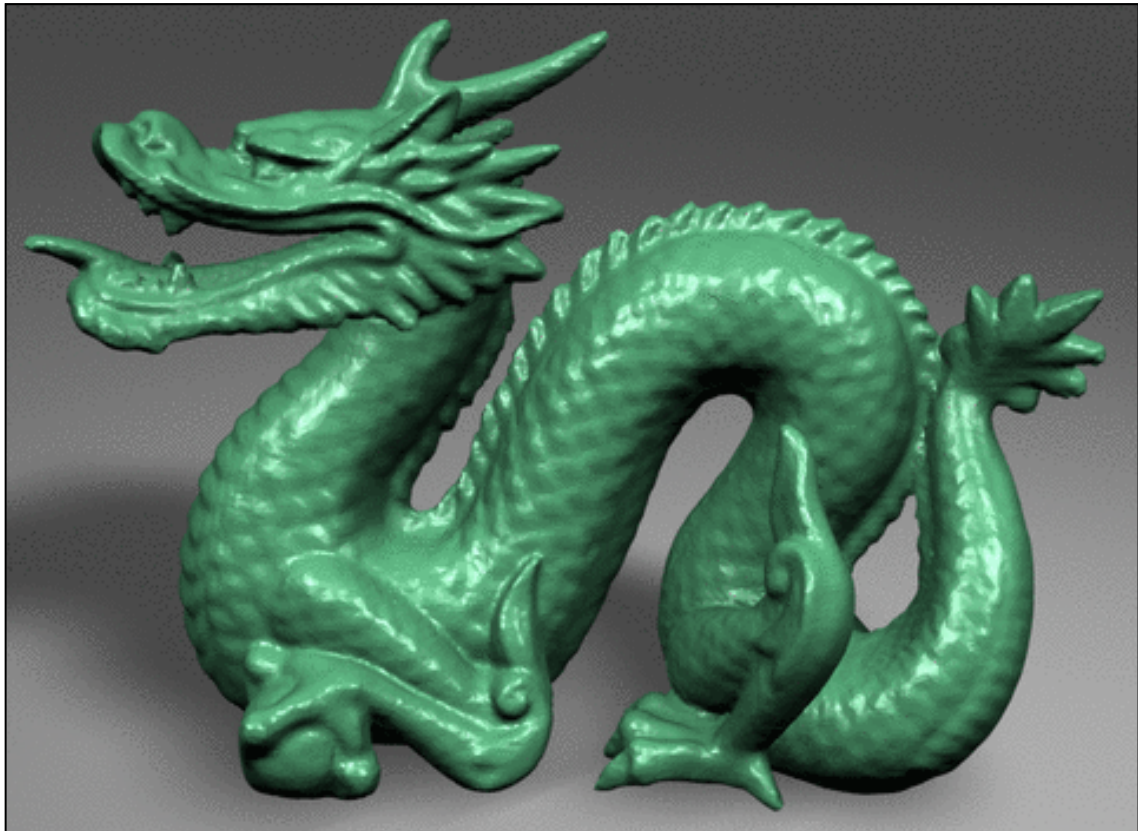


Figure 5.17: Eigencrust reconstruction of the Stanford dragon model after MLS smoothing.

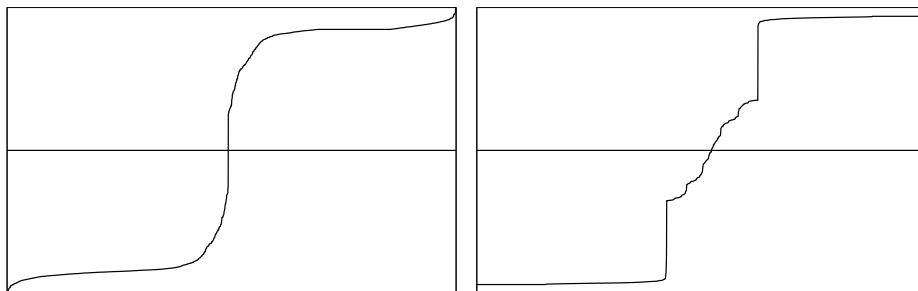
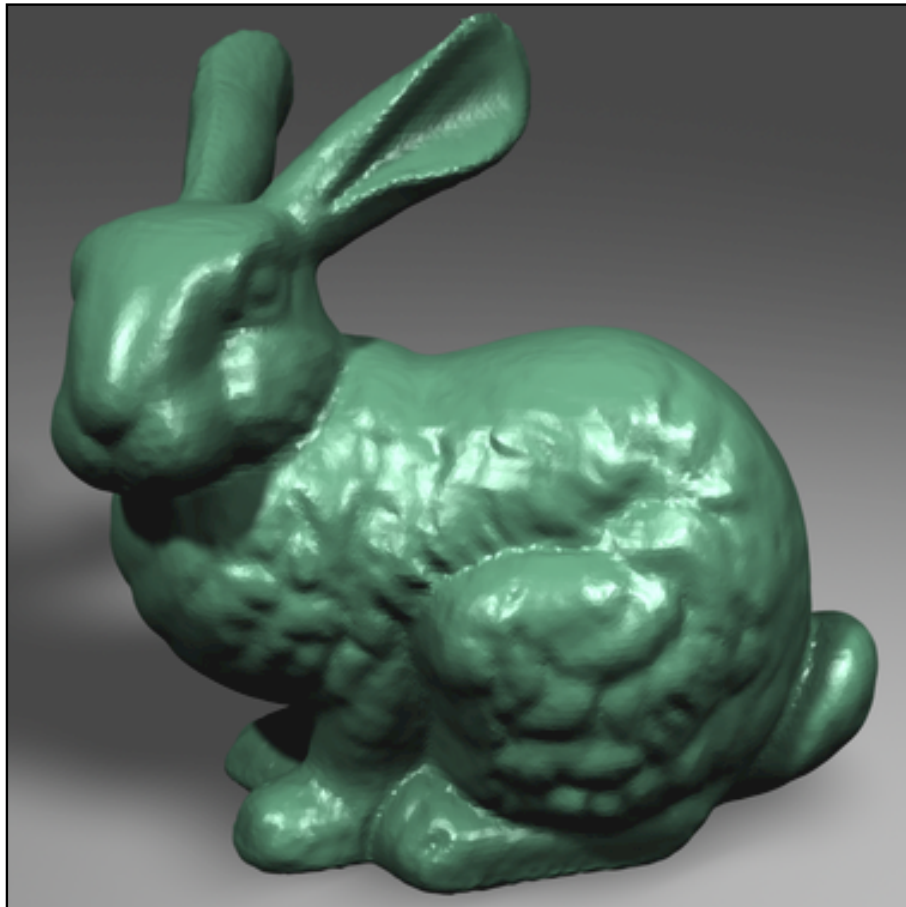


Figure 5.18: Eigencrust reconstructions of the Stanford bunny after MLS smoothing.

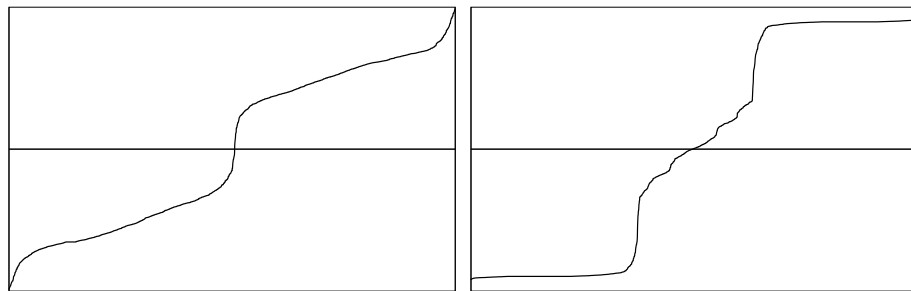
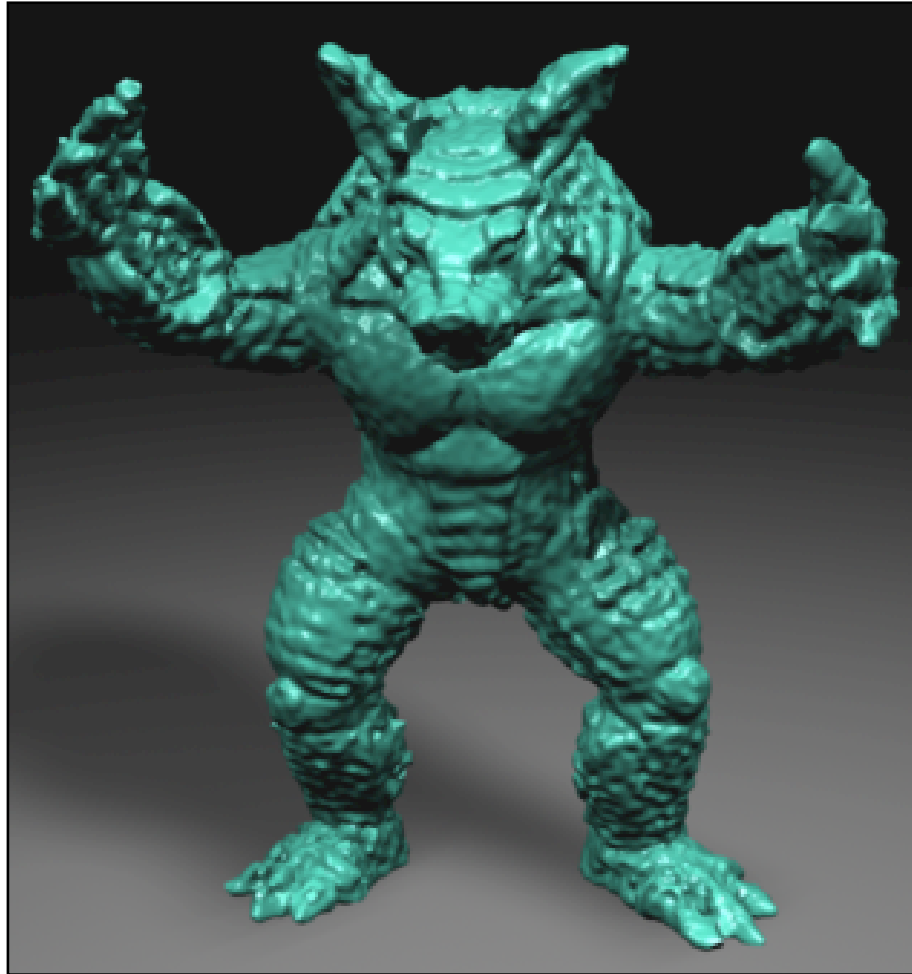


Figure 5.19: Eigencrust reconstructions of the armadillo model after MLS smoothing.

