

Diss. ETH No. 20905

# **Meshless Sampling and Reconstruction of Manifolds and Patterns**

A dissertation submitted to  
**ETH Zürich**

for the Degree of  
**Doctor of Sciences**

presented by

**A. Cengiz Öztireli**

M.Sc. Informatik-Ing., ETH Zürich, Switzerland

born 30. September 1983

citizen of Türkiye

accepted on the recommendation of

**Prof. Dr. Markus Gross**, examiner

**Prof. Dr. Marc Alexa**, co-examiner

**Prof. Dr. Mark Pauly**, co-examiner

2013



# Abstract

It is one of the main goals of Computer Graphics in particular, and science in general, to understand and mimic the complexity in the real world. Over the past decades, it has been proven that the mathematical structures of *manifolds*, and *stochastic point patterns* result in accurate and efficient computational representations for the geometric complexity of the world, and modeling these structures with meshless methods offers a versatile and unified treatment. In this thesis, we develop techniques and algorithms to tackle the fundamental problems of meshless *sampling* and *reconstruction* of manifolds and point patterns.

The acquired data sampled from manifold surfaces of objects is often noisy, corrupted with outliers, and sparse in some parts of the surface. It is thus very challenging to generate accurate reconstructions of the underlying surface. The first problem we address is the generation of *robust*, and *sharp feature* and *high frequency detail* preserving reconstructions of point sampled manifolds. Due to the common smoothness assumption, most approximation methods, when directly applied to the manifold surface reconstruction problem, can only generate smooth surfaces without such features, and are significantly affected by outliers. We propose to reformulate the *moving least squares* based point set surface reconstruction methods in the framework of *local kernel regression*, which enables us to incorporate methods from *robust statistics* to arrive at a feature preserving and robust point set surface definition. The new implicit surface definition can preserve fine details and all types of sharp features with controllable sharpness, has a simple analytic form, is robust to outliers and sparse sampling, and efficient and simple to compute. Since the definition is continuous, it is amenable to further processing without any special treatment.

The accuracy of the reconstruction of a surface is necessarily determined by the density and distribution of the points sampled from it. It is thus essential to ensure a dense enough sampling for faithful reconstructions. On the other hand, typical datasets can be massive and redundant in some parts with billions of points, which significantly degrades the performance of the reconstruction algorithms. Hence, finding *optimal sampling conditions* for a given reconstruction method is essential for efficient and accurate reconstructions. We propose new *simplification* and *resampling* algorithms that result in accurate reconstructions while minimizing redundancy. The algorithms are out-of-core, efficient, simple to implement, feature sensitive,

and generate high quality blue noise distributions. They utilize a new measure that quantifies the effect a point has on the definition of a manifold, if it is added to the set defining the manifold, by considering the change in the Laplace-Beltrami spectrum. We derive an approximation of this measure by a novel technique that combines *spectral analysis of manifolds* and *kernel methods*. Although the measure is conceptually global, it requires only local computations, making the algorithms time and memory efficient.

Not all structures of the real world admit a deterministic manifold form. Indeed, many structures, from the distribution of trees in a forest or pores in a piece of Swiss cheese to those of molecular particles or movements of humans in a crowd are best modeled in a distributional sense by *stochastic point patterns*. Reconstruction of such patterns from given example distributions is thus of utmost importance. To achieve this, we first propose a new *unified analysis of point distributions* based on a kernel based approximation of the *pair correlation function* (PCF). This analysis shows that the PCF is sufficient for unique determination and discrimination of most point patterns, and that there is a quantifiable relation between patterns depending on a new measure of their *irregularity*. Following this analysis, we propose the first algorithms that can synthesize point distributions with characteristics matching those of provided examples, by minimizing a certain distance between the PCFs. Our first algorithm is a *generalized dart throwing* method that accepts or rejects added points depending on the PCF. The second *gradient descent* based algorithm takes the output of the first algorithm, and moves the points so as to minimize the distance between the target PCF and the PCF of the final output point set. The resulting point distributions have the characteristics of the target patterns to be reconstructed.

# Zusammenfassung

Eines der Hauptziele spezifisch in der Computergrafik aber auch allgemein in der Wissenschaft, besteht darin, die Komplexität der realen Welt zu verstehen, zu modellieren und nachzuahmen. In den letzten Jahrzehnten wurde bewiesen, dass die mathematischen Strukturen von Mannigfaltigkeiten und stochastischen Punktverteilungen genaue und effizient berechenbare Repräsentationen dieser Komplexität ermöglichen, und dass die Modellierung dieser Strukturen mit punkt-basierten Ansätzen eine vereinheitlichte und allgemeine Behandlung erlaubt. In dieser Dissertation entwickeln wir Techniken und Algorithmen um fundamentale Probleme von punkt-basiertem Sampling und Rekonstruktion von Mannigfaltigkeiten und Punktverteilungen zu lösen.

Die Daten welche wir durch die Abtastung von mannigfaltigen Oberflächen erhalten sind oft verrauscht, durch Ausreisser verfälscht, sowie auch teilweise spärlich diskretisiert. Dies macht die Rekonstruktion der Oberfläche sehr schwierig. Als erstes fokussieren wir uns auf die Berechnung von punkt-basierten Oberflächen welche die Robustheit, scharfe Kanten sowie hochfrequente Oberflächendetails erhalten. Angewendet auf mannigfaltige Oberflächen erzeugen die meisten existierenden Rekonstruktionsmethoden glatte Oberflächen jedoch ohne Erhaltung von Merkmalen, und erweisen sich des Weiteren als anfällig auf Rauschen und Ausreisser. Wir stellen daher eine Methode vor, welche das Moving Least Squares-Verfahren basierend auf Punktdaten mit linearen Kernel-Regressionen ausführt. Dies erlaubt uns die Anwendung von robusten Schätzverfahren um eine punkt-basierte Oberflächendefinition herzuleiten, welche einerseits Details erhalten kann und andererseits robust ist. Die neue implizite Oberflächendefinition erzeugt hochaufgelöste Merkmale und erhält scharfe sowie auch kontrollierbare Oberflächendetails, hat eine einfache analytische Form, ist robust gegenüber Ausreissern und spärlicher Diskretisierung, und ist effizient und einfach zu berechnen. Durch die kontinuierliche Definition ist es ausserdem möglich weitere Verarbeitungsschritte ohne spezielle Zusatzbehandlung anzuwenden.

Die Genauigkeit der Oberflächenrekonstruktion wird hauptsächlich durch die Dichte und Verteilung der Punkte bestimmt welche die Oberfläche repräsentieren. Um eine gute Qualität zu erhalten ist es daher wichtig eine genügend dichte Abtastung zu verwenden. Typische Datensätze sind jedoch sehr gross und bestehen aus Milliarden von Punkten, und weisen in einigen Teilen eine grosse Redundanz

auf. Dies beeinflusst die Effizienz der Algorithmen merklich. Aufgrund von diesen Merkmalen ist es entscheidend dass man eine optimale Punkte-Diskretisierung findet. Wir stellen neue Algorithmen vor um die Datenmenge zu vereinfachen und neue Abtastungspunkte zu definieren, welche die Redundanz verkleinern sowie auch eine akkurate Rekonstruktion erlauben. Die Algorithmen sind Out-of-Core, effizient, einfach zu implementieren, berücksichtigen Merkmale, und erzeugen eine hohe Qualität bezüglich der Verteilung von Blauem Rauschen. Sie verwenden dabei ein neues Mass, welches den Effekt quantisiert welcher ein Punkt auf die resultierende mannigfaltige Oberfläche hat. Dabei wird der Einfluss des Punktes durch die Änderung des Laplace-Beltrami Spektrums analysiert. Von diesem Mass ausgehend leiten wir eine neue Annäherung her, welche die spektrale Analyse von mannigfaltigen Oberflächen mit Kernel-Methoden kombiniert. Obwohl dieses Mass konzeptionell global ist werden nur lokale Berechnungen benötigt, was den Algorithmus zeit- und speichereffizient macht.

Nicht alle Strukturen der realen Welt entsprechen der mannigfaltigen Form. Viele Strukturen wie zum Beispiel die Verteilung von Bäumen in einem Wald, Löcher in einem Schweizer Käse, molekulare Partikel, oder auch die Bewegungen einer Personenmenge können am Besten durch stochastische Punktemuster beschrieben werden. Die Rekonstruktion von solchen Verteilungen ist daher von höchster Wichtigkeit. Um dies zu erreichen, führen wir eine neue allgemeine Analyse von Punktmengen ein, basierend auf der kernelbasierten approximierten Paar-Korrelationsfunktion (PCF). Unsere Analyse hat ergeben, dass PCF ausreichend ist um die meisten Punktemuster eindeutig zu bestimmen und zu unterscheiden, und dass eine quantifizierbare Relation zwischen diesen Mustern existiert welche abhängig ist von der Irregularität der Punkte. Dieser Analyse folgend stellen wir einen neuen Algorithmus vor welcher beliebige Punktverteilungen synthetisieren kann, basierend auf der Charakteristik einer gegebenen Beispielsmenge. Dabei werden die PCF Distanzen der erzeugten Punkteverteilung und der Beispielsmenge minimiert. Unser erster Algorithmus den wir vorstellen entspricht einer verallgemeinerten Pfeilwurf-Methode welche die hinzugefügten Punkte entweder akzeptiert oder ablehnt, abhängig vom PCF. Die zweite Methode basiert auf dem Gradientenverfahren das die Ausgabe des vorhergehenden Algorithmus verwendet und die Punkte so bewegt dass die Distanz zwischen dem angestrebten PCF und dem PCF der resultierenden Punktemenge minimiert wird. Die resultierende Punktemenge weist dabei die selbe Charakteristik auf wie die des vorgegebenen Musters.

# Acknowledgments

First and foremost, I would like to thank my advisor, Prof. Markus Gross. He introduced me into a fascinating field, and continuously supported and helped me throughout my Ph.D to overcome any problems encountered, and produce works of high value. I feel privileged to have been able to gain from his knowledge, experience, talent, scientific curiosity, and enthusiasm.

I would like to thank my collaborators, and past advisors. I feel extremely lucky to have worked with great scientists. It has been a pleasure to collaborate with Prof. Marc Alexa, and learn from his deep understanding, innovative ideas, and practical solutions. I had the great chance of being advised by Prof. Mark Pauly during my master studies, who introduced me to computer graphics. I learned a lot from his approach towards science, scientific curiosity and thinking. Prof. Çağatay Başdoğan's support as my undergraduate advisor was invaluable for the development of my scientific career. I am also deeply grateful to my collaborators Prof. Alla Sheffer, Dr. Gaél Guennebaud, Dr. Tiberiu Popa, and Umut Uyumaz.

It was a pleasure to have Dr. Sebastian Martin, Dr. Marcel Germann, Dr. Tobias Pfaff, Dr. Thomas Oskam, Dr. Bernd Bickel, Peter Kaufmann, Prof. Hao Li, Dr. Balint Miklos, and Dr. Michael Eigensatz as friends throughout my Ph.D. I will miss our discussions on about anything and everything. Special thanks go to Dr. Tobias Pfaff for helping me on submissions, and Dr. Barbara Solenthaler and Dr. Sebastian Martin for translating the abstract. I also would like to thank all past and current CGL, DRZ, AGG, CVG, and IGL members for making the work place a fun and friendly environment.

Finally, I would like to express my deep thanks to my friends, especially Ali Sengül, Nihal Dindar, Nida Rizwan Farid, and M. Fatih Toy, my dear parents and brother, and Amina. This thesis would have been impossible without her love and support.

This thesis was supported by the Swiss National Science Foundation (Der Schweizerische Nationalfonds) (grant No. 200021-132438 / 1).



# Contents

<b>Introduction</b>	<b>1</b>
1.1 Contributions . . . . .	5
1.2 Publications . . . . .	7
<b>Background and Related Work</b>	<b>9</b>
2.1 Manifold Geometry Reconstruction . . . . .	11
2.1.1 Combinatorial Methods . . . . .	11
2.1.2 Scattered Data Approximation Methods . . . . .	12
2.1.3 Global and Variational Methods . . . . .	15
2.1.4 Methods with Structural Assumptions . . . . .	15
2.1.5 Issues with the Reconstruction Methods . . . . .	15
2.2 Manifold Geometry Sampling . . . . .	19
2.2.1 Sampling for Combinatorial Reconstruction . . . . .	19
2.2.2 Sampling for Surface Approximation . . . . .	19
2.2.3 Sampling in Manifold Learning . . . . .	21
2.3 Stochastic Point Patterns . . . . .	22
2.3.1 Blue Noise Patterns . . . . .	22
2.3.2 Point Distributions in Computer Graphics . . . . .	23
2.3.3 General Point Patterns with Point Processes . . . . .	25
2.3.4 Point Process Statistics . . . . .	25
2.3.5 Synthesis of Point Patterns . . . . .	26
<b>Meshless Reconstruction of Manifolds</b>	<b>27</b>
3.1 Differential Geometry of Manifolds . . . . .	29
3.1.1 Submanifolds . . . . .	29
3.1.2 Implicit Surfaces . . . . .	29
3.2 Reconstruction by Local Approximations . . . . .	32
3.2.1 Moving Least Squares . . . . .	32
3.2.2 Local Kernel Regression . . . . .	33
3.2.3 Derivation of MLS Surfaces from LKR . . . . .	34
3.3 Robust Local Approximations for Feature Preservation . . . . .	37
3.3.1 Robust Statistics . . . . .	37
3.3.2 Robust Local Kernel Regression . . . . .	38
3.4 Robust and Feature Preserving Point Set Surfaces . . . . .	40

## Contents

3.4.1	Derivation of the Surface Definition . . . . .	40
3.4.2	Computation of the Surface . . . . .	42
3.4.3	Robust Normal Mollification . . . . .	43
3.5	Results and Analysis . . . . .	45
3.5.1	Implementation and Performance Analysis . . . . .	45
3.5.2	Reconstruction Results . . . . .	48
3.6	An Application of Local Approximations with Global Constraints .	53
3.7	Discussion and Outlook . . . . .	56
<b>Spectral Sampling of Manifolds</b>		<b>59</b>
4.1	Spectral Analysis of Manifolds . . . . .	62
4.1.1	Harmonic Analysis . . . . .	62
4.1.2	The Laplace-Beltrami Operator . . . . .	63
4.1.3	The Heat Kernel . . . . .	65
4.2	Kernel Techniques . . . . .	68
4.2.1	Positive Definite Kernels . . . . .	69
4.2.2	Reproducing Kernel Hilbert Spaces . . . . .	71
4.2.3	Mercer's Theorem and the Heat Kernel . . . . .	72
4.3	Spectral Measures for Manifold Sampling . . . . .	74
4.3.1	Discretization of the Laplace-Beltrami Operator . . . . .	75
4.3.2	Spectral Change due to a Point . . . . .	76
4.3.3	Computation of the Measure . . . . .	78
4.4	Coupling Sampling to Reconstruction . . . . .	81
4.4.1	Relation to Kernel Regression . . . . .	81
4.4.2	Reconstruction Method of Chapter 3 . . . . .	82
4.5	Algorithms for Sampling . . . . .	83
4.5.1	Randomized Linear Scan . . . . .	83
4.5.2	Iterative Gradient Ascent . . . . .	84
4.5.3	Parameters and Data Structures . . . . .	85
4.5.4	Multi-resolution and Progressive Sampling . . . . .	86
4.6	Results and Analysis . . . . .	88
4.6.1	Quality of Samplings . . . . .	88
4.6.2	Quality of Remeshings . . . . .	89
4.6.3	Accuracy of Reconstructions . . . . .	92
4.6.4	Performance . . . . .	93
4.7	Application to Facial Rendering . . . . .	96
4.8	Discussion and Outlook . . . . .	98
<b>Reconstruction of Point Patterns</b>		<b>101</b>
5.1	Point Processes . . . . .	104
5.1.1	Hard-core Processes . . . . .	105
5.1.2	Clustering Processes . . . . .	106

5.1.3	General Process Models . . . . .	107
5.2	Statistics for Point Processes . . . . .	109
5.2.1	Definitions . . . . .	109
5.2.2	First Order Statistics . . . . .	110
5.2.3	Second Order Statistics . . . . .	110
5.2.4	Higher Order Statistics . . . . .	111
5.3	Analysis of Point Distributions . . . . .	113
5.3.1	The Pair Correlation Function . . . . .	113
5.3.2	Estimation of the PCF . . . . .	114
5.3.3	The Pair Correlation Space . . . . .	117
5.3.4	Characterization of the Distributions in the PCS . . . . .	118
5.3.5	A Measure of Irregularity . . . . .	120
5.4	Synthesis of Point Distributions . . . . .	122
5.4.1	Generalized Dart Throwing . . . . .	123
5.4.2	PCF Fitting by Gradient Descent . . . . .	124
5.4.3	Analysis . . . . .	124
5.4.4	Extensions . . . . .	126
5.5	Results . . . . .	127
5.6	Applications . . . . .	131
5.6.1	Synthesizing Real World Structures . . . . .	131
5.6.2	Turbulent Fluid Simulations . . . . .	132
5.7	Discussion and Outlook . . . . .	135
<b>Conclusions</b>		<b>137</b>
6.1	Wider Outlook . . . . .	138
<b>Appendix</b>		<b>141</b>
A.1	Iteratively Reweighted Least Squares . . . . .	141
A.2	Pseudocode for Projecting a Point onto the Surface . . . . .	144
A.3	Distribution of Effect of a Point on the Spectrum . . . . .	145
A.4	Projection onto a Span in a Reproducing Kernel Hilbert Space . . . . .	146
A.5	Notation and Glossary . . . . .	147
A.6	Notation . . . . .	147
A.6.1	Spaces . . . . .	147
A.6.2	Operators . . . . .	147
A.6.3	Other Definitions . . . . .	148
A.6.4	Background and Related Work . . . . .	148
A.6.5	Meshless Reconstruction of Manifolds . . . . .	149
A.6.6	Spectral Sampling of Manifolds . . . . .	150
A.6.7	Reconstruction of Point Patterns . . . . .	151
A.7	Glossary . . . . .	152
A.8	Curriculum Vitae . . . . .	153

*Contents*

**Bibliography**

**157**

---

# C H A P T E R

# 1

## Introduction

Understanding, analyzing, modeling, and recreating complex geometric structures of the world are some of the prominent goals of Computer Graphics. Researchers have been working on how to represent and process the seemingly continuous data gathered from the real world in the discrete world of the computers in the most effective way. The data can represent complex geometric structures and patterns. To further complicate the matters, it is often noisy, inaccurate, incomplete in some parts, and redundant in others.

*Sampling* and *reconstruction* are two fundamental problems underlying effective processing of data. Reconstruction refers to defining a mathematical structure and computational method that accurately capture the geometry or pattern the data represents. This is only possible with sufficient information in the form of samples gathered. Sampling deals with how to use the optimum amount and distribution of samples such that information loss and redundancy are avoided.

Many geometric structures can be represented by *manifold* surfaces (Figure 1.1, top row), which constitute one of the focuses of this thesis. Such surfaces bounding objects can be captured by gathering point samples using various techniques. These samples are then used in approximation algorithms to reconstruct a mathematical definition of the surface, evaluated algorithmically. The algorithms should tolerate noise and outliers, and output accurate reconstructions. It has been proven repeatedly that globally or locally fitting

## Introduction



**Figure 1.1:** *Some examples of the structures that can be well represented by manifolds (top row, the boundary of a statue, an animal, and a house), or patterns (bottom row, patterns formed by distributions of trees, flowers, and pores in a piece of cheese).*<sup>1</sup>

a function with smoothness constraints imposed explicitly or implied by the functional space used is a flexible and accurate method. In particular, the method of *moving least squares* (MLS) has been in use extensively for reconstructing surfaces from point clouds. This method solves a local least squares system in which more weights are given to the sample points close to a given query point so as to achieve a local fit. This locally fitted proxy surface is then used to approximate the underlying surface at the query point.

Although the methods that depend on the solution of either global or local least squares systems can reconstruct smooth surfaces out of noisy point cloud data, the resulting surfaces lack some of the perceptually very important features such as sharp edges, corners, or small-scale details. Furthermore, the presence of outliers in the data greatly biases the surface due to their significant effect on the least squares systems. To address these limitations, we describe a new method that relies on using an error function that is *robust* in a statistical sense so that positional and normal outliers are downweighted. We show how this formulation leads to a feature and detail preserving smooth surface definition that is robust to noise and outliers, stable under sparse sampling, efficient to compute, and simple to implement (Chapter 3).

As with any surface definition, generating accurate surfaces with the proposed definition requires having sufficiently many samples. For reconstructing

---

<sup>1</sup>From left to right, top to bottom: Courtesy of Berthold Werner and BarceX, Allen McCloud, Wikimedia Commons, Hansueli Krapf, André Karwath, Jon Sullivan.

tions to have feasible time and memory complexity, it is also important to have the least possible amount of data to process. This is especially important for typical massive point sets acquired, which can have billions of points. Although techniques from signal processing can be used to derive optimum sampling rates for a given function, this is much harder for the case of sampling a manifold. There has been efforts to extend the results in classical signal processing to discrete representations of manifolds in the spectral graph theory and computational harmonic analysis literature, which has led to new concepts and algorithms for manifolds in geometry processing. However, generalization of the concepts paralleling the sampling theory is non-trivial, and also computationally challenging due to the global nature of spectral methods.

We address this problem and propose effective sampling algorithms (Chapter 4). By relying on results from spectral analysis of manifolds, kernel methods, and matrix perturbation theory, we derive a new measure that quantifies the change a manifold undergoes due to addition or removal of a single sample point, and design algorithms that utilize this measure to generate near-optimal samplings. Although the measure is conceptually global, all computations are local, making the resulting algorithms very efficient and out-of-core. The measure determines the change in the eigenspectrum of the Laplace-Beltrami operator of a manifold due to a single point. The algorithms try to get a sampling pattern where each point contributes maximally to the manifold definition and any unnecessary samples are eliminated. The samplings are coupled to MLS based reconstruction algorithms and thus produce very accurate reconstructions that preserve features. The distribution of samples also possesses high quality blue-noise characteristics, which allows direct high quality remeshing.

Manifold surfaces do not constitute the only way the geometric complexity is manifested in the real world. For some structures, the *pattern/distribution* of gathered point samples becomes the entity to be reconstructed. Such a reconstruction involves an analysis step for understanding certain characteristics of a known distribution, and a synthesis step to generate new distributions with the extracted characteristics. In contrast to the diversity of natural patterns (Figure 1.1, bottom row), existing works prior to ours on point distributions in computer graphics entirely focused on distributions with blue-noise characteristics, where points are distributed randomly with a minimum distance between each pair. Analysis of such distributions is typically performed by using periodograms and derived spectral measures that are confined to Euclidean domains and by qualitative assessment. Similarly, the synthesis algorithms have been designed to only generate blue-noise distributions with slightly varying characteristics, optimized for particular applications. In

## Introduction

summary, there has been a need for a general analysis method that can work in any domain, provide compact representations for patterns, and explain the characteristics and relations of patterns; and a synthesis algorithm that can generate point patterns with arbitrary complexity and characteristics.

We present novel and unified analysis tools and the first general synthesis algorithms for point distributions (Chapter 5). Our methods depend on extracting a global statistical measure, the *pair correlation function*, that describes the joint probability of having pairs of point samples at particular locations in space. An embedding space implied by this measure is then used to provide a unified analysis of distributions, and to show that the pair correlation function is an accurate representation for characteristics of patterns. Based on this analysis, synthesis algorithms for general distributions are proposed. The algorithms match pair correlation functions of output and given example distributions. They work for multi-class and adaptive distributions with target and output point sets of different sizes and dimensions. These properties allow us to reconstruct a large range of patterns from distributions of different types of trees in a forest to vortex particles for turbulent fluid simulations.

## 1.1 Contributions

Our main contributions can be classified as theoretical and practical. On the theoretical side, we established and utilized relations between: *moving least squares surfaces* and *local kernel regression*, and *spectral analysis of manifolds* and *kernel methods*. We also explored the pair correlation function in a new light that led to the development of a *unified analysis of point patterns*. On the practical side, we proposed a *sharp feature preserving point set surface* reconstruction method that has a simple analytic form with local computations and without explicit marking of the features, a *manifold sampling algorithm based on spectral measures* that generates accurate reconstructions and high quality isotropic samplings of manifold surfaces, and a *general point pattern reconstruction algorithm* that can synthesize possibly multi-class distributions with any given characteristics on general metric spaces. In addition to these main results, we also proposed a method for *3D reconstruction of sketched objects* and an *adaptive rendering algorithm for faces*. Below we detail the main contributions:

- A unified derivation of some meshless surface reconstruction methods that interprets them as instances of a *local regression* analysis. In addition to contributing to the understanding of the reconstruction methods, this unification also opens doors to the statistics literature on extensions of local regression.
- A surface reconstruction algorithm that can reconstruct sharp features and fine details, is robust to noise and outliers, stable under sparse sampling, and efficient to compute. This definition is based on integrating a *robust statistics* approach into the framework of local kernel regression.
- A discrete spectral analysis of manifolds that utilize results from *harmonic analysis* and *kernel methods*. This analysis relies on defining a feature space for the heat kernel and operating in this space to define new spectral measures.
- Subsampling and resampling algorithms that are time and memory efficient, easy to control with intuitive parameters tailored to the reconstruction method used, result in accurate feature sensitive reconstructions and high quality samplings. These algorithms utilize a spectral measure that quantifies the change a point makes on the *Laplace-Beltrami spectrum* of a manifold. The measure is computed via approximations derived from our discrete spectral analysis of manifolds.
- An analysis technique for point patterns that can explain variability of general point distributions in a unified way by describing them in an embedding space implied by the *pair correlation function*. We showed that the degrees of freedom for distributions generated by various natural or computational

## Introduction

processes are actually low and directly related to regularity, and the pair correlation function is a compact and unique representation for most practical patterns. The analysis also led to a new *irregularity* measure.

- A point distribution synthesis algorithm for synthesizing general multi-class distributions. We generalize the well-known dart throwing and relaxation methods to synthesize distributions with *arbitrary characteristics*. The characteristics can be given by the user or extracted from an example distribution. The example and output point sets can be of different sizes and dimensions, reside on different domains, and contain multiple classes of points. The algorithms are efficient with  $O(n)$  complexity for  $n$  points, and simple to implement.

## 1.2 Publications

The following works were produced in the course of this thesis.

Peer-reviewed publications in journals and conferences:

- A. C. ÖZTIRELI, G. GUENNEBAUD, and M. GROSS. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. In *Proceedings of Eurographics (Munich, Germany, March 30 - April 3, 2009)*, *Computer Graphics Forum*, vol. 28, no. 2, pp. 493-501 (Best Student Paper Award).
- A. C. ÖZTIRELI, M. ALEXA, and M. GROSS. Spectral Sampling of Manifolds. In *Proceedings of ACM SIGGRAPH Asia (Seoul, Korea, December 15-18, 2010)*, *ACM Transactions on Graphics*, vol. 29, no. 5, pp. 168:1-168:8.
- A. C. ÖZTIRELI, U. UYUMAZ, T. POPA, A. SHEFFER, and M. GROSS. 3D Modeling with a Symmetric Sketch. In *Proceedings of SBIM (Vancouver, Canada, August 5-7, 2011)*.
- H. KIM, A. C. ÖZTIRELI, M. GROSS, and S. CHOI. Adaptive surface splatting for facial rendering. In *Proceedings of CASA 2012 (Singapore, May 9-11, 2012)*, *Computer Animation and Virtual Worlds*, vol. 23, no. 3-4, pp. 363-373.
- A. C. ÖZTIRELI, and M. GROSS. Analysis and Synthesis of Point Distributions based on Pair Correlation. In *Proceedings of ACM SIGGRAPH Asia (Singapore, November 28-December 1, 2012)*, *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 170:1-170:10.

Technical reports:

- A. C. ÖZTIRELI, M. ALEXA, and M. GROSS. Spectral Sampling of Manifolds: Extended Version. *Technical Report No. 683, Institute of Visual Computing, ETH Zürich, 2010.*

## *Introduction*

---

# C H A P T E R

# 2

## Background and Related Work

In this chapter, we provide the fundamental techniques used for processing manifold geometry and patterns acquired in the form of point samples, in the scope of this thesis. In addition to the ideas and works in the computer graphics literature, we also review related works in other fields in order to provide a coherent picture of the state-of-the-art.

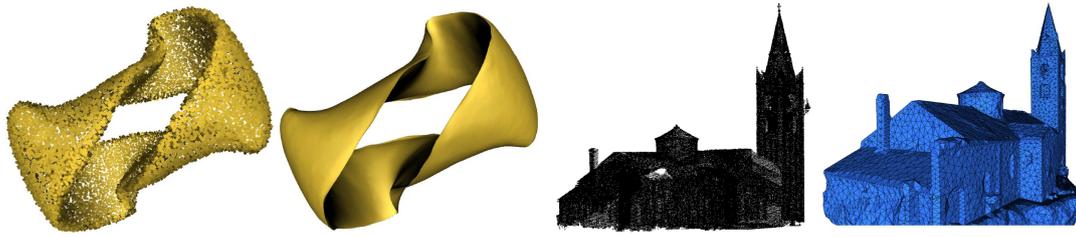
We start with a review of the current approaches to solve the problem of manifold geometry reconstruction under sparse sampling, noise, and outliers (Section 2.1). We explain that the *scattered data approximation* based methods, in particular *moving least squares* based *point set surfaces*, perform the best for typical corrupted real world data, and that some of the fundamental problems of the current scattered data approximation based manifold reconstruction methods can be solved by introducing ideas from the statistics literature.

Next, we develop an understanding of the existing sampling algorithms for accurate reconstructions with minimum data redundancy, point out their weaknesses, and state how they can be alleviated (Section 2.2). Our main focus is on *sampling algorithms for kernel based reconstruction* methods. We point out how the concept of a kernel unify different methods and can be utilized to develop new sampling algorithms with superior properties.

Finally, we study the methods for analysis and synthesis of point patterns (or distributions) (Section 2.3). Since most techniques in computer graphics are

## *Background and Related Work*

focused on blue noise distributions, we also review the relevant works from statistics and physics literature for handling general points distributions. We state how these works are related to our novel *unified analysis* method and *general synthesis/reconstruction* algorithms.



**Figure 2.1:** Points sampled on surfaces and the corresponding reconstructions (left is courtesy of Avron et al. [2010], right is courtesy of Salman et al. [2010])

## 2.1 Manifold Geometry Reconstruction

Reconstructing manifold geometry underlying a set of given sample points is a well-studied problem due to the abundance of such data coming from various sources. However, growing complexity of data with delicate structures, and high levels of noise and outliers due to the spread of low cost capturing devices constantly challenge the proposed reconstruction methods. Examples of acquired datasets and possible reconstructions of them are shown in Figure 2.1. We mainly focus on *general purpose reconstruction methods*, which assume that the only information available about the underlying geometry is the sample point locations and additional attributes such as surface normals. Hence, these algorithms do not rely on a template shape, databases of shapes, or other prior information on the specific geometry to be reconstructed. These general algorithms are applicable to a wide range of problems and thus have a large impact. However, the generality also makes the reconstruction problem ill-posed for many cases. This has led to the development of a multitude of methods with different characteristics.

### 2.1.1 Combinatorial Methods

These methods generate a mesh interpolating a possibly filtered subset of sample points. They are based on Voronoi diagram based constructs, Delaunay triangulations, or alpha shapes [Edelsbrunner and Mücke, 1994; Amenta and Bern, 1998; Amenta et al., 2000; 2001; Dey and Goswami, 2003; Dey, 2006]. A simple form of the algorithms works by constructing a Delaunay triangulation of the points in 3D, segmenting the tetrahedra as inside or outside, and taking the union of each triangle where an inside and outside tetrahedron meet as the reconstructed mesh.

These methods work well for noiseless data and some of them come with geometry or topology guarantees under certain sampling conditions. They

have also been extended to preserve sharp features [Kuo and Yau, 2006], intersections, and boundaries [Dey et al., 2012]. However, typical data acquired from the real world is often contaminated with noise and outliers, which degrades the accuracy and robustness of these methods significantly. To solve this problem, there has been efforts to robustify the reconstructions by filtering the initial points [Dey and Goswami, 2004], identifying inside/outside regions robustly by a global graph [Kolluri et al., 2004], or simplifying an initial simplicial complex via optimal transport [Digne et al., 2012]. Nevertheless, most reconstructions (with the exception of Digne et al.’s [Digne et al., 2012]) produced are still sensitive to the sampling rate and noise in the data. Moreover, constructing the diagrams needed is computation and memory intensive, making the application of these methods to large datasets very difficult.

### 2.1.2 Scattered Data Approximation Methods

To increase robustness to noise and outliers, reconstruction methods based on scattered data approximation are used. The general problem can be formulated as defining an implicit surface as the zero set of a function,  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ ,  $f(\mathbf{x}) = 0$ , that fits the data well while satisfying some assumptions such as the smoothness level. Some of these methods are global, where a large minimization problem is solved, and others are local, where a simple surface is fitted to the neighborhood of a given query point. In general, global methods are more resilient to large portions of missing data and non-uniform samplings, while local methods are more efficient, parallelizable, and thus applicable to large datasets.

**Radial Basis Functions.** One common approach that can lead to global or local fitting defines the fitted function as a linear combination of *radial basis functions* (RBF) and minimizes a fitting energy by solving a global linear system [Carr et al., 2001; Turk and O’Brien, 2002; Ohtake et al., 2005] to find the coefficients for the basis functions. The implicit function is thus written as:

$$f(\mathbf{x}) = \sum \alpha_i k(\|\mathbf{x} - \mathbf{x}_i\|), \quad (2.1)$$

where  $k$  is the radial basis function used,  $\|\mathbf{x}\|$  denotes the Euclidean norm of  $\mathbf{x}$ , and  $\alpha_i$ ’s are the coefficients to be solved for. Hence, the function  $f$  stays in the span of the basis functions  $k(\|\mathbf{x} - \mathbf{x}_i\|)$ , which are chosen to satisfy certain properties such as a certain degree of smoothness. Once this form is assumed, data constraints in the form of  $Af(\mathbf{x}_i) = a_i$  can be imposed for some operator

$A$  and the corresponding attributes  $a_i$  extracted from the data. As an example, the operator can be the identity operator, and  $a_i$  can be set to the expected distance of the sample points to the surface. These constraints result in a linear system to solve for the coefficients:  $\mathbf{A}\alpha = \mathbf{a}$ . This system can be sparse or dense, depending on the support of the basis functions used. Locally supported basis functions lead to much more efficiently solvable systems and local evaluations, while basis functions with global support result in more stable surfaces under non-uniform sampling.

**Partition of Unity Implicit.** Instead of solving a large linear system, *partition of unity implicit* [Ohtake et al., 2003; Nagai et al., 2009] blend locally defined functions. In this case, the implicit function is defined as:

$$f(\mathbf{x}) = \frac{\sum w_i(\mathbf{x})g_i(\mathbf{x})}{\sum w_i(\mathbf{x})} \quad (2.2)$$

for some local approximations  $g_i$  and weights  $w_i$ . Unfortunately, the reconstructions are very sensitive to the design choices used to construct the local approximations and blending weights, necessitating to post-process the resulting reconstructions [Nagai et al., 2009].

**Moving Least Squares.** Another approach to reconstruct an implicit function for a surface utilizes the *moving least squares* method. Moving least squares (MLS) [Shepard, 1968] is a popular method for functional approximation of irregular data. It has been more recently extended for the manifold reconstruction problem [Alexa et al., 2001; Levin, 2003; Alexa et al., 2003]. The fundamental idea of MLS based surface reconstruction is fitting a proxy surface to the local neighborhood of a given evaluation point  $\mathbf{x}$ , and using this proxy surface to project  $\mathbf{x}$  onto the surface iteratively, or to compute the value of the implicit function  $f(\mathbf{x})$ .

Alexa et al. [Alexa et al., 2003] defined the MLS based surface as the set of stationary points of an iterative projection operator: at each step, a projection onto a polynomial approximation of the local neighborhood from a local planar parametrization is performed. By omitting the polynomial fitting step, Amenta and Kil [Amenta and Kil, 2004a] showed that the same surface can be defined and computed by weighted centroids and a smooth gradient field. This definition avoids the planar parametrization issues in the case of sparse sampling, and simplifies the representation, especially in the presence of normals [Alexa and Adamson, 2004; 2007].

However, plane fit cannot perform tight approximations and becomes unstable when the sampling rate drops [Amenta and Kil, 2004b; Guennebaud and Gross, 2007]. To overcome these limitations, Guennebaud and Gross

## Background and Related Work

proposed a generalization to directly fit higher order algebraic surfaces such as spheres [Guennebaud and Gross, 2007]. This approach yields an efficient closed form solution for the underlying surface [Guennebaud et al., 2008]. The resulting surface is given by the function  $f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{u}(\mathbf{x})$ , where  $\mathbf{b}(\mathbf{x})$  is the vector of basis functions for the algebraic distance to the locally fit surface, and the vector  $\mathbf{u}(\mathbf{x})$  contains the corresponding coefficients. For example, if a plane is used as the locally fit surface, then  $\mathbf{b}(\mathbf{x}) = [1 \ x \ y \ z]^T$ . The coefficients  $\mathbf{u}(\mathbf{x})$  depend on the point  $\mathbf{x}$ , and computed by solving the following minimization problem:

$$\mathbf{u}(\mathbf{x}) = \arg \min_{\mathbf{u}} \sum \left( \mathbf{b}(\mathbf{x}_i)^T \mathbf{u} - y_i \right)^2 \phi_i(\mathbf{x}), \quad (2.3)$$

where  $y_i$ 's are the estimated values for  $f$  at the sample points  $\mathbf{x}_i$ , and  $\phi$  is a function giving less weights to points away from  $\mathbf{x}$ . Note that since the sample points are assumed to be close to the surface (i.e.  $y_i = 0 \ \forall i$ ), the minimization can result in the trivial solution of zero for all the coefficients. To avoid this, constraints should be imposed on  $\mathbf{u}(\mathbf{x})$ . Typical constraints involve setting the magnitude of the gradient of the locally fitted surface to a constant, or using estimated or provided sample surface normals (Please see Sections 3.2.1 and 3.2.3 for a detailed discussion on these constraints.)

Finally, a quite different approach of utilizing MLS was proposed by Shen et al. [Shen et al., 2004]: instead of trying to fit trivariate polynomials to the data, they proposed to use standard MLS to reconstruct tangential implicit planes prescribed at each input sample position. When constant polynomials are used as the MLS basis, this method yields a simple weighted average [Kolluri, 2005]:

$$f(\mathbf{x}) = \frac{\sum \mathbf{n}_i^T (\mathbf{x} - \mathbf{x}_i) \phi_i(\mathbf{x})}{\sum \phi_i(\mathbf{x})} \quad (2.4)$$

with  $\mathbf{n}_i$  denoting the surface normal at the sample point  $\mathbf{x}_i$ . The simplicity of this expression allows efficient and simple evaluation and differentials for the surface. Note that this definition can also be regarded as a partition of unity approach where the local approximating functions are distances to the planes at the sample points. There are theoretical results on the accuracy of the reconstructed surfaces under certain sampling conditions [Kolluri, 2005]. However, in practice, the reconstructions exhibit expansion and shrinking effects leading to inaccuracies and instabilities [Guennebaud and Gross, 2007].

### 2.1.3 Global and Variational Methods

Apart from the mentioned RBF-based global methods, there exist further algorithms that take a global view. Most of these algorithms discretize the problem such that a large and sparse linear system needs to be solved, or the space around the surface is represented and processed. One successful method is recovering the indicator function of a surface [Kazhdan, 2005; Kazhdan et al., 2006; Manson et al., 2008]. Extensions that use basis functions at multiple scales [Kazhdan et al., 2006; Manson et al., 2008] have been proposed and proved to be quite effective at handling sparse data. Another global reconstruction technique is defining the surface as the solution of a variational optimization problem [Alliez et al., 2007; Mullen et al., 2010; Calakli and Taubin, 2011]. This problem is discretized to obtain a global solution, usually by solving a sparse linear system.

### 2.1.4 Methods with Structural Assumptions

So far, we have focused on general purpose reconstruction methods that does not assume any specific property of the surfaces other than smoothness. In this section, we explain some recent methods that rely on the more restrictive assumption that the surfaces contain certain local primitives or repetitive structures.

These methods rely on fitting primitives such as planes, spheres, cylinders, or other extracted shapes to patches of points [Schnabel et al., 2007; Gal et al., 2007] and learning their relations [Li et al., 2011], or using the assumption of repetitive structures to non-locally filter the geometry for a complete reconstruction under missing data and noise [Zheng et al., 2010; Friedman and Stamos, 2012; Digne, 2012; Guillemot et al., 2012]. However, primitive dependent methods cannot perform well on geometries without the assumed primitives, and the success of non-local filtering methods heavily depend on the amount and reliable detection of the repetitive structures in the data. Moreover, most of these methods do not generate a single mathematical definition for the overall surface, and are more suited for completing and filtering the geometry prior to reconstruction of the final surface definition.

### 2.1.5 Issues with the Reconstruction Methods

In spite of the significant progress in surface reconstruction, there remain several challenges to be addressed. Here, we explain the problems and briefly mention our solutions, please see Chapter 3 for a detailed discussion.

**Sharp Feature Preservation.** Reconstruction methods based on scattered data approximation are robust to noise and sparse sampling. This is possible because of the underlying *smoothness* assumption. However, this assumption also means that important features such as sharp features and other high frequency details are lost.

To overcome this limitation, various approaches have been proposed. Some of them rely on an explicit representation of the sharp creases using either cell complexes [Adamson and Alexa, 2006b] or tagged point clouds [Reuter et al., 2005; Guennebaud and Gross, 2007] to separate the input samples into different components. A more challenging task, however, is to automatically detect or enhance features present in the input point cloud.

As initially observed in the context of anisotropic smoothing [Jones et al., 2003; Boris Mederos and de Figueiredo, 2003], samples belonging to different smooth patches across a sharp feature can be seen as outliers. This suggests the use of *robust statistics* both to deal with real outliers and reconstruct sharp features. Following this idea, Fleishman et al. [Fleishman et al., 2005] designed an iterative refitting algorithm which locally classifies the samples across discontinuities. While constituting an important progress, the method requires very dense sampling, as well as special and complex handling to locally combine the different patches making the approach relatively expensive. Furthermore, it offers only limited flexibility to the user, and the lack of global consistency of the classifications yields  $C^{-1}$  discontinuities and jagged edges. While the latter limitation can be overcome using more advanced techniques [Daniels et al., 2007; Lipman et al., 2007], their inherent complexity makes them only suitable for surface reconstruction and not to produce an effective surface representation.

A quite different global approach is using techniques from sparse coding to construct sample normals and positions by solving an  $l - 1$  sparse problem [Avron et al., 2010]. Although this can resolve some of the ambiguities associated with local computations, the computational complexity of the solver limits the utility of the reconstructions. It is also possible to detect and preserve sharp features when triangulating an implicit surface via specialized Delaunay triangulations [Salman et al., 2010]. However, constructing such a triangulation can be expensive, and does not provide a continuous definition of the underlying surface.

We have proposed a novel approach that can naturally preserve any kind of high frequency features, from sharp edges to fine details, without any special handling or segmentation [Öztireli et al., 2009]. It reconstructs a continuous implicit function with a simple closed-form, which can be efficiently eval-

uated by local computations. The method is based on the observation that surface normals, not points, should be considered as outliers near a sharp feature. We explain this approach in Chapter 3 in detail.

**Surface Normal Estimation.** For some datasets, only point locations are available. Since many reconstruction methods require surface normals, it is essential to estimate the *direction* and *orientation* of the normals. The direction of the normals can be estimated by *fitting local surfaces* [Hoppe et al., 1992; Pauly et al., 2002; Alexa et al., 2003; Mitra and Nguyen, 2003; Guennebaud and Gross, 2007], *using Voronoi diagrams* [Amenta and Bern, 1998; Dey and Sun., 2005], or combining the two approaches [Alliez et al., 2007].

The basic idea of local surface fitting is locally approximating a manifold by a plane or a sphere around a sample point and using the normal of the fitted primitive as the estimate of the normal at the sample point. A plane fitting corresponds to performing a *principle component analysis* (PCA) on the local neighborhood [Hoppe et al., 1992; Pauly et al., 2002; Alexa et al., 2003; Mitra and Nguyen, 2003]. Hence, as long as the neighborhood size captures the local shape well, this fitting gives a robust estimate of the normal due to the inherent low-pass filtering. Although robust to noise, this method can give wrong estimates under the influence of outliers, high curvatures, and sparse sampling. To tackle these problems, we have proposed a feature preserving and robust filtering method [Öztireli et al., 2009] based on *robust statistics*, followed by other estimation methods utilizing similar concepts [Li et al., 2010a; Boulch and Marlet, 2012] (see Chapter 3). The main idea of these methods is that normals of a smooth patch near a sharp feature can be thought of as outliers for the normals of the other smooth patches. Hence, methods from robust statistics can be applied to recover accurate normals.

The Voronoi diagram based normal estimation uses the furthest away vertices of the Voronoi cell of a sample point to get an estimate of the normal [Amenta and Bern, 1998; Dey and Sun., 2005]. Since the Voronoi cells get slimmer as the sampling rate increases, this estimation gives more accurate results for densely sampled point sets. Unlike the fitting based method, this method cannot handle noise well. In addition, outliers and sparse sampling also degrades the accuracy of the estimates significantly. An interesting approach for normal estimation is combining the two approaches of using PCA and Voronoi diagrams [Alliez et al., 2007]. For this method, the normal is estimated by performing a PCA on the shape of the Voronoi cell of a sample point. For robustness, the region is grown to include the Voronoi cells of the neighboring points. However, constructing the Voronoi diagram remains a burden and sharp features and details cannot be accurately reconstructed.

Although the mentioned methods can be used to robustly estimate the

## *Background and Related Work*

direction of the sample normals  $\mathbf{n}_i$ , there remains the ambiguity of the orientation, i.e. whether to use  $\mathbf{n}_i$  or  $-\mathbf{n}_i$  for each sample normal. This sign ambiguity cannot be solved by considering local information since the solution requires the knowledge of inside/outside regions of the surface. A common approach encodes consistencies of neighboring normals in a weighted graph and tries to maximize the total weight [Hoppe et al., 1992; Huang et al., 2009]. This approach can fail if the measure of consistency and the distribution of normals in the neighborhoods violate the assumed conditions. Other methods try to estimate a consistent orientation by determining outside/inside Delaunay tetrahedra via spectral partitioning [Kolluri et al., 2004], or by using global harmonic functions [Seversky et al., 2011].

## 2.2 Manifold Geometry Sampling

Ensuring accurate reconstructions requires to have sufficiently dense samplings of the surface. On the other hand, redundant sampling makes the methods suffer from infeasible time and memory complexity. Hence, an important problem is finding optimal sampling conditions for a given reconstruction technique. Although this is a classical problem in signal processing, extensions of the results to manifolds have been challenging. Two main approaches for deriving optimal sampling conditions reflect the two different ways of reconstructing surfaces. The first approach assumes that a combinatorial algorithm such as the ones explained in Section 2.1.1 is used, and derives sampling conditions based on the underlying structures. In the second category, we describe algorithms that do not depend on the properties of such combinatorial structures. These algorithms utilize a scattered data approximation technique to assess the quality of the samplings. We also put methods that do not assume a particular reconstruction method and propose generation of a sampling pattern instead, and some remeshing algorithms under this category.

### 2.2.1 Sampling for Combinatorial Reconstruction

For some of the reconstruction methods that depend on a combinatorial structure such as the Voronoi diagram, there exists sampling conditions that ensure a given geometric and topological accuracy [Amenta and Bern, 1998]. The sampling conditions are shown to be related to the distance of points on the surface to their closest points on the medial axis. However, since the medial axis of a surface is hard to obtain, the conditions are difficult to check or be used in algorithms for sampling. Although looser conditions have been developed [Boissonnat and Oudot, 2005] for efficient checking and construction of optimal samplings, the resulting algorithms inherit the disadvantages of combinatorial reconstructions, being time and memory intensive, and not robust to noise and outliers.

### 2.2.2 Sampling for Surface Approximation

Real datasets can be massive, with billions of points, and contain noise and outliers. Hence, efficient and accurate simplification of data is essential to process large datasets. The fastest methods for simplification are based on clustering of points [Pauly et al., 2002]. They are simple to implement and out-of-core, thus suitable for very large datasets. However, sampling quality

## *Background and Related Work*

is not sufficient for accurate reconstructions. An efficient and still accurate set of sampling algorithms simplify the point set by iteratively removing or adding a sample at a time based on a measure derived from the geometry and sampling rate. This measure can be defined as the distance of a point to the surface [Alexa et al., 2001; Carr et al., 2001], density and curvature based heuristics [Ohtake et al., 2004; Kitago and Gopi, 2006], or quadratic error metrics [Garland and Heckbert, 1997; Pauly et al., 2002]. The algorithms provide better accuracy for the final surfaces, but computing the initial scores and updating can become prohibitively expensive [Kitago and Gopi, 2006].

After simplifying the data, further resampling can be applied to improve quality. A common approach to generate high quality isotropic samplings of a surface is distributing points using relaxation techniques. These methods first compute an initial distribution of points on the surface, and then refine this distribution by techniques such as variants of the Lloyd’s method [Alliez et al., 2003; Valette et al., 2008; Yan et al., 2009], particle systems [Turk, 1992; Witkin and Heckbert, 1994; Pauly et al., 2002], or advancing front algorithms [Schreiner et al., 2006]. In spite of their good sampling properties, computational cost or critical dependence on parameters or initial distributions hinder their use. Instead of operating on the surface, some relaxation algorithms first parametrize the surface and then use well established methods to generate distributions with blue noise characteristics on the parameter domain [Alliez et al., 2002]. However, parametrization is a hard problem that may introduce distortions. This has led to algorithms that directly compute distributions with blue noise properties on a meshed surface using geodesic distances [Fu and Zhou, 2008], albeit at a high computational cost. For sharp feature preservation, these methods can be augmented with specialized projection operators [Huang et al., 2012].

Another class of resampling algorithms greedily places new samples so as to satisfy some constraints. Farthest point sampling [Boissonnat and Oudot, 2003] places a new sample at a time such that it is maximally away from all other inserted samples. For meshes, the algorithm has been extended using geodesic distances to improve both sampling quality and running time [Peyré and Cohen, 2006]. However, these greedy algorithms generally perform worse than relaxation based methods while computational cost remains high.

We have developed efficient and out-of-core simplification and resampling algorithms [Öztireli et al., 2010]. In Chapter 4, we will present in detail that the new algorithms significantly outperform the others in terms of reconstruction accuracy and sampling characteristics, benefiting from their theoretical

foundation and relation to our reconstruction method [Öztireli et al., 2009] (Chapter 3).

### 2.2.3 Sampling in Manifold Learning

Apart from the focus on 2-manifolds in computer graphics, sampling problem has also been tackled for general higher dimensional manifolds and ambient spaces. The term manifold learning refers to discovering the intrinsic manifold structure of a given point set [Coifman and Lafon, 2006]. Many learning techniques first construct a weighted graph from the points and form the graph Laplacian or a related matrix. Eigendecomposition of this matrix can then be used to estimate embeddings of the data points into a new space such that various learning tasks can be done easily [Ham et al., 2004]. Since these operations are computationally very demanding, several methods for sampling datasets have been proposed, but so far they mostly depend on random sampling [Karoui and d’Aspremont, 2009; Drineas and Mahoney, 2005; Achlioptas et al., 2001]. Notable exceptions try to exploit the low-rank structure of the diffusion matrix [Coifman and Lafon, 2006] or find subsets of the samples and use the Nyström method to approximate the eigenvectors [Liu et al., 2006]. However, the algorithms are computationally expensive and not suitable for high quality samplings of manifolds.

Our sampling techniques [Öztireli et al., 2010] are inspired by the recent developments in this field. We have introduced a new space implied by heat diffusion on manifolds, and showed how it can be used to define a measure that quantifies the change in the manifold definition due to a single point (see Chapter 4).



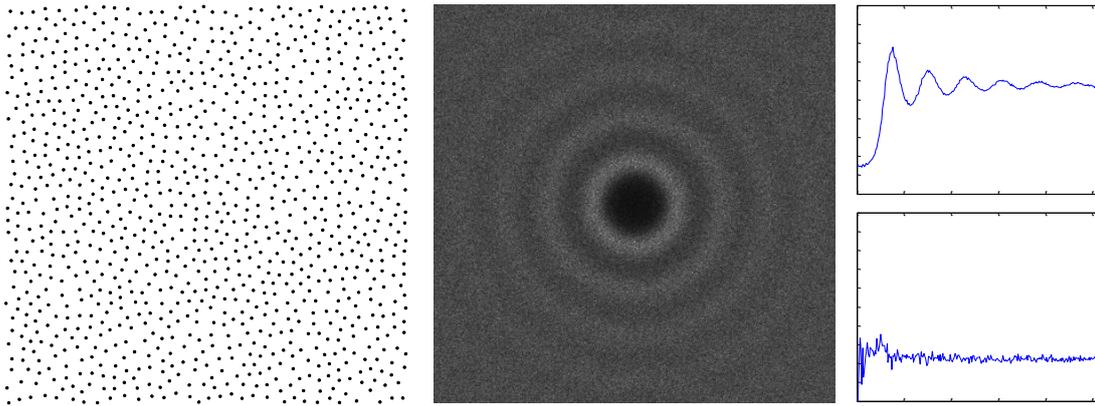
**Figure 2.2:** *Patterns are abundant in nature. Left: galaxies, middle: a sunflower, right: crystallines in a 2D colloid, (Left figure is courtesy of NASA, middle is courtesy of L. Shyamal, right is courtesy of Richard Wheeler.)*

## 2.3 Stochastic Point Patterns

Point patterns are abundant in the real world, and constitute probably the most common form of geometric complexity along with manifolds. Indeed, many different point distributions are found in nature ranging from locations of stars and galaxies to movements of people in a crowd (Figure 2.2). The ability to analyze and synthesize these structures is central for many applications. Statisticians and physicists have been dealing with general point distributions for decades. The developed methods have been collected under the name *point processes*. However, only a few of these developments have been utilized in computer graphics, where the methods focus on qualitative evaluations and reconstruction of point distributions with *blue noise* characteristics. In contrast to all other approaches, we have proposed a method that can reconstruct distributions with general characteristics by utilizing novel analysis and synthesis techniques [Öztireli and Gross, 2012]. We mention its characteristics in comparison to other methods below (see Chapter 5 for the detailed discussion).

### 2.3.1 Blue Noise Patterns

One recurrent pattern in nature, that is also extensively studied in computer graphics, is a blue noise point pattern. Blue noise is a loosely used term in computer graphics to name a family of distributions, which have a minimum distance between pairs of points, and do not exhibit regular structures. A typical blue noise distribution is shown in Figure 2.3 (a). Point patterns with blue noise characteristics have been introduced into computer graphics as a means for anti-aliasing [Dippé and Wold, 1985; Cook, 1986; Mitchell,



**Figure 2.3:** *Left: A typical blue noise distribution, middle: its mean periodogram, and right: the corresponding power and anisotropy plots.*

1987]. The anti-aliasing properties of such distributions result from their unique ability to preserve low-frequency details and map high-frequency aliasing patterns (such as jagged edges) to noise, which is much less disturbing to the human eye. They are also used in many contexts such as image sampling [Cook, 1986], geometry processing and synthesis [Alliez et al., 2002; Ma et al., 2011], object placement [Deussen et al., 1998; Lagae and Dutré, 2006; Wei, 2010], or procedural noise generation [Lewis, 1989]. Blue noise distributions are abundant in nature and even in the human visual system [Yellott, 1983].

The name blue noise stems from the shape of the mean *periodogram* [Ulichney, 1987] with a low energy region in the middle and an almost constant magnitude energy away from this region (Figure 2.3 (b)). The periodogram is simply computed by considering each point as an impulse and taking the Fourier transform of the resulting sum of impulses at the point locations. The radial average (power diagram) and anisotropy plots can be obtained by building a histogram with bins of concentric shells around the origin (Figure 2.3 (c)). It is desirable that the periodogram is radially symmetric with low anisotropy for high quality blue noise distributions.

### 2.3.2 Point Distributions in Computer Graphics

Works on point distributions in computer graphics mostly focused on several implementations of blue noise distribution generation algorithms, and qualitative analysis of blue noise patterns.

Spectral measures based on periodograms, along with simple scalar measures such as the packing density, have been the prevalent tools for analysis of point

## Background and Related Work

distributions in graphics [Ulichney, 1987; Lagae and Dutré, 2008]. Recently, diagrams with similar analysis power have been proposed for distributions in general non-Euclidean domains by considering the distribution of difference vectors between points [Wei and Wang, 2011]. We have proposed a new analysis that can quantitatively explain and relate distributions by mapping them into a space constructed by using distances between points, where the distance metric can belong to an arbitrary metric space [Öztireli and Gross, 2012].

Synthesis algorithms in graphics are mostly focused on distributions with blue-noise characteristics. The earliest thread of methods to generate blue-noise distributions is based on randomly generating points in space and accepting or rejecting based on a criterion. This algorithm is known as *simple sequential inhibition* in statistics, *random sequential adsorption* in physics [Illian et al., 2008], and dart throwing in graphics. The initial dart throwing algorithm [Cook, 1986] has been accelerated using a hierarchy of allowable radiuses [McCool and Eugene, 1992], specialized data structures [Dunbar and Humphreys, 2006], parallelization [Wei, 2008], adaptive trees [Jones, 2006; White et al., 2007; Gamito and Maddock, 2009], or explicit void region representations [Ebeida et al., 2011]. It has also been extended to multiple classes of objects [Wei, 2010] such that distributions of points in different classes as well as that of all points have blue-noise characteristics. We have extended the standard and multi-class dart throwing algorithms such that general distributions with arbitrary characteristics can be handled [Öztireli and Gross, 2012].

In order to increase density and regularity, relaxation methods where points are iteratively moved so as to optimize an energy function and satisfy certain constraints are used. Lloyd's algorithm [Lloyd, 1982] minimizes the quantization error and thus can be utilized to obtain optimum placement of points that capture the whole space well [McCool and Eugene, 1992]. To avoid the optimum involving regular structures, an equal area constraint for the Voronoi regions [Balzer et al., 2009], or injection of randomness into the relaxation algorithms [Schmaltz et al., 2010; Fattal, 2011] can be used. Each of these methods generate distributions with certain blue-noise characteristics dictated by the construction of the algorithms. In contrast, our gradient descent based relaxation algorithm can be used to generate distributions with general characteristics directly controllable via specifying example distributions or statistics [Öztireli and Gross, 2012].

As a fast alternative to the mentioned approaches, tiling based methods have been introduced [Ostromoukhov et al., 2004; Kopf et al., 2006]. These

methods can generate distributions in real time. The characteristics of the distributions depend on the initial distributions used on the tiles, which should be generated by the previously mentioned algorithms.

### 2.3.3 General Point Patterns with Point Processes

Blue noise distributions constitute only a small part of the set of point patterns. In fact, the diversity of distributions in nature is staggering, with complex structures at various levels. Exploring patterns in point distributions is thus very important in a diverse selection of disciplines such as physics, chemistry, sociology, geology, and astronomy. The analysis of point patterns has been attracting attention of statisticians and physicists for a long time. The underlying model of point patterns are called *point processes* [Illian et al., 2008]. A point process can be considered as the generating process of an observed pattern. Conversely, a point distribution can be seen as a particular instance generated by a point process.

The point process statistics are mainly concerned about correlations of point locations and marks of points in space. The correlations of point locations result from interactions among the points that can happen in many ways including repulsion, attraction, inhibition of nearby points, or generation of child points. Depending on these interactions, the point processes can generate *hard-core* or *clustering* point patterns, or a combination of the two at different levels. In hard-core distributions, there is a minimum distance between each pair of points. Hence, blue noise distributions constitute a subset of these patterns. In contrast, clustering distributions are characterized by lumps of points in random parts of the space with varying sizes and number of points.

### 2.3.4 Point Process Statistics

The statistical measures for point processes range from intensity of the points to pairwise or higher order correlations and topological characteristics such as statistics related to the Voronoi regions. These measures can be used to describe and discover patterns, as well as to infer the underlying process that generates the observed patterns. At long distances, the correlations diminish and hence short range correlations are more important to describe distributions. Among the measures, the *pair correlation function* (PCF), which measures the probability of having a pair of points at certain locations in space, has been widely accepted as the most informative [Illian et al., 2008].

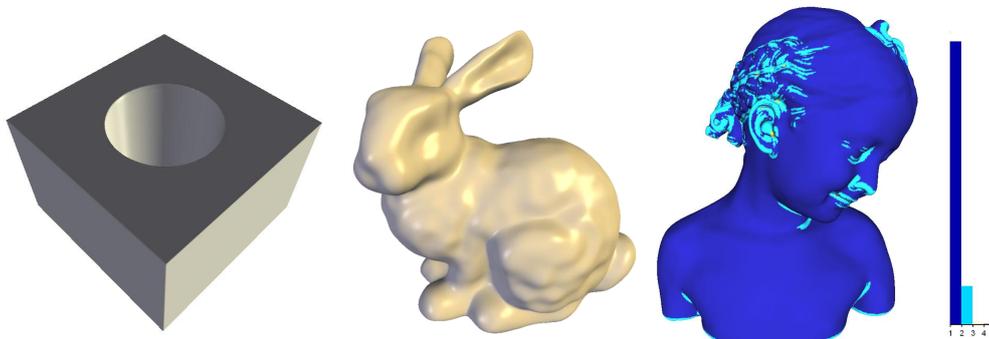
Indeed, we have showed that the PCF is sufficient to describe the characteristics of distributions with diverse properties [Öztireli and Gross, 2012]. This statistical measure was introduced in the physics literature in the beginning of the 20<sup>th</sup> century and is also known as the *radial distribution function* [Torquato, 2002]. It reduces to a function of the distances between point pairs for stationary and isotropic point processes. Hence, for these processes, it is a global measure that quantifies the distribution of distances in a point distribution. In spite of its simple interpretation, it surprisingly contains almost all information about the underlying process, and hence can be used to develop a unified analysis of patterns, as will be detailed in Chapter 5.

### 2.3.5 Synthesis of Point Patterns

The results of an analysis can also be used to synthesize new patterns following the underlying point process. A typical approach for synthesis is to assume a particular model for the underlying process, infer the parameters of the model using the data, and generate distributions from the fitted model. The models used can be generally categorized as *hard-core* or *clustering* processes depending on the repulsion or attraction between the points. Examples of some models are Gibbs, Cox, Matérn hard-core and clustering, and Neyman-Scott processes, random sequential adsorption (dart throwing), and force based algorithms [Jodrey and Tory, 1985]. Once the model parameters are estimated, variants of Markov Chain Monte Carlo methods can be used to generate points from the models.

As with all model based approaches, synthesis via models can only generate a limited subset of the diversity of patterns in nature. Instead of relying on estimating parameters of a model, it is possible to randomly explore the configuration space of all point locations and marks by adding or removing a point at a time so as to satisfy a given condition [Torquato, 2002]. However, this direct approach is not feasible and scalable for many cases. Our algorithms [Öztireli and Gross, 2012] do not rely on a model and still produce precise reconstruction of characteristics for general metric spaces and multiple classes of objects efficiently.

## Meshless Reconstruction of Manifolds



Given a point cloud sampled from the surface of an object, the goal of reconstruction is recovering a representation for the underlying surface. Since data is often contaminated with noise and outliers, or insufficiently sampled, it is challenging to obtain a reconstruction that captures the details and important features of the underlying surface well. Reconstruction is an inherently ill-posed problem unless certain assumptions on the surface to be reconstructed are made, and the associated sampling and noise conditions are met. To make the problem well-posed, often regularizations based on the smoothness assumption are employed. This allows the algorithms to infer a smooth manifold surface from the data with a convenient mathematical representation.

However, the smoothness assumption also leads to losing sharp features and other high frequency details.

In this chapter, we present a sharp feature and detail preserving surface reconstruction method. The method extends *moving least squares* (MLS) based surface reconstruction techniques. It can be used to reconstruct smooth surfaces, while preserving sharp features including edges, corners, and peaks, and small scale details. The sharpness of the reconstructed features can be controlled with a single intuitive parameter. We show that stable and efficient reconstructions under the influence of noise/outliers and sparse sampling is possible with the new definition. The fundamental observation leading to the new surface definition is that surface normals near a sharp feature deviate significantly from each other, if they are on different smooth surface patches meeting at the sharp feature. Hence, methods from statistics that deal with outliers can be employed to recover the features. We take special care to keep the resulting surface representation continuous, and its computation simple. This requires treating all features in a unified framework, and integrating the extension directly into existing MLS based mathematical definitions.

After summarizing relevant concepts from differential geometry of manifolds (Section 3.1), we will review MLS based local surface approximation methods and show their equivalence to *local kernel regression* (LKR) used in statistics for scattered data approximation (Section 3.2). This link opens up the way to borrow various concepts from the statistics literature for improving MLS surfaces. We describe how to integrate the fruitful concept of *robust statistics* into the LKR procedure (Section 3.3), and how the link between MLS surfaces and LKR can be used to derive a feature preserving surface definition (Section 3.4). This is followed by an analysis of the reconstructions produced and performance of the algorithms under various conditions (Section 3.5). In addition, we illustrate the use of local approximations in a different setting for an application that imposes quite different constraints (Section 3.6).

### 3.1 Differential Geometry of Manifolds

Our main object of analysis will be the concept of a *manifold* since we assume that the sample points represent a manifold surface. Here, we state the relevant definitions and theorems of the differential geometry of manifolds, and in particular manifolds represented by implicit functions.<sup>1</sup>

#### 3.1.1 Submanifolds

The concept of a (*sub*)*manifold* is intuitively defined as a space that is locally equivalent to an Euclidean space of dimension  $d$ ,  $\mathbb{R}^d$ . Many types of real world data that are represented by high dimensional vectors are actually generated by processes with much less degrees of freedom. The concept of a manifold naturally captures the intrinsic degrees of freedom. Hence, manifolds are very important constructs to investigate complex data. Perhaps the most familiar examples of manifolds are surfaces bounding objects. Although scanning these surfaces results in points living in  $\mathbb{R}^3$ , the actual degrees of freedom is 2. This can be seen by considering a local patch on the surface. As we zoom in more, the local patch will increasingly approach  $\mathbb{R}^2$ . This local space at a given point  $x$  is called the *tangent space* of the manifold at  $x$ .

**Implicit Definition.** We will mostly work with *implicit* definitions of manifolds. This means a manifold  $M$  embedded in  $\mathbb{R}^n$  is given by the zero (or level) set of a function  $f$  defined in some subspace of  $\mathbb{R}^n$ . Formally, for an open set  $W \subset \mathbb{R}^n$  and a map  $f : W \rightarrow \mathbb{R}^k$  that is a  $C^\infty$  diffeomorphism with  $n \geq k$ , a point  $\mathbf{x} \in \mathbb{R}^n$  of this map is called a regular point if  $Df_{\mathbf{x}}$  is surjective, and a value  $\mathbf{v} \in \mathbb{R}^k$  is called a regular value if all points  $\mathbf{x} \in f^{-1}\{\mathbf{v}\}$  are regular points. For any regular value  $\mathbf{v} \in \mathbb{R}^k$  of the map  $f$ , the set of points  $f^{-1}\{\mathbf{v}\}$  defines a submanifold embedded in  $\mathbb{R}^n$  with intrinsic dimension  $n - k$ .

For surfaces of objects in the real world,  $n = 3$ ,  $k = 1$ , and thus the dimension of the manifold surface is 2. This can also be understood by observing that the tangent spaces are 2 dimensional planes. A curve embedded in the same 3 dimensional space, on the other hand, is a 1 dimensional manifold.

#### 3.1.2 Implicit Surfaces

**Definition.** We will mostly work on surfaces bounding objects, that is, two dimensional manifolds embedded in  $\mathbb{R}^3$  with  $n = 3$  and  $k = 1$ . For this case,

---

<sup>1</sup>For an extensive treatment of the differential geometry of manifolds, we refer the reader to Do Carmo's excellent book [Carmo, 1976].

the differential is simply given by the gradient  $\nabla f$ . Since this differential is surjective, it is true that  $\nabla f(\mathbf{x}) \neq \mathbf{0} \ \forall \mathbf{x} \in f^{-1}\{v\}$  for a regular value  $v$ . Each regular value of  $f$  defines a different isocontour and a different manifold. We will assume that the manifold we would like to reconstruct corresponds to the regular value of 0. Hence, the implicit surface is given by the following formula:

$$S = \{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0\}, \quad (3.1)$$

where  $\nabla f(\mathbf{x}) \neq \mathbf{0}$  for  $\mathbf{x} \in S$ .

**Differentials for Implicit Surfaces.** The most commonly used differentials are first order and are called the gradient and normal. These define tangent planes, and are very important for almost all reconstruction algorithms. The normal of an implicit surface is given by the simple formula:

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}, \quad \|\nabla f(\mathbf{x})\| \neq 0. \quad (3.2)$$

The implicit equation of the tangent plane at the point  $\mathbf{x}$  is then given by  $f_{T_{\mathbf{x}}}(\mathbf{y}) = \mathbf{n}(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$ .

Second order derivatives of the implicit function can be gathered into the Hessian matrix:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix} \quad (3.3)$$

that can be used to compute how curved the manifold is at a point.

**Signed Distance Functions.** A signed distance function gives the signed Euclidean distance to the surface from each point in space. Since we assume closed manifold surfaces, the space  $\mathbb{R}^3$  is divided into two regions: inside and outside of the surface. We use the convention that the distance is positive outside, and negative inside. Then, the signed distance function for a surface  $S$  is defined as:

$$f(\mathbf{x}) = \text{sign}(\mathbf{x}) \inf_{\mathbf{s} \in S} \|\mathbf{x} - \mathbf{s}\| \quad (3.4)$$

### 3.1 Differential Geometry of Manifolds

with  $\text{sign}(\mathbf{x}) = 1$  when  $\mathbf{x}$  is outside, and  $-1$  otherwise. A fundamental property of the gradient of this function is that it satisfies the Eikonal equation:

$$\|\nabla f(\mathbf{x})\| = 1 \tag{3.5}$$

except when  $\mathbf{x}$  is on the medial axis of the surface. Since such points have more than one closest point on the surface, this gradient is not well defined.

Note that an implicit function defining a surface is not in general a signed distance function. Due to the stable gradient magnitude, it is desirable to keep the reconstructed implicit function close to the signed distance function.

## 3.2 Reconstruction by Local Approximations

Our goal is reconstructing an implicit function defining a closed manifold from the given points sampled on the surface. A versatile, robust, and efficient method of achieving this is using local approximations. In this section, we review a widely used local approximation method for implicit surface reconstruction, and show that it is equivalent to a similar local method used in statistics. This allows us to interpret existing surface definitions in a new statistical framework.

### 3.2.1 Moving Least Squares

Moving Least Squares (MLS) [Shepard, 1968] is a widely used method to approximate a function  $f$ , from its scattered point samples  $\{\mathbf{x}_i, y_i\}$ , where  $y_i$  is the known value of the function at  $\mathbf{x}_i$ . The main idea of this method is fitting a proxy function to the local neighborhood of an evaluation point  $\mathbf{x}$ , to approximate  $f(\mathbf{x})$ . Specifically, it solves the following minimization problem for a given point  $\mathbf{x}$ :

$$f_{MLS}(\mathbf{x}) = \min_f \sum_i (f(\mathbf{x}_i) - y_i)^2 \phi(\|\mathbf{x} - \mathbf{x}_i\|). \quad (3.6)$$

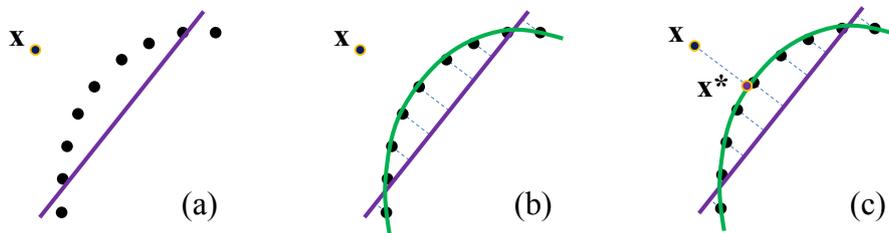
Locality of this system is established by a decaying weight function  $\phi$ , which gives more weight to the points near the evaluation point  $\mathbf{x}$ . Since the weights depend on  $\mathbf{x}$ , for each different  $\mathbf{x}$ , a different least squares system needs to be solved, hence the name *moving least squares*.

**Moving least squares surfaces.** The MLS method was later adapted for the surface reconstruction problem [Alexa et al., 2001]. Since a manifold is locally well represented by a plane, a natural choice is fitting a local plane by solving a weighted least squares problem similar to equation 3.6 (Figure 3.1 (a)), with  $f$  replaced by the implicit equation of a plane. This plane can then be used as a parameter domain to further fit a bivariate function to the local patch by again using MLS (Figure 3.1 (b)). Finally, the point  $\mathbf{x}$  is projected onto this fitted local patch [Alexa et al., 2001] (Figure 3.1 (c)). This idea was later extended such that distance to the fitted plane is defined as the distance to the surface [Amenta and Kil, 2004a]. One advantage of this extension is that an implicit function of the surface is obtained, which can also be used for projecting a point if needed. Since the plane fitting can get unstable in high curvature areas, fitting higher order proxy surfaces was also proposed [Guennebaud and Gross, 2007].

### 3.2 Reconstruction by Local Approximations

One fundamental problem when fitting a local proxy surface using MLS is that the known values of the function,  $y_i$ , are assumed to be all zero, since we assume that the sample points are close to the surface. This results in the following minimization:  $\min_f \sum_i (f(\mathbf{x}_i))^2 \phi(\|\mathbf{x} - \mathbf{x}_i\|)$ . To avoid the trivial solution of  $f(\mathbf{x}) = 0$ , some constraints on the function  $f$  are needed. A commonly used constraint is requiring  $\|\nabla f\| = 1$ . This is motivated by the fact that an implicit function with normalized gradient is also a signed distance function for the represented surface. It has been shown that this constraint works well for fitting local planes and spheres [Amenta and Kil, 2004a; Guennebaud and Gross, 2007]. However, a linear system needs to be solved for each evaluation, slowing down the methods and complicating the expressions. Another way to avoid the trivial solution is computing the normals first, and using them to constrain the gradient of the fitted proxy surface (normal constraints) [Kolluri, 2005; Guennebaud et al., 2008]. This results in simpler and efficiently computable expressions for the implicit function, and hence preferred if the normals have already been estimated.

Respecting these constraints, the MLS based surface reconstruction methods solve the weighted least squares problem  $\min_f \sum_i (f(\mathbf{x}_i))^2 \phi(\|\mathbf{x} - \mathbf{x}_i\|)$  to get the value of the implicit function  $f$  at the point  $\mathbf{x}$ . Approximations of this kind have also been used in statistics. In the next section, we explain one such method that has direct connections to MLS.



**Figure 3.1:** Example MLS projection: (a) A least squares plane is fit. (b) The fitted plane is used as a reference domain to fit a polynomial. (c) The point  $\mathbf{x}$  is projected onto this polynomial.

#### 3.2.2 Local Kernel Regression

Similar to MLS, local kernel regression (LKR) is a method to approximate a function  $f(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$  given its values  $y_i \in \mathbb{R}$  at the sample points  $\mathbf{x}_i \in \mathbb{R}^d$ , by local fits. The input data might be corrupted with noise such that  $y_i = f(\mathbf{x}_i) + \epsilon$ , where  $\epsilon$  is a random variable with zero mean.

The essence of the method is to approximate the unknown function  $f(\mathbf{x}_i)$  around the evaluation point  $\mathbf{x}$  in terms of a Taylor expansion:

$$f(\mathbf{x}_i) \approx f(\mathbf{x}) + (\mathbf{x}_i - \mathbf{x})^T \nabla f(\mathbf{x}) + \frac{1}{2} (\mathbf{x}_i - \mathbf{x})^T \mathbf{H} f(\mathbf{x}) (\mathbf{x}_i - \mathbf{x}) + \dots,$$

where  $\mathbf{H} f(\mathbf{x})$  denotes the Hessian matrix of  $f(\mathbf{x})$ . The order  $o$  of the expansion is defined as the number of terms used in the Taylor expansion minus one. This equation can be reformulated as a sum of inner products:

$$f(\mathbf{x}_i) \approx s_0 + \mathbf{a}_i^T \mathbf{s}_1 + \mathbf{b}_i^T \mathbf{s}_2 + \dots, \quad (3.7)$$

where  $\mathbf{a}_i = (\mathbf{x}_i - \mathbf{x})$ , and  $\mathbf{b}_i = [\dots (\mathbf{a}_i)_j (\mathbf{a}_i)_k \dots]^T$  with  $k \geq j$ . The local approximation is by definition more accurate nearby the point  $\mathbf{x}$ . This suggests the use of a weighted least squares minimization to find the unknown parameters  $\mathbf{s} = [s_0 \ \mathbf{s}_1^T \ \mathbf{s}_2^T \ \dots]^T$ :

$$\arg \min_{\mathbf{s}} \sum (y_i - (s_0 + \mathbf{a}_i^T \mathbf{s}_1 + \mathbf{b}_i^T \mathbf{s}_2 + \dots))^2 \phi_i(\mathbf{x}), \quad (3.8)$$

where  $\phi_i(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_i\|)$ , and  $\phi(x)$  is a symmetric and decreasing weight function giving more weight to samples near  $\mathbf{x}$  in the minimization.