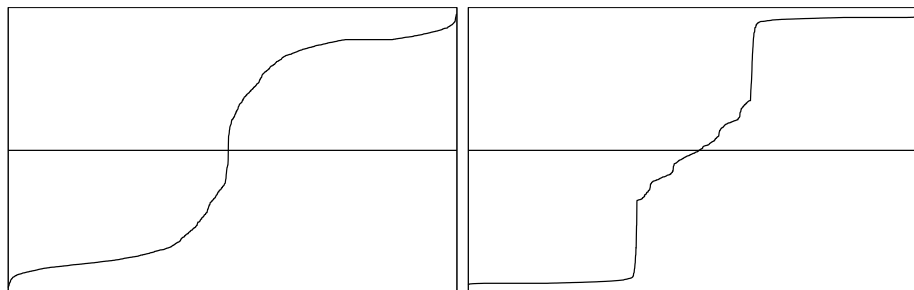
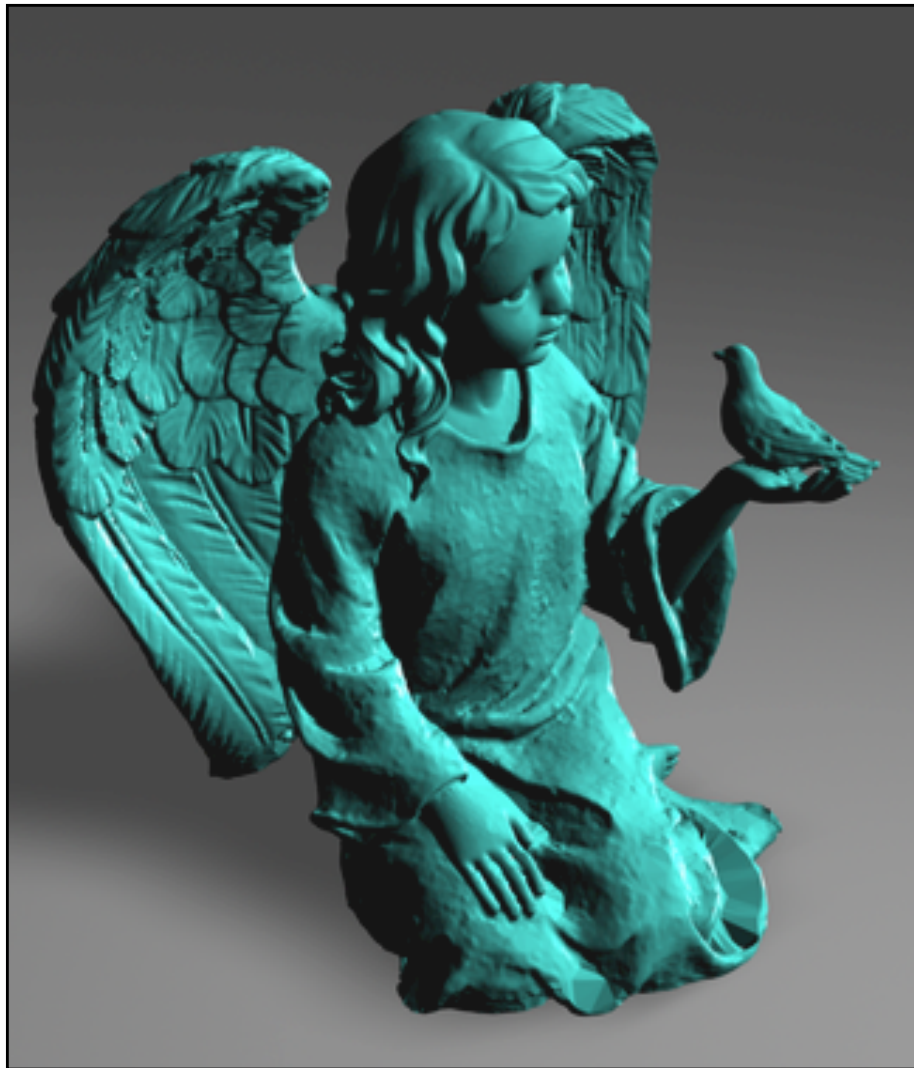


Figure 5.20: Eigencrust reconstructions of the armadillo model after MLS smoothing.



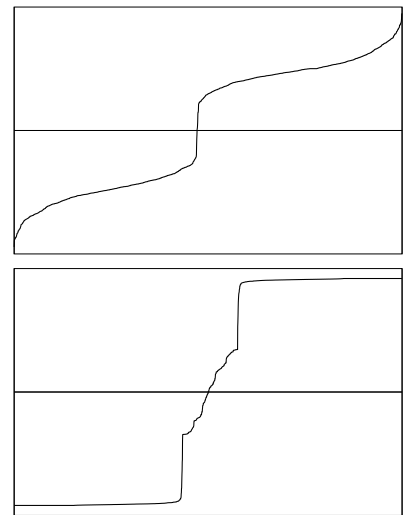


Figure 5.21: Eigencrust reconstructions of the happy buddha model after MLS smoothing.

# Chapter 6

## Registration

### 6.1 Introduction

Laser range scanners cannot scan an entire object at once because the whole surface is not visible from a single point of view. In this chapter we describe our implementation of a registration algorithm that aligns range images into a common coordinate system.

Automatic surface registration is typically done in two stages. *Coarse registration* provides a rough positioning of the surfaces, and *fine registration* brings the scans into tight alignment. The difference between the two is akin to the difference between global and local optimization: coarse registration is concerned primarily with determining which general region of one scan represents the same portion of an object as a selected region of another scan, while fine registration is concerned with minimizing the disparity between “overlapping” regions of different scans.

We present here a coarse registration algorithm that uses *signatures* that we call *harmonic shape contexts* to match similarly shaped regions of different surfaces. A signature encodes the surroundings of a particular point in a particular surface in a manner that makes it easy to find similar regions in different surfaces while discriminating between dissimilar

regions. Our signatures are computed by forming a *shape context*—a rough model of the shape of the surface that a scan represents—and computing the spherical harmonics of the shape context. Shape contexts began life as successful two-dimensional signatures in the computer vision literature; we generalize them to three dimensions. We choose spherical harmonics because they allow us to define signatures that are invariant to rotation as well as translation, yet retain most of the geometric information in the shape context.

Given two surfaces with similarly shaped regions, we register the surfaces relative to each other through *correspondences*—pairs of approximately matching signatures taken from the two surfaces. For the scan registration application, we expect the overlapping regions of the two scans to be similarly shaped throughout the region of overlap. The quality of a match might depend on the area of overlap. Our method is fully automatic and works consistently without manual assistance. Because translation- and rotation-invariance are built into our signatures, we achieve excellent accuracy and competitive speed.

## 6.2 Related Work

Computer vision researchers have extensively studied the problem of object recognition and matching, in two and three dimensions. Campbell and Flynn [25] offer a survey of three-dimensional shape representations used for object recognition.

Most shape signatures are one of two types. *Object signatures* attempt to summarize the shape of an entire object, and are a fast way to find similar shapes in a database. *Point signatures* [29] attempt to summarize the shape around a specific point, a necessary task for registering two scans. (The terms *signature* and *descriptor* appear to be used interchangeably.) Many object signatures have been proposed, including superquadrics, parabolic curves [82], medial axis representations, and spherical harmonic representations. Our point signatures are inspired by the spherical harmonic object signatures of Kazhdan, Funkhouser, and Rusinkiewicz [56]. Vranic [84] presents object signatures similar to har-

monic shape contexts for searching in shape databases. Object signatures provide a fast way to identify similar shapes, but are rarely suitable for finding small similarities (e.g. a shared part) between objects that are otherwise different.

Point signatures can be either “local” or “global” in nature, depending on how much of the object’s shape is encoded. The curvature of a surface is an example of a purely local point signature. Unfortunately, purely local signatures are not descriptive enough to disambiguate many shapes. Global point signatures can perform better, but they are sensitive to occlusion (i.e. the deletion of a large part of a surface) and positional noise.

We use point signatures that characterize an object’s shape in a region surrounding a point of interest. The most successful three-dimensional point signatures to date are the spin images of Johnson and Hebert [54]. At the end of Section 6.4, we describe spin images and compare them with harmonic shape contexts. See Frome, Huber, Kolluri, Bulow and Malik [44] for a comparison of the performance of several point signatures on the task of object recognition in range images.

Fine registration has been well studied. Most authors use some variant of the Iterated Closest Point (ICP) approach for aligning a pair of scans, which was introduced independently by Chen and Medioni [28] and Besl and Mckay [16]. In the ICP method, each scan is approximated by a continuous surface. Typically, a pair of scans partly “overlap” each other, and each scan also has sample points that are not near the surface covered by the other scan. The goal of ICP is to ensure that each sample point in one scan (excepting points not in the overlap region) is as close as possible to some surface that represents the other scan. The ICP method iteratively minimizes an energy function that estimates how well two scans are aligned in their region of overlap. Several variants of the ICP algorithm have been proposed, such as the fast algorithm that Rusinkiewicz and Levoy [75] developed for real-time scanning.

Of course, there are usually more than two scans to align. Algorithms are available that simultaneously align several scans. For coarse multiview registration, Huber and

Hebert [52; 53] present an aggressive alignment algorithm that uses a consistency criterion to combine pairs of scans that have been aligned using spin images. Pulli [73] offers a notable fine multiview registration algorithm, which is well-suited for large models that cannot fit in memory. We use Pulli’s algorithm in our registration software.

### 6.3 Overview of Our Algorithm

Our software for scan registration executes two stages, solving global and local optimization problems. First, the software aligns pairs of range scans using point signatures called harmonic shape contexts. The aligned pairs are combined into a single model by placing the scans in a shared coordinate system. Second, a multiview ICP algorithm refines the registration and improves its accuracy.

Our signatures are based on the *shape contexts* proposed by Belongie, Malik, and Puzicha [12] for problems arising in computer vision. Shape contexts appear to perform better than most of their competitors, as they are able to make finer discriminations of shape while omitting irrelevant detail. To make shape contexts invariant to rotation, we use spherical harmonics. The idea of using spherical harmonics to make object signatures rotation-invariant was introduced by Kazhdan, Funkhouser, and Rusinkiewicz [56] (in the context of querying databases of shapes). Spherical harmonics are a strong approach because they maintain nearly all rotation-invariant information. Our use of spherical harmonics differs from Kazhdan’s in several important ways, described in Section 6.5.

The coarse registration stage itself consists of four steps. First, our algorithm computes harmonic shape contexts at a random subset of sample points on each surface. Second, pairs of signatures from different surfaces are tested to find *correspondences*—pairs of points that are likely to come from the same spot on a physical object or part. Correspondences are indicated by highly correlated signature pairs. Candidate *poses*—transforms (translation and rotation) that align one scan with another—are computed from pairs of

correspondences. Poses are evaluated by determining how close each point in one scan is to its nearest neighbor in the other, transformed scan. We select the best candidate pose for each pair of surfaces.

Third, for each pair of surfaces, our algorithm improves the pose (which is computed from a single correspondence) by simultaneously considering all the “good” correspondences between the two surfaces, and optimizing the least-squares distance between correspondence pairs.

Fourth, we register all the scans in a single global coordinate system by choosing the most reliable pairwise poses.

### 6.4 Shape Contexts as Signatures

A signature is a local description of the shape of a scan around a chosen sample point in the scan. Signatures are used to find correspondences between different scans. Ideal signatures should be insensitive to any rotation or translation of a scan, because a single spot on an object may be scanned from many different directions.

Our signatures begin with the notion of *shape contexts*, suggested by Belongie, Malik, and Puzicha [12] for finding correspondences among two-dimensional images. Their idea is to divide space into a collection of bins centered at a chosen sample point. Space is subdivided by radial lines passing through the sample point, and also by concentric spheres centered at the sample point. The concentric spheres have exponentially increasing radii, making the signature more sensitive to nearby sample points than faraway ones. The *shape context* is a histogram listing how many sample points fall into each bin.

A three-dimensional shape context is based on a division of space into bins by choosing a sample point and independently partitioning the radial, azimuthal, and polar dimensions around the point.

There are several reasons why shape contexts in their original form are not suitable for

scan registration. First, they are not invariant to rotation. Second, the number of sample points in each bin is not a reliable measure for finding correspondences between scans. Different scans may be taken from different distances, and therefore each scan samples a different density of points on an object's surface. A surface is typically scanned from several different angles, so the spacing of sample points on the surface is often anisotropic, and the anisotropy differs from scan to scan, influencing the number of sample points that fall in each bucket. The remainder of this section discusses how we build signatures that are invariant to sampling density and sampling anisotropy. Rotation invariance is discussed in Section 6.5.

Suppose we have a scan and a sample point  $p$  in that scan. Discretize the space around  $p$  into  $u$  radial divisions between the radii  $r_0 = 0, r_1, \dots, r_u$ ;  $v$  polar divisions between the angles  $\theta_0 = 0, \theta_1, \dots, \theta_v = \pi$ , as measured from the normal vector  $n$ ; and  $w$  azimuthal divisions between the angles  $\phi_0 = 0, \phi_1, \dots, \phi_w = 2\pi$ . The largest radius  $r_u$  is chosen slightly larger than the diameter (i.e. the greatest inter-sample distance) of any scan, and the smallest radius  $r_1$  is chosen to be somewhat greater than the resolution of the scans, so that the innermost bins are not overly vulnerable to aliasing effects. The polar and azimuthal divisions are equiangular, but the radii are geometrically spaced, with

$$r_i = \exp \left\{ \ln r_1 + \frac{i-1}{u-1} \ln \frac{r_u}{r_1} \right\}.$$

Let  $b_{jkl}$  be the value assigned to the bin covering the points  $(r, \theta, \phi)$  satisfying  $r_j \leq r < r_{j+1}$ ,  $\theta_k \leq \theta < \theta_{k+1}$ , and  $\phi_l \leq \phi < \phi_{l+1}$ . As we have observed, it is a poor idea to set  $b_{jkl}$  to be the number of sample points that fall into the corresponding bin. However, range scanners sample points on a square grid (as viewed from the scanner), which forms a natural triangulation. We treat the triangulation as a piecewise linear surface embedded in three-dimensional space, and assign to each bin the amount of surface area lying in the bin. This insulates the bin values against the effects of sampling density or anisotropic

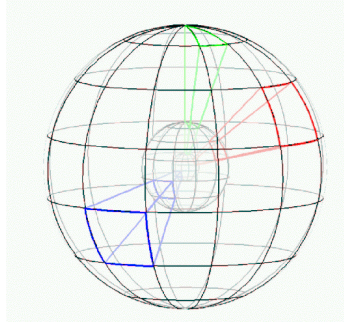


Figure 6.1: The bins of a three-dimensional shape context.

sampling.

In our implementation, we modify this idea in two ways. First, very long triangles appear where the scanner crosses a silhouette of the object being scanned, and these triangles are rarely representative of the object's shape. We compute a *grid spacing*  $\ell$  equal to the median length of the diagonal edges of the triangulated grid, and we discard any triangle whose greatest edge length exceeds  $4\ell$ .

Second, it is unnecessarily expensive to compute the area of intersection of a triangle with a bin. As a heuristic, we compute the area of each triangle (that is not too long), and assign its area once to each of the three bins containing the triangle's vertices. Frequently, all three vertices of a triangle are in one bin, so the bin is assigned thrice the triangle's area. In the end, the bin value  $b_{jkl}$  is a reasonable approximation to thrice the amount of triangulation surface area that falls in the bin.

Spin images [54] are similar to shape contexts, but they use cylindrical coordinates instead of spherical coordinates, their radial divisions are uniformly spaced, and most importantly, they obtain rotation-invariance by not using divisions of the azimuthal coordinate (i.e. each bin is ring-shaped). For this last reason, our harmonic shape contexts are more descriptive than spin images.



## 6.5 Coarse Registration, Step 1:

### Harmonic Shape Contexts

In two dimensions, computing discrete Fourier transforms over a two-dimensional shape context yields a rotation-invariant signature. For a ring of bins at a fixed radius interval, let  $b$  be a vector whose components are the values of the bins in the ring. Let  $f$  be a vector whose components are the complex coefficients of the discrete Fourier expansion of  $b$ . The magnitude of each component of  $f$  reflects the predominance of a particular frequency in the ring of bins, whereas the phase (angle in the complex plane) of a component reflects the phase of that frequency. If the scan is rotated, the phase of each component changes accordingly, but (if the number of bins is large enough) the magnitude of each component is virtually unaffected.

In three dimensions, we can build a rotation-invariant signature using normal estimation and either Fourier expansions or spherical harmonics. We call these signatures *Fourier shape contexts* and *harmonic shape contexts*, respectively, and evaluate both of them in Section 6.9. Both signatures use normal estimation to provide rotation invariance in the polar direction. Rotation invariance in the azimuthal direction is provided by the Fourier or harmonic basis.

To compute a signature at a point  $p$ , we begin by estimating a normal  $n$  to the surface at  $p$ . These normals are computed from the triangulation of each range image. The normal at each sample point  $s$  is given by an average of the normals of all faces incident on  $s$ . We then compute a shape context using  $n$  to specify the north pole of our coordinate system, so the polar angle  $\theta$  is measured from  $n$ , as illustrated in Figure 6.5. Thus our signatures are rotation invariant in the polar direction (except in the few parts of a scan where an accurate normal computation is difficult).

Three-dimensional Fourier shape contexts confer azimuthal rotation invariance in the same manner as their two-dimensional kin. The subset of bins covering a fixed radial

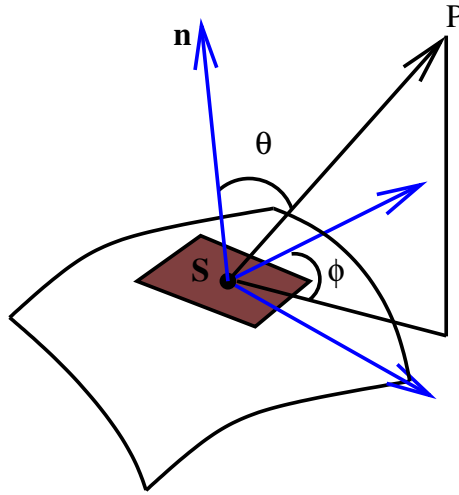


Figure 6.2: The polar angle  $\theta$  of a point  $p$  is determined by the approximate normal  $n$  at a sample point  $s$ . The coordinate axes along the azimuth angle  $\phi$  are ambiguous.

interval and polar interval forms an azimuthal ring around  $n$ . We compute a discrete Fourier transform of the azimuthal bin values, and store the absolute magnitudes of the Fourier coefficients.