By replacing the function  $f$  in equation 3.6 by its Taylor expansion terms, equation 3.8 can be exactly obtained. Hence, the MLS and LKR methods are the same if a local Taylor approximation of the fitted function is used. As explained in Section 3.2.1, MLS based surface reconstruction methods fit a plane, sphere, or higher order local proxy surfaces. These surfaces correspond to lower order terms of the Taylor expansion of the implicit function. Hence, these surface definitions can be derived in the framework of LKR with the proper constraints. We show derivations of some MLS based surfaces in the next section.

### 3.2.3 Derivation of MLS Surfaces from LKR

As explained in Section 3.2.1, the problem of implicit surface reconstruction from point clouds consists of approximating the signed distance function to a surface given points sampled from the surface. To approximate the signed distance function using LKR, we assume that the sampled points are close

### 3.2 Reconstruction by Local Approximations

to the surface so that  $f(\mathbf{x}_i) \approx 0$  which implies to take  $y_i = 0$ . Similar to the MLS case, in order to avoid the trivial solution  $\mathbf{s} = 0$ , one can constrain the norm of the gradient to be one:  $\|\nabla f(\mathbf{x})\| = \|\mathbf{s}_1\| = 1$ , or make the gradient of the implicit function approximate the prescribed normals, if available. Different MLS based surfaces can be obtained by changing the degree of local approximations  $o$ , and the constraints used. By incorporating these two factors, we derive some of the proposed implicit MLS based surface definitions.

**Deriving Kolluri's [2005] MLS definition.** Given the Taylor expansion of the gradient:

$$\nabla f(\mathbf{x}_i) \approx \nabla f(\mathbf{x}) + \mathbf{H}f(\mathbf{x})(\mathbf{x}_i - \mathbf{x}) + \dots, \quad (3.9)$$

taking a zero order approximation results in  $\nabla f(\mathbf{x}_i) \approx \nabla f(\mathbf{x})$ . Since we assume the function  $f$  approximates a signed distance field, we can set the normal constraint  $\nabla f(\mathbf{x}_i) = \mathbf{n}_i$ , where  $\mathbf{n}_i$  is the normal at the sample point  $\mathbf{x}_i$ . In this case, the first order LKR minimization degenerates to a zero order LKR:

$$\begin{aligned} & \arg \min_{s_0, \mathbf{s}_1} \sum (y_i - (s_0 + \mathbf{a}_i^T \mathbf{s}_1))^2 \phi_i(\mathbf{x}) \\ &= \arg \min_{s_0} \sum (s_0 + (\mathbf{x}_i - \mathbf{x})^T \mathbf{n}_i)^2 \phi_i(\mathbf{x}) . \end{aligned}$$

Solving this minimization yields the following explicit formula for  $f(\mathbf{x})$ :

$$f(\mathbf{x}) = s_0 = \frac{\sum \mathbf{n}_i^T (\mathbf{x} - \mathbf{x}_i) \phi_i(\mathbf{x})}{\sum \phi_i(\mathbf{x})}, \quad (3.10)$$

which is exactly Kolluri's [2005] definition, which we call IMLS.

**Deriving Adamson and Alexa's [2003] MLS definition.** For deriving this surface definition, the gradient constraint  $\|\nabla f(\mathbf{x})\| = \|\mathbf{s}_1\| = 1$ , and  $o = 1$  are assumed. Under these assumptions, the LKR minimization becomes:

$$\min_{s_0, \mathbf{s}_1} \sum (y_i - (s_0 + \mathbf{a}_i^T \mathbf{s}_1))^2 \phi_i(\mathbf{x}) = \min_{s_0, \mathbf{s}_1} \sum (s_0 + \mathbf{a}_i^T \mathbf{s}_1)^2 \phi_i(\mathbf{x}) \quad (3.11)$$

with the constraint  $\|\mathbf{s}_1\| = 1$ .

This constrained minimization has the following solution:

$$(\mathbf{C} - \mathbf{c}\mathbf{c}^T)\mathbf{s}_1 = 0, \quad \|\mathbf{s}_1\| = 1$$

$$s_0 = \mathbf{s}_1^T(\mathbf{x} - \mathbf{c}) \quad (3.12)$$

with  $\mathbf{c} = \frac{\sum \mathbf{x}_i \phi_i(\mathbf{x})}{\sum \phi_i(\mathbf{x})}$ , and  $\mathbf{C} = \frac{\sum \mathbf{x}_i \mathbf{x}_i^T \phi_i(\mathbf{x})}{\sum \phi_i(\mathbf{x})}$ . Hence,  $\mathbf{s}_1$  can be approximated by the eigenvector of the matrix  $(\mathbf{C} - \mathbf{c}\mathbf{c}^T)$  with the smallest eigenvalue. Here,  $s_0$  corresponds to the MLS based surface of Adamson and Alexa [2003].

Note that  $\mathbf{s}_1$  is also the normal of the plane fitted using the following least squares error:

$$\min_{\mathbf{n}, d} \sum (\mathbf{n}^T \mathbf{x}_i - d)^2 \phi_i(\mathbf{x}) \quad (3.13)$$

with the constraint that  $\mathbf{n}$  has unit length. Fitting a plane this way has been used in other works as well [Pauly et al., 2003].

**Deriving Higher Order Surfaces.** In general, higher order implicit surface fitting methods [Guennebaud and Gross, 2007; Guennebaud et al., 2008] can also be interpreted in this framework by just adding more terms to the Taylor expansion and using specific constraints. As an example, the sphere fitting can be realized by using second order terms, and constraining the Hessian to be identity:

$$f(\mathbf{x}_i) \approx f(\mathbf{x}) + (\mathbf{x}_i - \mathbf{x})^T \nabla f(\mathbf{x}) + \frac{1}{2} (\mathbf{x}_i - \mathbf{x})^T \mathbf{I} (\mathbf{x}_i - \mathbf{x}) + \dots,$$

where  $\mathbf{I}$  is the identity matrix.

### 3.3 Robust Local Approximations for Feature Preservation

The formulation of the MLS surfaces using LKR allows us to incorporate techniques from the regression literature to improve the surface definitions. In this section, we focus on a particular technique, *robust statistics*, and show how it can be integrated into LKR without disturbing the continuity and simplicity of the formulation. Below we explain the relevant concepts from robust statistics for our purposes (please refer to [Huber, 2004] for an extensive treatment), and then show its application to LKR.

#### 3.3.1 Robust Statistics

In many fields, the word *robust* is used to indicate various properties of approximations, and has qualitative and imprecise meanings in most cases. This, however, is not true for statistics. Robust statistics deals specifically with *outliers* in the data, and a method or measure that is robust in this sense will be resilient to unexpected data that does not follow the assumed model. This is a fundamental deviation from classical statistical measures, which can be affected by a single off-model data. In fact, classical methods are affected more, as outliers become more distant from the majority of data points. This is due to the inherent assumption that the data follows a particular model. However, real world data is almost always contaminated with outliers, making classical measures significantly inaccurate.

Instead of attempting to remove outliers prior to computing statistical measures, robust statistics solves this problem by proposing measures and methods that work under the influence of outliers. A well-known example of a robust measure is the *median*. Using the classical counterpart, the mean, a single wrong measurement that is very far away from the others results in a completely different estimate, such as the mean of the numbers  $\text{mean}(0, 1, 899) = 300$ . With the same data, the median is 1.

**Breakdown Point.** The example with the mean and median shows that the mean is affected even by one outlier, while the median can tolerate outliers even if they approach 50% of the data. The amount of outliers that a statistical measure can tolerate before giving arbitrary results is called the breakdown point. The breakdown point of the mean is 0%, while that of the median is 50%. Note that 50% is the maximum breakdown point that can be achieved, since if the outliers constitute more than half of the data, that is, if they are the majority, they can no longer be regarded as outliers by any method.

**Robust Regression Techniques.** Standard local kernel regression, as

presented in section 3.2.2, assumes the data follow a smooth model with uniform noise. As a consequence, even a single outlier in the data can significantly influence the solution. Robust statistics can be used to alleviate this problem. Some popular robust regression methods are *least median of squares*, *least trimmed squares*, and *repeated median* (e.g. [Morgenthaler, 2007]). However, since the objective functions of these methods are not differentiable, they likely generate discontinuous solutions (i.e.,  $C^{-1}$ ) and require expensive minimizations. Instead, we use the  $\psi$ -type M-Estimators to make LKR robust to outliers while keeping the continuity and analytic expressibility of the solution, as explained in the next section.

### 3.3.2 Robust Local Kernel Regression

We extend LKR to *robust local kernel regression* (RLKR) using  $\psi$ -type M-Estimators, which have the significant advantage of leading to simple and efficient minimization procedures [Huber, 2004]. Instead of the ordinary least squares criterion,  $\psi$ -type M-Estimators minimize a different but still differentiable objective function such that outliers are given less weight. Using M-Estimation, the general LKR *objective function* of eq. (3.8) becomes:

$$\arg \min_{\mathbf{s}} \sum \rho(y_i - g_{\mathbf{s}}(\mathbf{x}_i)) \phi_i(\mathbf{x}), \quad (3.14)$$

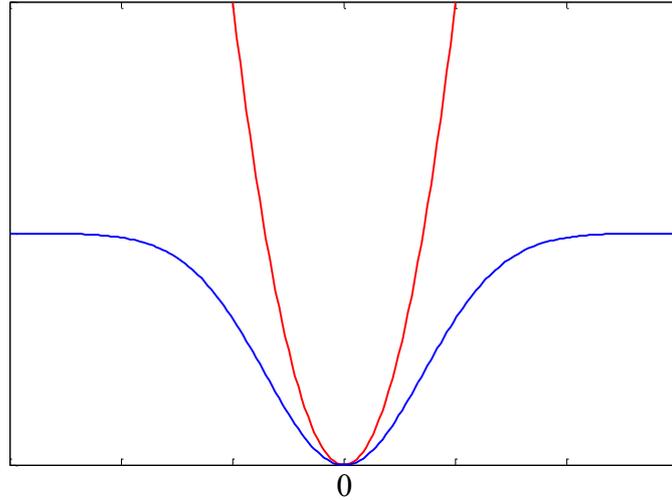
where  $g_{\mathbf{s}} = s_0 + \mathbf{a}_i^T \mathbf{s}_1 + \mathbf{b}_i^T \mathbf{s}_2 + \dots$  corresponds to the local approximation of  $f$ , and  $\rho$  is an arbitrary function. Assuming  $\rho$  is differentiable, and taking  $w(x) = \frac{d\rho}{dx}/x$ , this non-linear problem can be solved using the following Iteratively Reweighted Least Squares (IRLS) [Cleveland, 1979] minimization (see Appendix A.1 for the derivation):

$$\mathbf{s}^k = \arg \min_{\mathbf{s}} \sum \phi_i(\mathbf{x}) w(r_i^{k-1}) (y_i - g_{\mathbf{s}}^k(\mathbf{x}_i))^2, \quad (3.15)$$

where  $r_i^{k-1} = y_i - g_{\mathbf{s}}^{k-1}(\mathbf{x}_i)$  is the  $i^{th}$  residual at the  $k - 1^{th}$  iteration. Each iteration is equivalent to a weighted least square minimization. The behavior of this method depends on the choices of the function  $\rho$  and initial conditions, which are discussed below.

**Choice of  $\rho$ .** To be robust to outliers, the function  $\rho$  should grow slowly such that  $w(x)$  decreases to zero as  $x \rightarrow \infty$ . Moreover, for our purposes, its continuity degree is also an important criteria. In this paper we used Welsch's function  $\rho(x) = \frac{\sigma_r^2}{2} (1 - e^{-\frac{x}{\sigma_r}})^2$  which is  $C^\infty$ , and yields a Gaussian weight

### 3.3 Robust Local Approximations for Feature Preservation



**Figure 3.2:**  $L_2$  error (red) versus the robust Welsh's function (blue).

function  $w(x) = e^{-\left(\frac{x}{\sigma_r}\right)^2}$ . A comparison between the  $L_2$  norm and Welsh's function is given in Figure 3.2. The  $L_2$  norm grows arbitrarily large for large residuals, giving more importance for outliers. In contrast, Welsh's function stays approximately constant after a certain residual value, giving less weight to the outliers in the minimization.

**Choice of the starting point.** As with any non-linear optimization, a critical choice is the starting point. In the context of IRLS, it is usually recommended to initialize the iterations with another *robust* method such as the *median* or *median absolute deviation*. However, in the context of MLS, the continuity of the solution is of utmost importance. Therefore, we use solutions of the least square minimization of eq. (3.8) as the starting points. This has the significant advantage of leading to a continuous reconstruction at any iteration step, while being both simpler and much more efficient. In practice, this choice is equivalent to setting all the initial weights to one (i.e.,  $w(r_i^0) = 1$ ).

### 3.4 Robust and Feature Preserving Point Set Surfaces

In the previous sections, we showed the equivalence of MLS based surface definitions and LKR, and how LKR can be improved using robust statistics. In this section, we combine these formulations to develop a surface definition that preserves sharp features and details, and also robust to outliers in the data.

#### 3.4.1 Derivation of the Surface Definition

Combining the IMLS surface definition as presented in Section 3.2.3 with the general RLKR approach of Section 3.3.2 yields a robust IMLS surface (RIMLS) defined by the following IRLS minimization:

$$f^k(\mathbf{x}) = \arg \min_{s_0} \sum (s_0 + (\mathbf{x}_i - \mathbf{x})^T \mathbf{n}_i)^2 \phi_i(\mathbf{x}) w(r_i^{k-1}) \quad (3.16)$$

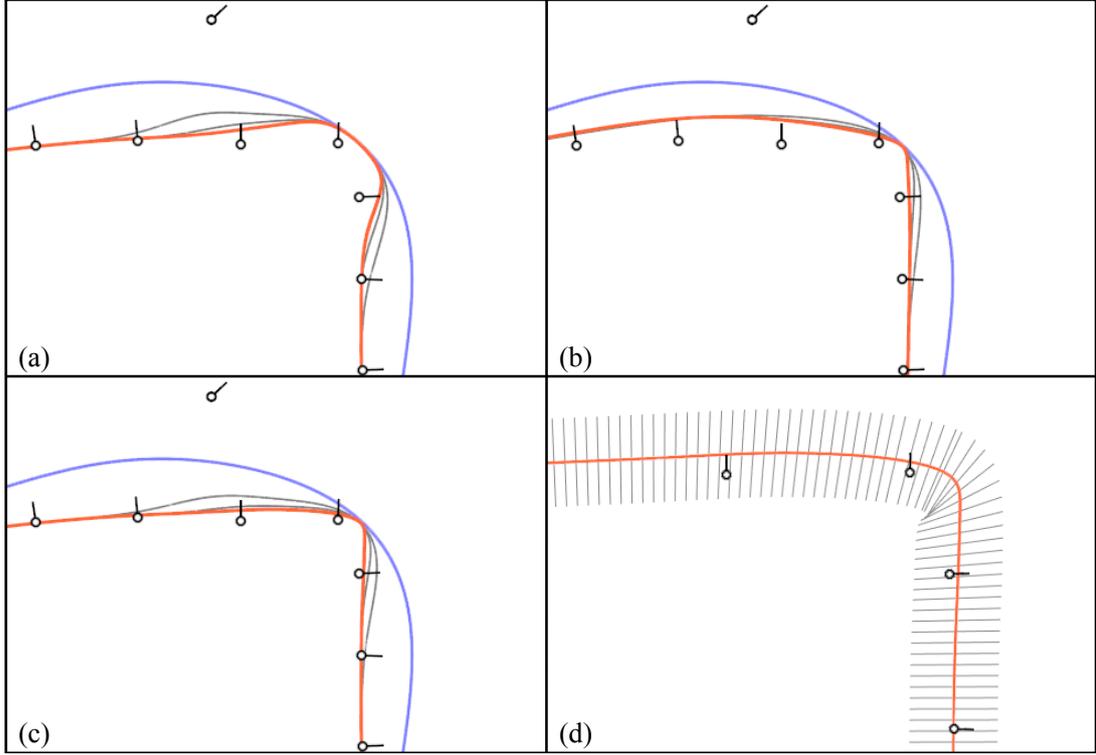
with the residuals  $r_i^{k-1} = f^{k-1}(\mathbf{x}) - (\mathbf{x} - \mathbf{x}_i)^T \mathbf{n}_i$ . This definition is robust to spatial outliers which is already a great improvement over the initial version.

However, our main motivation is to increase the accuracy of the reconstruction in the case of sharp features. While it is common to assume that samples belonging to a different surface patch across a discontinuity can also be interpreted as spatial outliers, it is easy to observe that this assumption does not hold nearby the actual first order discontinuity (Figure 3.3a). We overcome this limitation by making the observation that, across a sharp feature, the actual outliers are not the samples themselves, but the sample normals. This suggests the addition of a second re-weighting term penalizing samples having normals far away from the predicted gradient of the RIMLS surface.

Given  $\Delta \mathbf{n}_i^k = \|\nabla f^k(\mathbf{x}) - \mathbf{n}_i\|$  as the measure of the difference between the predicted gradient and a sample normal, we define a new weight function  $w_n$  as:

$$w_n(\Delta \mathbf{n}_i^k) = e^{-\frac{(\Delta \mathbf{n}_i^k)^2}{\sigma_n^2}}. \quad (3.17)$$

The iterative formula of our final RIMLS definition becomes:



**Figure 3.3:** A corner with one outlier reconstructed using IMLS (blue) and RIMLS (orange). Grey curves show RIMLS after one and two iterations. (a) is with the residual term only, (b) is with the gradient term only, while (c) includes both terms. (d) Plot of the approximate gradient in a worst case scenario.

$$f^k(\mathbf{x}) = \frac{\sum \mathbf{n}_i^T (\mathbf{x} - \mathbf{x}_i) \phi_i(\mathbf{x}) w(r_i^{k-1}) w_n(\Delta \mathbf{n}_i^{k-1})}{\sum \phi_i(\mathbf{x}) w(r_i^{k-1}) w_n(\Delta \mathbf{n}_i^{k-1})}. \quad (3.18)$$

The effect of these refitting weights is illustrated in Figure 3.3. If only the weights  $w$  are used (Figure 3.3 (a)), the surface is not effected by the outlier, but the sharp corner is not preserved. On the other hand, if only  $w_n$  is used, the surface is attracted towards the outlier, although the corner is accurately reconstructed (Figure 3.3 (b)). Finally, using both weights leads to a robust and feature preserving surface (Figure 3.3 (c)).

Further insight into the robust weights  $w_n$  can be gained by considering the first order regularization term  $\nabla f(\mathbf{x}) = \mathbf{n}_i$  we used to derive the initial IMLS definition in section 3.2.3. The error made on this constraint is not explicitly addressed in the residuals  $r_i^{k-1}$ . In fact, it corresponds to the norm of the residual gradient, which is equal to  $\Delta \mathbf{n}_i^{k-1}$ . Finally, it is interesting to note that

equation (3.18) can also be seen as an iterative trilateral filtering including both zero order and first order penalizing weights.

**Gradient term and the isophotic manifold.** The weights  $w_n$  depend on the difference of normal vectors. Assuming that both  $\phi_i$  and  $w_n$  are selected to be Gaussians, we can combine these two weights to get a single weight. Assuming that the gradient has approximately unit length, this combined weight will depend on the distance between vectors of the form  $[\mathbf{x} \ \mathbf{c}\mathbf{n}]^T$ , up to a scaling term. Ignoring the term  $w$ , this means that the filtering is done on a 2-manifold embedded in  $\mathbb{R}^6$ , with the regularized isophotic metric [Pottmann et al., 2004]. This metric is known to be sensitive to features [Lai et al., 2007]. This relation will allow us to design sampling algorithms that explicitly work on this manifold with the isophotic metric in Chapter 4.

### 3.4.2 Computation of the Surface

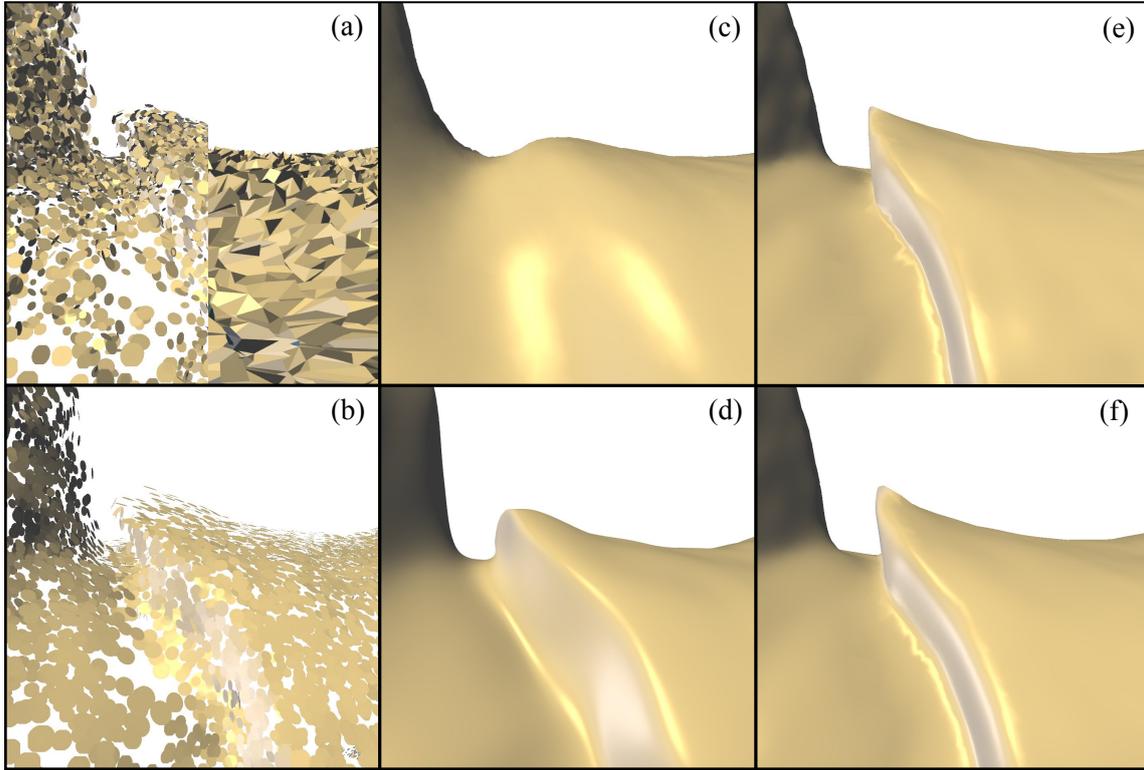
The previous equation (3.18) entirely defines our RIMLS surface as the zero set of the signed scalar field  $f$ . Computing the surface, or parts of it, can therefore be achieved using either a marching cube algorithm, by casting rays, or by projecting points onto the surface. In any case, evaluating the scalar field  $f(\mathbf{x})$  requires the computation of the surface gradient at each iteration.

Owing to the recursive nature of our definition, computing exact derivatives leads to rather complicated and expensive expressions. Fortunately, we show that tolerating a minor approximation, one can get pretty accurate results in a fairly easy manner.

The key idea is to consider that, in the definition of the current implicit function  $f^k(\mathbf{x})$ , the refitting weights are constant values. Thus, fixing  $w_i = w(r_i^{k-1})w_n(\Delta\mathbf{n}_i^{k-1})$ , the gradient  $\nabla f^k$  becomes:

$$\nabla f^k(x) = \frac{\sum w_i \phi_i(\mathbf{x}) \mathbf{n}_i + \sum w_i \nabla \phi_i(\mathbf{x}) (\mathbf{n}_i^T (\mathbf{x} - \mathbf{x}_i) - f^k(x))}{\sum w_i \phi_i(\mathbf{x})}.$$

The error due to this approximation is maximal when the refitting weights vary quickly, i.e., nearby very sharp edges. Figure 3.3d shows the accuracy of this approximation in such a worst case scenario. Higher order derivatives can be easily computed in the same manner.



**Figure 3.4:** (a) Illustration of the robust normal smoothing algorithm on a highly noisy input rendered as both a set of splats and a mesh. (b) After the normal filtering pass, the normals stay accurate near the edges. (c, d) The final RIMLS reconstruction before and after the normal smoothing step respectively. (e, f) RIMLS in the presence of 25% and 40% outliers, respectively.

### 3.4.3 Robust Normal Mollification

Our RIMLS surface definition, like many others, relies on the input surface normals. Since RIMLS embeds a low pass filter, it naturally deals very well with both spatial and normal noise. Nevertheless, when the input normals are extremely noisy it might be helpful to first filter them. In order to preserve sharp features, we perform this filtering by applying the robust optimization procedure derived in the previous sections to the normal vector field. In particular, given a point  $\mathbf{p}_j$ , we define its smooth normal  $\tilde{\mathbf{n}}_j$  as the solution of the following IRLS minimization:

$$\mathbf{n}_j^k = \frac{\sum_i \phi_i(\mathbf{p}_j) w_n(\|\mathbf{n}_j^{k-1} - \mathbf{n}_i\|) \mathbf{n}_i}{\sum_i \phi_i(\mathbf{p}_j) w_n(\|\mathbf{n}_j^{k-1} - \mathbf{n}_i\|)}, \quad (3.19)$$

where  $\phi_i$  and  $w_n$  are defined as in the previous section. Again, a critical choice is the starting point  $\mathbf{n}_j^0$ . In the bilateral filtering setting,  $\mathbf{n}_j^0$  would be set to the initial sample normal  $\mathbf{n}_j$ . However, it is well known that this strategy preserves outliers: for instance let's pick a sample  $\mathbf{p}_j$  with a flipped normal, all the similarity weights  $w_n(\|\mathbf{n}_j - \mathbf{n}_i\|)$  with  $i \neq j$  will be negligible, and its normal  $\mathbf{n}_j$  will stay unchanged. Robust starting points aim to overcome this issue, but come with additional complexity. As a much faster approximation, we propose to initialize the iterations with the weighted mean of the neighbor normals without considering the current normal itself:

$$\mathbf{n}_j^0 = \frac{\sum_{i,i \neq j} \phi_i(\mathbf{p}_j) \mathbf{n}_i}{\sum_{i,i \neq j} \phi_i(\mathbf{p}_j)} . \quad (3.20)$$

The effectiveness of our normal filtering technique is illustrated in Figure 3.4 on a highly noisy dataset. This procedure is similar to the one of Choudhury and Tumblin [Choudhury and Tumblin, 2003], and mainly differs in the fact that we are doing a full minimization and in the choice of a more robust starting point.

## 3.5 Results and Analysis

This section is devoted to investigating the properties of the new surface definition. We illustrate the robustness, feature preservation, efficiency and convenience of the approach with various examples.

### 3.5.1 Implementation and Performance Analysis

Our RIMLS algorithm is straightforward to implement: as shown by equation (3.18), evaluating the scalar field  $f(\mathbf{x})$  is essentially a simple weighted average over the neighbors of  $\mathbf{x}$  repeated until some condition vanishes. In Appendix A.2, we provide a pseudocode for the projection of a point onto the underlying RIMLS surface using a steepest gradient descent strategy. In the remaining parts of this section, we discuss various implementation choices and performance.

**Spatial weight function.** For the spatial weight function  $\phi_i(\mathbf{x})$  we used the following  $C^3$  continuous polynomial approximation of a Gaussian:

$$\phi_i(\mathbf{x}) = \left(1 - \frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h_i^2}\right)^4, \quad (3.21)$$

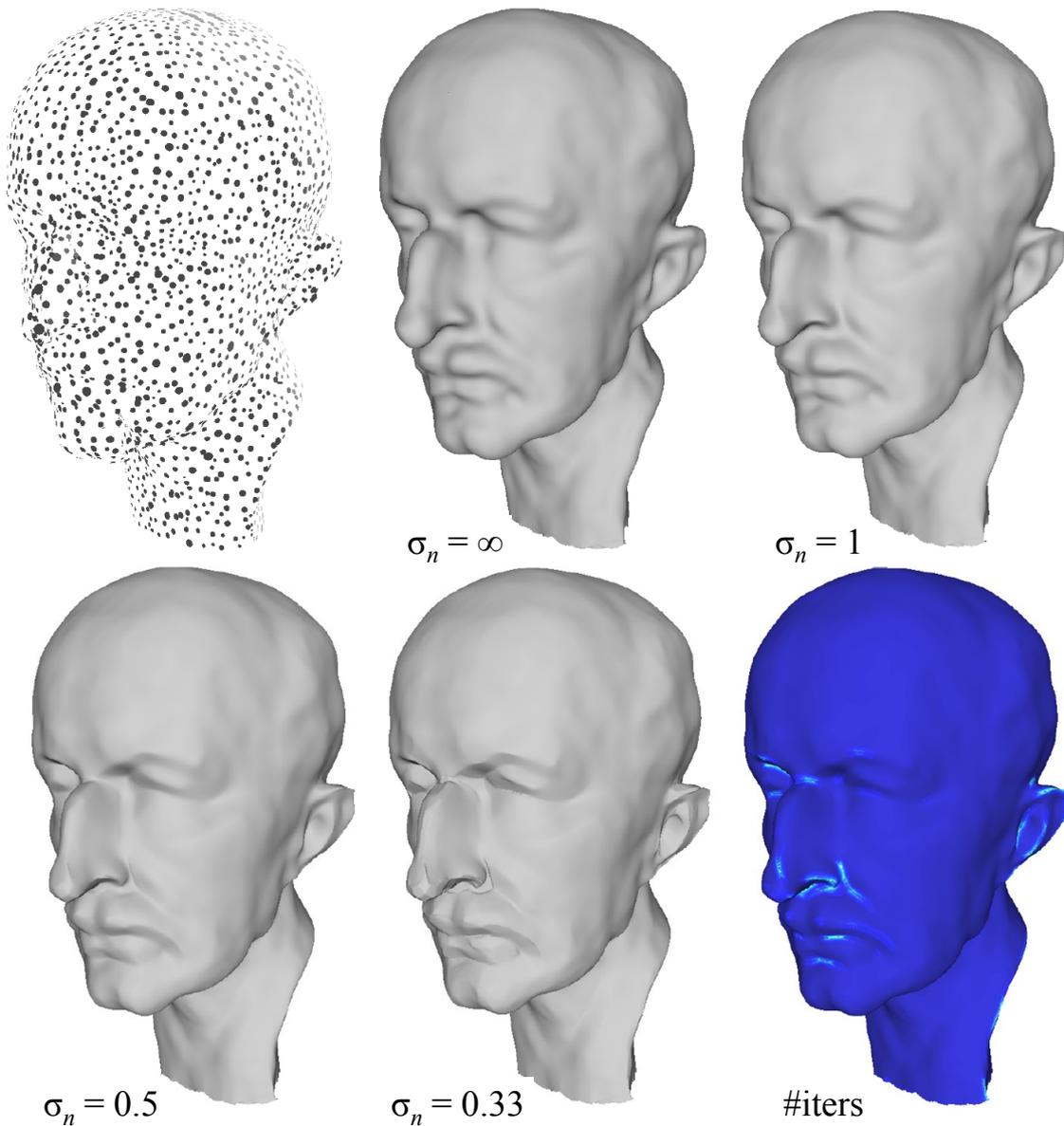
where the weight radii  $h_i$  allow to adapt to the local density. The choice of  $h_i$  depends on the amount of noise, and typical values range from 1.4 to 4 times the local point spacing. For fast neighbor retrieval, we store the input samples in a kd-tree data structure (e.g. [Adamson and Alexa, 2006a]).

**Robust weight radii.** Our robust definition introduces two additional weight radii  $\sigma_r$  and  $\sigma_n$ . Since  $\sigma_r$  is used to scale the residual term, which represents a difference of distance to the surface, it can be set locally as a fraction of the spatial radius  $h_i$ . Thus, the refitting weight  $w$  becomes:

$$w(x) = e^{-\left(\frac{x}{\sigma_r h_i}\right)^2}. \quad (3.22)$$

This way, the choice of  $\sigma_r$  does not depend on the object scale anymore, and it can be set once for all. In this paper we used  $\sigma_r = 0.5$ .

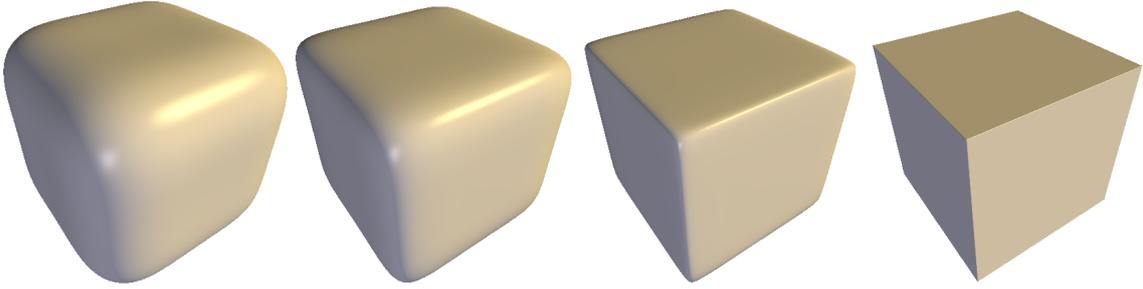
The choice of  $\sigma_n$  is much more subjective and depends on the desired degree of sharpness. If we assume the norm of the gradient is close to one,  $\sigma_n$  is used to scale the difference between unit vectors, and then typical choices range from 0.5 to 1.5. Smaller values lead to sharper results. Figures 3.5 and 3.6



**Figure 3.5:** Illustration of the effect of the sharpness parameter  $\sigma_n$  on the reconstructions. Note that in this example we set  $\sigma_r = \infty$  and therefore, the case  $\sigma_n = \infty$  corresponds to IMLS. The last picture illustrates the number of iterations to convergence for the case  $\sigma_n = 0.5$ .

illustrate this effect. We also emphasize that this sharpness parameter can be set locally for each sample, allowing the user to locally adjust the degree of sharpness, via, for instance, a painting-like tool [Pauly et al., 2003].

**Termination criteria.** The last implementation choice concerns the stopping criterion of the iterations. One option would be to iterate until convergence



**Figure 3.6:** A cube sampled with 4 sample points per face is reconstructed by RIMLS with  $\sigma_n = 3, 1.2, 0.75$  and  $0.55$ .

is detected. We detect convergence by tracking the relative change of the refitting weights, i.e.:

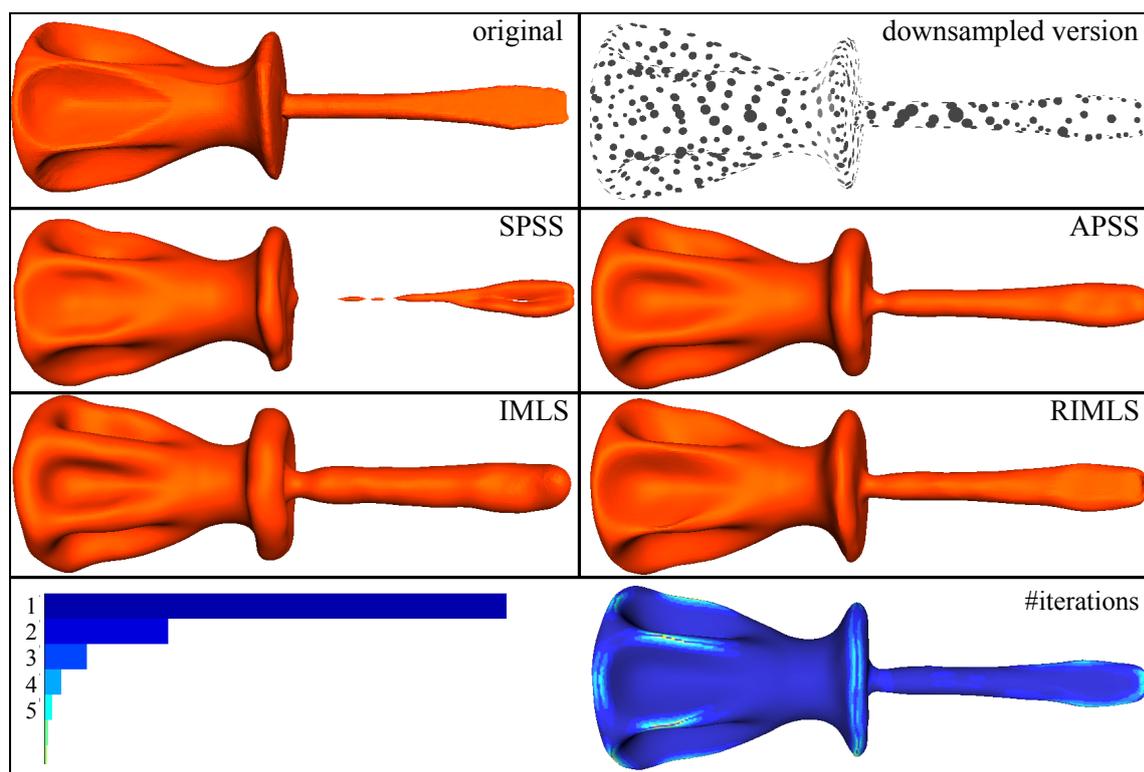
$$\max_i \left| \frac{w(r_i^k)w_n(\Delta\mathbf{n}_i^k)}{\sum w(r_i^k)w_n(\Delta\mathbf{n}_i^k)} - \frac{w(r_i^{k-1})w_n(\Delta\mathbf{n}_i^{k-1})}{\sum w(r_i^{k-1})w_n(\Delta\mathbf{n}_i^{k-1})} \right| < t,$$

where  $t$  is a user defined threshold (eg,  $t = 10^{-4}$ ).

On the other hand, we took a special care to build RIMLS such that at any iteration it generates a reasonable and continuous surface. Therefore, one may fix a constant number of iterations while still keeping the convergence test as an early termination optimization. Note that using a fixed number of iterations, preferably small, also allows to extract a closed form formula of the implicit scalar field  $f$ .

As demonstrated in Figures 3.3, 3.5, and 3.7, we observed that the converge rate of our algorithm is extremely high. Not surprisingly, the number of refitting iterations for full convergence slightly increases near sharp features or high frequency details, while only one iteration is needed in smooth regions. In fact, according to our experience, in most cases the differences after one refitting iteration are seldom perceptible (e.g. Figure 3.8).

**Performance.** Thanks to the very high convergence rate and simplicity of our approach, we can achieve performance of similar order as the fastest MLS definitions. In practice, the neighbor search part still remains the main computational cost. We emphasize that our RIMLS algorithm could perfectly fit into the real-time upsampling and rendering framework of algebraic point set surfaces (APSS) [Guennebaud et al., 2008]. The following table summarizes for both APSS and RIMLS the number of arithmetic operations required by one evaluation as a function of the number of neighbors  $k$  and refitting iterations  $i$ .