Alternatively, spherical harmonics also confer rotational invariance. Spherical harmonics are analogous to the Fourier basis functions, defined on the sphere. Whereas the Fourier bases are univariate sine and cosine functions, spherical harmonics are inherently bivariate. Our signatures treat each of the concentric spheres independently—spherical harmonics are computed separately for each fixed radius. For a radius  $r$ , let  $f(\theta, \phi)$  be a function defined over the sphere. In particular, we choose  $f(\theta, \phi)$  to be the value assigned to the bin holding the point  $(r, \theta, \phi)$ ; but the following comments apply to any choice of  $f$ .

Any square-integrable complex function on the sphere can be expressed as a linear combination of spherical harmonic basis functions  $Y_m^n$ , with  $m \geq 0$  and  $-m \leq n \leq m$ .

The basis functions are orthonormal; that is,

$$\int_S Y_m^n \overline{Y_{m'}^{n'}} = \begin{cases} 1 & m = m' \text{ and } n = n', \\ 0 & \text{otherwise,} \end{cases} \quad (6.1)$$

where  $S$  is the unit sphere and  $\overline{Y_{m'}^{n'}}$  is the complex conjugate of  $Y_{m'}^{n'}$ . The basis functions are

$$Y_m^n = c_{mn} P_m^n(\theta) e^{in\phi},$$

where  $i = \sqrt{-1}$ ,  $e$  is the base of the natural logarithm, the  $P_m^n$  are the Legendre polynomials, and the  $c_{mn}$  are normalization constants chosen so that (6.1) holds. See Arfken [10] for the details of both Legendre polynomials and spherical harmonics.

We thus write  $f(\theta, \phi)$  in terms of spherical harmonics,

$$f(\theta, \phi) = \sum_{m=0}^{\infty} \sum_{n=-m}^{n=m} a_m^n Y_m^n(\theta, \phi),$$

where the  $a_m^n$  are complex coefficients. Intuitively, the subscript  $m$  is roughly a specifier of the “frequency” of a basis function  $Y_m^n$ , whereas the superscript  $n$  specifies a mode within a particular “frequency.”

An important property of the spherical harmonic basis is that if a function  $f$  is “rotated” in the azimuthal coordinate  $\phi$ , the magnitude of the coefficients  $a_m^n$  do not change, but their phase (angle in the complex plane) changes to reflect the rotation of  $f$ . Therefore, if we build a signature using the coefficients  $a_m^n$ , we can make the signature invariant to rotations in the azimuthal coordinate by storing only the absolute value  $|a_m^n|$  of each coefficient.

Practically, we need to transform not a continuous function, but a function  $b$  defined at a discrete number of points, and we wish to compute only a finite number of coefficients. The standard discretization is to assume that there is a continuous function  $f(\theta, \phi)$  such

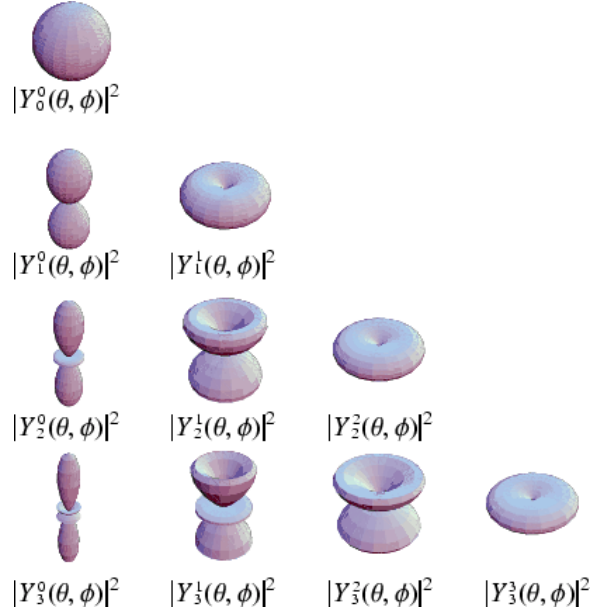


Figure 6.3: Plots of  $|Y_m^n|^2$  for small values of  $m$ . Observe that the magnitudes of the basis functions do not depend on the azimuth angle  $\phi$ .

that the bin values are the values of  $f$  at the center of each bin:

$$b_{jkl} = f\left(\frac{\theta_k + \theta_{k+1}}{2}, \frac{\phi_l + \phi_{l+1}}{2}\right)$$

(for a fixed  $j$ ), and there is a maximum bandwidth  $z$  so that  $f$  has no “high-frequency” components with  $m \geq z$ .

Given this assumption, the coefficients  $a_m^n$  (for  $0 \leq m < z$ ,  $-m \leq n \leq m$ ) can be computed from the bin values alone by a combination of discrete Fourier transforms and discrete Legendre transforms, for which we use the publicly available SpharmonicKit software [48].

We define a signature to be a vector  $\tilde{a}$ , where  $\tilde{a}_{jmn} = |a_m^n|$ , with  $a_m^n$  being the complex coefficients of the spherical harmonics for the shell  $j$ . A property of spherical harmonics is that if the function  $f$  is real-valued (as it is in our application), then  $a_m^n = a_m^{-n}$ , so there is

no need to separately store the coefficients for which  $n$  is negative.

An important difference between our signatures and those used by Kazhdan et al. [56] is that they obtain rotation invariance in the polar direction by summing the energies of the modes within each frequency. Thus, Kazhdan's signatures are the values  $\sum_{n=-m}^m \tilde{a}_{jmn}^2$  for every  $0 \leq j < u$  and  $0 \leq m < z$ . In our application, we are able to obtain rotation invariance by estimating surface normals, so our signatures maintain all the values  $\tilde{a}_{jmn}$  separately. Therefore, our signatures can make much finer discriminations between shapes.

The representation of a function in the spherical harmonic basis is more expensive to compute than the Fourier transform, but in Section 6.9 we see that signatures based on spherical harmonics are more descriptive. We believe the difference in descriptive power arises because the Fourier shape context does not change if one of its azimuthal rings is rotated relative to another, whereas a harmonic shape context does.

## 6.6 Coarse Registration, Step 2:

### Computing Correspondences and Poses

Here, we describe our method for determining a pose (translation and rotation) that attempts to bring one scan into coarse alignment with another. The method usually succeeds if the scans have a significant area of overlap (say, at least 20% of the surface area of each scan). Because scan pairs sometimes yield very few correspondences, the coarse-grained stage is designed to find a pose even if only two good correspondences exist. A pose improvement step, described in Section 6.7, uses multiple correspondences to improve a pose if more than two are available.

If the scans are large (collectively, hundreds of thousands of points or more), we subsample the scans with a coarser rectangular grid. For our results in Section 6.9, we use every second column and every second row of each scan, thereby reducing the data to a

quarter. The grid spacing  $\ell$  discussed in Section 6.4 is computed from this coarser grid.

To further reduce the running time, we compute signatures for only 20% of the points in each scan; these points are selected randomly.

Each point signature is a histogram, and the  $\chi^2$  histogram difference operator [74] defines a distance measure for the signature vectors. The  $\chi^2$  distance between each pair  $x, y$  of signature vectors is given by

$$\chi^2(x, y) = \sum_{x_i + y_i > 0} \frac{(x_i - y_i)^2}{2(x_i + y_i)},$$

where  $x_i$  and  $y_i$  are the components of the harmonic shape contexts  $x$  and  $y$ . (We tried several different scoring methods; this distance measure proved best.)

In one scan, we randomly choose 100 points (from the 20% whose signatures were computed). For each of these points, we find the best matching signature in the other scan. We treat each match as a correspondence. Typically, some of these correspondences are good, some are erroneous, and the quality of most of them depends on how much the two scans overlap. Of course, if the two scans do not overlap, all the correspondences are poor. Unfortunately, it is difficult to recognize this circumstance from the correlation scores alone. Occasionally, incorrect correspondences give the appearance of being excellent, and the mistake is not realized until the global scan registration stage.

We use these candidate correspondences to compute a set of candidate poses, and we test the quality of the candidate poses. Let  $S_1$  and  $S_2$  be two (subsamped) scans, and let  $T$  be a transform (rotation and translation) that defines a pose of  $S_2$ . To judge the quality of  $T$ , we assign it an overlap score of

$$O(T) = \sum_{p \in S_1} \exp - \frac{d(p, \text{nn}(p, T(S_2)))^2}{\ell^2}, \quad (6.2)$$

where  $\text{nn}(p, T(S_2))$  is the nearest neighbor to  $p$  in the transformed scan  $T(S_2)$ , and  $d(p, q)$

denotes the Euclidean distance between  $p$  and  $q$ . Note that the sum is over all points in  $S_1$ , and not just those for which signatures were computed. We use a  $k$ -d tree to efficiently find the nearest neighbor of a point in a scan.

Observe that the overlap score  $O(T)$  is credited mainly by points that have very close neighbors, and receives little credit from points that do not. If  $S_1$  has  $n$  points, the maximum possible value of  $O(T)$  is  $n$ , achieved if every point in  $S_1$  is matched by a point in  $S_2$  with the same coordinates.