**Figure 3.7:** Comparison of different definitions and our definition under very sparse sampling conditions. Bottom row shows the number of RIMLS iterations with the corresponding histogram.

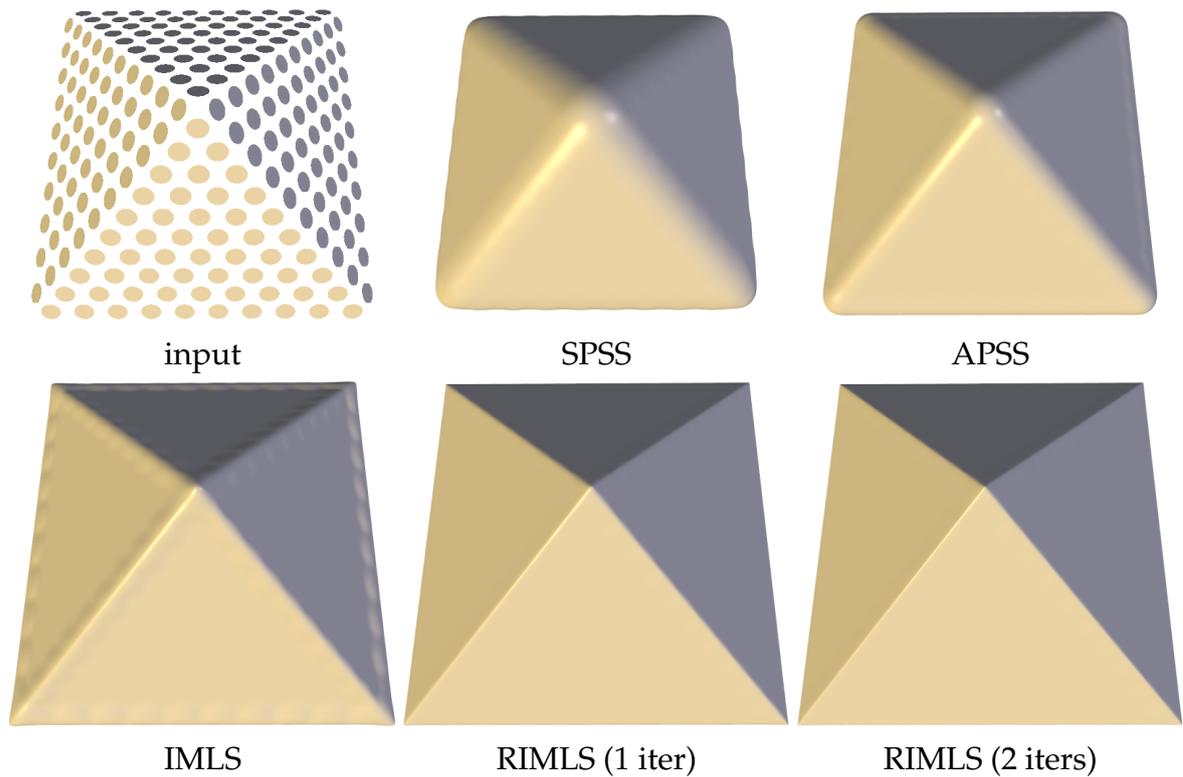
APSS (approx normals)	APSS (exact normals)	RIMLS
$k * 38 + 45$	$k * 130 + 309$	$i * (k * 56 + 13)$

### 3.5.2 Reconstruction Results

We evaluated the capacity of our approach to reconstruct and preserve surface features on a wide variety of models. In the following comparisons, SPSS stands for Adamson et al.’s *simple point set surfaces* based on normal averaging [Adamson and Alexa, 2003].

Moreover, unless stated otherwise, all the results presented here have been produced using the raw normals without using the pre-smoothing procedure described in Section 3.4.3.

In contrast to previous sharp feature preserving reconstruction methods for point set surfaces, we emphasize that the goal of our algorithm is not to generate surfaces with precise normal discontinuities, but rather to improve

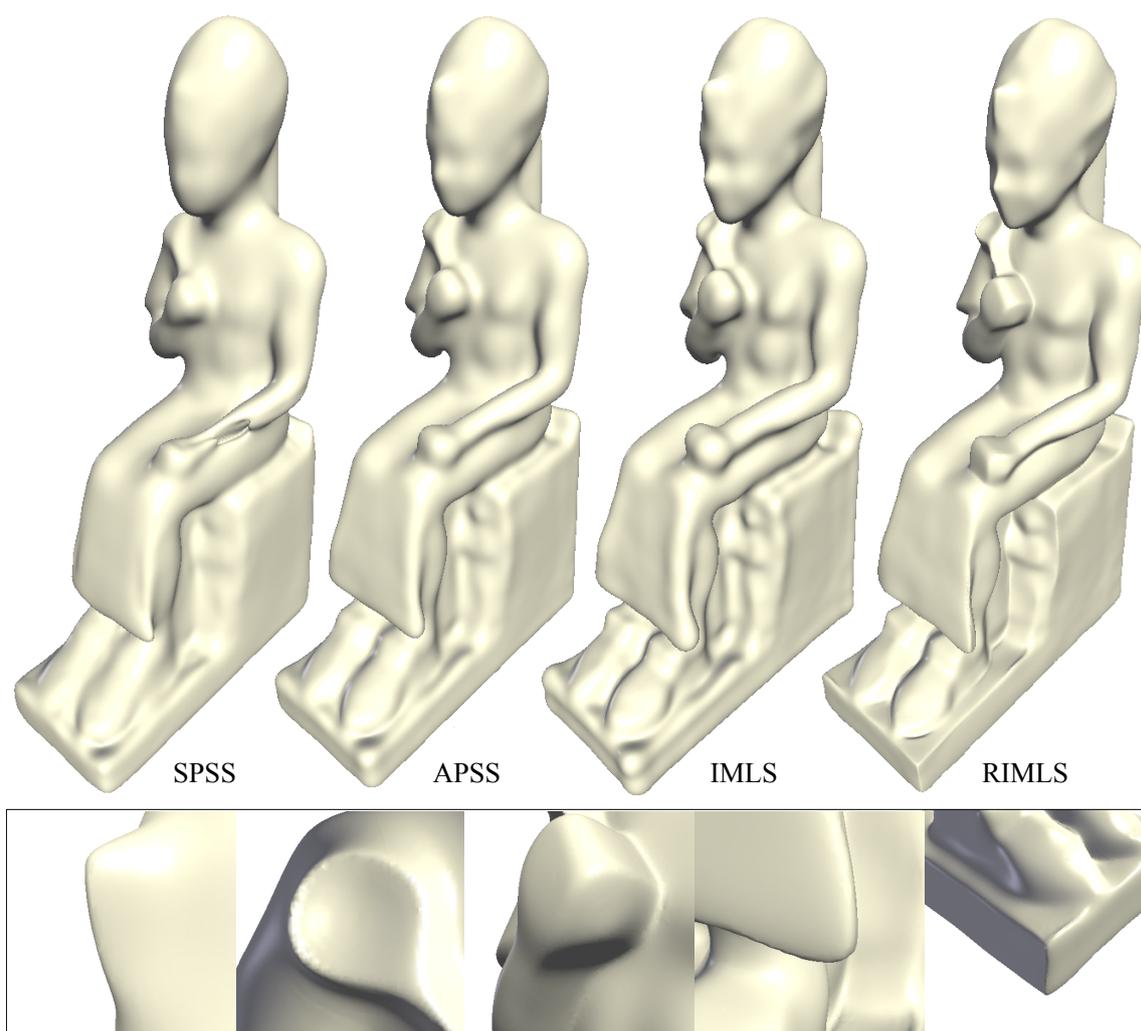


**Figure 3.8:** *Reconstruction of sharp features for the difficult case of four intersecting planes.*

the faithfulness of the approximation at any scale. This behavior is depicted in Figure 3.9 on a sparsely sampled model. The shape of many parts such as the hands and chest are distorted by previous MLS definitions. In contrast, RIMLS successfully recovers these parts, while smoothing out lower scale details and noise. Sharp features are preserved as can be seen at the corners and edges of the stand, and the tip of the head. The expanding and shrinking effect at the tip of the skirt is also prevented. As can be seen in Figure 3.5, this behavior of RIMLS is also effective in the case of very smooth objects.

The ability of our approach to recover sharp features from sparse sampling is further demonstrated in Figure 3.7. The original model is subsampled to contain approximately 4% of its samples and the resulting model is reconstructed with several MLS definitions. RIMLS preserves fine details and sharp features on the body while providing a more stable overall shape. Note the similarity of the RIMLS reconstruction from the subsampled model to the actual model.

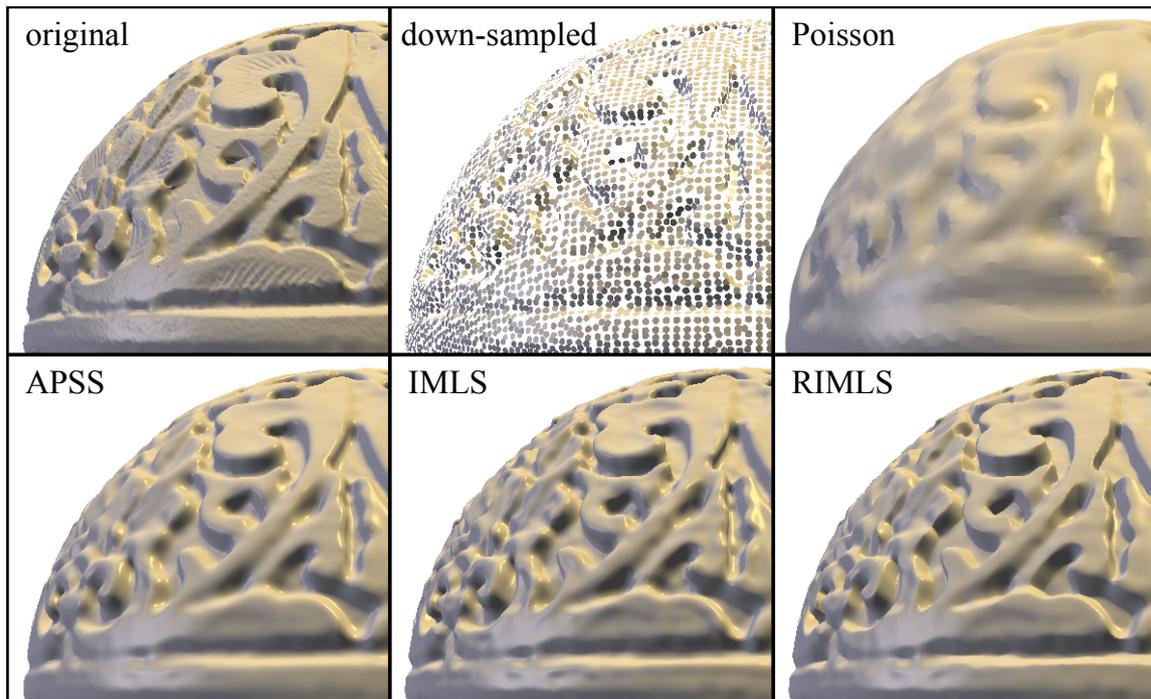
In the same vein, Figure 3.10 includes a comparison to the Poisson surface reconstruction (PSR) technique [Kazhdan et al., 2006] which also takes into



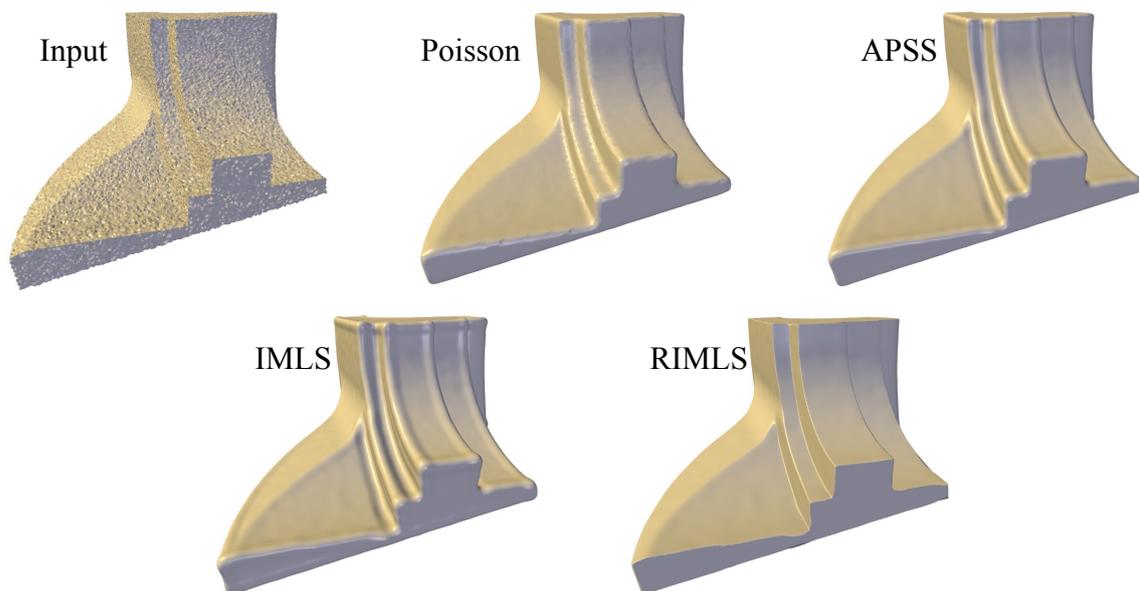
**Figure 3.9:** Top row shows a comparison of various MLS definitions on a low sampled Ramses model. Bottom row shows details of the RIMLS reconstruction.

account input normals. While PSR works well in extracting fine details out of densely sampled datasets, when sampling density drops, MLS based approaches, and RIMLS in particular, are clearly superior.

Figure 3.11 illustrates the ability of our algorithm to reconstruct sharp features from a highly noisy CAD model. As can be seen, because of the large amount of noise, a large spatial filter has to be employed, whence the large undesirable effects produced by both APSS and IMLS near the edges and corners. For the same reason, PSR also produced oversmoothed edges in spite of the relatively dense sampling. Figure 3.4 demonstrates the stability of the reconstructions under the influence of high levels of noise and outliers. For this difficult case, the advantage of utilizing the pre-smoothing procedure (Section 3.4.3) is evident.

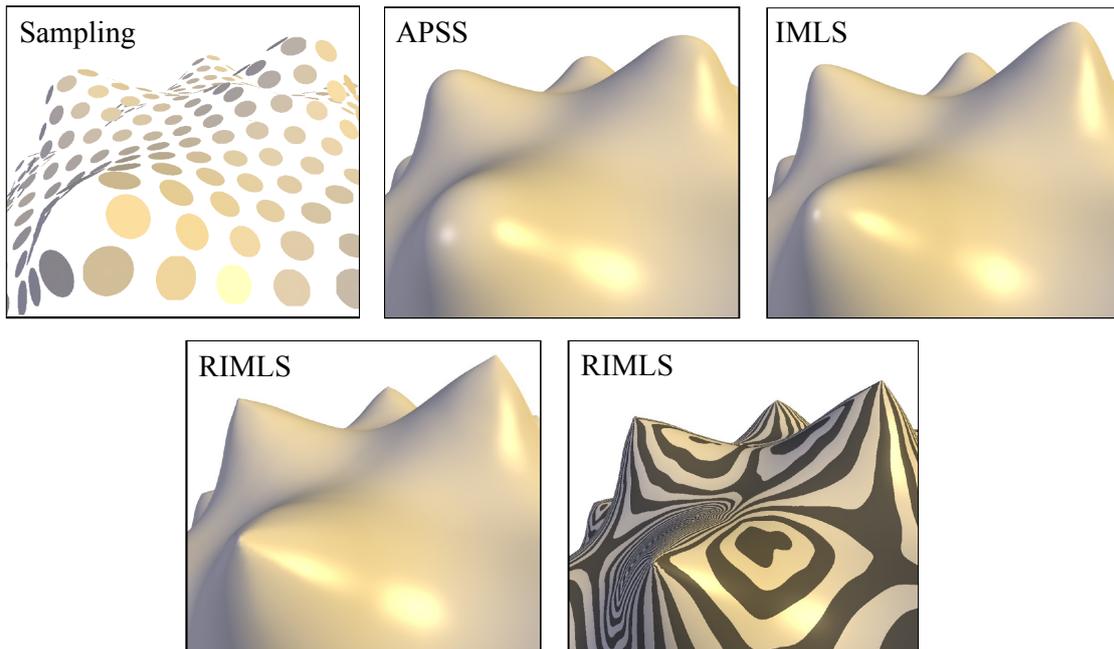


**Figure 3.10:** Comparison of various reconstruction methods on a model downsampled to about 2.8% of the original data. The Poisson reconstruction has been generated with a limit of one sample per node and depth 12.



**Figure 3.11:** Various reconstructions of the fandisk model after being randomly corrupted by noise of magnitude 0.5% of the object size.

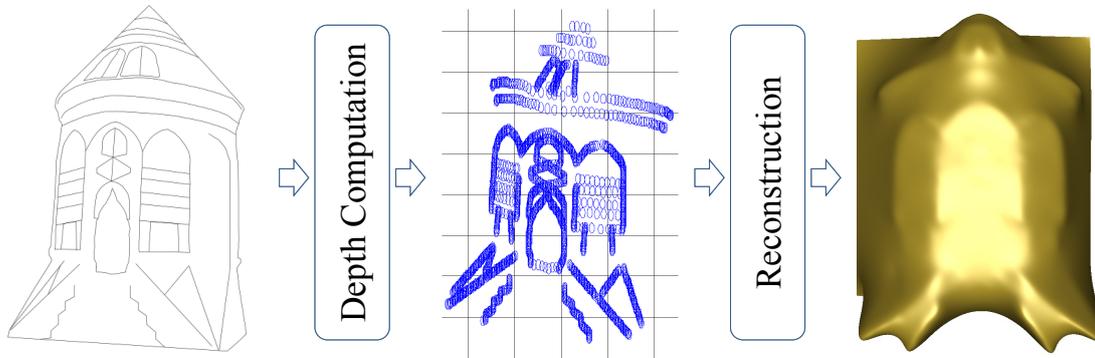
## Meshless Reconstruction of Manifolds



**Figure 3.12:** *Illustration of the reconstruction of coarsely sampled peaks. Bottom right is a rendering of the RIMLS reconstruction with reflection lines.*

Furthermore, Figures 3.8 and 3.12 show that our approach can naturally handle high order corners and peaks respectively, which is particularly difficult with previous sharp feature reconstruction methods.

### 3.6 An Application of Local Approximations with Global Constraints



**Figure 3.13:** Given an input sketch (left), depths for the points on the curves are computed (middle), which are then used to reconstruct a surface approximating the sketched objects's surface.

Although local approximations are fast and can be made robust with the methods explained in the previous sections, they become unstable when the point set exhibits high degrees of non-uniformness. For these cases, it is beneficial to have global terms to have a reliable reconstruction. With this approach, evaluations of the function still remains local, while a global system needs to be solved once to get the reconstruction coefficients.

In this section, we would like to illustrate an interesting application of this approach using local radial basis functions (RBF) and several global energy terms. The goal is to reconstruct a surface of an object from its sketched illustration. The method starts with extracting a set of symmetric curves and points (Figure 3.13 (left)). Assuming that the sketch represents an object that is bilaterally symmetric, the depth values at the symmetric pairs of points can be obtained (Figure 3.13 (middle)). Once we have the depths, the problem becomes reconstructing a depth-map parametrized on the image plane using these sparse set of depth constraints (Figure 3.13 (right)). The challenge lies in the sparsity of these constraints. We omit the details of the method to get the depth values and refer the reader to the relevant paper [Öztireli et al., 2011], and focus on the reconstruction of the depth-map using the computed depths.

Given the sparse set of points with depths, we would like to reconstruct a smooth 3D surface that interpolates these points. We propose to combine a continuous formulation [Ng et al., 2010] with energy terms from surface approximation and deformation literature that are known to produce tight and

smooth reconstructions. We discretize the energy terms and minimize them along with other terms and constraints using compactly supported radial basis functions centered at each pixel, to get a continuous function for the depth map. Although this formulation results in a big linear system to solve for the coefficients, it is efficiently solvable thanks to the extreme sparseness. Once the coefficients are computed, each evaluation of the function at a pixel involves only a small set of neighboring pixels around the given pixel.

**Function representation and constraints.** We represent the depth map function as a sum of compactly supported radial basis functions  $k(\mathbf{x}, \mathbf{x}_i) = e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/\sigma^2}$  such that  $f(\mathbf{x}) = \sum f_i k(\mathbf{x}, \mathbf{x}_i)$  for pixels  $\mathbf{x}_i$ . Although bigger  $\sigma$  gives smoother reconstructions, it results in much higher computation times. Since we can determine the smoothness of  $f$  through the energy terms, we fixed  $\sigma = 1.2$  such that a three-ring neighborhood of each pixel is covered by the effective support of the Gaussian (assuming the size of a pixel is 1). To make  $f(\mathbf{x})$  interpolate the known depth values, we should have  $f(\mathbf{x}_i^d) = d_i$  where we denote the points (pixels) with depth values by  $\mathbf{x}^d$ . By writing the expression for  $f(\mathbf{x}_i^d) = \sum f_j k(\mathbf{x}_i^d, \mathbf{x}_j) = d_i$ , and gathering all equations, we get a linear system  $\mathbf{K}\mathbf{f} = \mathbf{d}$ , where  $\mathbf{K}$  is an  $n_d$  by  $n$  matrix with  $n_d$  denoting the number of depth values and  $n$  the number of pixels.

**Energy terms and minimization.** We discretize the thin plate spline energy with the following sum:  $\sum_k f_{xx}^2(\mathbf{x}_k) + 2f_{xy}^2(\mathbf{x}_k) + f_{yy}^2(\mathbf{x}_k)$ . Minimization of this sum leads to a quadratic form  $\mathbf{f}^T \mathbf{T} \mathbf{f}$ , where  $\mathbf{T}_{ij} = \sum_k k_{i,xx}(\mathbf{x}_k)k_{j,xx}(\mathbf{x}_k) + k_{i,xy}(\mathbf{x}_k)k_{j,xy}(\mathbf{x}_k) + k_{i,yy}(\mathbf{x}_k)k_{j,yy}(\mathbf{x}_k)$  and the subscripts denote derivatives.

Similarly, we discretize a specific case of the Sobolev norm,  $\int \|\nabla f(\mathbf{x})\|^2$ , which is also known to approximate the membrane or stretching energy [Terzopoulos et al., 1987; Celniker and Gossard, 1991]. This leads to another quadratic form  $\mathbf{f}^T \mathbf{B} \mathbf{f}$  with  $\mathbf{B}_{ij} = \sum_k \nabla k_i(\mathbf{x}_k)^T \nabla k_j(\mathbf{x}_k)$ .

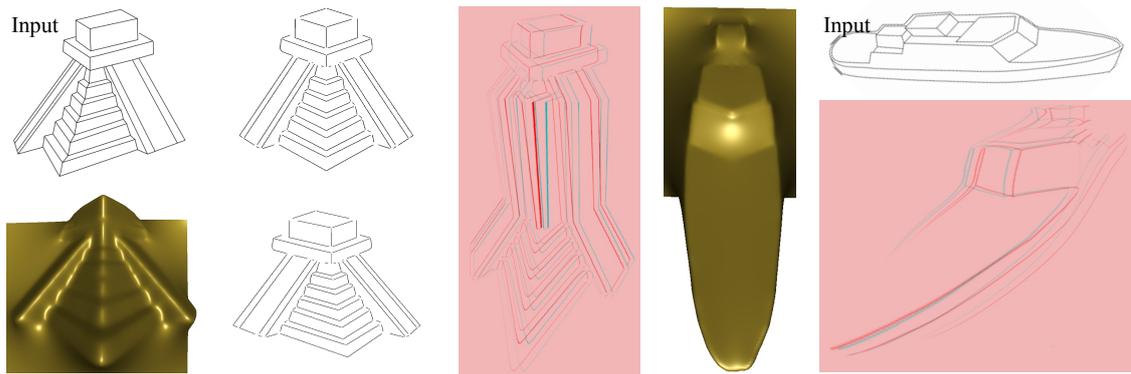
We gather the terms arising from the constraints and the energies as follows:

$$E(\mathbf{f}) = \|\mathbf{K}\mathbf{f} - \mathbf{d}\|^2 + \mathbf{f}^T (\lambda_T \mathbf{T} + \lambda_B \mathbf{B}) \mathbf{f}. \quad (3.23)$$

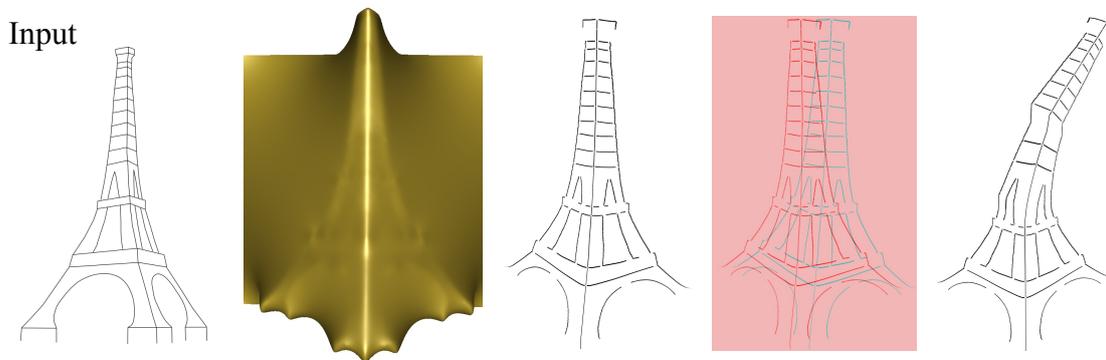
By solving the linear system resulting from this minimization, different continuous functions can be defined depending on the parameters used. For the results presented, we set  $\lambda_T = 3$  and  $\lambda_B = 1$ .

**Example reconstructions.** Some example reconstructions are shown in Figures 3.14 and 3.15. The reconstructed models capture the overall shape of the depicted objects well. The sketch-textured depictions with the reconstructions match our intuition of the sketch faithfully as illustrated by the rotated views of the sketches. Having a depth map allows us to construct

### 3.6 An Application of Local Approximations with Global Constraints



**Figure 3.14:** *Given input sketches, approximate geometries can be reliably reconstructed. This allows to change viewing directions or generate anaglyph images.*



**Figure 3.15:** *Left to right: The input sketch, reconstructed geometry, view of the sketch from a different angle, an anaglyph image of the sketch, and a physically-based deformation applied to the sketch.*

stereoscopic images of the sketches from different angles. It is interesting to see that, even without any 3D cues from shading, a good sense of depth can be obtained from these images. Using the reconstructed models, one can also deform the geometry or perform physically based simulations as in Figure 3.15.

### 3.7 Discussion and Outlook

We have demonstrated that MLS and local kernel regression can be advantageously combined to design a novel surface representation with great feature preserving ability. Our approach is particularly robust in the presence of noise, outliers and sparse sampling. While previous *sharp feature* reconstruction techniques for MLS focused on the generation of exact  $C^0$  discontinuities at edges and corners, our approach *sharpifies* the initial over-smoothed solution to a more faithful approximation, keeping the surface differentiable. As a result, our approach can improve the representation of sharp features and fine details of any frequency, and naturally deals with complex situations such as high order corners and peaks. A central advantage of our approach is its extreme simplicity, making it particularly appealing for interactive applications.

Though our results show the reconstructed surface is continuous for a wide range of sampling density and parameters, theoretical bounds on these are yet to be derived. We show a possible approach of guaranteeing a stable reconstruction with near-optimal sampling in Chapter 4. As most of the non-linear techniques, ours depends on the choice of a starting point. While our current solution already provides satisfactory results, we believe it would still be interesting to investigate more robust alternatives.

In addition to the novel surface definition, we showed that MLS based point set surfaces can be interpreted in terms of LKR, suggesting to adopt further methods from regression literature to solve various problems in the geometric setting. By changing the order of regression and constraints, or by investigating unsupervised kernel regression, we believe that superior MLS reconstructions will be possible.

Furthermore, we showed how such local approximations can be combined with global energy functions to obtain reliable reconstructions for the case of very sparse and non-uniform constraints. Other methods have also tried to incorporate global information for better reconstructions (e.g. [Avron et al., 2010; Dey et al., 2012]). We believe that integration of the LKR framework with global statistics gathered from the sample points can significantly improve the reconstructions when the sampling is highly non-uniform.

So far we considered a single source of data, that is, the set of points sampled from a surface, possibly equipped with normals, as the input. For many applications, richer data can be acquired from different sensors. Incorporation of such data can considerably improve the reconstructions for applications that impose specific constraints. An example of this approach applied to

rendering where facial texture and features are considered in addition to geometry is illustrated in Section 4.7.

Finally, we believe that considering multiple shapes, their relations, and human visual system in addition to the data for a single shape can provide valuable priors for reconstruction. Integration of relevant machine learning and statistics techniques into the LKR framework can provide reconstructions matching our intuition well even under extreme levels of noise, without sacrificing the convenience and efficiency of the surface definitions.



---

# C H A P T E R

# 4

## Spectral Sampling of Manifolds



Accuracy of reconstructions ultimately depends on how much information is at hand in the form of point samples on a surface. The amount and distribution of point samples, coupled with the reconstruction method used determines the quality of the reconstructed surfaces. In this chapter, we show how sampling conditions based on the spectrum of the Laplace-Beltrami operator can be derived by using a kernel-based approximation. This approximation allows us to connect the produced sampling pattern to the reconstruction method presented in Chapter 3, resulting in very accurate reconstructions.

Finding an optimal sampling that ensures accurate reconstructions with minimum redundancy is one of the fundamental issues for scattered data ap-

proximation. The sampling problem is especially hard for general manifolds since no parametric domain is defined and standard methods from signal processing do not apply [Saucan et al., 2008].

There have been significant efforts in spectral graph theory and computational harmonic analysis domains to extend signal processing results to discrete approximations of manifolds by utilizing techniques from spectral analysis [Lafon and Lee, 2006]. Following the seminal work of Taubin [1995], the developments in harmonic analysis have been inspiring many successful algorithms in geometry processing [Zhang et al., 2007; Sun et al., 2009]. Despite their excellent performance and theoretical foundations, spectral methods are global and need eigendecomposition of a large matrix. Although there have been local patching approaches to approximate the spectral properties [Pauly and Gross, 2001], finding a decomposition of a manifold into patches from which *global spectral properties* can be derived is a difficult problem. This global nature of spectral methods makes their application to massive datasets unfeasible even though specialized methods are employed [Vallet and Lévy, 2008]. A more fundamental problem when using spectral methods for sampling is coupling the samplings to reconstructions. Finding a direct link between the spectral structure of a manifold and the reconstruction methods commonly used in geometry processing is a highly non-trivial task. These have been hindering use of current spectral approaches in a sampling algorithm.

In this chapter, we introduce an analysis and sampling algorithms that overcome these drawbacks. The proposed sampling algorithms utilize new measures derived from spectral analysis of manifolds. Combining results from spectral analysis, kernel methods and matrix perturbation theory, we derive our measure that quantifies the change a point causes to the manifold, if it is added to the point set defining the manifold. Although the derived measure conceptually quantifies *global* changes to the manifold, it only requires *local* computations. This allows us to use the theoretical framework within efficient algorithms. The measure also has strong connections to approximation by kernel regression. Hence, generated samplings result in accurate reconstructions using kernel based algorithms such as Moving Least Squares (MLS) based Point Set Surfaces (PSS) [Levin, 2003; Alexa et al., 2003], and in particular the reconstruction method we introduced in Chapter 3.

We start this chapter by an overview of the fields of spectral analysis of manifolds (Section 4.1) and kernel techniques (Section 4.2) in the scope of this thesis. Building on results from these two fields, we then propose a spectral measure that quantifies the effect of a point on the definition of a

manifold (Section 4.3), and show its relation to the reconstruction method we presented in Chapter 3 (Section 4.4). Based on this measure, we develop efficient sampling algorithms (Section 4.5), and analyze the performance of the algorithms, distribution of generated points, and accuracy of resulting reconstructions in Section 4.6. Finally, we present an application of the algorithms to high quality facial rendering in Section 4.7.

## 4.1 Spectral Analysis of Manifolds

Our analysis and algorithms depend on spectral analysis of manifolds based on the eigenstructure of the Laplace-Beltrami operator. The Laplace-Beltrami operator can be seen as an extension of the Laplace operator to manifolds. It is used widely to describe the intrinsic structure of a manifold since its eigenstructure conveniently encodes all information about the manifold up to isometry. Below we summarize relevant concepts and results related to the spectral analysis of manifolds.

### 4.1.1 Harmonic Analysis

Classical signal processing relies on constructing a basis  $b_i(\mathbf{x})$  for a space of functions, and representing a function  $f(\mathbf{x})$  in this basis, where the domain of the functions are assumed to be an Euclidean space,  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ . The new representation of the function is given as  $\tilde{f}(\mathbf{x}) = \sum c_i b_i(\mathbf{x})$ . Here, the coefficients are computed by an appropriate inner product  $c_i = \langle f(\mathbf{x}), b_i(\mathbf{x}) \rangle$ .

This framework provides a high degree of flexibility to define different spaces and bases for different applications. An important example of this analysis is the Fourier transform of one dimensional functions,  $f : \mathbb{R} \rightarrow \mathbb{R}$ , where the basis functions are sines and cosines, which are compactly written as  $b(w, x) = e^{-2\pi i x w}$ . Note that in this case we have a continuous set of basis functions. Assuming an integrable function  $f$ , the coefficients of its expansion in terms of the basis functions are given as  $c(w) = \langle f(x), b(w, x) \rangle = \int_{\mathbb{R}} f(x) e^{-2\pi i x w} dx$ . Operations on this Fourier expansion have been proven to be powerful in many fields, since it provides a tool to separate a given function into *frequency* components. Each frequency, denoted by  $w$  in the above definition, represents a different level of detail in the function. The lowest frequency,  $w = 0$ , gives a constant basis  $b(0, x) = 1$ , and hence the coefficient  $c(w)$  is just the integral of the function  $f$ . This is just a very global summary of the function  $f$ . As the frequency is increased, the basis functions oscillate more and more, and higher oscillations in the function  $f$  are captured. The frequencies naturally parallel our notion of detail in a function.

The Fourier transform can be extended to higher dimensional Euclidean domains  $\mathbb{R}^d$ . It can be further extended to manifolds such that the transform acts on the functions living on a given manifold. Such extensions are the subject of *harmonic analysis*. In this setting, the basis functions are called *waves*. The goal is to define these basis functions, as extensions from the classical setting. This can be achieved by the fundamental property of wave functions

that they are eigenfunctions of the Laplace operator. As an example, it is easy to derive that when applied to  $\sin(2\pi wx)$ , the usual Laplace operator in  $\mathbb{R}$ , that is the second derivative, results in  $\Delta \sin(2\pi wx) = -(2\pi w)^2 \sin(2\pi wx)$ . Hence, the Laplace operator applied to the wave functions only scales them.

This connection of the wave functions to the Laplace operator is used to derive wave functions of spaces represented by manifolds through the extension of the Laplace operator. In the following section, we present the extension of it to Riemannian manifolds, called the Laplace-Beltrami operator.

### 4.1.2 The Laplace-Beltrami Operator

The Laplace-Beltrami operator, which we denote by  $\Delta_M$  for a manifold  $M$ , is the generalization of the Laplace operator in Euclidean spaces to curved Riemannian manifolds. Specifically, it is defined in terms of the divergence and gradient operators on a manifold  $M$  as  $\Delta f = \text{div grad} f$ , where  $f$  is a function defined as  $f : M \rightarrow \mathbb{R}$ . The Laplace-Beltrami operator can be precisely defined by taking the intrinsic structure of the manifold into account via the metric tensor for the manifold. It inherits many useful properties of the Laplace operator. In particular, it is a linear and Hermitian operator, implying that it has an orthogonal set of eigenfunctions for different eigenvalues.

Once defined, its eigenfunctions (waves on the manifold) and eigenvalues (frequencies) can be computed. This is done by solving the equation

$$-\Delta_M u(x) = \lambda u(x) \quad (4.1)$$

for  $x \in M$  and  $u : M \rightarrow \mathbb{R}$ . The eigenfunctions and eigenvalues define the manifold up to isometry. Hence, they are sufficient to describe the intrinsic properties of the manifold. Unfortunately, analytical solutions of this equation are not known for most manifolds except for simple cases. Three important special manifolds where analytic forms of the eigenstructure is known are a fixed-length interval, a square region in  $2D$ , and a sphere in  $3D$ . Assuming Dirichlet boundary conditions, i.e.  $u(x) = 0$  on the boundaries, the eigenfunctions of an interval  $(0, l)$  can be computed as  $\sqrt{\frac{2}{l}} \sin\left(\frac{n\pi}{l} x\right)$  with  $n \in \mathbb{N}$ . Note that, similar to the case of  $\mathbb{R}$ , this finite interval results in sines as the eigenfunctions. A more interesting case is a  $2D$  square. The eigenfunctions of this manifold have already been studied by Ernst Florens Friedrich Chladni in the 19<sup>th</sup> century. He used a vibrating plate of square shape to observe the patterns formed by the sand particles on the plate. These patterns actually correspond to the zero-crossings of the eigenfunctions of the Laplace-Beltrami operator.

For a square  $(0, \pi) \times (0, \pi)$ , closed-form solutions for the eigenfunctions can be derived as

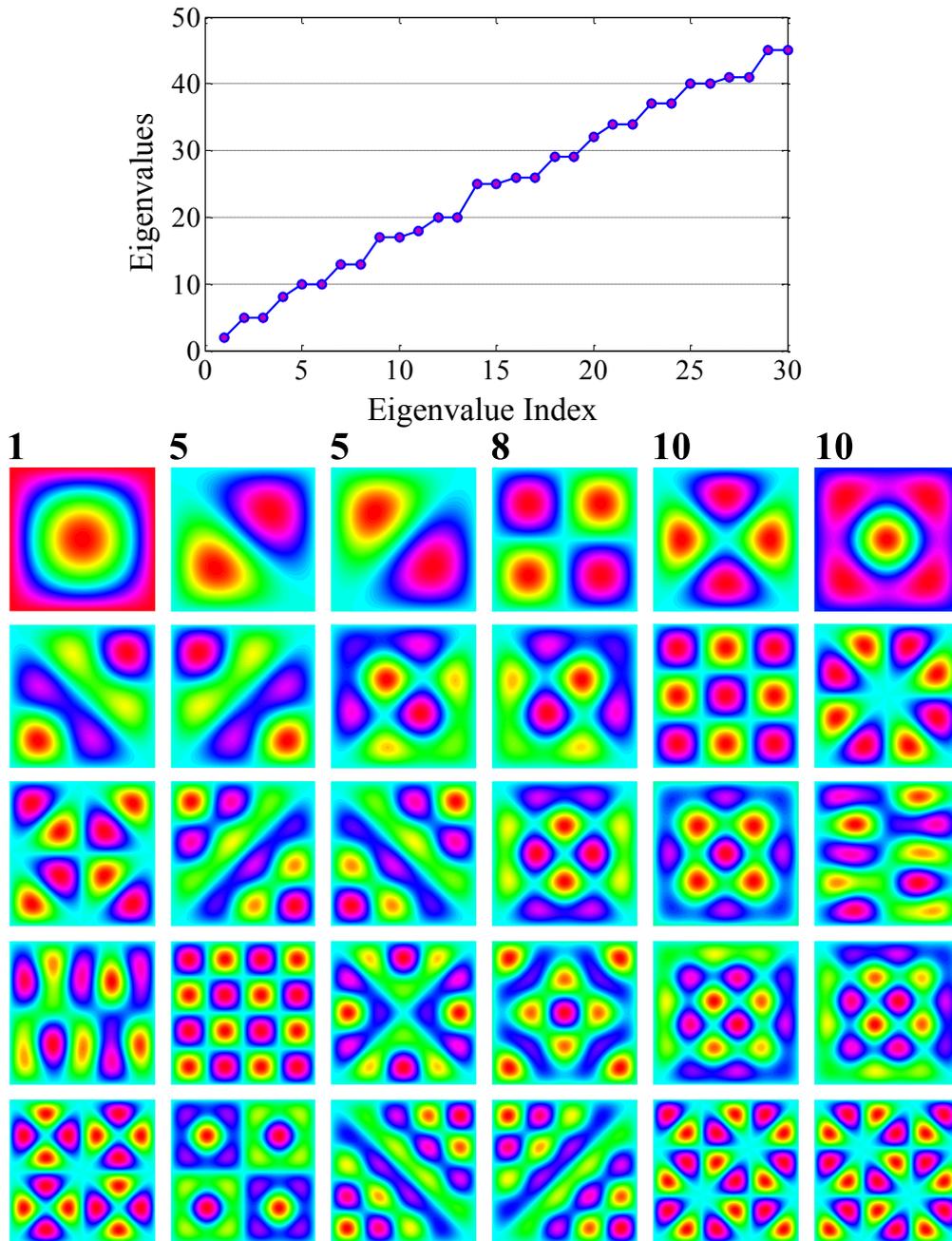
$$u_{mn}(x, y) = a \sin(nx) \sin(my) \quad (4.2)$$

with the eigenvalues  $\lambda_{mn} = n^2 + m^2$  and a certain constant  $a$ . The indices  $n$  and  $m$  can be reduced to a single index, and one gets  $\lambda = 1, 5, 5, 8, 10, 10, \dots$  for the eigenvalues. Note that some of these eigenvalues are repeated twice, and hence the corresponding eigenfunctions span a two-dimensional subspace. For example, for  $\lambda = 5$ , one has the linear combination  $au_{12}(x, y) + bu_{21}(x, y)$ . Thus, the eigenfunctions for repeated eigenvalues can be set in different ways. The first 30 eigenvalues and corresponding eigenfunctions are shown in Figure 4.1.

Another important example of Laplace-Beltrami eigenvalues/functions are the spherical harmonics, extensively used in computer graphics. The manifold in this case is a sphere and the eigenvalues are given as  $\lambda = l(l + 1)$ ,  $l = 0, 1, \dots$ . Each eigenvalue is repeated  $2l + 1$  times, as illustrated in Figure 4.2 along with some of the eigenfunctions for three different eigenvalues.

Cases like the above examples where analytic forms of the eigenfunctions and eigenvalues are conveniently available are very rare. For this reason, several discrete techniques that can reliably compute these on general surfaces represented by meshes have been developed. An example computation is illustrated in Figure 4.3. The plots again reveal the multiresolution nature of the eigenfunctions.

As illustrated by these examples, the eigenfunctions of the Laplace-Beltrami operator adapt to and carry interesting information about the geometry of the manifold they are defined on. The structure of the eigenfunctions can be described in terms of *nodal domains* and *nodal sets*. A nodal domain of an eigenfunction  $u_i(x)$  is a region of space with a constant sign for  $u_i(x)$ , and the nodal sets are the zero crossings  $\{x | u_i(x) = 0\}$ . The geometry of the nodal domains/sets are characterized by several theorems. It is known that nodal sets have dimension one less than the (intrinsic) dimension of the manifold, and they intersect at constant angles. Hence, they are curves on a 2-dimensional manifold, and they capture various symmetries of the manifold. Another important result states that an eigenfunction  $u_i(x)$  can have at most  $i$  nodal domains. This means that more and more oscillations are expected for larger indices, and thus the eigenfunctions form a basis suitable for multi-resolution analysis, with the corresponding eigenvalues resembling frequencies.

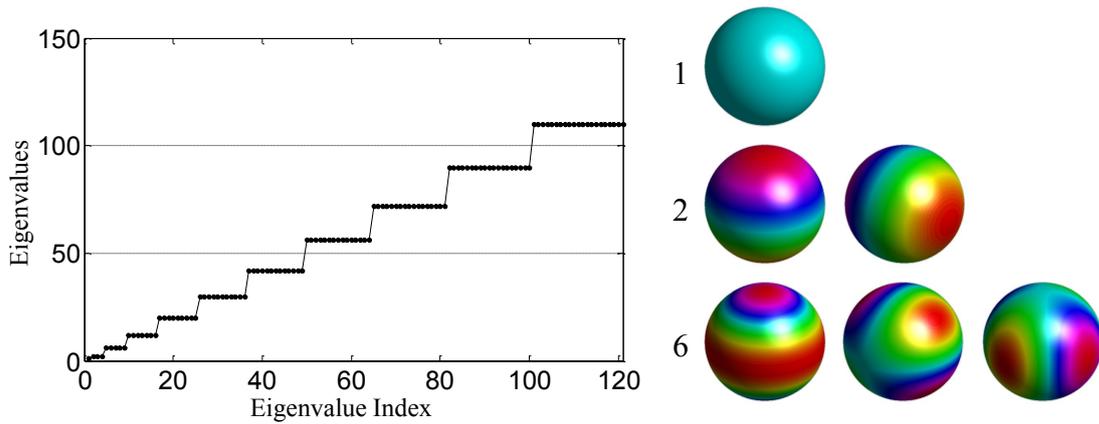


**Figure 4.1:** First 30 eigenvalues and eigenfunctions of the square. The eigenvalues for the eigenfunctions on the first row are also indicated on the upper-left corner of each eigenfunction.

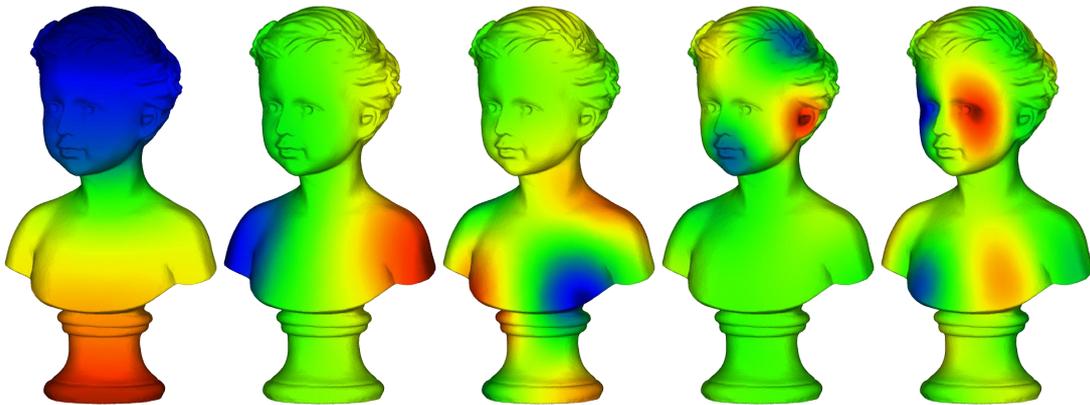
### 4.1.3 The Heat Kernel

The eigenfunctions and eigenvalues of the Laplace-Beltrami operator can be used to construct other useful operators. One prominent example of

## Spectral Sampling of Manifolds



**Figure 4.2:** First 120 eigenvalues of the sphere, and some eigenfunctions corresponding to  $\lambda = 1, 2, 6$ .



**Figure 4.3:** Some eigenfunctions on a surface for increasing eigenvalues from left to right.

such a construct follows from the diffusion equation on the manifold. This differential equation explains how heat is diffused on the manifold and thus has a very intuitive behavior. As diffusion progresses, the heat becomes more and more evenly distributed. This results in a natural multi-resolution structure parametrized by the time of the diffusion.

In mathematical terms, diffusion of heat on a manifold is described by the equation (for a detailed account on the heat kernel and its properties, please see e.g. [Grigor'yan, 1998])

$$\Delta_M f(x, t) = -\frac{\delta f(x, t)}{\delta t}. \quad (4.3)$$

The solution of this equation can be expressed in terms of an operator applied

## 4.1 Spectral Analysis of Manifolds

to the initial heat distribution at time  $t = 0$ . Starting from this initial heat distribution  $f(x,0) = f(x)$ , the distribution at time  $t$ ,  $f(x,t)$ , is given by applying the *heat operator* to the initial distribution as  $H_t f(x)$ . This operator spreads the heat distribution over the manifold while respecting its intrinsic structure.

It can be shown [Belkin and Niyogi, 2006] that for small  $t$ , the heat operator converges to the Laplace-Beltrami operator as  $\Delta_M = \lim_{t \rightarrow 0} \frac{I - H_t}{t}$ . Hence, the two operators have the same eigenfunctions and for short time, eigenvalues of  $H_t$  can be used to approximate those of  $\Delta_M$ .

Associated with the heat operator, there is the *heat kernel*  $h_t(x,y) : M \times M \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$  that satisfies

$$H_t f(x) = \int_M h_t(x,y) f(y) dy, \quad (4.4)$$

where  $f : M \rightarrow \mathbb{R}$  and  $dy$  is the volume form on  $M$ . Hence, intuitively the operator  $H_t$  smoothes functions on the manifold by convolving them with the heat kernel, and  $h_t(x,y)$  measures the amount of heat transferred from a unit source at  $x$  to  $y$  at time  $t$ . Depending on the geometry of the manifold, the rate of heat transfer and hence the distribution of heat will change. Note that once the heat kernel is known, the solution to the diffusion equation is also known.

Among other interesting properties, the most important property of the heat kernel for our purposes is the expansion of it in terms of the eigenfunctions and eigenvalues of the Laplace-Beltrami operator:

$$h_t(x,y) = \sum_{i=0}^{\infty} e^{-\lambda_i t} u_i(x) u_i(y). \quad (4.5)$$

Note that an immediate consequence of this property is that the heat kernel is symmetric, and can be written as a dot product  $h_t(x,y) = \phi_t(x)^T \phi_t(y)$ , where  $\phi_t(x)$  is a vector such that the  $i^{\text{th}}$  component is given by  $\sqrt{e^{-\lambda_i t}} u_i(x)$ <sup>1</sup>. This interpretation allows us to define a new space where these vectors live and establishes a direct connection to the kernel techniques, as explained in the next section.

---

<sup>1</sup>For technical reasons, it is convenient to assume that the vector is finite dimensional. This is not a limitation, since for all  $t > 0$ , one can safely truncate the series at  $n < \infty$  without significantly affecting the accuracy, as is done in practice.

## 4.2 Kernel Techniques

Kernel techniques have been in use in meshless methods, probabilistic models, and machine learning for decades [Scholkopf and Smola, 2001; Schaback and Wendland, 2006; Hofmann et al., 2008]. They are used to approximate functions, provide trial spaces for solving partial differential equations, define covariances in probabilistic models, and form the base for many successful learning algorithms. The basic idea of all kernel methods starts with mapping the data to another space, a *feature space* with a useful structure for the problem at hand. The power of the methods stems from the representation of this space. In fact, the space itself is not explicitly represented. It is only known via an inner product. This product is given by the kernel function, which can be evaluated very efficiently. Hence, every algorithm that only depends on inner products can be applied to the vectors in the feature space to develop non-linear techniques. This idea is called the *kernel trick* in the machine learning literature.

As explained in Chapter 2, our methods are also based on kernels. In this chapter, we concentrate on a specific class of kernels, the *positive definite kernels* and in particular *Mercer kernels*, and relate it to the heat kernel of manifolds. This treatment of the heat kernel forms the basis of our spectral measures for sampling. Before explaining these in detail in further sections, we would like to explain the relevant facts and theorems on kernel techniques in this section.

Central to all kernel methods is the kernel function. This is a symmetric function of the form  $k(x, y) : \Omega \times \Omega \rightarrow \mathbb{R}$  for  $x, y \in \Omega$  with  $\Omega$  some space, and  $k(x, y) = k(y, x)$ . This function is assumed to correspond to an inner product in some *feature space*  $\mathcal{F}$ :

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{F}} \quad (4.6)$$

for the *feature map*  $\phi(x) \in \mathcal{F}$ . Typically, the relation between the spaces  $\Omega$  and  $\mathcal{F}$  is very complex. However, the simple form of the kernel provides an easy and efficient way to access the space.

Characterizing kernels of the form in equation 4.6 has been well-studied and it has been shown that the kernels that can be written as an inner product should necessarily have the *positive definiteness* property. We define this property and its consequences in the next section.

### 4.2.1 Positive Definite Kernels

The definition of a positive definite kernel follows from that of a positive definite matrix. For some given points  $x_i \in \Omega$ , the matrix with the entries

$$\mathbf{K}_{ij} = k(x_i, x_j) \quad (4.7)$$

is called the *Gram matrix* of the symmetric kernel function  $k$ . If

$$\sum_{ij} c_i c_j \mathbf{K}_{ij} \geq 0 \quad (4.8)$$

for all  $c_i \in \mathbb{R}$  and for all sets of distinct  $x_i$ , then the kernel  $k$  is called a *positive definite kernel*. Although it might seem difficult to check if a kernel is positive definite or not due to the arbitrariness of the points  $x_i$ , in practice, there exist several easy ways to construct such kernels. An important construction uses the property of positive definite kernels that states sums and multiplications of positive definite kernels are positive definite, such that if  $k_1(x, y)$  and  $k_2(x, y)$  are positive definite, then so are  $k_1(x, y)k_2(x, y)$  and  $\alpha_1 k_1(x, y) + \alpha_2 k_2(x, y)$  for  $\alpha_1, \alpha_2 \geq 0$ .

Note that the kernels used for regression or density estimation in statistics do not in general have this property. However, we utilize computational approximations of positive definite kernels in all our methods throughout this work (such as for local kernel regression as explained in Section 3.2.2, or for density estimation in Section 5.3.2). Although this restriction to the class of positive definite kernels might seem as a limitation, it actually provides a quite flexible method to define kernels with useful structure in vastly different spaces and forms. We present some important examples of positive definite kernels below.

**Inner Product Kernels.** An important observation is that all inner product kernels (that can be written in the form of equation 4.6) are positive definite. To see this, one can simply expand the expression in equation 4.8 as follows

$$\begin{aligned} \sum_{ij} c_i c_j \mathbf{K}_{ij} &= \sum_{ij} c_i c_j \langle \phi(x_i), \phi(x_j) \rangle \\ &= \langle \sum_i c_i \phi(x_i), \sum_j c_j \phi(x_j) \rangle \\ &= \left\| \sum_i c_i \phi(x_i) \right\|^2 \geq 0. \end{aligned}$$

This is a direct result of the properties of the inner product space  $\mathcal{F}$ . Going in the other direction, all positive definite kernels are inner product kernels, where the feature space can be set to the reproducing kernel Hilbert space of the kernel, as we will explain in Section 4.2.2. The equivalence of positive definiteness and inner product form opened the way to design a vast variety of kernels in different domains. The examples below can all be seen as special cases of inner product kernels.

**Polynomial Kernels.** A simple class of positive definite kernels is formed by  $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y})^p$  for  $p \in \mathbb{N}$  and  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ . An interesting property of this simple kernel is that a feature map  $\phi(\mathbf{x})$  of the kernel can be explicitly constructed.

**The Gaussian Kernel.** This is perhaps the most widely used kernel in many fields. It is given as

$$k(x, y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}, \quad (4.9)$$

where  $\|x\|$  denotes the Euclidean norm and  $\sigma$  is a parameter. The Gaussian kernel is widely used in local approximations via local kernel regression and MLS surface definitions, as explained in Sections 3.2 and 3.4. It is also used for the approximations in Chapters 3 and 5. Hence, it forms an interesting link between approximation methods and kernel techniques as detailed in Chapter 2. In addition, it is also the heat kernel of the Euclidean space (see Section 4.1.3).

**Diffusion Kernels.** A useful property of positive definite kernels is that if  $k(x, y)$  is a kernel, then  $f(x)k(x, y)f(y)$  is also a kernel. To see this, one can first prove that the kernel  $f(x)f(y)$  is positive definite by writing  $\sum f(x_i)f(x_j)c_i c_j = \sum f(x_i)c_i \sum f(x_j)c_j = (\sum f(x_i)c_i)^2 \geq 0$ . Since multiplication of two positive definite kernels is also positive definite, the result follows. In particular, if the space  $\Omega$  is equipped with a measure  $\mu$ , the following is a positive definite kernel

$$\frac{k(x, y)}{d^\alpha(x)d^\alpha(y)}, \quad (4.10)$$

where  $d(x) = \int_\Omega k(x, y)\mu(y)$ . This kernel is used to approximate diffusion operators on manifolds, regardless of the distribution of points [Coifman and Lafon, 2006]. We will also use it in our approximations of the heat kernel in Section 4.3.3.

**Graph Kernels.** The generality of the kernel construction allows us to define kernels on more abstract structures such as graphs. To define a positive definite kernel, first a graph Laplacian is constructed. Assuming that we have an undirected graph, the normalized graph Laplacian is given by the expression

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}, \quad (4.11)$$

where  $\mathbf{A}$  is the adjacency matrix, or a similarity matrix for the vertices of the graph. The kernel matrix can then be constructed in several ways based on  $\mathbf{L}$ . One way is mimicking the relation between the heat operator and the Laplace-Beltrami operator of a manifold and taking  $K = e^{-t\mathbf{L}}$  for some  $t \geq 0$  [Kondor and Lafferty, 2002]. Intuitively, this matrix describes a diffusion process among the vertices of the graph. If one assumes that the graph is defined on a point cloud lying near a manifold and Gaussian weights are used for the similarity matrix such that  $\mathbf{A}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  with  $k$  defined as the Gaussian kernel, the matrix  $\mathbf{L}$ , its unnormalized version  $\mathbf{D} - \mathbf{A}$ , or other variants can be used to approximate the heat kernel matrix of the underlying manifold [Belkin and Niyogi, 2006; Coifman and Lafon, 2006].

#### 4.2.2 Reproducing Kernel Hilbert Spaces

A fundamental property of positive definite kernels is that they are associated with a Hilbert space called the *reproducing kernel Hilbert space* (RKHS) that we denote by  $\mathcal{H}$ . One way to specify this space is starting with a function  $k_x(y) = k(x, y)$  for each  $x \in \Omega$ . Then one can define the RKHS for the kernel  $k$  as

$$\mathcal{H} = \overline{\text{span}\{k_x : x \in \Omega\}}. \quad (4.12)$$

Here,  $\overline{\mathcal{S}}$  denotes the completion of the space  $\mathcal{S}$ , which is formed by adding the limit points of the Cauchy sequences (see for e.g. [Schaback and Wendland, 2006]).

This space has an associated inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  such that for every  $x \in \Omega$  and  $f \in \mathcal{H}$ , the following is true

$$f(x) = \langle k_x, f \rangle_{\mathcal{H}}. \quad (4.13)$$

Thus, the kernel  $k$  reproduces the functions in the Hilbert space  $\mathcal{H}$ , hence the name reproducing kernel Hilbert space. In particular, since  $k_y \in \mathcal{H}$ , one can write

$$\langle k_x, k_y \rangle_{\mathcal{H}} = k_y(x) = k(x, y). \quad (4.14)$$

Thus, the inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  is given by the kernel function  $k$ . This also implies that the norm is given by  $\|k_x\|_{\mathcal{H}} = \sqrt{\langle k_x, k_x \rangle_{\mathcal{H}}} = \sqrt{k(x, x)}$ .

This form shows that every positive definite kernel can be written as an inner product in some feature space by setting  $\mathcal{F} = \mathcal{H}$ . In general,  $\mathcal{F}$  and  $\mathcal{H}$  need not be the same. To see this, consider a space  $\mathcal{F}$ , such as the Euclidean space, and the implied kernel of the form  $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{F}}$ , as in equation 4.6. For this kernel, one can either operate in the space  $\mathcal{F}$ , or the associated RKHS  $\mathcal{H}$ . This, however, does not result in any change in the kernel based algorithms, since the value of the inner product, given by the kernel, stays the same regardless of the interpretation of the kernel.

### 4.2.3 Mercer's Theorem and the Heat Kernel

In this section, we will discuss a special type of positive definite kernels, given by an explicit expansion formula. In the following sections, this will allow us to treat the heat kernel (see Section 4.1.3) in the framework of kernel techniques. We first state a slightly specialized form of the Mercer's theorem:

**Mercer's Theorem.** *Assume that  $\Omega \in \mathbb{R}^n$  is closed,  $k(x, y) : \Omega \times \Omega \rightarrow \mathbb{R}$  is a continuous positive definite kernel, and  $\mu$  is a strictly positive Borel measure on  $\Omega$ . If the following is true*

$$\int_{\Omega} \int_{\Omega} k(x, y)^2 d\mu(x) d\mu(y) < \infty,$$

*then the kernel can be expanded as*

$$k(x, y) = \sum_{i=0}^{\infty} \lambda_i \phi_i(x) \phi_i(y),$$

*where  $\lambda_i \geq 0$ ,  $\sum_{i=1}^{\infty} \lambda_i^2 < \infty$ , and  $\phi_i$  are normalized with the norm in  $L^2_{\mu}(\Omega)$ .*

Here,  $L_\mu^2(\Omega) = \{f : \Omega \rightarrow \mathbb{R} : \int |f(x)|^2 d\mu(x) < \infty\}$ . Note that associated with such a kernel, there is also the integral operator acting on the functions in  $L_\mu^2(\Omega)$ :

$$Kf(x) = \int_{\Omega} k(x,y)f(y)d\mu(y). \quad (4.15)$$

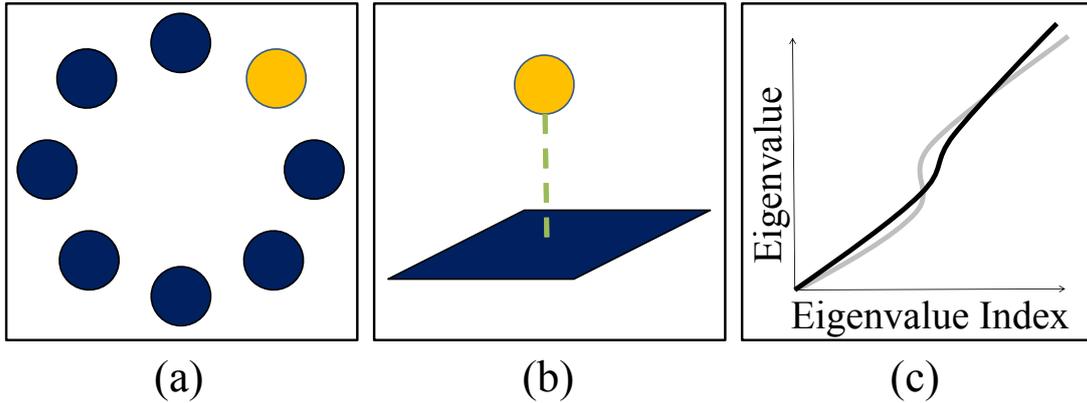
This is a compact, positive, and self-adjoint operator with the eigenfunctions  $\phi_i$  and eigenvalues  $\lambda_i$ . Hence, the eigenfunctions  $\phi_i$  form an orthonormal basis for the space  $L_\mu^2(\Omega)$ , according to the Hilbert–Schmidt theorem.

The importance of Mercer’s theorem for our purposes stems from the expansion of the kernel  $k$ . This expansion allows us to construct an explicit feature space for  $k$  by writing

$$\phi(x) = \left[ \sqrt{\lambda_1}\phi_1(x) \cdots \sqrt{\lambda_{n_T}}\phi_{n_T}(x) \right]^T. \quad (4.16)$$

For technical reasons, we used an approximation of the expansion by truncating it at the  $n_T$ th term. This is not a limitation since the series converges uniformly (e.g. [Scholkopf and Smola, 2001; Minh et al., 2006]) and hence arbitrary accuracy can be obtained by choosing a big enough  $n_T < \infty$ . It is easy to see that the usual inner product of Euclidean spaces gives the kernel  $k(x,y) = \phi(x)^T \phi(y)$ .

A very interesting observation is that *the heat kernel* (see Section 4.1.3) of a manifold also has a similar expansion and thus a feature space for it can be defined accordingly. This provides a fundamental bridge between the spectral analysis methods and kernel methods. We will use this link to develop our measures in the next section.



**Figure 4.4:** *Conceptual Overview: (a) Given a set of points, (b) we map them to a higher dimensional space implied by the heat kernel of the underlying manifold such that the distance of a point to the span of others can be used to measure the (c) influence of that point on the Laplace-Beltrami spectrum.*

### 4.3 Spectral Measures for Manifold Sampling

The essence of our method is measuring the effect of a point on the global properties of the manifold using the Laplace-Beltrami spectrum, based on the link between spectral and kernel methods. We already established this link in the previous sections, and now move on to the derivation of our measure.

Ideally, our measure quantifies the change in the Laplace-Beltrami spectrum of a manifold due to a single point. It is well-known that eigenvalues of the Laplace-Beltrami operator provide an almost unique identification of the manifold up to isometry [Reuter et al., 2006; Rustamov, 2007]. Although there exist isospectral manifolds (i.e. sharing the same spectrum) that are not isometric, these cases are very rare, and many geometric and topological properties of a manifold can be extracted from the spectrum [Kesavan, 1998; Lévy, 2006]. The eigenspectrum is also stable under perturbations of the manifold, thus similar manifolds will have close spectra [Dey et al., 2010] and changes in the implicit structure of the manifold will be reflected to the spectrum. We utilize these facts to measure changes in a manifold through changes in its spectrum. Measuring this change due to a single point is performed by considering the high dimensional feature space implied by the heat kernel (Figure 4.4, see also Section 4.2.3).

As stated earlier, a sampling should necessarily be coupled to reconstructions. We take special care in approximating the sampling measure such that the resulting measure is closely linked to kernel based reconstruction methods, in particular the moving least squares based reconstruction method we de-

veloped in Chapter 3. To be useful in practical applications, it should also be efficiently computable, which we achieve by developing local approximations. In this section, we show how to derive a measure exhibiting these properties. First, we present our choice of the discrete approximation for the spectrum of the Laplace-Beltrami operator via the heat kernel. Next, we derive our measure in terms of the heat kernel, and finally show how to compute it with appropriate approximations.

#### 4.3.1 Discretization of the Laplace-Beltrami Operator

In order to compute the spectrum of the Laplace-Beltrami operator in practice, we need to discretize the operator into a matrix. For samples forming a simplicial mesh, several discrete approximations with nice invariance and convergence properties have been derived [Pinkall and Polthier, 1993; Cohen-Steiner and Morvan, 2006; Wardetzky et al., 2007; Bobenko and Springborn, 2007; Bauer et al., 2009]. However, during sampling or re-sampling we wish to avoid repeatedly constructing such a mesh, as it can be a challenging problem in itself. Even if we ignore the meshing problem, these mesh based approximations do not allow us to establish a direct connection to the kernel based reconstruction methods, which is one of our goals. For these reasons, we rather follow the line of approximations motivated by methods in manifold learning, spectral graph theory, and computational harmonic analysis [Belkin and Niyogi, 2006; Belkin et al., 2009; Coifman and Lafon, 2006]. These approaches rely on the connection between the Laplace-Beltrami operator and the heat operator.

To compute an approximation of the heat operator, we can discretize the integral defining the heat operator (equation (4.4)) to get

$$(\mathbf{H}_t \mathbf{f})_i = \sum_j h_t(x_i, x_j) \mathbf{f}_j, \quad (4.17)$$

where  $\mathbf{f}_j = f(x_j)$ . The entries of this heat kernel matrix are thus given by  $(\mathbf{H}_t)_{ij} = h_t(x_i, x_j)$ . By expanding the heat kernel in terms of the eigenfunctions and eigenvalues of the Laplace-Beltrami operator (as in equation 4.5), it can be shown [Braun, 2006; von Luxburg, 2004] that the eigenvalues of the matrix  $\mathbf{H}_t$  converge to  $e^{-\lambda_i t}$  up to a constant, assuming the eigenvectors of  $\mathbf{H}_t$  are good approximations to the eigenfunctions of  $H_t$ , which is the case as the sample set becomes denser. For small  $t$ , this implies that the eigenvalues of  $\mathbf{H}_t$  approximate those of the Laplace-Beltrami operator well (see also Section 4.1.3).

Concluding, the eigenvalues of the Laplace-Beltrami operator can be approximated by computing the eigenfunctions of the discrete heat operator, i.e. the matrix  $\mathbf{H}_t$ . It remains to measure the change in the spectrum of  $\mathbf{H}_t$ .

### 4.3.2 Spectral Change due to a Point

Given points  $x_i$  on the manifold that are used to form  $\mathbf{H}_t$ , we seek to quantify the change in the spectrum of  $\mathbf{H}_t$  due to the addition of a new point  $x$ . We reformulate this problem in another space and use matrix perturbation theory and properties of the eigenfunctions of the heat kernel to arrive at our measure.

As explained in Section 4.2, for every symmetric positive definite kernel, there exist feature spaces where the inner products are given by the kernel, and the expansion of the heat kernel allows us to use a similar approach by writing it as an inner product  $h_t(x, y) = \phi_t(x)^T \phi_t(y)$  in a high dimensional feature space. For brevity, we will drop the time dependency from all identities and write  $\phi_i = \phi_t(x_i)$  for the rest of this section. We also denote eigenvalues of a matrix  $\mathbf{M}$  as  $\lambda_i(\mathbf{M})$ .

With these definitions, heat kernel matrix can be written as  $\mathbf{H}_{ij} = h(x_i, x_j) = \phi_i^T \phi_j$ . More interestingly, we can define the covariance matrix  $\mathbf{C} = \sum \phi_i \phi_i^T$ . Non-zero eigenvalues  $\lambda_i(\mathbf{C})$  of  $\mathbf{C}$  are the same as those of  $\mathbf{H}$  [Schölkopf et al., 1998]. Thus we can equivalently consider the change in the spectrum of  $\mathbf{C}$ .

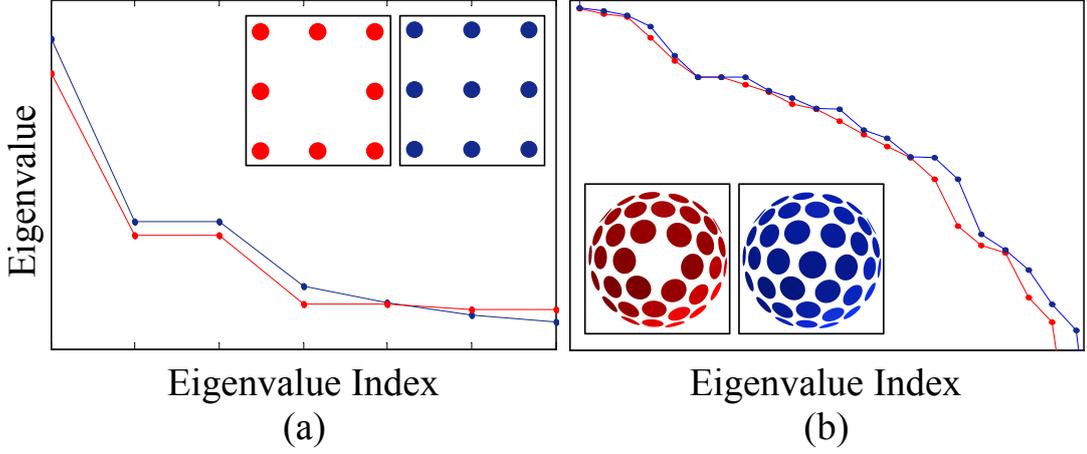
Adding a sample  $x$  to the point set means forming a new covariance matrix  $\mathbf{C}' = \mathbf{C} + \phi\phi^T$  with  $\phi = \phi(x)$ . The vector  $\phi$  can be written as  $\phi = \mathbf{r} + \mathbf{o}$  for the projection  $\mathbf{r}$  onto the span of  $\phi_i$ 's and the orthogonal component  $\mathbf{o}$ . Thus we can expand the expression for  $\mathbf{C}'$  as

$$\mathbf{C}' = \mathbf{C} + (\mathbf{r} + \mathbf{o})(\mathbf{r} + \mathbf{o})^T = \mathbf{C} + \mathbf{o}\mathbf{o}^T + \mathbf{E}. \quad (4.18)$$

The matrix  $\mathbf{C} + \mathbf{o}\mathbf{o}^T$  has eigenvalues  $\lambda_i$  and  $\|\mathbf{o}\|^2$ . We want to study how the eigenspectrums of  $\mathbf{C} + \mathbf{o}\mathbf{o}^T$  and  $\mathbf{C}'$  differ.

Let  $\mathbf{v}_i$  denote an eigenvector of  $\mathbf{C}$  corresponding to the eigenvalue  $\lambda_i(\mathbf{C})$ . The change in the eigenvalues can be studied using the following result from matrix perturbation theory [Ipsen and Nadler, 2009]:

$$\min_j |\lambda_i(\mathbf{C} + \mathbf{o}\mathbf{o}^T) - \lambda_j(\mathbf{C}')| \leq \|\mathbf{E}\mathbf{v}_i\|^2. \quad (4.19)$$



**Figure 4.5:** Effect of adding a point to the point set on the eigenvalues of the heat kernel matrix for (a)  $\mathbb{R}^2$  and (b) sphere (zoomed).

Substituting the expression for  $\mathbf{E}$ , the bound can be computed as

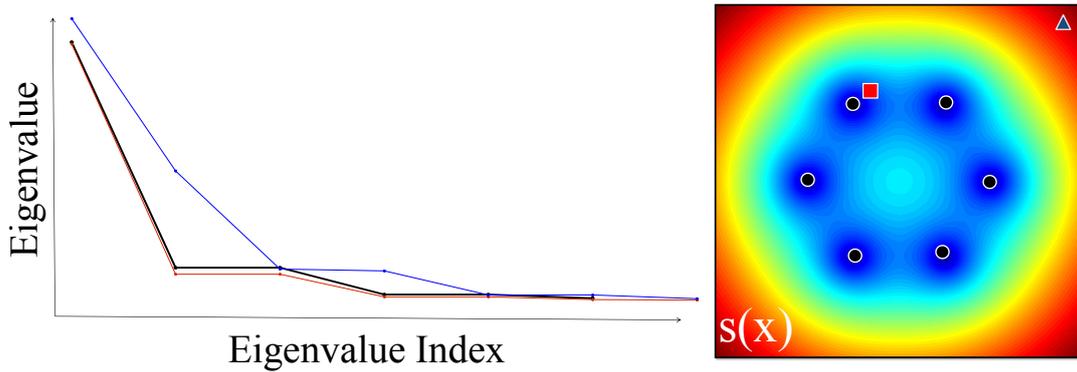
$$\|\mathbf{E}\mathbf{v}_i\|^2 = \|(\phi\mathbf{r}^T + \mathbf{r}\mathbf{o}^T)\mathbf{v}_i\|^2 = \|\phi\|^2(\mathbf{r}^T\mathbf{v}_i)^2,$$

where we used the fact that  $\mathbf{o}$  is orthogonal to  $\mathbf{v}_i$ . Thus the change in  $\lambda_i(\mathbf{C})$  due to adding the point  $x$  to the set depends on  $\delta_i = (\mathbf{r}^T\mathbf{v}_i)^2 = (\phi^T\mathbf{v}_i)^2$ . In Appendix A.3, we derive that if the eigenvectors of  $\mathbf{H}$  well approximate eigenfunctions of the heat operator, the change in the spectrum will diffuse to many eigenvalues and effect of a point on the spectrum will not be significant. Examples of this behavior are shown in Figure 4.5 for the cases of  $\mathbb{R}^2$  and sphere.

This analysis implies that if  $\sum\delta_i = \|\mathbf{r}\|^2$  is close to  $\|\phi\|^2$ ,  $x$  will not disturb the distribution of the eigenvalues considerably. On the other hand, if  $\|\mathbf{o}\|^2$  is large, then the spectrum will have a new large eigenvalue, which will alter the distribution of the eigenvalues. Thus we define our measure as  $\|\mathbf{o}\|^2/\|\phi\|^2 \in [0,1]$ . Although one could define other measures based on the individual bounds  $\delta_i$ , this particular choice results in very efficient computations as will be shown in the next section.

In Appendix A.4, we show that this measure that quantifies the change in the spectrum due to the addition of the point  $x$  can be computed as

$$s(x) = 1 - \mathbf{h}^T\mathbf{H}^{-1}\mathbf{h}/h(x,x), \quad (4.20)$$



**Figure 4.6:** On the left, black curve shows the spectrum of the heat kernel matrix and on the right, the measure  $s(x)$  is plotted for the six black points and the heat kernel of  $\mathbb{R}^2$ . When the blue point (represented with a small triangle) in a region of high  $s(x)$  is added to the set of black points, the eigenspectrum becomes the blue curve, and hence changes significantly. In contrast, the red point (represented with a small square) does not cause much change. The measure  $s(x)$  correctly captures this behavior.

where  $\mathbf{h}_i = h(x, x_i)$ . We illustrate that  $s(x)$  correctly captures changes to the spectrum in Figure 4.6 for two cases where explicit expressions for the heat kernel is available. If  $s(x)$  is low at a location, placing a new sample at that location does not change the spectrum significantly. In particular, it is easy to see that if  $x = x_i \exists i$ , then  $s(x) = 0$ . This is in accordance with the expectation that adding a point that is already in the point set does not give any new information about the spectrum.

Note that since we do not know the heat kernel of a given manifold, a direct computation of this expression is not possible. In the next section, we utilize approximation methods from spectral processing of graphs to arrive at our final measure.