How do we obtain candidate poses from the candidate correspondences? Pairs of scans occasionally yield few good correspondences, so we choose a method that will find one good pose even if there are only two good correspondences. Let  $(p_1, p_2)$  and  $(q_1, q_2)$  be two candidate correspondences, where  $p_i$  and  $q_i$  are points in scan  $S_i$ . As a sanity check, we do not compute a pose if the distances  $|p_1q_1|$  and  $|p_2q_2|$  differ by more than 10%. (We find that this check saves us 25–50% of the pose computations when the scans overlap well, and 80–90% when there is little overlap.) To compute a transform  $T$ , we first translate  $S_2$  so that  $p_2$  has the same coordinates as  $p_1$ . We then rotate  $S_2$  around  $p_2$  so that the normal vectors to  $p_1$  and  $p_2$  are parallel. Finally, we perform an azimuthal rotation of  $S_2$  (through the  $\phi$  coordinate, leaving the normal vector unchanged) so that the vector  $p_1q_1$  is aligned (azimuthally) with the vector  $p_2q_2$ .

Of course, a normal vector can point in either of two directions, so this procedure can determine two different transforms  $T, T'$ . We disambiguate by ensuring that every normal points toward the range scanner. If this information were not available, we would test both possibilities and calculate the overlap score of each.

For a pair of scans, we calculate 100 candidate correspondences, yielding 4,950 candidate transforms. We compute the score of all these transforms and choose the best—except that if we find a transform  $T$  for which  $O(T) \geq 0.75n$  (i.e. three-quarters the maximum possible score), we stop early and choose  $T$ . (This is usually the case, except when  $S_1$  and  $S_2$  barely overlap at all.)

There is a strong probabilistic justification for using a method that requires only two good correspondences to yield a good pose, even though we could obtain much more accurate poses using three. Assume a hypothetical model in which a pose is satisfactory if and only if the correspondences used to calculate it are all good. For example, suppose that for two scans with little overlap, the probability of any correspondence being good is 1%. Given 100 randomly generated correspondences, the probability of two being good is about 26%, whereas the probability of three being good is less than 8%. Because it is very difficult to determine whether a correspondence is good (correlation scores are an unreliable method of doing so), we might need to test 161,700 poses to find if each pose is computed from three correspondences, as opposed to 4,950 poses that are computed from two correspondences.

Thus, we follow the pose computations with a pose improvement step.

### **6.7 Coarse Registration, Step 3: Pose Improvement**

The pose produced by the coarse-grained stage is sensitive to small misalignments in the original correspondences, especially in the estimated normal vectors. To improve the pose, we test the 100 original correspondence pairs to see which pairs have been brought together by the coarse pose. If the distance between a pair of correspondence points  $p_1$  and  $p_2$  does not exceed  $8\ell$  (where  $\ell$  is the grid spacing discussed in Section 6.4), we call the correspondence “good”; otherwise we call it “bad.”

We compute an improved pose that minimizes the sum of the squares of the separations of the good correspondence pairs, using the standard method of Horn [51]. The pose improvement step takes little time compared to the other steps.



## 6.8 Coarse Registration, Step 4: Global Scan Registration

We use standard methods to register all the scans within a single coordinate system. Let  $G$  be a graph whose nodes represent the scans. For each pair of scans, we find a pairwise registration using the methods of Sections 6.6 and 6.7, and we connect the corresponding nodes of  $G$  with an edge holding two pieces of information: the relative pose produced by our pairwise registration algorithm, and the overlap score (6.2) for that pose.

Next, we compute a spanning tree  $T$  of  $G$ , and use the relative poses on the edges of  $T$  to register all the scans in a global coordinate system. To compute  $T$ , we recommend the sophisticated global registration algorithm of Huber and Hebert [53] (Section 6). For our implementation and results in Section 6.9, however, we simply used Prim’s maximum spanning tree algorithm to find the spanning tree  $T$  that maximizes the sum of overlap scores.

Figure 6.9 shows the global coarse registration of several scans in each of four different models. It also shows the same scans after fine multiview registration by Pulli’s algorithm [73].

## 6.9 Signature Comparison and Results

We use precision-recall curves—popular in computational learning theory—to compare signatures. The distance measure defined by a signature is a classifier that classifies each correspondence as correct or wrong, depending on whether the distance measure is under or over some threshold. To test this classifier, we give it two types of correspondences: those we know to be correct (nearest neighbor points from correctly registered pairs of scans), and pairs of points selected at random from two different scans. The histogram in Figure 6.4 shows distances between pairs of harmonic shape contexts obtained in these

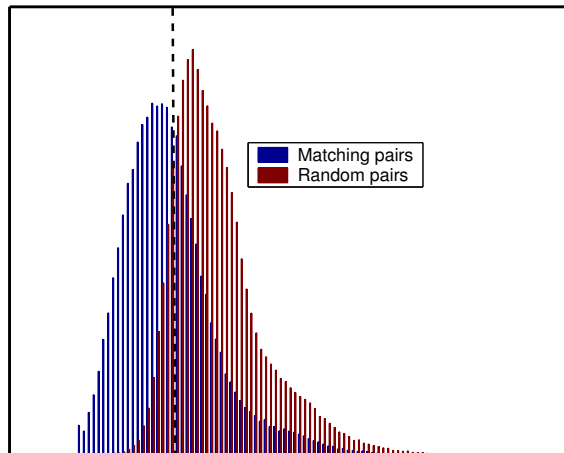


Figure 6.4: A histogram showing the distances between harmonic shape contexts for random pairs (right distribution) and correctly matched pairs (left distribution) in scans from the Happy Buddha model. The dashed line represents the decision boundary of a *likelihood ratio classifier*.

two ways from the Happy Buddha model. Observe that the distance measure is typically smaller for true correspondences than for randomly chosen pair of points, but it is not a perfect classifier.

The *precision* of a classifier is the fraction of signature pairs labeled as matches (below threshold) by the classifier that are true correspondences (i.e. 1 minus the number of false positives). The *recall* value is the fraction of true correspondences that were labeled as matches by the classifier. Clearly, the precision and recall of a classifier depend on what threshold the classifier chooses. We plot the precision and recall of signatures for different values of the threshold. A good signature yields a classifier that has high precision for increasing values of recall. For each signature, correct and random correspondences were obtained from 1000 pairs of scans in the Bunny, Dragon, Happy Buddha and Armadillo models. We chose pairs of scans that overlap by at least 50%.

In Figure 6.5(a) we plot precision-recall curves for spin images of different sizes. Notice that spin images with 16 planar divisions and 16 cylindrical divisions have the best

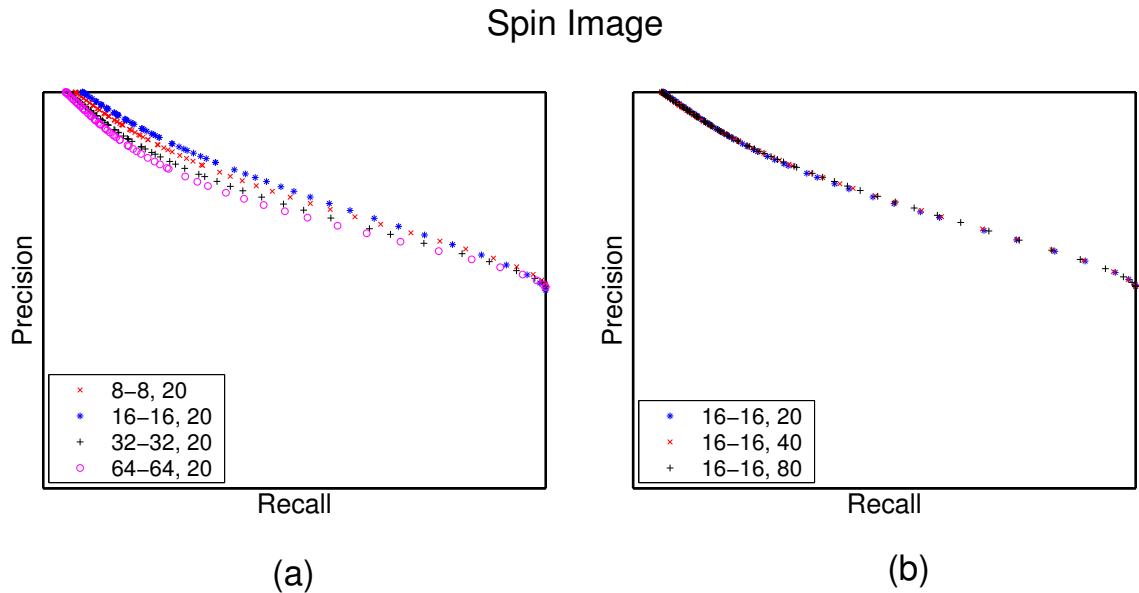


Figure 6.5: (a) Precision-recall curves for spin images of different sizes. In the legend we show the number of planar divisions, the number of cylindrical divisions, and the percentage of sample points with signatures. (b) Precision-recall curves for spin images of same size but with increasing percentage of sample points with signatures.

performance. Increasing the number of divisions makes the signature less effective as small bins introduce aliasing effects. In Figure 6.5(b) we plot precision-recall curves for spin images of fixed size and increase the number of sample points with signatures. Increasing the number of sample points with signatures has little effect on the precision of the classifier.

In Figure 6.6(a) and Figure 6.6(b) we plot precision-recall curves for fourier shape contexts of different sizes. Fourier shape contexts with 8 radial divisions, 8 polar divisions and 8 azimuthal divisions have the best precision-recall curves.

In Figure 6.7(a) and Figure 6.7(b) we plot precision-recall curves of harmonic shape contexts of different sizes. Harmonic shape contexts with 8 radial divisions, 16 polar divisions and 16 azimuthal divisions have the best precision-recall curves.