### 4.3.3 Computation of the Measure

For a given set of points  $\{\mathbf{x}_i \in \mathbb{R}^d\}_1^n$  lying near a manifold, graph based methods try to approximate the Laplace-Beltrami operator of the manifold with that of the weighted graph constructed from the point set. It has been shown that under the assumption that the points are sampled according to a uniform probability distribution on the manifold, this approximation converges to the Laplace-Beltrami operator as the sample set becomes denser [Belkin and Niyogi, 2006; Coifman and Lafon, 2006]. One of the fundamental observations in their proof is that heat kernel operator of the manifold is well-

### 4.3 Spectral Measures for Manifold Sampling

approximated by the matrix  $\mathbf{D}^{-1}\mathbf{K}$ , for a kernel function  $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$  encoding similarity between points, the kernel matrix  $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  and the diagonal matrix  $\mathbf{D}$  with entries  $\mathbf{D}_{ii} = \sum_{j=1}^n \mathbf{K}_{ij}$ . Hence, spectrum of the matrix  $\mathbf{D}^{-1}\mathbf{K}$  provides an approximation for that of the heat kernel matrix  $\mathbf{H}$ .

It is easy to show that the matrices  $\mathbf{D}^{-1}\mathbf{K}$  and  $\mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}$  share the same eigenvalues. Since we are only interested in the eigenvalues, we can equivalently use the latter in our analysis. Let us call the latter matrix as  $\tilde{\mathbf{H}}$ . Although we could directly substitute  $\tilde{\mathbf{H}}$  for  $\mathbf{H}$  in the definition of  $s(x)$ , it is not clear how to define  $\mathbf{h}_i = h(x, x_i)$ .

Instead, we consider the continuous kernel from which the entries of  $\tilde{\mathbf{H}}$  can be generated by sampling and discretization [von Luxburg, 2004; Coifman and Lafon, 2006],  $\tilde{h}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{y}) / \sqrt{d(\mathbf{x})d(\mathbf{y})}$  where  $d(\mathbf{x}) = \int_{\mathbb{R}^d} k(\mathbf{x}, \mathbf{y}) d\mu(\mathbf{y})$  for some measure  $\mu$  (this is a positive definite kernel, as explained in Section 4.2.1). Let us define the diagonal matrix  $\bar{\mathbf{D}}_{ii} = d(\mathbf{x}_i)$  and the vector  $\mathbf{k} = [k(\mathbf{x}, \mathbf{x}_1) \cdots k(\mathbf{x}, \mathbf{x}_n)]^T$ . By direct substitution of  $\tilde{h}$  for  $h$  in the expression for  $s(x)$  (equation (4.20)), we can compute the following expression

$$\tilde{s}(\mathbf{x}) = 1 - \left( \frac{1}{\sqrt{d(\mathbf{x})}} \mathbf{k}^T \bar{\mathbf{D}}^{-1/2} \bar{\mathbf{D}}^{-1/2} \mathbf{K}^{-1} \bar{\mathbf{D}}^{1/2} \bar{\mathbf{D}}^{-1/2} \mathbf{k} \frac{1}{\sqrt{d(\mathbf{x})}} \right) / \frac{k(\mathbf{x}, \mathbf{x})}{d(\mathbf{x})},$$

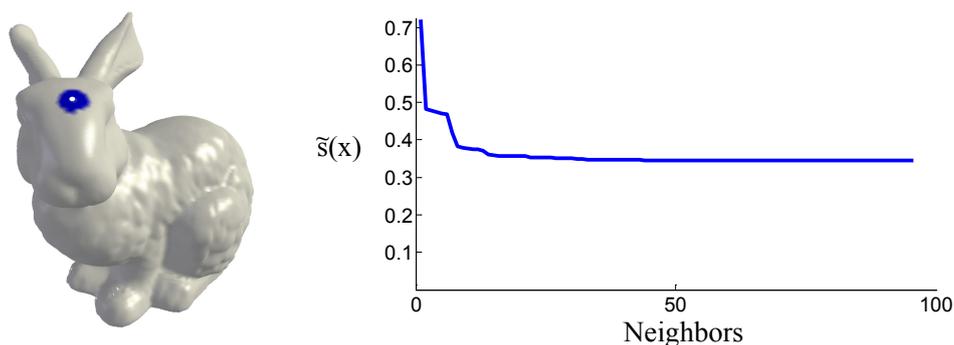
which leads to the final expression for our measure

$$\tilde{s}(\mathbf{x}) = 1 - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} / k(\mathbf{x}, \mathbf{x}). \quad (4.21)$$

The measure  $\tilde{s}(\mathbf{x})$  can be easily computed once the kernel  $k$  is decided upon. However, for arbitrary kernels, computation of it will involve inversion of the  $n \times n$  matrix  $\mathbf{K}$ . Since  $\tilde{s}(\mathbf{x})$  measures contribution of a single point  $\mathbf{x}$  to the spectrum, this inversion will be performed many times, which will make the algorithms utilizing it very inefficient. Fortunately, this global matrix can be substituted by a local matrix constructed from the neighbors of  $\mathbf{x}$  for a specific class of kernels.

Assume that  $k$  is a positive definite and symmetric kernel. Then,  $k$  can be written as an inner product  $k(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x})^T \varphi(\mathbf{y})$  (see Section 4.2). As derived in Appendix A.4,  $\mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$  is the norm of the projection of  $\varphi(\mathbf{x})$  onto the span of  $\varphi(\mathbf{x}_i)$ 's. Further assume that  $k$  is approximately locally supported, meaning that  $k(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x})^T \varphi(\mathbf{x}_i) \approx 0$  for  $\|\mathbf{x} - \mathbf{x}_i\| > r$  for a support radius  $r$ . This means that the vectors  $\varphi(\mathbf{x}_i)$  for points  $\|\mathbf{x} - \mathbf{x}_i\| > r$  will be almost

## Spectral Sampling of Manifolds



**Figure 4.7:** Using a Gaussian kernel, the measure  $\tilde{s}(\mathbf{x})$  for the white point converges to a value as more neighboring points are used for the computation. The points in the blue region are sufficient for an almost exact computation, illustrating the local nature of the measure.

orthogonal to  $\varphi(\mathbf{x})$ . Hence, the projection of  $\varphi(\mathbf{x})$  onto the span of all vectors can be well-captured by the projection onto the span of local neighbors of  $\mathbf{x}$ . Thus  $\tilde{s}(\mathbf{x})$  can be well approximated by using  $\mathbf{K}$  and  $\mathbf{k}$  constructed from the neighboring points. An example of this behavior is shown in Figure 4.7. Since number of neighboring points in the support of  $k$  for a point  $\mathbf{x}$  will be much smaller than total number of points in the set, this local computation makes our sampling very efficient.

The measure  $\tilde{s}(\mathbf{x})$  also leads to an essential link between the approximation of the spectrum and kernel based approximation methods that utilize the same kernel function  $k$ , such as the surface reconstruction methods we presented in Chapter 3. In the next section, we make this link more precise.

## 4.4 Coupling Sampling to Reconstruction

For many problems in computer graphics, sampling is followed by reconstruction or inferring new data. A versatile approach to surface reconstruction is using weighted averages of kernels. Prominent examples are radial basis function and moving least squares (MLS) based reconstruction methods. In this section, we show that there is a close link between the kernel based approximation of our measure, and kernel based reconstruction methods. We then explain how our measure can be adapted to work with the MLS based reconstruction method we developed in Chapter 3.

### 4.4.1 Relation to Kernel Regression

Consider a general weighted sum of a positive definite and symmetric kernel  $\sum w_i k_i(\mathbf{x}, \mathbf{x}_i)$ . Since we can write  $k$  as a dot product in the feature space of it, we can rewrite this expression as  $\sum w_i \varphi(\mathbf{x})^T \varphi(\mathbf{x}_i) = \varphi(\mathbf{x})^T \sum w_i \varphi(\mathbf{x}_i)$ .

Thus, this approximation becomes nothing but an inner product of  $\varphi(\mathbf{x})$  with a weighted average of the vectors  $\varphi(\mathbf{x}_i)$ . Similar to our analysis on the heat kernel, we can decompose any arbitrary vector  $\varphi(\mathbf{x}_j)$  into a component  $\mathbf{d}_j$  in the span of other vectors and a component  $\mathbf{o}_j$  orthogonal to the span. Furthermore,  $\mathbf{d}_j$  can be written as a linear combination of others as  $\mathbf{d}_j = \sum_{i \neq j} a_i^j \varphi(\mathbf{x}_i)$ . With these definitions, the expression for the weighted average becomes  $\sum_{i \neq j} w_i \varphi(\mathbf{x}_i) + w_j \mathbf{d}_j + w_j \mathbf{o}_j = \sum_{i \neq j} (w_i + w_j a_i^j) \varphi(\mathbf{x}_i) + w_j \mathbf{o}_j$ .

As proved in Appendix A.4,  $\tilde{s}(\mathbf{x}_j) = \|\mathbf{o}_j\|^2 / \|\varphi(\mathbf{x}_j)\|^2$ . If  $\tilde{s}(\mathbf{x}_j)$  is small, then we can ignore the term  $w_j \mathbf{o}_j$  and the weighted average can be computed by modifying the weights and storing only the vectors  $\varphi(\mathbf{x}_i), i \neq j$ . Hence, small  $\tilde{s}(\mathbf{x})$  implies the kernel  $k(\mathbf{x}, \mathbf{x}_j)$  can be ignored in the reconstruction.

Another important connection is due to the nature of our approximation of the heat kernel with the normalized adjacency matrix of the graph. The decay of the spectrum of this matrix is directly related to how strongly the underlying graph is connected via the edge weights [Coifman and Lafon, 2006]. Since our measure tries to quantify the change a point makes to the spectrum, it implicitly encodes connectivity of the graph. For kernel based reconstruction algorithms to produce accurate results, this connectivity is essential.

This analysis implies that samplings generated by algorithms that utilize the measure  $\tilde{s}(\mathbf{x})$  can be coupled to reconstructions generated by kernel based methods, by using the same kernel for both.

### **4.4.2 Reconstruction Method of Chapter 3**

Although other kernel based methods can also be used for reconstructing manifolds from the samplings obtained, we will focus on the MLS based reconstruction method we introduced in Chapter 3, and adapt our algorithms accordingly.

Specifically, let  $\mathbf{x} = [\mathbf{p}/\sigma_p \ \mathbf{n}/\sigma_n]^T$ , where  $\mathbf{p}$  is the position of a point,  $\mathbf{n}$  is the normal vector at that point, and  $\sigma_p$  and  $\sigma_n$  are user provided smoothness parameters, then the kernel used for sampling and reconstruction is given by  $k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2}$ . Higher values of  $\sigma_p$  leads to smoother surfaces and lower values of  $\sigma_n$  causes more pronounced sharp features. This kernel definition allows us to generate samplings adapted to the MLS based reconstruction method introduced in Chapter 3 by operating on a 2-dimensional manifold embedded in  $\mathbb{R}^6$  [Lai et al., 2007]. When used for reconstructions, it leads to sharp feature and detail preserving surfaces. For a detailed analysis and discussion of the resulting reconstructions, please see Chapter 3.

We will use this reconstruction method and the adapted version of our measure for all results in this chapter.

## 4.5 Algorithms for Sampling

Having derived our measure and demonstrated how it can be efficiently computed, we move on to algorithms utilizing this measure for sampling. The first algorithm is a simple subsampling algorithm developed for out-of-core and efficient sampling of huge datasets. The second one is a gradient ascent procedure that iteratively moves the points to maximize and equalize their contribution to the manifold. The dense point set is first input to the subsampling algorithm. The output point set is then input to the resampling algorithm. The subsampling algorithm is designed to avoid expensive computations that involves all points in the input point set, making the whole sampling process very efficient.

### 4.5.1 Randomized Linear Scan

Point sets acquired from the real world can be very large and thus out-of-core and fast algorithms are needed to sample these datasets effectively for a given detail level. For this reason, we designed a simple linear scan algorithm. The algorithm starts with an empty output point set. It randomly selects a point  $\mathbf{x}_k$  from the input set and computes  $\tilde{s}(\mathbf{x}_k)$  using the already added points to the output set. If  $\tilde{s}(\mathbf{x}_k) > \epsilon$ , then  $\mathbf{x}_k$  is added to the output point set. The algorithm stops when all input points have been considered. A pseudo code of the algorithm is given in Algorithm 1.

```

Input: Initial Point set  $X$ 
Output: Subsampled point set  $O$ 
 $O = \emptyset$ 
while ( $X \neq \emptyset$ )
  remove a random point  $\mathbf{x}_k$  from  $X$ 
  find the set of local neighbors  $N_k$  of  $\mathbf{x}_k$  among the points in  $O$ 
  if ( $N_k \neq \emptyset$ )
    compute  $\tilde{s}(\mathbf{x}_k)$  (see equation (4.21)) using points in  $N_k$ 
  if ( $\tilde{s}(\mathbf{x}_k) > \epsilon$  or  $N_k = \emptyset$ )
    add  $\mathbf{x}_k$  to  $O$ 

```

**Algorithm 1:** Simplification by Randomized Linear Scan

### Iterative Inversion

The main computational burden of the linear scan algorithm comes from finding local neighbors and inverting the local kernel matrix. To keep this cost at a minimum, we propose to use an iterative algorithm. To compute  $\tilde{s}(\mathbf{x}_k)$ , first, the points in the neighborhood of  $\mathbf{x}_k$  are sorted according to their distance to  $\mathbf{x}_k$  in ascending order. This is because points closer to  $\mathbf{x}_k$  will contribute to  $\tilde{s}(\mathbf{x}_k)$  more. Starting from the closest point, at each iteration, a new neighboring point is considered, and the inverse and  $\tilde{s}(\mathbf{x}_k)$  is updated. Since every added point decreases  $\tilde{s}(\mathbf{x}_k)$  (this is because every added point contributes to the span of the vectors  $\varphi$ , see also Section 4.3.3), once  $\tilde{s}(\mathbf{x}_k) \leq \epsilon$ , no further iterations are needed because it is for certain that  $\mathbf{x}_k$  will not be added to the output point set.

To compute the inverse iteratively, one can use the block matrix inversion formula that has been proven to be very effective for similar cases [Moghaddam et al., 2008], which reads as follows for our case:

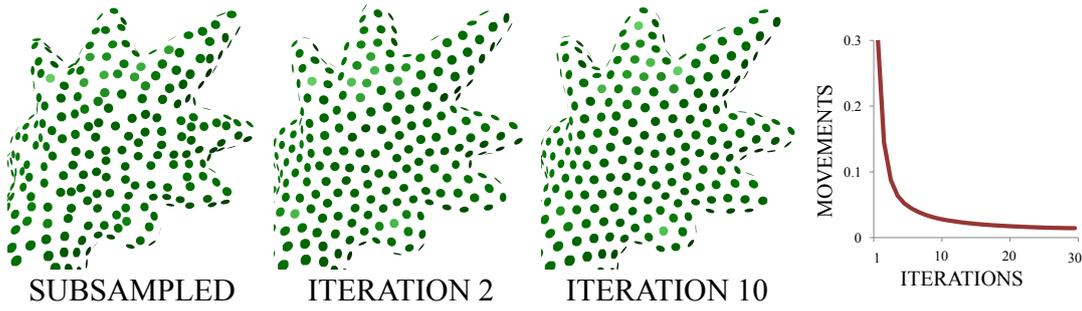
$$\mathbf{K}_{n+1}^{-1} = \begin{bmatrix} \mathbf{K}_n^{-1} + g_n \mathbf{a}_n \mathbf{a}_n^T & -g_n \mathbf{a}_n \\ -g_n \mathbf{a}_n^T & g_n \end{bmatrix}, \quad (4.22)$$

where  $\mathbf{K}_n$  is an  $n$  by  $n$  matrix with elements  $(\mathbf{K}_n)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  for  $i, j \leq n$ ,  $\mathbf{a}_n = \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}_{n+1})$ ,  $g_n = (k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) - \mathbf{k}_n(\mathbf{x}_{n+1})^T \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}_{n+1}))^{-1}$ , and  $(\mathbf{k}_n(\mathbf{x}_{n+1}))_i = k(\mathbf{x}_{n+1}, \mathbf{x}_i)$  for  $i \leq n$ .

Note that in this update rule, instabilities arise when  $g_n^{-1}$  is close to zero. We can safely avoid these cases by omitting the point  $\mathbf{x}_{n+1}$  for which  $g_n^{-1}$  is close to zero since this means the mapping of  $\mathbf{x}_{n+1}$ ,  $\varphi(\mathbf{x}_{n+1})$  (as defined in Section 4.3.3) is almost in the span of already considered  $\varphi(\mathbf{x}_i)$ 's for  $i < n + 1$  and thus do not contribute to the projection of  $\varphi(\mathbf{x})$  onto this span. This allows us to compute the measure even when the kernel matrix is close to singular.

### 4.5.2 Iterative Gradient Ascent

Recall that  $\tilde{s}(\mathbf{x})$  measures the contribution of a point to the manifold definition. Hence, by maximizing and equalizing  $\tilde{s}(\mathbf{x})$  for all points, we can make sure that each point is contributing equally to the surface. In general, this is a difficult non-linear optimization problem involving modifying positions of points to reach a global optimum. Instead of a global minimization, we use



**Figure 4.8:** Points are moved so as to maximize and equalize their contribution to the surface iteratively. Sum of movements of the points decrease quickly.

local operations and move points in a simple gradient ascent iteration on  $\tilde{s}(\mathbf{x})$ . Specifically, at each step, a point  $\mathbf{x}$  is selected, moved to the position

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{1}{2} \nabla \tilde{s}(\mathbf{x}^k) \quad (4.23)$$

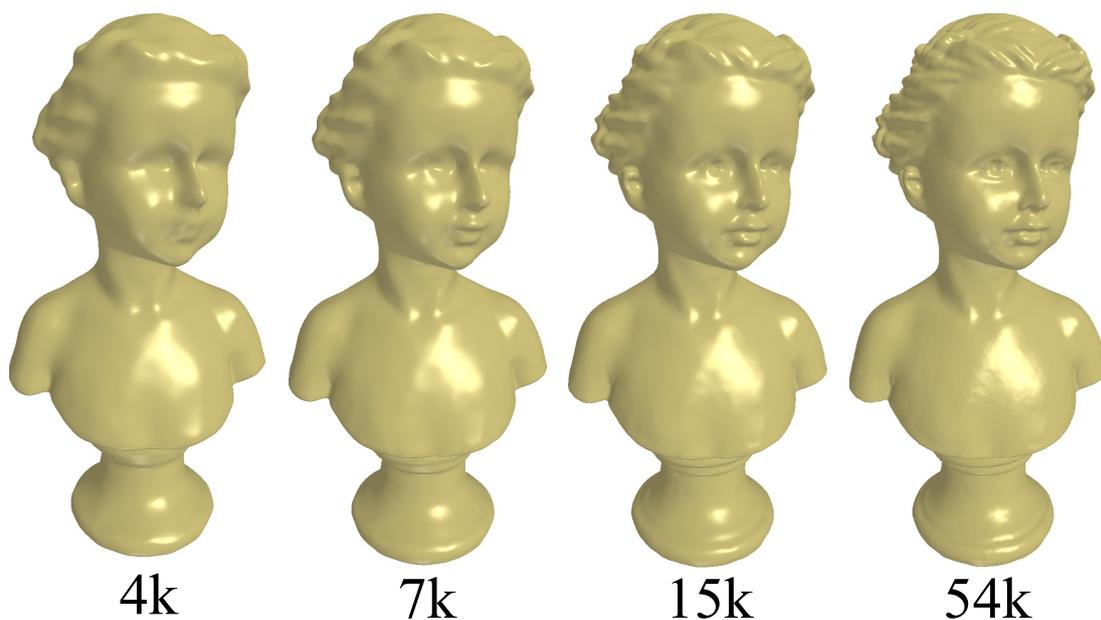
and projected onto the surface. The algorithm then continues with the next point randomly chosen among the points that have not been moved. Once all points are exhausted, the algorithm continues with another iteration until a criterion is met.

In practice, this algorithm converges very fast and produces high quality samplings with blue noise properties. An example resampling process is shown in Figure 4.8. We illustrate and further explain properties of the samplings in Section 4.6. For reference, one can easily compute the gradient for the Gaussian kernel  $k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$  as

$$\nabla \tilde{s}(\mathbf{x}) = (-2/\sigma^2) \sum (2\mathbf{x} - \mathbf{x}_i - \mathbf{x}_j) k(\mathbf{x}, \mathbf{x}_i) k(\mathbf{x}, \mathbf{x}_j) \mathbf{K}_{ij}^{-1}. \quad (4.24)$$

### 4.5.3 Parameters and Data Structures

**Parameters.** For our simplification algorithm (Algorithm 1), the threshold  $\epsilon$  is the only free parameter apart from the kernel parameters. Since  $\tilde{s}(\mathbf{x}) \in [0, 1]$ , we set  $\epsilon = 0.5$  for all results in this paper. For resampling, due to the high convergence rate of the algorithm, we use 10 iterations. According to the decay of the Gaussian  $k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2}$ , the neighborhood size is set to  $r = 2.5$ .



**Figure 4.9:** A multiresolution hierarchy of point set surfaces is obtained by progressive sampling and reconstruction, with indicated number of points.

**Data Structures.** During simplification by the linear scan, points are added one by one to the output point set, and the measure computation for a new point is done using only its neighbors among the points in the output set. Thus we need a dynamic data structure that allows local neighbor retrieval. We used a dynamic kd-tree for these reasons. For the iterative gradient ascent algorithm, the data structure should also allow to alter positions of the points already added. Although a grid or octree can be used efficiently for this case, we chose to simply use a kd-tree for computing indices of neighboring points and assume the same neighborhoods in all iterations. This assumption holds in practice since points are already near optimum positions after the subsampling algorithm and only a few iterations are necessary for the resampling algorithm to converge.

#### 4.5.4 Multi-resolution and Progressive Sampling

The linear scan algorithm can be trivially extended to sample progressively similar to EZW encoding [Shapiro, 1993], such that a series of point sets  $P_i$  with the property that  $P_i \subset P_{i+1}$ ,  $1 \leq i \leq n$  is obtained. In a coarse-to-fine approach, starting with a large smoothing parameter, a first linear scan is performed and the resulting set is defined as  $P_1$ . To get a more detailed version, the smoothing parameter is decreased and another linear scan is

performed on the remaining points in the input point set, adding more points to  $P_1$  to get  $P_2$ .

One can also start with a small smoothing parameter so as to obtain a point set  $P_n$  for a detailed surface, then reduce the parameter and run a linear scan on  $P_n$  to subsample it and get  $P_{n-1}$ , and continue in this fashion until  $P_1$ , the coarsest level is reached. For each  $P_i$ , the surface can be reconstructed using the same smoothing parameters as used for subsampling. An example of such a multiresolution reconstruction is shown in Figure 4.9.

## 4.6 Results and Analysis

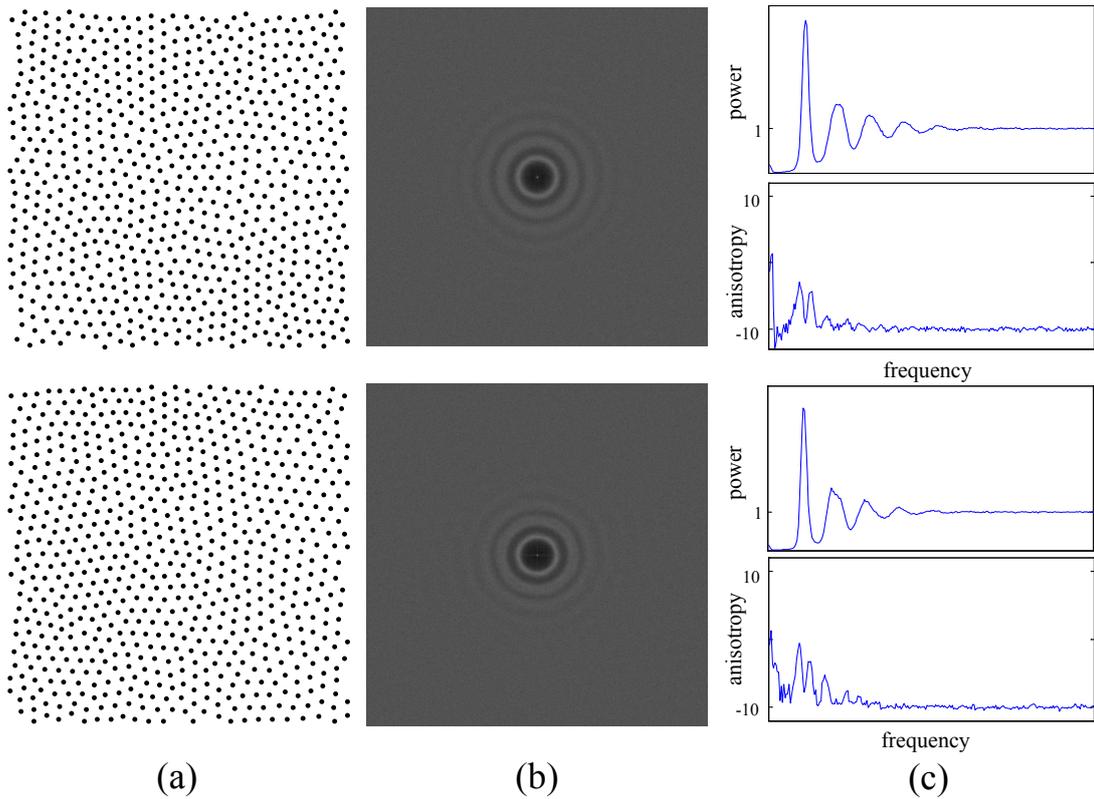
To assess the effectiveness of our algorithms, we test the quality of the samplings, accuracy of the reconstructions resulting from the sampled points, and performance of the algorithms quantitatively in extensive experiments. For all experiments, models are scaled such that their bounding box has a maximum length of 100.

### 4.6.1 Quality of Samplings

To test the quality of the distributions generated by our sampling algorithms, we first show that they possess high quality blue noise characteristics on a toroidal square. We then show that the same characteristics exist when sampling general surfaces. Finally, we illustrate that the algorithms generate state-of-the-art results when applied to the remeshing problem with well-shaped triangles. To our knowledge, our algorithm is the first to generate such high quality remeshing results directly from point samples with little time and space complexity.

**Sampling in  $\mathbb{R}^2$ .** One of the important special cases of isotropic manifold sampling is sampling the plane or a bounded region in  $\mathbb{R}^2$ . For this case, established quantitative measures to assess the quality of distributions exist. In Figure 4.10, we show an example distribution on a toroidal square, mean periodogram [Ulichney, 1987], power and anisotropy plots computed using 10 different random initial distributions for our and Lloyd’s (100 iterations) method. The distributions obtained by our algorithms have characteristics similar to Lloyd’s method, with an average normalized Poisson disk radius of  $\rho = 0.793$ . To obtain this distribution, we start with a random sampling of  $n$  points and resample by gradient ascent for 10 iterations. The width of the Gaussian kernel is set to  $\sigma = 2.5r$ , where  $r = \rho / \sqrt{(2\sqrt{3})n}$  with  $\rho = 0.75$  the optimal Poisson disk radius [Lagae and Dutré, 2008] to ensure there are enough points in the support of the kernel. We can get the same distributions if we start from a dense sampling of the domain, set a kernel width, subsample and then resample using our algorithms.

**Sampling Surfaces.** We tested our sampling algorithms on different models with different parameters. The parameter  $\sigma_p$  can be tuned to get different smoothness and number of points, and  $\sigma_n$  controls the adaptivity of the samplings. Example samplings are shown in Figures 4.12 and 4.11. Setting  $\sigma_n = \infty$ , one can get uniform sampling of the surface with well-distributed points. Lower values of  $\sigma_n$  causes the algorithms place more samples in the curved regions and features, resulting in preservation of details. As



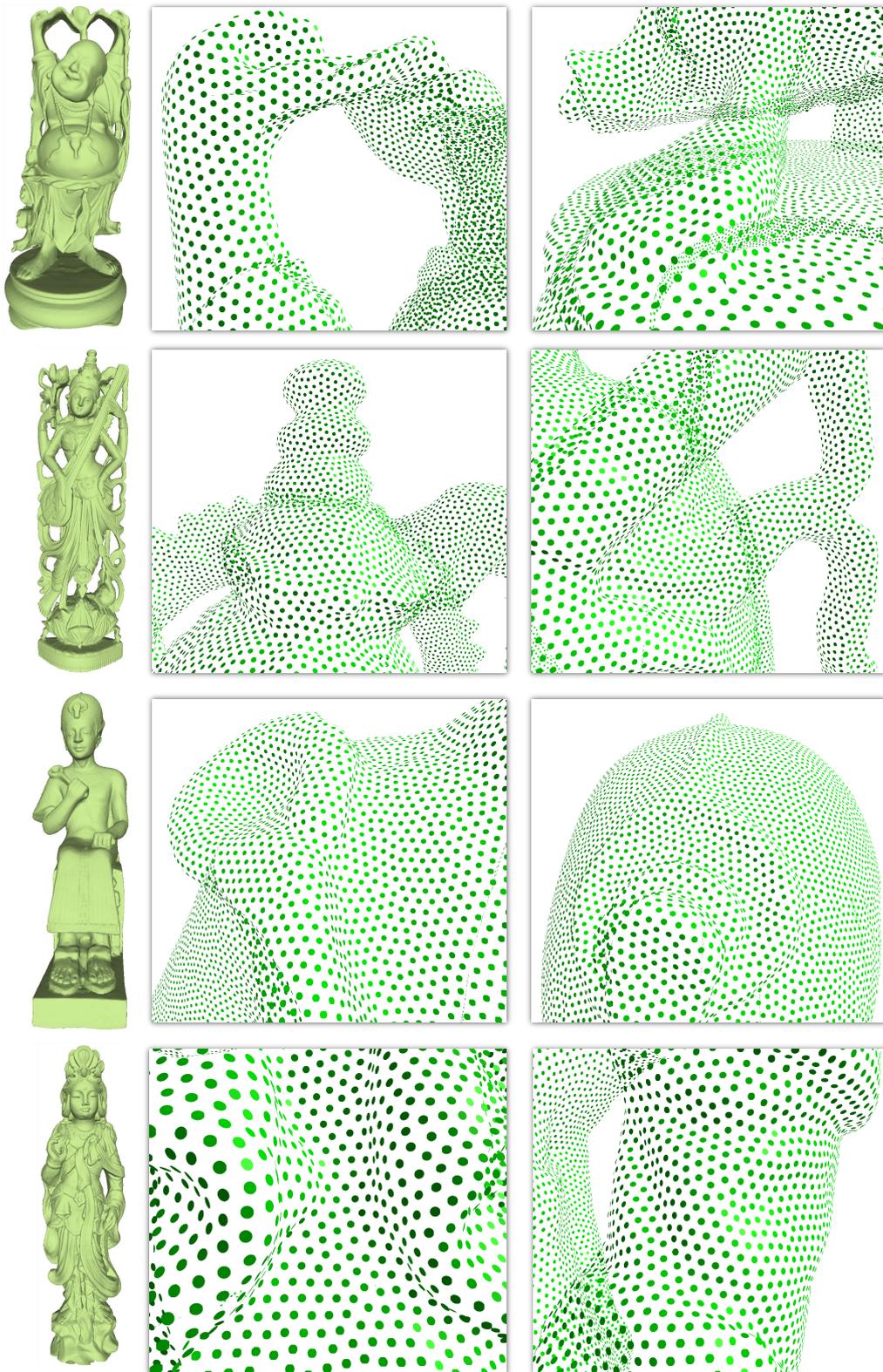
**Figure 4.10:** (a) A distribution of points, (b) mean periodogram, and (c) power and anisotropy graphs for our algorithm (top) and Lloyd's algorithm (bottom).

illustrated in Figure 4.13 and Figure 4.15, our sampling algorithms are also resilient to noise. Our algorithms depend on both the point set and the kernel to determine the manifold they are working on. Hence, whether a point with noise is considered important or not depends on the smoothness level of the kernel. In Figure 4.13 (top row), the input to our sampling algorithms contains a point with high normal noise. This point is consistently kept for smaller values of  $\sigma_p$  (middle two figures) and the corresponding bump is present on the circle. For bigger  $\sigma_p$  (rightmost), it is no longer selected as important by the subsampling algorithm and the bump is eliminated.

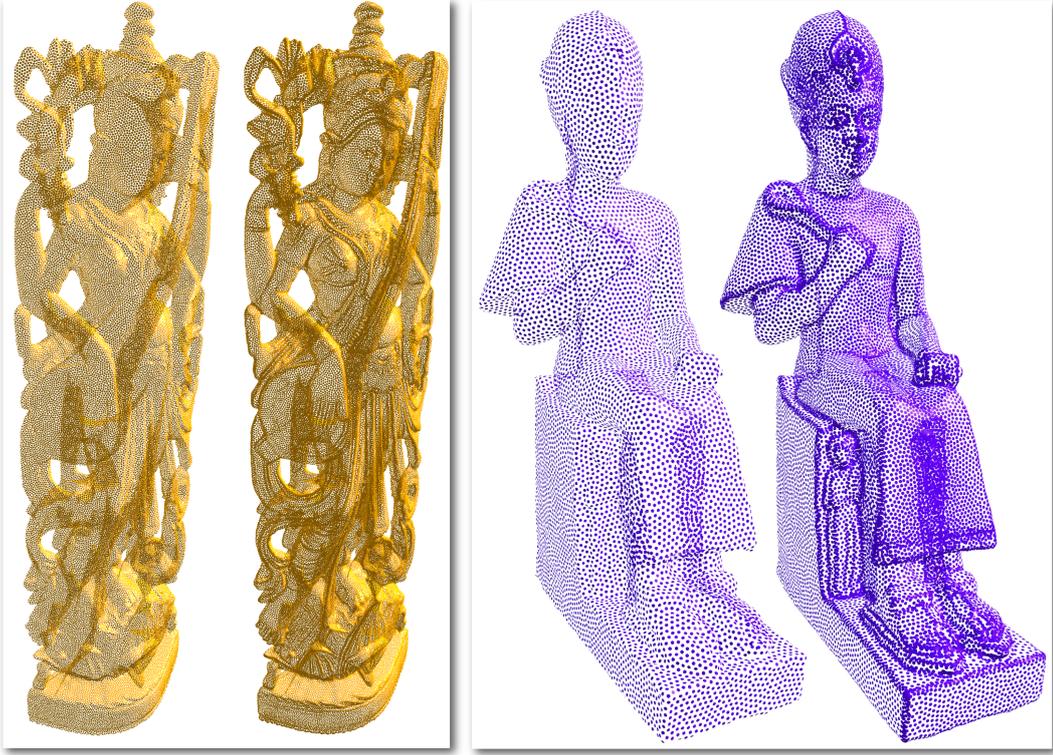
## 4.6.2 Quality of Remeshings

Isotropic distributions are used to remesh surfaces with well-shaped triangles [Yan et al., 2009; Valette et al., 2008] with approximations of the geodesic Centroidal Voronoi Diagram [Du et al., 1999]. To quantitatively measure the quality of our samplings, we compare meshes obtained by using the sampled points output from our algorithms as vertices. We use Tight

*Spectral Sampling of Manifolds*



**Figure 4.11:** *Uniform samplings of various models by our algorithm.*

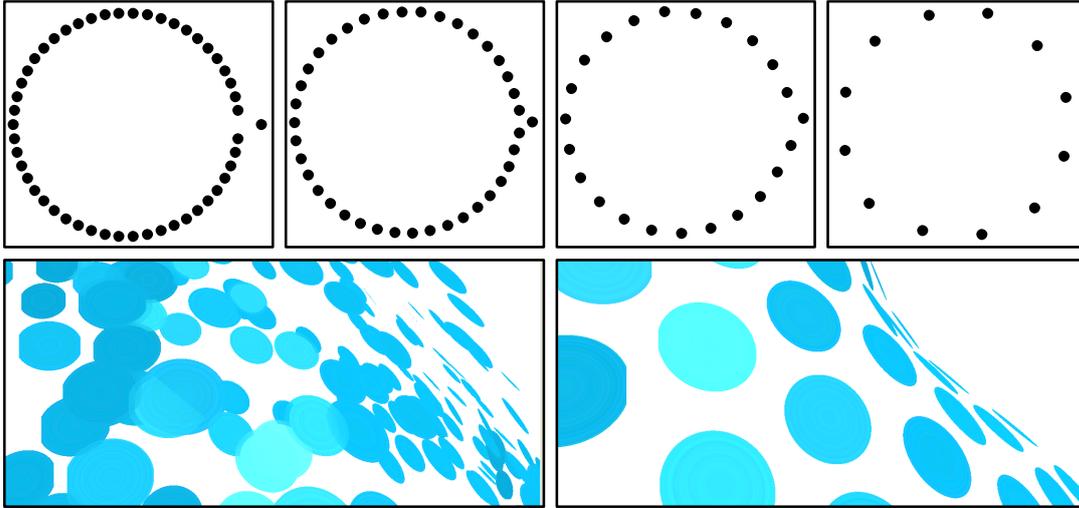


**Figure 4.12:** Feature adaptive sampling ( $\sigma_p = 0.25$ , for each model from left to right:  $\sigma_n = \infty, 0.5$ ).

Cocone [Dey and Goswami, 2003] to triangulate the samples and compare triangle qualities to those generated by a fast clustering based approach (VR) [Valette et al., 2008] and a higher quality but slower method (YR) [Yan et al., 2009].

We use the same measures as used in the papers [Yan et al., 2009; Valette et al., 2008]. The  $\theta_{min}$  is the smallest angle in the mesh,  $\theta_{min,ave}$  is the average of minimum angles of all triangles, and  $\theta < 30^\circ$  is the percentage of triangles with minimum angle below  $30^\circ$ .  $Q$  denotes quality of a triangle and measured as  $6S_t / (\sqrt{3}p_t l_t)$ , where  $S_t$  is the area of the triangle  $t$ ,  $p_t$  is its half-perimeter, and  $l_t$  is the length of the longest edge.  $Q_{min}$  and  $Q_{ave}$  are the minimum and average of  $Q$ 's of all triangles in the mesh.