The running time to register many scans is dominated by the cost of comparing signatures, which is linear in the size of the signature. Therefore, we compare signatures of

### Fourier Shape Context

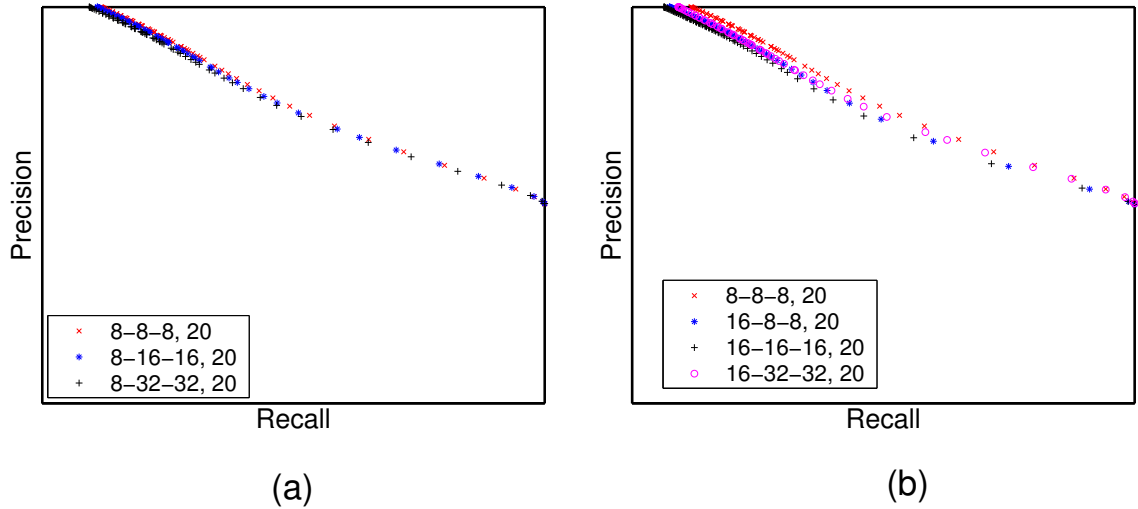


Figure 6.6: Precision-recall curves for fourier shape contexts of different sizes. In the legend of each plot we show the number of radial divisions, the number of polar divisions, the number of azimuthal divisions, and the percentage of sample points with signatures.

### Harmonic Shape Context

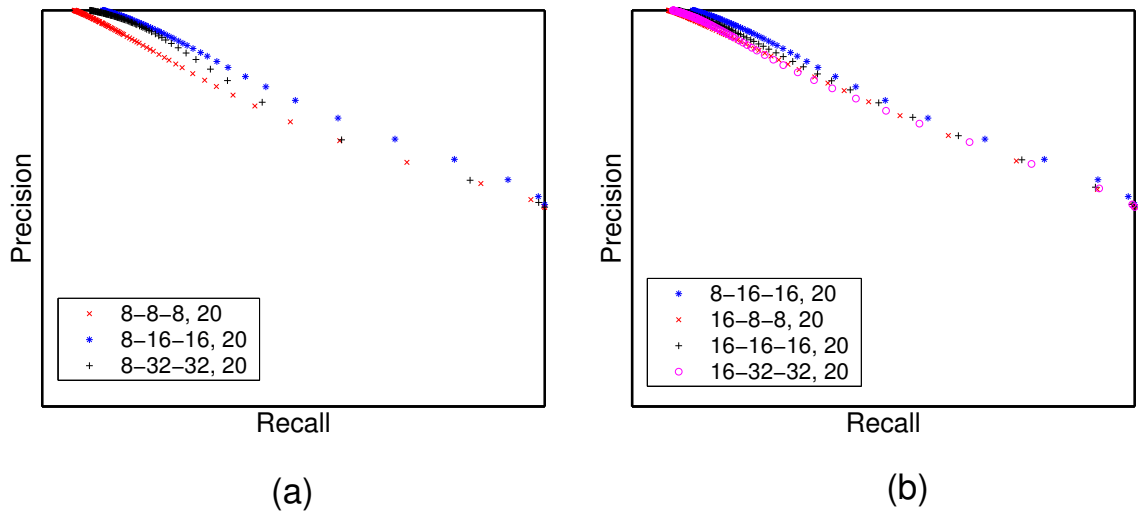


Figure 6.7: Precision-recall curves for harmonic shape contexts of different sizes. In the legend of each plot we show the number of radial divisions, the number of polar divisions, the number of azimuthal divisions, and the percentage of sample points with signatures.

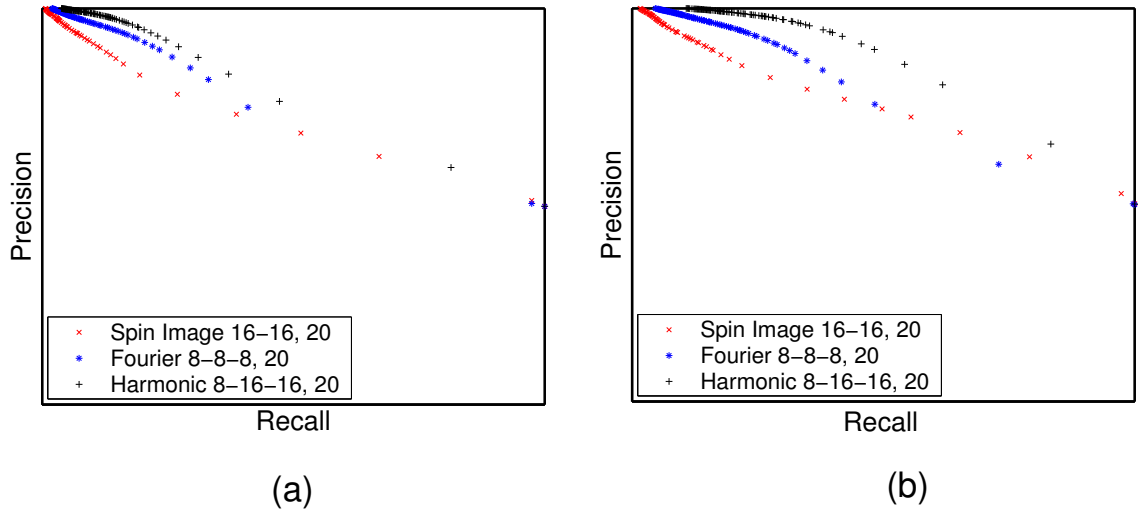


Figure 6.8: (a) Precision-recall curves of the three signatures computed for scans with 50% overlap. (b) Precision-recall curves of the same three signatures computed for scans with 75% overlap.

roughly equal sizes. Our spin images have 16 planar divisions and 16 cylindrical divisions, yielding a point signature of size 256. Our Fourier shape contexts have 8 radial divisions, 8 polar divisions, and 8 azimuthal divisions. Because the bin values are real, half the coefficients of the Fourier shape contexts are complex conjugates of the other half, so we can discard half of them, and the Fourier shape context is a vector of 256 coefficients. The harmonic shape contexts used in our experiments divide the radial, polar, and azimuthal coordinates into  $u = 8$ ,  $v = 16$ , and  $w = 16$  bins respectively. The maximum bandwidth of the spherical harmonics is  $z = 8$ , which means that each harmonic shape context signature has 288 coefficients.

In Figure 6.8(a) we plot the precision-recall curves of the three signatures. Harmonic shape contexts have the best precision-recall curves. In Figure 6.8(b) we show precision-recall curves for the same signatures computed from pairs of scans with at least 75% overlap. For pairs of scans with large overlap, harmonic shape contexts show a large increase in precision compared to the other signatures. In our coarse registration implementation,

Model	scans	points	Registration (Seconds)
Bunny	10	362272	632
Dragon	61	1765913	5986
Buddha	48	2643108	9201
Armadillo	106	1706919	12089

Table 6.1: Running times of the coarse registration algorithm

we chose harmonic shape contexts with  $u = 8$ ,  $v = 16$ , and  $w = 16$  bins. Table 6.1 we shows the running times of the coarse registration algorithm. In Figure 6.9 we show pairs of registered scans from the four data sets.

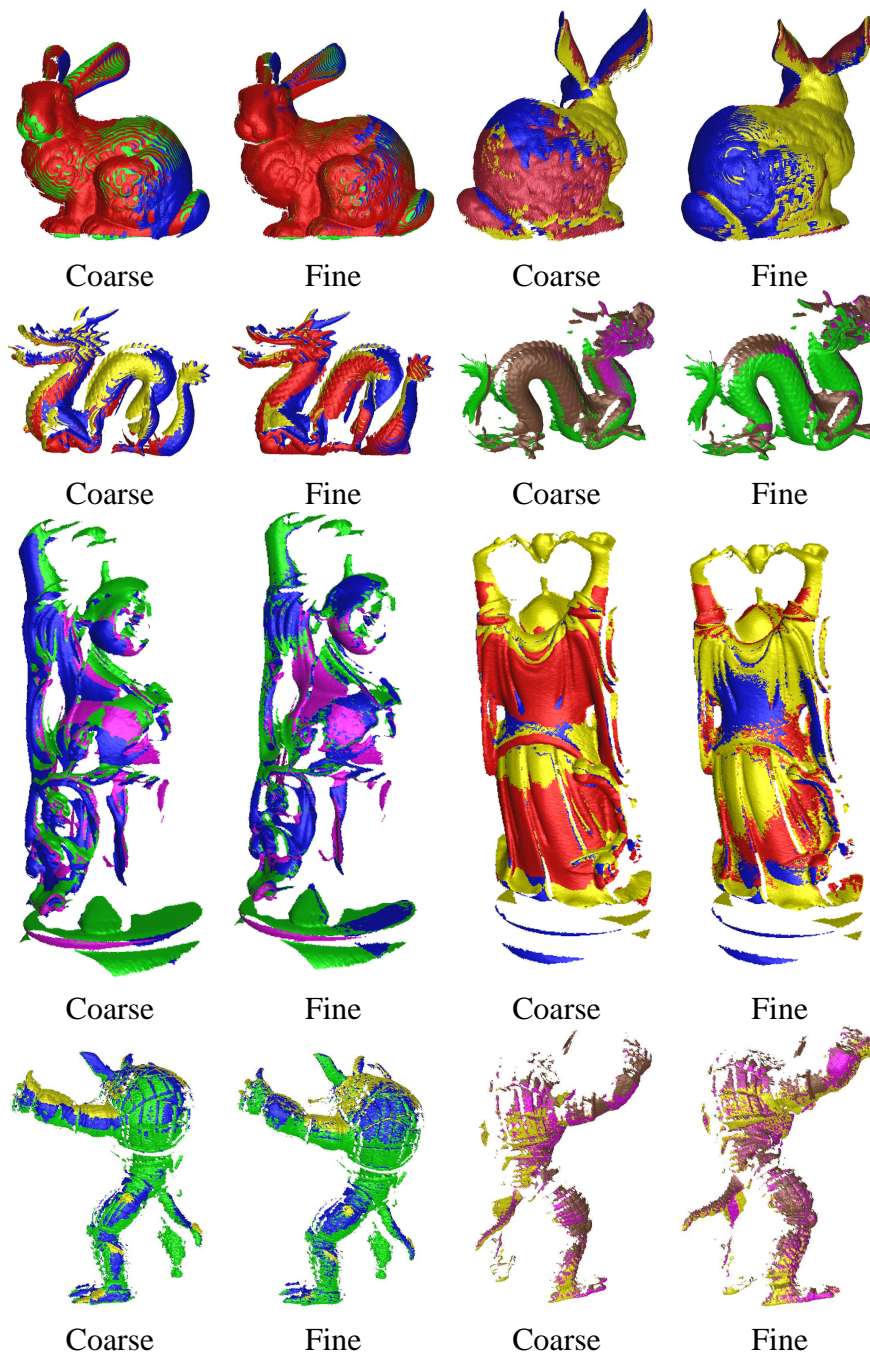


Figure 6.9: Scans in the four test models after coarse and fine registration. The rippling effect in the coarsely aligned scans suggests that the scans are already quite closely aligned.

# Bibliography

- [1] ADAMSON, A., AND ALEXA, M. Approximating and Intersecting Surfaces from Points. In *Proceedings of the Eurographics Symposium on Geometry Processing* (2003), Eurographics Association, pp. 230–239.
- [2] ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND T. SILVA, C. Computing and Rendering Point Set Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 9, 1 (2003), 3–15.
- [3] AMENTA, N., AND BERN, M. Surface Reconstruction by Voronoi Filtering. *Discrete and Computational Geometry* 22 (1999), 481–504.
- [4] AMENTA, N., BERN, M., AND KAMVYSSELIS, M. A New Voronoi-Based Surface Reconstruction Algorithm. In *Computer Graphics (SIGGRAPH '98 Proceedings)* (July 1998), pp. 415–421.
- [5] AMENTA, N., CHOI, S., DEY, T. K., AND LEEKHA, N. A Simple Algorithm for Homeomorphic Surface Reconstruction. *International Journal of Computational Geometry and Applications* 12, 1–2 (2002), 125–141.
- [6] AMENTA, N., CHOI, S., AND KOLLURI, R. The Power Crust. In *Proceedings of the Sixth Symposium on Solid Modeling* (2001), Association for Computing Machinery, pp. 249–260.
- [7] AMENTA, N., AND KIL, Y. Defining Point-Set Surfaces. In *Proceedings of ACM SIGGRAPH 2004* (Aug. 2004), ACM.
- [8] AMENTA, N., AND KOLLURI, R. Accurate and Efficient Unions of Balls. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry* (June 2000), Association for Computing Machinery, pp. 119–128.
- [9] AMENTA, N., PETERS, T. J., AND RUSSELL, A. C. Computational Topology: Ambient Isotopic Approximation of 2-manifolds. *Theoretical Computer Science* 305, 1 (Aug. 2003), 3–15.



## BIBLIOGRAPHY

---

- [10] ARFKEN, G. B. Legendre Functions. In *Mathematical Methods for Physicists*, third ed. Academic Press, Orlando, Florida, 1985, ch. 12, pp. 637–711.
- [11] BAJAJ, C. L., BERNARDINI, F., AND XU, G. Automatic Reconstruction of Surfaces and Scalar fields from 3D scans. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 109–118.
- [12] BELONGIE, S., MALIK, J., AND PUZICHA, J. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 4 (Apr. 2002), 509–522.
- [13] BELYTSCHKO, T., KRONGAUZ, Y., FLEMING, M., ORGAN, D., AND LIU, W. K. Meshless Methods: An Overview and Recent Developments. *Journal of Computational and Applied Mathematics* 74 (1996), 111–126.
- [14] BERNARDINI, F., MITTLEMAN, J., RUSHMEIER, H., SILVA, C., AND TAUBIN, G. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 5, 4 (Oct. 1999), 349–359.
- [15] BERNARDINI, F., RUSHMEIER, H., MARTIN, I. M., MITTLEMAN, J., AND TAUBIN, G. Building a Digital Model of Michelangelo’s Florentine Pietà. *IEEE Computer Graphics and Applications* 22, 1 (2002), 59–67.
- [16] BESL, P., AND MCKAY, N. A Method of Registration of 3D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 4 (Apr. 1992), 509–522.
- [17] BLOOMENTHAL, J., Ed. *Introduction to Implicit Surfaces*. Morgan Kaufman, 1997.
- [18] BOISSONAT, J.-D. Geometric Structures for Three-Dimensional Shape Reconstruction. *ACM Transactions on Graphics* 3 (Oct. 1984), 266–286.
- [19] BOISSONAT, J.-D. Geometric Structures for Three-Dimensional Shape Reconstruction. *ACM Transactions on Graphics* 3 (Oct. 1984), 266–286.
- [20] BOISSONAT, J.-D., AND CAZALS, F. Smooth Surface Reconstruction via Natural Neighbour Interpolation of Distance Functions. In *Proceedings of the Sixteenth Annual symposium on Computational geometry* (2000), ACM, pp. 223–232.
- [21] BOISSONAT, J.-D., AND CAZALS, F. Natural Neighbor Coordinates of Points on a Surface. *Computational Geometry Theory and Applications* 19 (July 2001), 155–173.
- [22] BOISSONAT, J.-D., COHEN-STEINER, D., AND VEGTER, G. Isotopic Implicit Surface Meshing. In *Proceedings of the thirty-sixth Annual ACM symposium on Theory of computing* (2004), pp. 301–309.

## BIBLIOGRAPHY

---

- [23] BOISSONNAT, J. D., AND OUDOT, S. Provably Good Surface Sampling and Approximation. In *Proceedings of the Eurographics Symposium on Geometry Processing* (2003), Eurographics Association, pp. 9–18.
- [24] BREMER, P.-T., AND HART, J. C. A Sampling Theorem for MLS Surfaces. In *Eurographics Symposium on Point-Based Graphics* (2005), pp. 47–54.
- [25] CAMPBELL, R. J., AND FLYNN, P. J. A Survey of Free-form Object Representation and Recognition Techniques. *Comput. Vis. Image Underst.* 81, 2 (2001), 166–210.
- [26] CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. Reconstruction and Representation of 3D Objects with Radial Basis Functions. In *Computer Graphics (SIGGRAPH 2001 Proceedings)* (Aug. 2001), pp. 67–76.
- [27] CHAZAL, F., AND COHEN-STEINER, D. A Condition for Isotopic Approximation. In *Proceedings of the Tenth Symposium on Solid Modeling* (2004), Association for Computing Machinery.
- [28] CHEN, Y., AND MEDIONI, G. Object Modeling by Registration of Multiple Range Images. In *Proceedings of the IEEE Conference on Robotics and Automation* (1991), pp. 2724–2729.
- [29] CHUA, C. S., AND JARVIS, R. Point Signatures: A New Representation for 3D Object Recognition. *Int. J. Comput. Vision* 25, 1 (1997), 63–85.
- [30] CURLESS, B., AND LEVOY, M. Better Optical Triangulation through Spacetime Analysis. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision* (Washington, DC, USA, 1995), IEEE Computer Society, p. 987.
- [31] CURLESS, B., AND LEVOY, M. A Volumetric Method for Building Complex Models from Range Images. In *Computer Graphics (SIGGRAPH '96 Proceedings)* (1996), pp. 303–312.
- [32] DEY, T., AND SUN, J. An Adaptive MLS Surface for Reconstruction with Guarantees. In *Symposium on Geometry Processing* (2005). to appear.
- [33] DEY, T. K., AND GOSWAMI, S. Tight Cocone: A Water-tight Surface Reconstructor. *Journal of Computing and Information Science in Engineering* 3, 4 (Dec. 2003), 302–307.
- [34] DEY, T. K., AND GOSWAMI, S. Provable Surface Reconstruction from Noisy Samples. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry* (Brooklyn, New York, June 2004), Association for Computing Machinery.