All measures for the models in the figures are provided in Table 4.1. As can be observed from Table 4.1, our algorithms run in comparable times to VR but still provide triangles with qualities similar to YR. Owing to the inherent smoothing of our algorithms, it also works for very noisy cases without pre-smoothing as illustrated in Figure 4.15 (and the corresponding entry in Table 4.1). The triangle quality is higher and geometry is well-captured. The



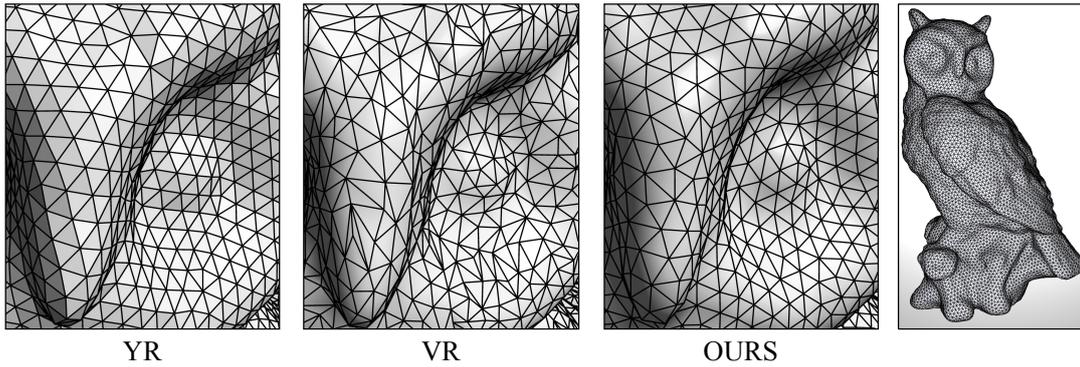
**Figure 4.13:** *Top: Input to our algorithms contains one point with high normal noise (leftmost). Resamplings are shown for increasing  $\sigma_p$ . Bottom: A noisy surface part and its resampling.*

blue noise characteristics of the distribution of the points generated by our algorithms result in more non-regular triangles as can be seen in Figure 4.14. Furthermore, since our algorithms are out-of-core and very efficient, they can be applied to very large datasets such as the Lucy model with 14 million points (Table 4.1).

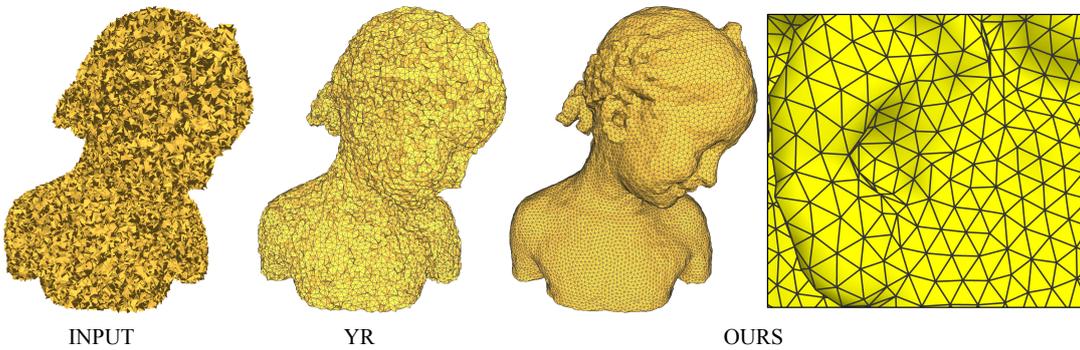
### 4.6.3 Accuracy of Reconstructions

We test the accuracy of the reconstructions by direct comparisons with two point-based iterative simplification algorithms which remove points according to their distance to the surface (AS) [Alexa et al., 2001] or a kernel based measure (KS) [Kitago and Gopi, 2006]. Reconstructions using the initial dense point set and the simplified point sets are compared in terms of the root mean square (RMS) error and Hausdorff distance. Note that we run only our simplification algorithm and not the resampling for fair comparisons.

We use a variety of models of different complexity, genus, and source and a range of  $\sigma_p$  values to illustrate the quality of the reconstructions under different conditions. The parameter  $\sigma_n$  is set to 0.75 for adaptive sampling. After densely triangulating the implicit function we use for reconstructions, Metro tool [Cignoni et al., 2001] is run for computing the errors between this ground truth mesh, and meshes generated by the algorithms. We plot the results of our tests in Table 4.2 and show examples of reconstructions obtained in



**Figure 4.14:** Remeshing results for the Owl model.



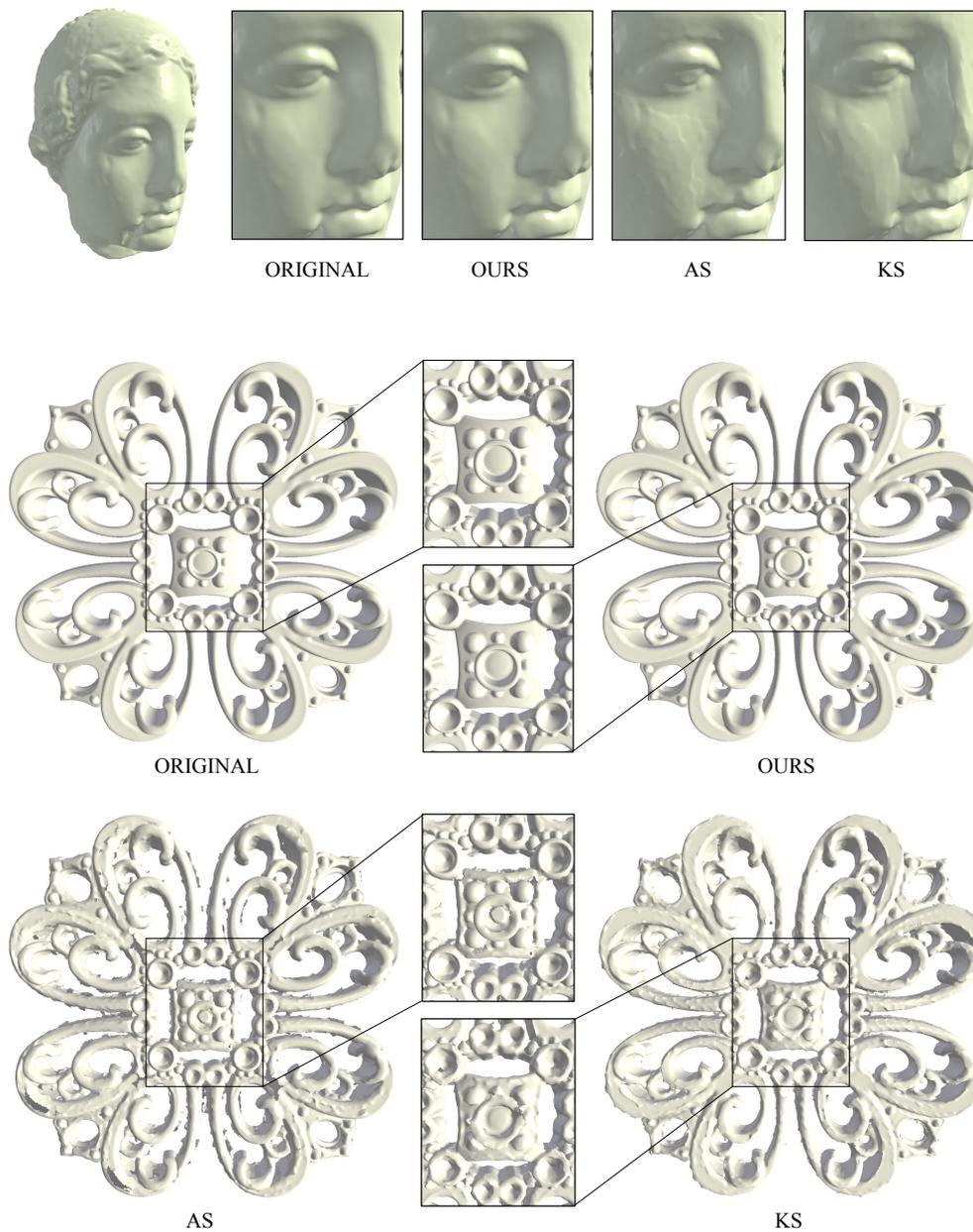
**Figure 4.15:** Remeshing results for the Bimba model with YR and our algorithm. Due to the inherent smoothing of our approach, the geometric shape is well-captured while achieving high triangle quality without pre-processing.

Figure 4.16. Our simplification algorithm results in more accurate reconstructions, with considerable improvements especially for complex models. In Figure 4.16 bottom row, the reconstructions using the simplified point sets of AS and KS result in extra surface parts and distortions on the surface while our algorithm almost exactly reproduces the original reconstruction.

#### 4.6.4 Performance

Linear scanning of the input points avoids costly operations such as finding neighbors and taking local kernel matrix inverses with all input points. This makes our algorithms run in comparable times to even mesh-based subsampling methods. Point based simplification algorithms AS and KS have much larger time complexity and become infeasible to use for large models and large  $\sigma_p$ .

Performance of our algorithms is illustrated in Figure 4.17. In Figure 4.17 (a),



**Figure 4.16:** Reconstructions of the Igea (top row,  $\sigma_p = 0.75$ ,  $\sigma_n = 0.75$ ) and Filigree ( $\sigma_p = 0.5$ ,  $\sigma_n = 0.75$ ) models using the full data, and subsampled data output from different algorithms.

## 4.6 Results and Analysis

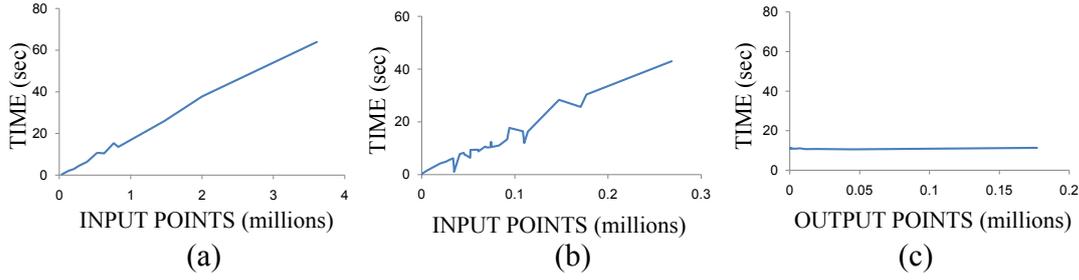
Model	Num Pts	Method	$\theta_{\min}$	$\theta_{\min,ave}$	$Q_{\min}$	$Q_{ave}$	$\theta < 30^\circ$	Time
Owl	13161	<b>Ours</b>	<b>19.34</b>	<b>52.62</b>	<b>0.3007</b>	<b>0.8998</b>	<b>0.0012</b>	<b>3.37 + 6.51</b>
		VR	0.61	39.19	0.0112	0.7194	6.3025	4.453
		YR	37.35	54.71	0.6548	0.9371	0	319
Horse	8600	<b>Ours</b>	<b>32.52</b>	<b>53.01</b>	<b>0.5844</b>	<b>0.9057</b>	<b>0</b>	<b>2.25 + 4.18</b>
		VR	14.76	46.24	0.3148	0.8250	0.3372	3.579
		YR	37.92	54.84	0.6562	0.9389	0	290
Rabbit	9635	<b>Ours</b>	<b>35.19</b>	<b>53.02</b>	<b>0.5573</b>	<b>0.9054</b>	<b>0</b>	<b>2.50 + 4.65</b>
		VR	10.69	44.67	0.2138	0.8036	1.1747	4.859
		YR	39.04	55.09	0.6649	0.9414	0	255
Bimba	13318	<b>Ours</b>	<b>17.10</b>	<b>52.68</b>	<b>0.2664</b>	<b>0.9016</b>	<b>0.0003</b>	<b>1.109 + 2.61</b>
		VR	0.91	37.73	0.0144	0.6993	0.0948	4.031
		YR	20.13	46.17	0.3722	0.8266	0.0030	1424
Lucy	3259	<b>Ours</b>	<b>13.58</b>	<b>51.81</b>	<b>0.2727</b>	<b>0.8915</b>	<b>0.2222</b>	<b>367.56 + 1.46</b>
	7395	<b>Ours</b>	<b>11.14</b>	<b>51.74</b>	<b>0.2538</b>	<b>0.8906</b>	<b>0.3547</b>	<b>367.29 + 3.79</b>
C. Lion	38342	<b>Ours</b>	<b>14.56</b>	<b>52.83</b>	<b>0.2216</b>	<b>0.9033</b>	<b>0.0082</b>	<b>18.83 + 22.58</b>

**Table 4.1:** Quantitative comparisons of the remeshing results. Timings for our method, as well as for Tight Cocone is given in the rows for our algorithm. “Num Pts” refers to the number of vertices in the output mesh.

		$\sigma_p = 0.5$			$\sigma_p = 0.75$			$\sigma_p = 1$		
Model	Error Metric	<b>Ours</b>	AS	KS	<b>Ours</b>	AS	KS	<b>Ours</b>	AS	KS
Igea	RMS	<b>0.0071</b>	0.0064	0.0091	<b>0.0114</b>	0.0310	0.0818	<b>0.0407</b>	0.0824	0.1251
	HAUS	<b>0.1075</b>	0.1415	0.1120	<b>0.1500</b>	0.4384	0.4640	<b>0.1907</b>	1.3302	2.0278
Hand	RMS	<b>0.0186</b>	0.1628	0.1464	<b>0.0286</b>	0.5405	0.5873	<b>0.0439</b>	1.0162	1.0382
	HAUS	<b>0.2116</b>	1.2040	1.1011	<b>0.2472</b>	2.0007	2.1906	<b>0.5616</b>	2.9348	2.7861
Statue	RMS	<b>0.0488</b>	0.2357	0.2796	<b>0.0949</b>	0.5980	0.6975	<b>0.1501</b>	1.0184	0.7677
	HAUS	<b>1.2853</b>	1.3792	1.4335	<b>1.9832</b>	2.0409	2.4409	<b>2.5519</b>	3.7673	2.5541
Dragon	RMS	<b>0.0526</b>	0.1011	0.3299	<b>0.0973</b>	0.2514	0.6862	<b>0.1495</b>	0.5747	1.4733
	HAUS	<b>1.4130</b>	1.3971	1.7840	<b>2.0595</b>	2.0062	2.6382	<b>2.6049</b>	2.8164	4.3328
Filigree	RMS	<b>0.0209</b>	0.3206	0.1623	<b>0.0457</b>	0.8481	0.4929	<b>0.0832</b>	1.3032	1.0696
	HAUS	<b>0.2591</b>	1.3294	1.1363	<b>0.4363</b>	2.2062	1.7476	<b>0.8408</b>	2.6989	2.5756

**Table 4.2:** Quantitative comparisons of the reconstructions.

total time (including kd-tree queries and disk reads) needed to subsample models of various sizes is plotted. Complexity grows linearly with input size due to the linear scan used in the subsampling algorithm. Figure 4.17 (b) illustrates the same performance analysis for the resampling algorithm, run after the subsampling algorithm. In Figure 4.17 (c), number of output points, hence  $\sigma_p$ , is changed for the subsampling algorithm. Since we use iterative inversion (as explained in Section 4.5.1) for computing our measure, iterations are cut earlier if  $\sigma_p$  is large and the complexity stays constant.



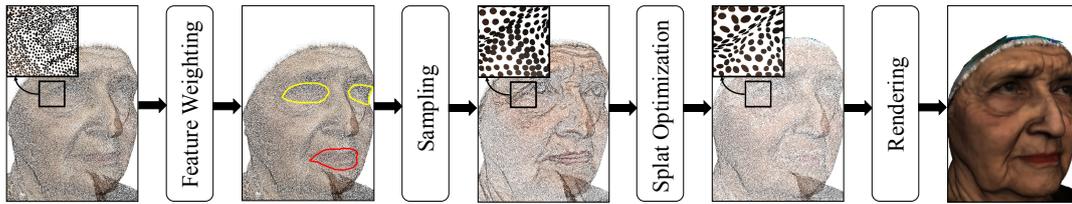
**Figure 4.17:** Effect of the input model size on the time complexity of the (a) subsampling algorithm and (b) resampling algorithm. (c) Timings for the subsampling algorithm for a fixed input model with changing output sample sizes.

## 4.7 Application to Facial Rendering

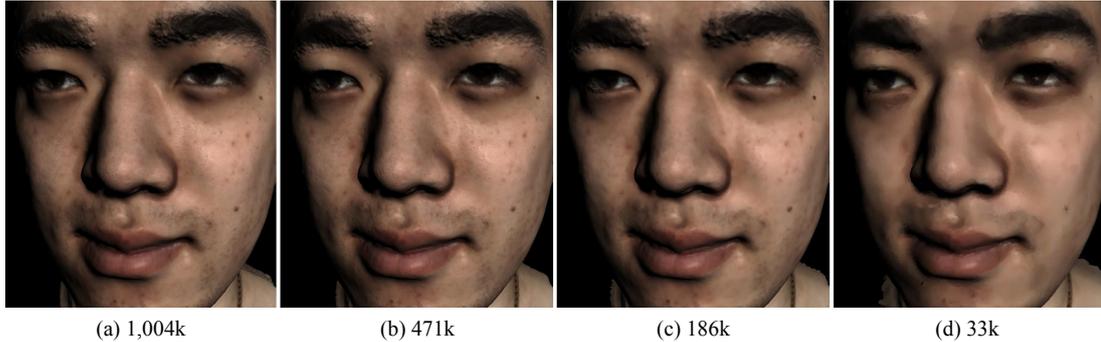
The samplings produced by our algorithms depend on the kernels used. Although the algorithms so far assumed that we only have information about the geometry to be reconstructed, further information can easily be taken into account by modifying the domains of the kernels accordingly. In particular, similar to aggregating normals and positions for defining the domain  $\Omega$  where the kernel operates, one can include more dimensions for any additional information present. As an important application of this flexibility, we developed a multi-resolution splat-based face renderer. High quality splatting is obtained by considering the texture and important features of the faces when sampling.

The pipeline of this algorithm is presented in Figure 4.18. After extracting perceptually important features of the face, the sampling algorithms are run with the kernels derived using the extracted features and colors associated with the sample points. Finally, splat shapes are optimized accordingly, and a dedicated splat based renderer is used for efficient and realistic rendering of faces. Below, we focus on how the extra information is used for the sampling algorithms we developed in the chapter.

Assume that the feature extraction is completed and for each sample point, the position  $\mathbf{p}_i$  normal  $\mathbf{n}_i$ , color  $\mathbf{c}_i$ , and a weight  $w_i$  that determines the importance of this point as a feature is stored:  $\mathbf{x}_i = (\mathbf{p}_i, \mathbf{n}_i, \mathbf{c}_i, w_i)$ . We extend the domain  $\Omega$  of the kernel to include color components and the feature weights, so that it becomes  $\mathbf{x} = [\mathbf{p} / (\sigma_p w) \quad \mathbf{n} / \sigma_n \quad \mathbf{c} / \sigma_c]^T$ . The weights  $w$  take two different values depending on the region of the face. If it is a perceptually important region such as eyes, lips, or eyebrows, it takes a low constant value. Otherwise, it takes a higher constant value. Hence, this weighting adjusts the sampling rate such that the density will be higher for perceptually important



**Figure 4.18:** *The pipeline of adaptive surface splatting for facial rendering.*



**Figure 4.19:** *Multi-resolution rendering of a face.*

facial features. This means we have two different kernel definitions for the feature and non-feature areas. Although these areas have boundaries, we did not encounter any instabilities in practice.

Note that we only have color values for the initial sample points, but values at arbitrary points in space are needed for resampling. Following the original derivation of the adaptive kernel definition from robust statistics (see also Section 3.4.3), we use the following iterative formula to get the value  $\mathbf{c}(\mathbf{x})$  for a point  $\mathbf{x}$ :

$$\mathbf{c}^{k+1} = \frac{\sum_i \mathbf{c}_i k(\mathbf{x}^k, \mathbf{x}_i)}{\sum_i k(\mathbf{x}^k, \mathbf{x}_i)}, \quad (4.25)$$

where  $\mathbf{x}^k$  includes the current estimate of the color  $\mathbf{c}^k$ . Starting from the usual kernel based approximation ( $\sigma_c = \infty$ ), the iterations continue until convergence. This gives a robust color estimate that accurately preserves discontinuities. When processing the point  $\mathbf{x}$  in the resampling algorithm, the color value for it is computed with this formula, and added to the vector  $\mathbf{x}$ .

A multi-resolution splatting result is shown in Figure 4.19. Since the distribution of the samples adapts well to the facial details, much less points are needed for high quality renderings.

## 4.8 Discussion and Outlook

We presented new algorithms for simplification and resampling of manifolds. The algorithms depend on a measure that restricts changes to the Laplace-Beltrami spectrum. By utilizing kernel methods and matrix perturbation theory, we were able to derive a local measure linked to the MLS based reconstruction method we introduced in Chapter 3. We then utilized this measure in efficient simplification and resampling algorithms. The generated samplings have high quality blue noise characteristics and result in accurate reconstructions. We also demonstrated an application of our algorithms to high quality multi-resolution facial rendering. By changing the domain of the kernel used to compute the spectral measure, our method can be easily adapted for preservation of application specific features.

We tested and validated the results of our algorithms for a diverse selection of models and conditions. Although we observed that our algorithms are efficient and accurate, they are greedy and thus not theoretically guaranteed to give the optimal sampling. We believe however, that our analysis of the Laplace-Beltrami spectrum can be utilized for more sophisticated algorithms as well. There exist some results on theoretical optimal sampling conditions when using Delaunay triangulation based reconstruction algorithms [Amenta and Bern, 1998; Amenta et al., 2000; 2001; Dey, 2006]. However, they depend on the local feature size, which is hard to estimate in general. For MLS or radial basis functions based reconstruction techniques, no such results exist. Hence, there is a general need in developing sampling conditions that are robustly computable. On the other hand, many works have focused on improving the Laplace-Beltrami operator for meshes [Zhang et al., 2007] and point sets [Belkin et al., 2009]. Deriving optimal sampling conditions based on the Laplace-Beltrami operator will allow the researchers to utilize these improvements and can lead to algorithms with theoretical guarantees.

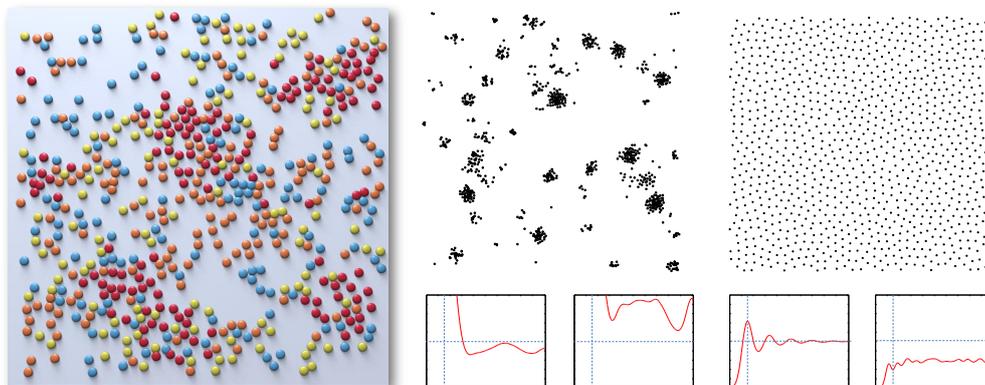
Sampling is ultimately related to signal processing and multiresolution analysis. Computational harmonic analysis gives us a unified framework to perform multiresolution analysis on general manifolds. We presented an effective application of it to sampling. We believe that these and similar ideas from data analysis, sparse coding, and kernel methods will be useful to understand the sampling problem better and develop practical algorithms with theoretical guarantees.

Apart from the sampling problem, the interpretation of the heat kernel in the framework of kernel techniques can have direct consequences for many techniques that utilize the eigenstructure of the Laplace-Beltrami operator [Zhang et al., 2007]. Beyond these works in the geometry processing literature,

the proposed sampling measure and interpretation of the heat kernel can also have impact on the data analysis and pattern recognition methods that work with high dimensional data. Many of these methods deal with vast amounts of data and hence developing new compact representations can considerably speed up the algorithms. In general, diffusion on manifolds has been one of the fundamental tools for analysis [Kondor and Lafferty, 2002; Coifman et al., 2005; Coifman and Lafon, 2006; Lafon and Lee, 2006; Coifman and Lafon, 2006], which is directly linked to the heat kernel. Hence, we anticipate further developments by utilizing and unifying the theoretical frameworks developed in data analysis and machine learning via the heat kernel.

## *Spectral Sampling of Manifolds*

## Reconstruction of Point Patterns



Many geometric structures exhibit repetitive and stochastic patterns that can be explained well in a distributional sense. Although an individual tree in a forest can be represented by a point set and reconstructed via the algorithms proposed in Chapter 3, synthesizing a virtual forest with the distribution of the trees learned from a real forest poses a completely different problem. The entity to be reconstructed in this case becomes a stochastic point pattern/distribution with certain characteristics matching those of given patterns. Such a reconstruction consists of analyzing point patterns

generated by a process to learn their characteristics, and synthesizing new distributions having the same characteristics.

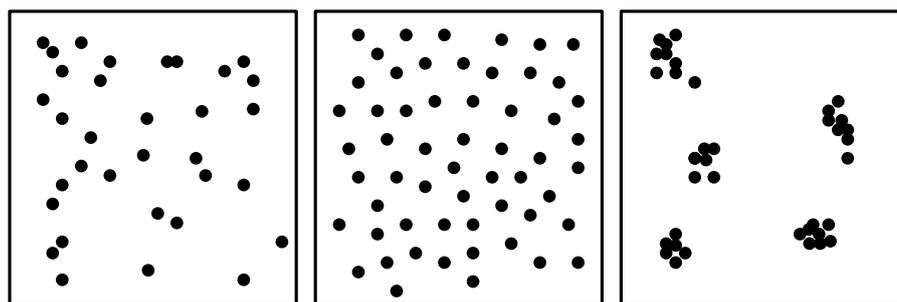
Although stochastic point distributions arise in many contexts, analysis tools and synthesis algorithms in computer graphics have been mostly focused on blue-noise distributions, in which points are distributed randomly with a minimum distance between pairs. Such distributions have been observed in many biological and natural structures including the human visual system [Yellott, 1983]. It is well-known that using such a distribution for sampling leads to high quality anti-aliasing. They are also used in many other applications, as explained in Section 2.3. This has led to extensive analysis and many synthesis algorithms that generate blue-noise distributions with different regularity, density, and randomness. However, these algorithms cannot generate general distributions with controllable characteristics.

Point distributions have also been extensively studied in physics and spatial statistics [Torquato, 2002; Illian et al., 2008]. The emphasis in these fields is put on analyzing general distributions and fitting models to understand natural processes. Statistics that depend on correlations of locations and marks of points are used to analyze a diversity of distributions and have been proven to be powerful and discriminative.

In this chapter, we introduce methods for analysis and synthesis of general multi-class point distributions based on the statistical measure *pair correlation function* (PCF). To explore the nature of this measure, we introduce an analysis based on the interpretation of it as a mean in a high dimensional vector space that we call the *pair correlation space* (PCS). We show that in this space, the degrees of freedom to characterize point distributions is low and directly linked to regularity. This analysis allows us to explain distributions and existing synthesis algorithms in a unified way, propose an irregularity measure, and show that the PCF provides a compact representation for the characteristics. Following this analysis, we propose two general synthesis algorithms. The first one is a generalization of dart throwing for arbitrary PCFs and the other is a gradient descent based fitting of the PCFs. The output of the first algorithm is used as the input for the second to facilitate convergence. These algorithms can generate point distributions with desired characteristics extracted from example distributions or synthesized. The example point sets and generated point sets can be of different dimensions and sizes, contain multiple classes, and reside on non-Euclidean domains. The algorithms are simple to implement and run in  $O(n)$  time.

We start the chapter by reviewing the relevant concepts and methods from the statistical field of *point processes* (Section 5.1). We then explain the theory behind the statistical analysis tools for point distributions in this field in Sec-

tion 5.2. Next, we propose a new interpretation of the PCF and define a new irregularity measure in Section 5.3. This new analysis allows us to develop synthesis algorithms for distributions in Section 5.4, which are evaluated extensively (Section 5.5), and used in applications where stochastic structures need to be synthesized (Section 5.6).



**Figure 5.1:** Examples of random (left), hard-core (middle), and clustering (right) distributions.

## 5.1 Point Processes

A point process is defined as the generating stochastic model for a point distribution/pattern. A distribution/pattern of points generated using a point process is one of the infinitely many realizations of that process. The goal of point process statistics is to describe, analyze, and compare point patterns in the most effective way. This branch of statistics differs from classical statistics. While classical statistics relies on moments of probability distributions such as the mean or standard deviation, point process statistics focuses on describing distributions of points using *correlations* among locations and marks of points. This field also has strong connections to physics and material sciences. Explaining repetitive structures such as the distribution of molecules in a gas, or pores in a concrete requires to rely on similar statistics in these fields. For an excellent review of the theory and applications of point processes, we refer the reader to Illian et al.'s book [Illian et al., 2008].

In this chapter, we consider infinite (i.e. the process can be used to generate patterns of any number of points covering arbitrarily large volumes in space), stationary (i.e. translation invariant), isotropic (i.e. rotation invariant), and ergodic (i.e. analysis on a particular generated pattern is sufficient) point processes. Hence, the analysis of the point set does not depend on the point, direction, or window chosen. We will call such point processes and generated point sets simply as isotropic. The cases where an adaptive or anisotropic distribution is needed can be considered as uniform sampling of a domain and warping that domain via a distance measure [Li et al., 2010b].

The characteristics of the distributions generated by point processes depend on the underlying rules used. One thread of rules involves interaction of points. Repulsion and attraction are two interactions that can exist among points. Point processes with repulsion result in distributions where each

pair of points is a minimum distance apart. Such distributions can possess blue-noise properties (see Chapter 4) and the underlying point processes are called *hard-core* processes, since each point resembles a hard sphere. In contrast, if attraction among the points is dominant, points tend to cluster together. These processes are called *clustering* processes. In nature, repulsion and clustering can happen at different scales, leading to more complex distributions. Nevertheless, many distributions can be modeled well by assuming a hard-core or clustering model. In addition to interactions, other operations such as *thinning*, that is, taking out points from an initial point set, or *superposition* of different point sets can be used as generating rules. We review some of the proposed models for hard-core and clustering point processes below. Examples of typical random, hard-core, and clustering distributions are shown in Figure 5.1.

### 5.1.1 Hard-core Processes

Hard-core processes can be generated in two fundamental ways. The first one involves *thinning* a given point cloud. Starting from a dense point set, thinning can be applied to get a new set with no points closer to each other than a specified distance. Although simple, this procedure often results in a low density distribution. A familiar example of this procedure is known as *dart throwing* in computer graphics, *simple sequential inhibition* in statistics, and *random sequential adsorption* in physics. The second method to obtain a hard-core process involves repulsion of points. In practice, this means the points are moved in space according to the interactions among them. Such simulations based on particle models have also been used in graphics [Turk, 1992; Witkin and Heckbert, 1994; Schmaltz et al., 2010].

**Matérn Hard-core Processes.** This process [Matern, 1960] is an instance of the thinning-based hard-core processes. There are three variants of the process depending on the thinning rule used. The process starts with an initial point set distributed according to a Poisson process (i.e. uniformly randomly distributed). The first type simply visits each point, and deletes the points that have neighbors within a specified radius independently. The second type generates a random mark in  $[0, 1]$  for each point in the initial dataset. The thinning is then performed by retaining only the points that have no neighbors with smaller marks within a specified radius. A mark for a given point can be seen as the time that point arrives. Then the second type can be interpreted as retaining the points that arrived earlier than their neighbors. The third type uses the same thinning procedure as the second type iteratively. In the first iteration, it is exactly equivalent to the second

type. For the next iteration, only the points that are not in the neighborhoods of the points that were kept at the first iteration are considered when thinning. The algorithm iterates by pruning the space of allowable points further at each iteration.

The third type generates the densest patterns, followed by the second and the first, respectively. Since all thinning operations are independent of each other, the first and second type can be implemented in an embarrassingly parallel way, while the third type needs multiple parallel steps corresponding to the iterations.

**The Dead Leaves Model.** This model [Matheron, 1968] follows the natural process of the piling up of dead leaves in nature. Each leaf is modeled as a circular disk and falls onto the others at a time in  $(-\infty, 0]$ . The leaves are allowed to overlap. After a given number of leaves fall, only the ones in the top layer that can be seen fully are retained. This process approaches the second Matérn hard-core process as the number of leaves goes to infinity.

**Random Sequential Adsorption (RSA).** This process is equivalent to the well-known dart throwing method in computer graphics. At each iteration, a new point is generated uniformly randomly in space, and it is retained if there is no other point in the neighborhood of it. Unlike the previous processes, RSA is an inherently sequential algorithm, although parallel algorithms that approximate the RSA process have been devised [Wei, 2008]. RSA can be interpreted as thinning an infinitely dense initial point set corresponding to the whole space.

**Force-based Interactions.** An interaction-based approach to generate hard-core distributions parallels the physical forces between particles. Depending on the forces used, many different algorithms have been devised in physics and statistics [Jodrey and Tory, 1985; Moscinski et al., 1989; Bezrukov et al., 2002], and computer graphics [Turk, 1992; Witkin and Heckbert, 1994; Schmaltz et al., 2010]. The algorithms move points according to the computed forces so as minimize the defined energy. Since a continuous movement is allowed, the distributions can have much higher density of points than the ones generated by the thinning-based algorithms. However, distributions converge to a regular pattern due to the deterministic nature of the algorithms, and they have typically higher time complexity.

### 5.1.2 Clustering Processes

Clustering processes are characterized by a heterogeneous point pattern with lumps of points. This is a very broad definition and without the assumption

of isotropy, is not well-defined. This is because clusters can also be seen as density variations in the point pattern. Many object distributions in nature such as distributions of galaxies, plants, or seeds, exhibit clustering patterns at various scales. Formation of a clustering pattern can be due to several processes. Thinning of an initial uniform pattern or generation of points around initial seed points are two instances of constructions. Apart from these, interactions among points in the form of attraction can also lead to clustering. Below we describe a seed-based clustering process, the Neyman-Scott process, and two other processes that can be seen as special cases of the Neyman-Scott process.

**Neyman-Scott Processes.** Neyman-Scott processes [Neyman and Scott, 1952] are based on generating secondary points around an initial set of seed points. The seed points are uniformly distributed according to a Poisson process. Around each seed point, a random number of secondary points are produced, following a specified distribution. Only the secondary points are retained for the final pattern. For isotropic processes, the distribution of the secondary points around each seed point is the same and radially symmetric. Assuming a seed point at the origin, this distribution can be described by a uniform sampling of directions, and some distribution of distances of the secondary points to the origin.

Since the generation of seed and secondary points is done independently, the process is quite simple to simulate from. First, a uniformly distributed set of seed points is generated. Then, for each seed point  $\mathbf{x}_s$ , a random number  $n$  from a given discrete probability distribution  $P(n)$  is drawn. Finally,  $n$  secondary points are placed around this seed point. Each secondary point's location is given by  $\mathbf{x} = \mathbf{x}_s + r\mathbf{d}$ , where  $\mathbf{d}$  is a direction drawn from a uniform distribution, and  $r$  is a distance following a given continuous probability distribution  $p(r)$ . Depending on the choice of these probability distributions, different processes can be obtained. A choice of uniform distribution for  $p(r)$  leads to the Matérn Clustering Process [Matern, 1960], whereas a normal distribution for  $p(r)$  leads to the Modified Thomas Process [Thomas, 1949].

### 5.1.3 General Process Models

Apart from the discussed dedicated models, there exist more general models that can be used to generate more complex patterns. We will describe two such processes: Cox processes and Gibbs processes. In particular, Gibbs processes are quite flexible and can be used to synthesize a variety of distributions once pairwise correlations are extracted. We start with Cox processes.

**Cox Processes.** A Cox process is constructed in two steps. In the first step, a stochastic intensity  $\Lambda(\mathbf{x})$  is set, and an instance  $\lambda(\mathbf{x})$  is drawn. Intuitively, this means the density of points at different locations in space is random and determined by drawing from a probability distribution for a particular instance of the process (please see Section 5.2.2 for a precise definition of intensity). In the second step, a random point set is generated according to the drawn intensity. This two-level construction results in a valid definition of an isotropic process and yet very flexible to model especially clustering processes.

**Gibbs Processes.** A fundamental limitation of Cox processes is that they ignore any interactions between points. Gibbs processes (also called Markov point processes) are specifically designed to take pairwise interactions into account. For modeling the interactions, a potential function  $\psi(d(\mathbf{x}_i, \mathbf{x}_j))$ , for a distance function  $d$ , is defined. For a particular distance  $r$  in the process,  $\psi(r) = 0$  means that there is no interaction between the points at this distance. The case  $\psi(r) > 0$  implies a repulsion, and conversely  $\psi(r) < 0$  leads to attraction between points. For simplicity, if we assume that the point process is finite and the number of points in the space under consideration is fixed to  $n$ , then the unnormalized probability  $p(x_1, \dots, x_n)$  of having the particular configuration  $\{x_i\}$  of points is given by the following expression:

$$p(x_1, \dots, x_n) = e^{-\sum_{i=1}^{n-1} \sum_{j=i+1}^n \psi(d(\mathbf{x}_i, \mathbf{x}_j))}. \quad (5.1)$$

Note that the points denoted by  $x_i$  are abstract identities for which  $\mathbf{x}_i$ 's are the locations. This probability is defined over the set of all possible point configurations. Using different potential functions  $\psi$ , very different processes can be obtained. An example is a hard-core process with separation distance  $r_0$  between the points. For this case, the potential function is  $\psi(r) = \infty$  for  $r \leq r_0$  and  $\psi(r) = 0$  for  $r > r_0$ . Hence, there is no interaction beyond the distance  $r_0$ , and the points repel each other with infinite strength for smaller distances.

The power of Gibbs processes comes from the definition of the probability in Equation 5.1. Once defined, it can be used in Markov Chain Monte Carlo algorithms that accept unnormalized densities for simulations. These algorithms are extensively used to sample from high dimensional distributions and typically involve proposals of addition/deletion of points at each iteration.

## 5.2 Statistics for Point Processes

For most cases, only a single distribution of points is known, and no information about the underlying point process is available. Hence, it is crucial to develop appropriate statistical measures to explain distributions without reference to the underlying process. Such statistical measures have been proposed and extensively used to analyze distribution of locations of objects ranging from atoms to galaxies. In this section, we focus on some important measures and their properties.

For isotropic point processes, there exist several commonly used statistics. Since correlations among points far away from each other are typically insignificant, the statistics focus on explaining short-distance correlations between the points. A fundamental property that shapes point process statistics is the *degree* of the correlations. A 1<sup>st</sup> order statistic considers each point at a time and hence interactions between points cannot be understood. A 2<sup>nd</sup> order statistic can be used to investigate interactions between *pairs* of points. Statistics of higher orders are similarly defined. In order to properly define the statistics, we first explain the probabilistic framework, and then summarize some of the important statistics commonly used in the statistics and physics literature.

### 5.2.1 Definitions

Statistical analysis of point processes calls for definitions of probabilities, from which average values can be obtained. One way to assign probabilities for processes is considering subsets of the space the points live in. Specifically, let  $V$  be a Borel set and  $N(V)$  be the random number of points in set  $V$ . Note that since the point process is stochastic,  $N(V)$  is a random variable for a fixed  $V$  (assuming that  $N(V) < \infty$  for a bounded  $V$ ). Then, the probability  $P(N(V) = n)$  of having  $n$  number of points in set  $V$ , the expected value  $\mathbf{E}[N(V)]$  of the number of points in set  $V$ , and higher order moments can be defined.