## BIBLIOGRAPHY

---

- [35] DEY, T. K., LI, G., AND SUN, J. Normal Estimation for Point Clouds: A Comparison Study for a Voronoi Based Method. In *Eurographics Symposium on Point-Based Graphics* (2005), pp. 39–46.
- [36] EDELSBRUNNER, H. Surface Reconstruction by Wrapping Finite Sets in Space. In *Ricky Pollack and Eli Goodman Festschrift*. Springer-Verlag, 2002.
- [37] EDELSBRUNNER, H., KIRKPATRICK, D., AND SEIDEL, R. On the Shape of a Set of Points in the Plane. *IEEE Transactions on Information Theory* 29 (1983), 551–559.
- [38] EDELSBRUNNER, H., AND MÜCKE, E. P. Three-dimensional Alpha Shapes. *ACM Transactions on Graphics* 13, 1 (1994), 43–72.
- [39] FIEDLER, M. Algebraic Connectivity of Graphs. *Czechoslovak Mathematical Journal* 23, 98 (1973), 298–305.
- [40] FLEISHMAN, S., ALEXA, M., COHEN-OR, D., AND SILVA, C. T. Progressive Point Set Surfaces. *ACM Transactions on Computer Graphics* 22, 4 (2003).
- [41] FLEISHMAN, S., COHEN-OR, D., AND SILVA, C. T. Robust Moving Least-squares Fitting with Sharp Features. *ACM Trans. Graph.* 24, 3 (2005), 544–552.
- [42] FORTUNE, S. Voronoi Diagrams and Delaunay Triangulations. In *Computing in Euclidean Geometry*, D.-Z. Du and F. Hwang, Eds., vol. 1 of *Lecture Notes Series on Computing*. World Scientific, Singapore, 1992, pp. 193–233.
- [43] FREIDMAN, J. H., BENTLEY, J. L., AND FINKEL, R. A. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software* 3, 3 (1977), 209–226.
- [44] FROME, A., HUBER, D., KOLLURI, R., BULOW, T., AND MALIK, J. Recognizing Objects in Range Data Using Regional Point Descriptors. In *European Conference on Computer Vision (to appear)*. (2004).
- [45] FRÜH, C., AND ZAKHOR, A. Data Processing Algorithms for Generating Textured 3D Building Façade Meshes from Laser Scans and Camera Images. In *Proceedings of 3D Data Processing, Visualization and Transmission* (2002).
- [46] GOPI, M., AND KRISHNAN, S. A Fast and Efficient Projection-Based Approach for Surface Reconstruction. In *SIBGRAPI '02: Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 179–186.

## BIBLIOGRAPHY

---

- [47] HALL, K. M. An  $r$ -Dimensional Quadratic Placement Algorithm. *Management Science* 11, 3 (1970), 219–229.
- [48] HEALY, JR., D., ROCKMORE, D., KOSTELEK, P., AND MOORE, S. S. B. FFTs for the 2-Sphere—Improvements and Variations. *Journal of Fourier Analysis and Applications* 9, 4 (2003), 341–385.
- [49] HERMANN, L. R. Laplacian-Isoparametric Grid Generation Scheme. *Journal of the Engineering Mechanics Division of the American Society of Civil Engineers* 102 (Oct. 1976), 749–756.
- [50] HOPPE, H., DEROSE, T., DUCHAMP, T., McDONALD, J., AND STUETZLE, W. Surface Reconstruction from Unorganized Points. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (1992), pp. 71–78.
- [51] HORN, B. K. P. Closed-Form Solution of Absolute Orientation Using Unit Quaternions. *Journal of the Optical Society of America A* 4, 4 (Apr. 1987), 629–642.
- [52] HUBER, D., AND HEBERT, M. 3D Modeling Using a Statistical Sensor Model and Stochastic Search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (June 2003), pp. 858–865.
- [53] HUBER, D., AND HEBERT, M. Fully Automatic Registration of Multiple 3D Data Sets. *Image and Vision Computing* 21, 7 (July 2003), 637–650.
- [54] JOHNSON, A. E., AND HEBERT, M. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 5 (May 1999).
- [55] JU, T., LOSASSO, F., SCHAEFER, S., AND WARREN, J. Dual Contouring of Hermite Data. *ACM Transactions on Graphics* 21, 3 (2002), 339–346.
- [56] KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In *Proceedings of the EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing* (2003), pp. 156–164.
- [57] KOBBELT, L. P., BOTSCH, M., SCHWANECKE, U., AND SEIDEL, H.-P. Feature-Sensitive Surface Extraction from Volume Data. In *Computer Graphics (SIGGRAPH 2001 Proceedings)* (Aug. 2001), pp. 57–66.
- [58] LANCASTER, P., AND SALKAUSKAS, K. Surfaces Generated by Moving Least Squares Methods. *Mathematics of Computation* 37, 155 (July 1981), 141–158.

## BIBLIOGRAPHY

---

- [59] LANCZOS, C. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *J. Res. Nat. Bur. Stand.* 45 (1950), 255–282.
- [60] LEVIN, D. Mesh-Independent Surface Interpolation. In *Geometric Modeling for Scientific Visualization*, G. Brunett, B. Hamann, K. Mueller, and L. Linsen, Eds. Springer-Verlag, 2003.
- [61] LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. The Digital Michelangelo Project: 3D Scanning of Large Statues. In *Computer Graphics (SIGGRAPH 2000 Proceedings)* (2000), pp. 131–144.
- [62] LOBAY, A., AND FORSYTH, D. A. Recovering Shape and Irradiance Maps from Rich Dense Texton Fields. In *CVPR (1)* (2004), pp. 400–406.
- [63] LORENSEN, W. E., AND CLINE, H. E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (July 1987), pp. 163–170.
- [64] MITRA, N. J., GELFAND, N., POTTMANN, H., AND GUIBAS, L. Registration of Point Cloud data from a Geometric Optimization Perspective. In *Symposium on Geometry Processing* (2004), pp. 23–32.
- [65] MITRA, N. J., NGUYEN, A., AND GUIBAS, L. Estimating Surface Normals in Noisy Point Cloud Data. *International Journal of Computational Geometry and Applications* 14, 4–5 (2004), 261–276.
- [66] NG, A. Y., JORDAN, M., AND WEISS, Y. On Spectral Clustering: Analysis and an Algorithm. In *Neural Information Processing Systems* (2001), pp. 849–856.
- [67] OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H.-P. Multi-Level Partition of Unity Implicits. *ACM Transactions on Graphics* 22, 3 (July 2003), 463–470.
- [68] OSHER, S., AND FEDKIW, R. *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag, New York, 2003.
- [69] PAULY, M., GROSS, M., AND KOBBELT, L. P. Efficient Simplification of Point-Sampled Surfaces. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 163–170.
- [70] PAULY, M., KEISER, R., KOBBELT, L. P., AND GROSS, M. Shape Modeling with Point-Sampled Geometry. *ACM Transactions on Graphics* 22, 3 (2003), 641–650.

## BIBLIOGRAPHY

---

- [71] POLLEFEYS, M., KOCH, R., VERGAUWEN, M., AND GOOL, L. J. V. Metric 3D Surface Reconstruction from Uncalibrated Image Sequences. In *SMILE'98: Proceedings of the European Workshop on 3D Structure from Multiple Images of Large-Scale Environments* (London, UK, 1998), Springer-Verlag, pp. 139–154.
- [72] POTHEN, A., SIMON, H. D., AND LIOU, K.-P. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM Journal on Matrix Analysis and Applications* 11, 3 (July 1990), 430–452.
- [73] PULLI, K. Multiview Registration for Large Sets. In *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling* (Ottawa, Canada, Oct. 1999), pp. 160–168.
- [74] PUZICHA, J., HOFMANN, T., AND BUHMANN, J. Non-Parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval. In *CVPR (1997)*, pp. 267–272.
- [75] RUSINKIEWICZ, S., AND LEVOY, M. Efficient Variants of the ICP Algorithm. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling* (Québec City, Canada, May 2001), pp. 145–152.
- [76] Scanalyze. <http://graphics.stanford.edu/software/scanalyze>.
- [77] SETHIAN, J. A. *Level Set Methods and Fast Marching Methods*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.
- [78] SHEN, C., O'BRIEN, J. F., AND SHEWCHUK, J. R. Interpolating and Approximating Implicit Surfaces from Polygon Soup. In *Proceedings of ACM SIGGRAPH 2004* (Aug. 2004), ACM.
- [79] SHEPARD, D. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. In *Proceedings of the 1968 23rd ACM national conference* (New York, NY, USA, 1968), ACM Press, pp. 517–524.
- [80] SHI, J., AND MALIK, J. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (Aug. 2000), 888–905.
- [81] TELLER, S., ANTONE, M., BODNAR, Z., BOSSE, M., COORG, S., JETHWA, M., AND MASTER, N. Calibrated, registered images of an extended urban area. *International Journal of Computer Vision* 53, 1 (2003), 93–107.
- [82] THIRION, J. The extremal mesh and the understanding of 3d surfaces. *IJCV* 19 (1996), 115–128.

## BIBLIOGRAPHY

---

- [83] TURK, G., AND O'BRIEN, J. Shape Transformation Using Variational Implicit Functions. In *Computer Graphics (SIGGRAPH '99 Proceedings)* (1999), pp. 335–342.
- [84] VRANIC, D. V. An Improvement of Rotation Invariant 3D Shape Descriptor Based on Functions on Concentric Spheres. In *Proceedings of the IEEE International Conference on Image Processing* (2003), pp. 757–760.
- [85] YU, S. X., AND SHI, J. Understanding Popout through Repulsion. In *IEEE Conference on Computer Vision and Pattern Recognition* (Dec. 2001).
- [86] ZHANG, R., TSAI, P., CRYER, J., AND SHAH, M. Shape from Shading: A survey. *Pattern Analysis and Machine Intelligence* 21, 8 (August 1999), 690–706.
- [87] ZHAO, H.-K., OSHER, S., AND FEDKIW, R. Fast Surface Reconstruction Using the Level Set Method. In *First IEEE Workshop on Variational and Level Set Methods* (2001), pp. 194–202.