For isotropic point sets, the choice of the origin does not affect the computed statistics. We assume that the point process is shifted such that there is a point of the process at the location  $\mathbf{o}$ . Then, probabilities and statistics can be defined with respect to the point  $o$ . For example,  $\mathbf{E}[N(b(\mathbf{o}, r) \setminus \{o\})]$  is defined as the expected number of points in a spherical neighborhood of radius  $r$  centered at  $\mathbf{o}$ , without counting  $o$ . Finally, we denote a distance metric of the space that the points live in with  $d(x, y)$  or  $d(\mathbf{x}, \mathbf{y})$ , depending on the implications.

## 5.2.2 First Order Statistics

A familiar 1<sup>st</sup> order statistic is the *intensity* of a point process. Specifically, given a Borel set  $V$ , the intensity is defined as:

$$\lambda = \frac{\mathbf{E}[N(V)]}{|V|}, \quad (5.2)$$

where  $|V|$  denotes the volume measure of the set  $V$ . For isotropic processes, this is a constant number that identifies how many points are to be expected in a given unit area. Hence, for isotropic point sets, simply counting the number of points in the given point set and dividing by the area of the observation window gives an estimate of intensity. Note that intensity is more commonly known as density in computer graphics. Intensity is very simple to compute and interpret, but is insufficient to describe distributions since it completely ignores any interactions among the points. This lead to the development of higher order statistics, as explained in the next sections.

## 5.2.3 Second Order Statistics

Statisticians and physicists regard 2<sup>nd</sup> order statistics and in particular the *pair correlation function* (PCF) as the most informative and for most distributions sufficient for unique determination (this is called the *second order dogma* [Illian et al., 2008]). This is supported by a theorem by Boutin et al. [Boutin et al., 2004; Boutin and Kemper, 2007]. The theorem states that the set of point distributions that are not uniquely determined by their distance distributions has Lebesgue measure zero in the  $nd$  dimensional space of point configurations, when  $n \geq \max(3, d + 2)$  for  $n$  points in  $d$  dimensions. Thus, for isotropic point sets, the distribution of pairwise distances, and hence the PCF, uniquely determines most of the point distributions (for the interpretation of the PCF as the distribution of pairwise distances, please see Section 5.3.1).

Below we explain three fundamental 2<sup>nd</sup> order statistics: Ripley's K-function, the L-function, and the PCF.<sup>1</sup> These measures are related to each other by simple formulas, which lead to the definition of the PCF. Hence, they allow to see how the PCF is derived starting from basic probabilities.

**Ripley's K-function.** This statistic measures the number of points in a spherical neighborhood of a point of the process, not counting the point itself. It is defined as follows:

---

<sup>1</sup>The pair correlation function will be investigated in detail in Section 5.3 from a slightly different perspective.

$$K(r) = \frac{\mathbf{E}[N(b(\mathbf{o}, r) \setminus \{o\})]}{\lambda}. \quad (5.3)$$

The statistic  $K(r)$  is proportional to  $r^d$  for points in  $\mathbb{R}^d$ . This means the fluctuations and hence the information content is more observable for small  $r$ 's. This limits its use in practical applications.

**The L function.** The L function is defined in terms of Ripley's K-function as:

$$L(r) = \left( \frac{K(r)}{|V_d|} \right)^{1/d}, \quad (5.4)$$

where  $|V_d|$  denotes the volume a unit sphere in  $d$  dimensions. Although  $L(r)$  contains exactly the same information as  $K(r)$ ,  $L(r)$  is more suitable for analysis since it always grows proportional to  $r$ .

**The Pair Correlation Function.** The pair correlation function can also be defined in terms of Ripley's K-function with a simple formula:

$$g(r) = \frac{K'(r)}{|\delta V_d| r^{d-1}}, \quad (5.5)$$

where  $|\delta V_d|$  denotes the volume of the boundary of a unit sphere in  $d$  dimensions. Similar to the L function, it grows more slowly with  $r$  and hence more amenable to analysis. We will give another interpretation of this measure, and investigate its properties in detail in Section 5.3.

### 5.2.4 Higher Order Statistics

It is possible, and necessary for some applications, to consider higher order statistics for analysis of some point patterns. These statistics measure various indicators of the correlations among points. However, most are rarely used in practice.

One possible way to get higher order statistics is utilizing the Voronoi diagram of the set of points in a given distribution. Voronoi-based statistics start with computing the Voronoi diagram or the Delaunay triangulation of the points. Next, quantities related to the shapes of the simplices are computed and compared [Medvedev and Naberukhin, 1987; Naberukhin et al., 1991]. In general, it is easier to distinguish fine-scale details by considering simplices in a Delaunay triangulation than in a Voronoi diagram [Illian et al., 2008].

## *Reconstruction of Point Patterns*

This is because a simplex in the Delaunay triangulation typically depends on less points.

An example usage of these statistics appeared in the paper by Balzer et al. [Balzer et al., 2009], where the authors use the number of neighbors of the Voronoi regions to determine regularity in the point set. The presence of large patches of hexagonal regions reveal that the point set has regularities.

## 5.3 Analysis of Point Distributions

As pointed out in Section 5.2.3, second order statistics and in particular the pair correlation function (PCF) is regarded as the most informative of all, and there is also a strong theoretical support for this conjecture. As will be shown below, the PCF has a very simple and smooth estimator that can be regarded as a kernel based estimator of the probability density function of distances between points. We base our analysis on this estimator of the PCF and show how it can be re-interpreted in a novel way to define a new and unified characterization of point patterns.

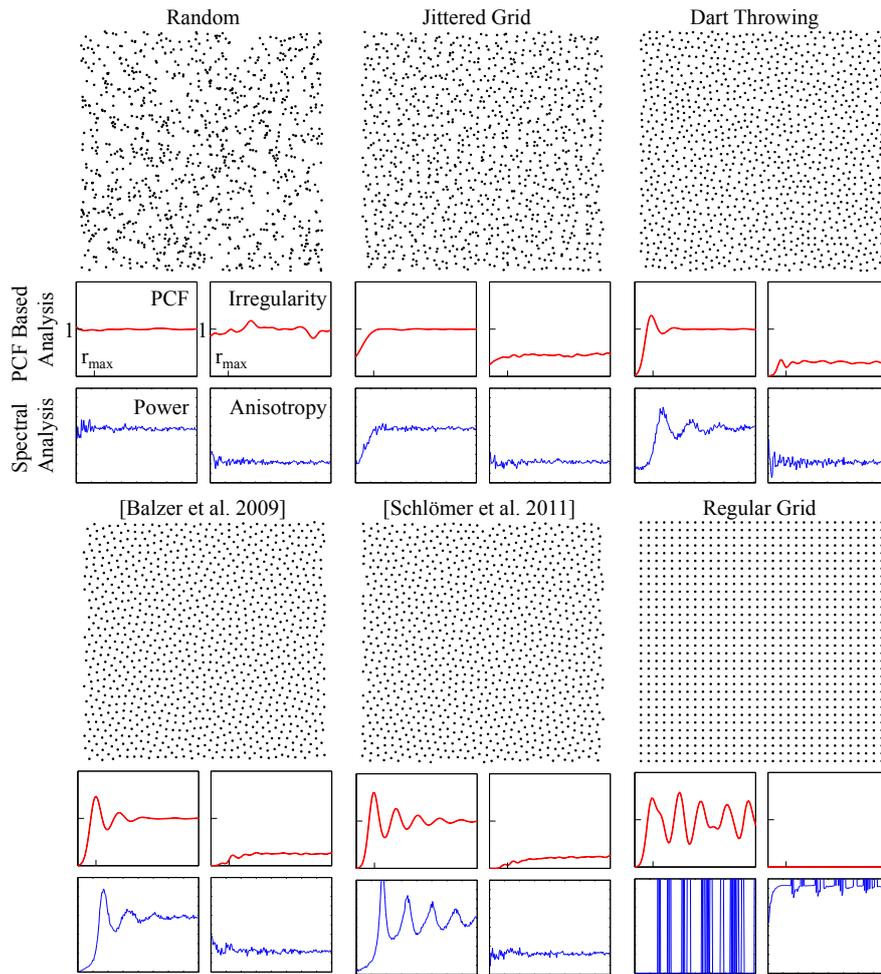
In Sections 5.3.1 and 5.3.2, we explain the concepts needed for our purposes. In Sections 5.3.3, 5.3.4, and 5.3.5 we introduce a new analysis of point distributions in a space implied by the PCF.

### 5.3.1 The Pair Correlation Function

Intuitively, the pair correlation function  $g(\mathbf{x}, \mathbf{y})$  describes the joint probability of having points at locations  $\mathbf{x}$  and  $\mathbf{y}$  at the same time. A precise definition of the PCF can be given in terms of the intensity  $\lambda$  and product density  $\varrho$  of a point process. The intensity  $\lambda(\mathbf{x})$  of a point process is the average number of points in an infinitesimal volume around  $\mathbf{x}$ . Hence, intuitively it measures the average density of the points. For isotropic point processes, this is a constant value  $\lambda$ . To define the product density, let  $\mathbf{x}_i$  denote the points,  $B_i$  infinitesimal spheres around the points, and  $dV_i$  the volume measures of  $B_i$ . Then  $p(\mathbf{x}_1, \dots, \mathbf{x}_n) = \varrho(\mathbf{x}_1, \dots, \mathbf{x}_n) dV_1 \cdots dV_n$  is defined as the probability of having  $\mathbf{x}_i$  in the infinitesimal spheres  $B_i$ . For a pair of points, a second order version of this probability is  $p(\mathbf{x}, \mathbf{y}) = \varrho(\mathbf{x}, \mathbf{y}) dV_x dV_y$ . In the isotropic case,  $\varrho$  only depends on the distance between the points, hence one can write  $\varrho(\mathbf{x}, \mathbf{y}) = \varrho(\|\mathbf{x} - \mathbf{y}\|) = \varrho(r)$  and  $p(r) = \varrho(r) dx dy$ . The PCF is then defined as

$$g(r) = \frac{\varrho(r)}{\lambda^2}. \quad (5.6)$$

For Poisson processes, there are no correlations between the point locations and thus  $p(r) = \lambda dx \lambda dy$ , which implies that  $g(r) = 1$ . Generally, the shape of the PCF depends on the clustering and repulsion among the points. It can be shown that as  $r \rightarrow \infty$ ,  $g(r) \rightarrow 1$ . For many point sets, there is a finite  $r_c$  such that  $g(r) = 1$  for  $r > r_c$  and hence, most information about the point set is contained in  $g(r)$  for the lower values of  $r$ .



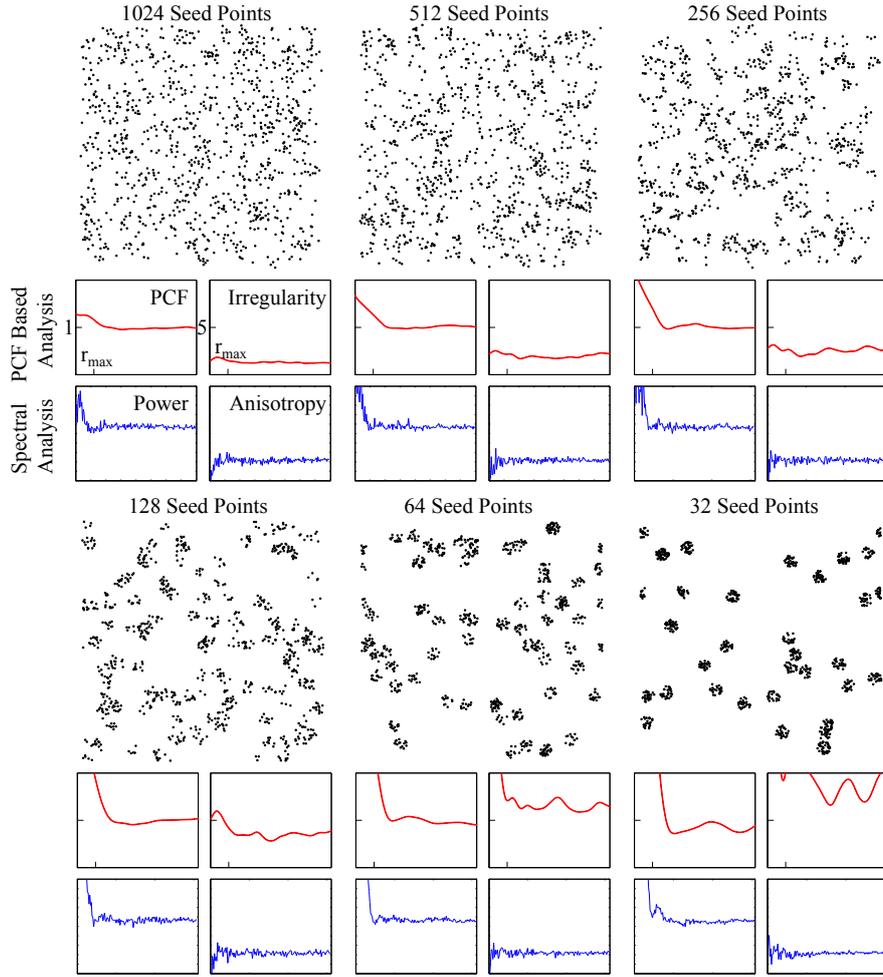
**Figure 5.2:** PCF, irregularity, and radial spectral measures for distributions generated by various algorithms. All point sets have approximately 1024 points. For the PCF and irregularity graphs,  $r \in [\sigma, 5] r_{max}$ .

### 5.3.2 Estimation of the PCF

In order to estimate the PCF, the intensity and product density should be estimated. The trivial way to estimate the intensity is dividing the number of points by the volume of the observation region, that is  $\hat{\lambda} = n/|V|$ , which provides an unbiased estimator. Estimating the product density is more involved and window edge effects should be taken into account. In practice, edge effects are less important when hard-core processes are considered.

We adapt an estimator designed for isotropic distributions [Ohser and Mcklich, 2000; Illian et al., 2008]. Disregarding the window edge effects, the estimator can be given by

### 5.3 Analysis of Point Distributions



**Figure 5.3:** *PCF, irregularity, and radial spectral measures for clustering distributions generated by the Matérn clustering process with clustering radius  $0.8r_{max}$ . All point sets have approximately 1024 points. For the PCF and irregularity graphs,  $r \in [\sigma, 5]r_{max}$ .*

$$\hat{g}(r) = \frac{|V|}{|\partial V_d| r^{d-1} n^2} \sum_{i \neq j} k_\sigma(r - d(\mathbf{x}_i, \mathbf{x}_j)). \quad (5.7)$$

Here  $|\partial V_d|$  denotes the volume of the boundary of a unit sphere in a  $d$  dimensional domain, and  $d(\mathbf{x}_i, \mathbf{x}_j)$  is its distance measure. We use the Gaussian kernel  $k_\sigma(x) = \frac{1}{\sqrt{\pi\sigma}} e^{-x^2/\sigma^2}$  in our estimators.

This estimator makes it clear that the PCF boils down to a density estimation of the distribution of the distances. The inverse weighting by the term  $|\partial V_d| r^{d-1}$  normalizes the distribution by the volume of an infinitesimal spher-

ical shell of radius  $r$  since there will be naturally more distances at larger radiuses.

In Figures 5.2 and 5.3, PCFs of some point distributions are plotted. Unlike the spectral measures, the smoothing level we set makes the PCF estimates smooth and indistinguishable for different instances of the same distribution. This property is important if only a single distribution is all we have to extract the properties of the underlying point process.

**Relation to Other Analysis Methods.** The power spectrum and the differential domain analysis of Wei and Wang [2011] are two proposed tools for the analysis of distributions in computer graphics. We also utilized the power spectrum to analyze the distributions obtained in Chapter 4. It can be shown [Wei and Wang, 2011] that the power spectrum is the cosine transform of the function used for differential domain analysis. Hence, both contain the same information. The power spectrum and the differential domain function are computed using the difference vectors  $\mathbf{x}_i - \mathbf{x}_j$  (assuming  $\mathbb{R}^d$  for simplicity). The differential domain function can be defined in terms of the probability density function of these difference vectors, which implies that power spectrum also depends on this density. For analysis, radial averages around the origin are computed to generate  $\mathbb{R} \rightarrow \mathbb{R}$  functions via a histogram with a bin for each concentric shell. The variation of the function values within each bin is defined as anisotropy, which measures the deviation of the distribution from being isotropic. Thus, the main statistics used for discriminating isotropic distributions are the radial mean plots.

The estimator  $\hat{g}$  approximates the probability density function of the magnitude of the difference vectors,  $\|\mathbf{x}_i - \mathbf{x}_j\|$ . This means it contains the same information as the radial averages for the mentioned analysis methods, and one can be obtained from the other. For isotropic distributions, the magnitudes of the difference vectors are the only quantities that need to be used in a 2nd order statistic, since they are rigid motion invariant.

**Parameters.** The most important parameter of the estimation is  $\sigma$ . In point process statistics, there is no general consensus on how to choose this parameter [Illian et al., 2008]. Small values will cause fluctuations in the density estimation and make the estimator change from one instance of a point distribution to another. Although this is desired for the analysis of a particular instance, obtaining a general characteristic for a particular type of distributions requires to choose a certain degree of smoothness. Another parameter is the range  $[r_a, r_b]$  of the  $r$  values. This range should ideally capture enough of the characteristics to distinguish different processes without redundancy.

In order to define these parameters in relative terms, we first normalize the distances by the distance  $r_{max}$  defined as the minimum distance between pairs of points for the maximum packing of points in a given volume [Lagae and Dutré, 2008; Gamito and Maddock, 2009]. This normalization ensures that the number of points and the volume considered do not affect the PCFs. We then assign  $\sigma = 0.25$  and  $r_b = 2.5$  or  $5$ , for all results obtained in this chapter (several PCF plots in the figures have  $r_b = 5$  for illustration purposes). In our experiments, values  $\sigma \in [0.1, 0.5]$  and  $r_b > 2.0$  provided good results. The lower limit  $r_a$  cannot be set to 0 due to the numerical problems. Although there exist solutions such as the reflection method to accurately handle the unstable range  $[0, \sigma]$  [Illian et al., 2008], we refrained from using them for simplicity of the algorithms and expressions. In practice, we used values as low as  $r_a = 0.01\sigma$  in our algorithms without problems. All plots in the figures also use this value for  $r_a$ , unless stated otherwise. Finally, a sampling of  $r$  should be specified to reliably capture the shape of the estimator for the given smoothness level. We use a simple regular sampling with a spacing of 0.05 between the samples.

### 5.3.3 The Pair Correlation Space

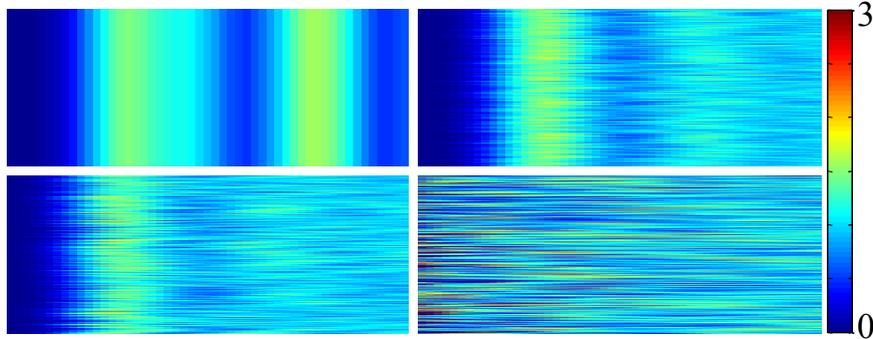
Equation 5.7 can be interpreted as the average of distance distributions for each point

$$\hat{g}(r) = \frac{1}{n} \sum_i \hat{g}_i(r), \quad (5.8)$$

where  $\hat{g}_i(r) = \frac{|V|}{n|\partial V_d| r^{d-1}} \sum_{j \neq i} k_\sigma(r - d(\mathbf{x}_i, \mathbf{x}_j))$ . This implies that we can consider all  $\hat{g}_i(r)$ 's to describe the distributional characteristics instead of just using the mean given by the PCF. Although one can work in the functional space, for the convenience of the exposition and its connection to the algorithms we use in practice, we consider a discretized PCF such that the functions are evaluated at discrete radiuses  $r_1 \cdots r_{n_r}$ . With this discretization, one can define the vectors  $\phi_i = [\hat{g}_i(r_1) \cdots \hat{g}_i(r_{n_r})]^T$  and their mean  $\bar{\phi} = \frac{1}{n} \sum_i \phi_i$  to describe the distribution of the points. We call the space where  $\phi_i$ 's live as the *pair correlation space* (PCS).

Since the PCS only depends on the distribution of the distances, it is rigid motion invariant. As long as the same discretization for  $r$  is used, one can map different point sets into the same space. Hence, each point set is described by a distribution of vectors in this space and the empirical mean of this distribution is given by  $\bar{\phi}$ . Precise matching of the distributional characteristics requires

matching of the probability distributions of  $\phi_i$ 's. However, as we will see in the next section, the distribution of  $\phi_i$ 's, the mean  $\bar{\phi}$ , and *regularity* are highly correlated.



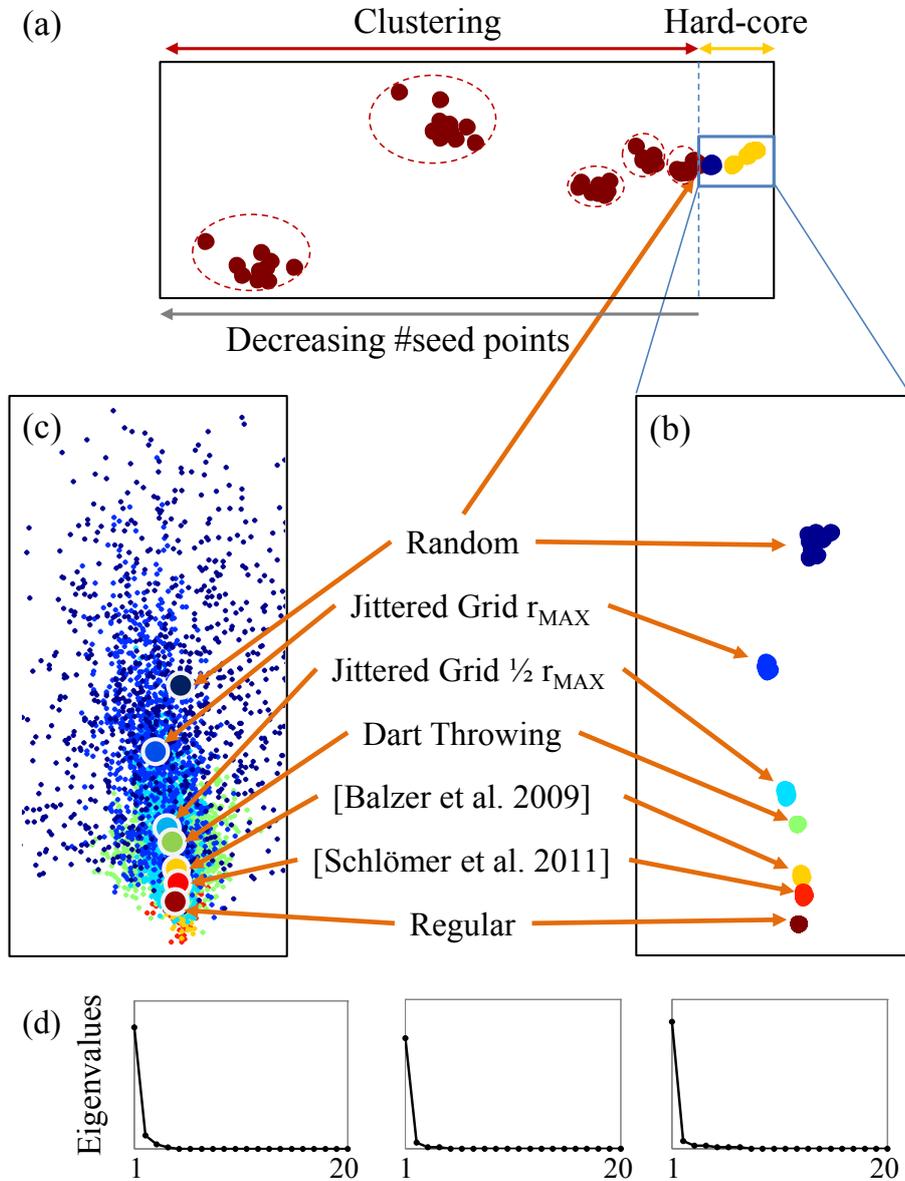
**Figure 5.4:** Vectors in the PCS for different distributions (left to right, top to bottom: Regular grid, dart throwing, [Balzer et al. 2009], and random distribution) are plotted. The  $r$  changes along the  $x$  axis, and index of points along the  $y$  axis.

### 5.3.4 Characterization of the Distributions in the PCS

In general, matching probability distributions requires parameter estimates that can be costly. In the high dimensional PCS, this will manifest itself as computational and algorithmic complexity when one tries to match point distributions. On the other hand, we expect that the mean  $\bar{\phi}$  might properly capture all characteristics of point distributions since it is widely used in many fields. To provide evidence for this conjecture, we perform an analysis of the distributions of the vectors in the PCS.

In Figure 5.5, PCS vectors for different distributions are plotted. We embedded the vectors into a 2D space using principle component analysis computed on all  $\phi_i$ 's (c) and means  $\bar{\phi}$  (a and b) of 10 instances of distributions with approximately 1000 points. To generate the clustered distributions, we used the Matérn clustering process with different seed points and radiuses of clusters. This process is simulated by seeding a number of points in space following a Poisson distribution and generating clusters of points around those seeds within a given clustering radius uniformly. The number of points in the clusters follows a Poisson distribution. The mean of this distribution is set to the number of points desired (1000) divided by the number of seed points.

The magnitude of the eigenvalues (d) clearly show that the data can be well explained using only 2 dimensions. Figure 5.5 (a) illustrates that there is a clear separation between clustering and hard-core processes, with the random



**Figure 5.5:** (a) 2D embedding of the vectors  $\bar{\phi}$  in the PCS for both clustering and hard-core point processes with 10 different distributions for each different point process. (b) Embedding of the hard-core part. (c) Embedding of the vectors  $\phi_i$  for the hard-core distributions. (d) Left to right: the eigenvalues of the covariance matrices for the three embeddings in (a), (b), and (c), respectively.

distribution on the border. For clustering processes, decreasing number of seed points for clusters results in scattering of  $\bar{\phi}$ 's. For hard-core processes, going from random to more regular point distributions, the variance of the  $\phi_i$  (c) and scattering of  $\bar{\phi}$  vectors (b) diminish and in particular, for the regular grid, the  $\phi_i$  distribution becomes a single spike. Hence, regularity and uniformity in point distributions result in less variance of the vectors in the PCS. This is further illustrated in Figure 5.4, where we plot all  $\phi_i$  vectors for different distributions. It can be clearly seen that the variance of the vectors is higher for less regular distributions.

Another interesting property of the PCS is that the  $\bar{\phi}$ 's effectively lie on a line, as can be observed from the magnitude difference between the first eigenvalue of the covariance matrices and the others in Figure 5.5 (d), and also from the embeddings in Figure 5.5 (a), (b), and (c). For the hard-core distributions, this line extends from the  $\bar{\phi}$  for random distributions to that of the regular grid. The parameter of the means on this line correctly determines the order of regularity in the point distributions. In particular, Schlömer et al.'s algorithm [2011] is found to be the most regular, followed by Balzer et al.'s algorithm [2009] and then dart throwing and different levels of jittering. This ordering matches the regularity and packing density observed for these algorithms in practice [Schlömer et al., 2011].

This analysis gives us a tool to characterize the distributions generated by different processes or algorithms. The means  $\bar{\phi}$  for a distribution are clustered with variance getting smaller as the means for different distributions start to approach each other. Hence, the  $\bar{\phi}$ 's are sufficient to discriminate different types of distributions. Furthermore, the means approximately lie on a 1D line in the high dimensional PCS, which provides an easy quantitative measure of the *closeness* of distributions generated by different algorithms. We quantify such a measure in the next section.

### 5.3.5 A Measure of Irregularity

Regularity in a point distribution can be intuitively described as the indistinguishability of the neighborhoods of points. As the distribution becomes more random, or exhibit clearly distinctive structures at different locations in space, the neighborhoods deviate more and more from each other. Following this intuition, irregularity in a distribution can be described by the vectors  $\phi_i$ , which effectively describe the neighborhoods of the points in terms of the distance distributions.

We illustrate that the variance in  $\phi_i$ 's correctly captures irregularity in Figure 5.5 (c). When points are regularly distributed, the components of different

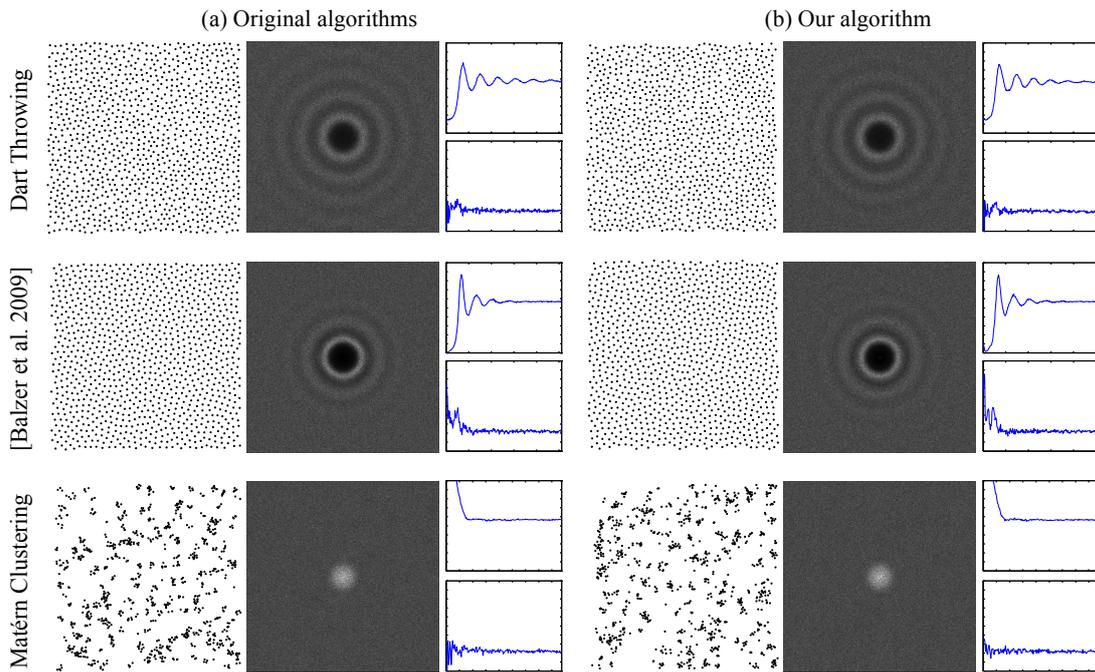
### 5.3 Analysis of Point Distributions

$\phi_i$ 's match since the distance distribution around each point  $x_i$  is the same. As more randomness is added, this regularity degrades and in the case of complete randomness, the vectors no longer correlate with each other. Clustering processes further increase irregularity by introducing different structures at different points.

This observation can be quantified as an irregularity measure as follows:

$$\iota_k = \frac{1}{n} \sum_i (\phi_{ik} - \bar{\phi}_k)^2. \quad (5.9)$$

This measure describes the observed irregularity in the point set at different radiuses  $r_k$ . In practice, we normalize this measure by the irregularity of the empirical random distribution obtained by averaging 10  $\iota$ 's. The  $\iota$  statistics (irregularity) for different distributions are plotted in Figures 5.2 and 5.3. The level of irregularity in hard-core and clustering processes considered exactly match the order and scattering of the means  $\bar{\phi}$  in Figure 5.5 (a) and (b), which shows that  $\iota$  is also an accurate measure of closeness in the PCS.



**Figure 5.6:** Given a single example point set generated by different algorithms shown on the top, our algorithm can generate new point sets with matching spectral characteristics. Average spectra of 10 distributions are shown.

## 5.4 Synthesis of Point Distributions

All statistics for point processes are aimed at providing a good summary of the distributional characteristics. Unlike other measures such as periodograms, the PCF has a simple form and interpretation directly linked to the distribution of the distances between pairs of points. The PCS analysis we presented also shows that it characterizes point distributions well. In this section, we build on these properties and propose two simple synthesis algorithms that use the PCF as a global statistic. The first one relies on simple random sampling and can provide an initial distribution for the second relaxation based method. We assume that a target PCF  $g_0(r)$  is computed using one or more examples, or given by the user. The goal of the algorithms is to transform an arbitrary input point set such that its PCF matches the target PCF. The reader will see similarities between the algorithms explained below and the algorithms proposed in Chapter 4. Indeed, our aim is to generalize the commonly used algorithmic techniques for generating blue-noise distributions to general distributions.

### 5.4.1 Generalized Dart Throwing

Our first algorithm generalizes the well-known dart throwing algorithm. In the original algorithm, at each step a random point is generated. If the distance of this point to the closest point in the set of already accepted points is smaller than desired, it is rejected and otherwise it is accepted. From this definition of the algorithm, it is clear that only the lower end of the PCF is used in the decision step. In order to extend this algorithm for a given target PCF  $g_0$ , one can simply impose the condition that at any given iteration,  $g(r_k) \leq g_0(r_k) \forall r_k$ . Here,  $g(r)$  is normalized by the target number of points.

The pseudo code of the resulting algorithm is provided in Algorithm 2. The algorithm reduces to standard dart throwing if the range upper limit  $r_b$  is set to the hard-core radius between the points. In practice, we use a parameter  $\epsilon$  and relax the condition as  $\max_k(g(r_k) - g_0(r_k)) \leq \epsilon$ . The  $\epsilon$  is changed at each iteration with a user defined function. We used a simple function  $f_\epsilon(\text{iteration}) = c \text{ iteration}$ , for some constant  $c$ . Using this relaxed version avoids the expected high rejection rates and deadlocks due to infeasible configurations but also distorts the desired characteristics. Nevertheless, this algorithm provides a very good initial distribution for our gradient descent based fitting algorithm.

Input: Target  $g_0(r_k)$ , number of points  $n_T$ , dimension  $d$ , volume  $V$   
Output: Point Set  $P$   
 $P = \emptyset, g(r_k) = 0 \forall k, \text{iter} = 0$   
**while** ( $|P| < n_T$ )  
   $\epsilon \leftarrow f_\epsilon(\text{iter})$   
  generate a random point  $\mathbf{x}$  in  $V$   
  update  $g(r_k) \forall k$  for  $\mathbf{x}$  using equation 5.11  
  with  $n = n_T$  in the normalization  
  **if** ( $\max_k(g(r_k) - g_0(r_k)) \leq \epsilon$ )  
    add  $\mathbf{x}$  to  $P$   
  **else**  
    restore previous  $g(r_k) \forall k$   
  iter  $\leftarrow$  iter + 1

**Algorithm 2:** Generalized Dart Throwing

### 5.4.2 PCF Fitting by Gradient Descent

This fitting algorithm takes a random point set, or the output of Algorithm 2 as the input, and tries to minimize the least squares fitting error  $E(\mathbf{x}_1, \dots, \mathbf{x}_n) = \int_0^\infty (g(r) - g_0(r))^2 dr$ . This corresponds to a least squares fitting of the means  $\|\bar{\phi} - \bar{\phi}_0\|^2$  in the PCS. Since we discretize  $r$  to  $r_k$ 's, the integral turns into a sum and the normalized gradient with respect to a point  $\mathbf{x}_m$  can be computed as follows:

$$\Delta_m = -\frac{\sum_{i \neq m} \mathbf{u}_{mi} w_{mi}}{|\sum_{i \neq m} w_{mi}|}, \quad (5.10)$$

$$w_{mi} = \sum_k \frac{g(r_k) - g_0(r_k)}{r_k^{d-1}} (d_{mi} - r_k) k_\sigma(d_{mi} - r_k).$$

Here,  $\mathbf{u}_{mi} = \nabla_{\mathbf{x}_m} d_{mi}$  (the unit vector from  $\mathbf{x}_i$  to  $\mathbf{x}_m$  for Euclidean spaces) and  $d_{mi} = d(\mathbf{x}_m, \mathbf{x}_i)$ , which is assumed to be symmetric for brevity of the expressions.

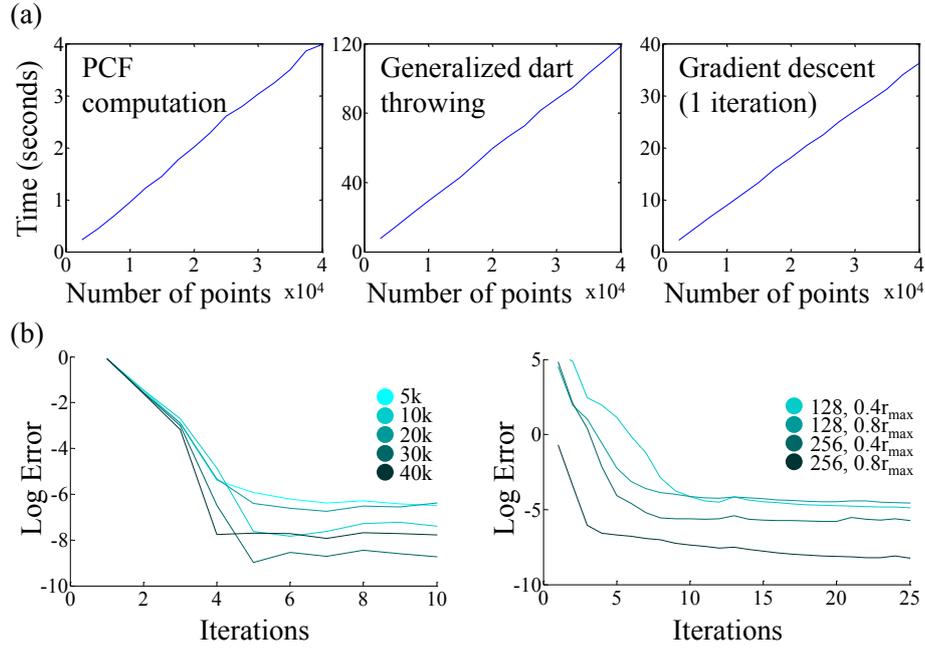
At each iteration, each point  $\mathbf{x}_m$  is moved by a gradient descent  $\mathbf{x}_m^{k+1} = \mathbf{x}_m^k - \lambda \Delta_m$ . To determine the step size  $\lambda$ , the algorithm performs a simple search by taking 5 different  $\lambda$  values  $10^{-i}$   $i = 1, \dots, 5$  and accepts the one that causes the most reduction in the error. When a random point set is used as the input, reaching convergence takes longer, but the characteristics of the output point sets are not affected significantly. However, computing the initial point set with Algorithm 2 improves convergence such that the number of iterations are not affected by the number of points [Schmaltz et al., 2010].

### 5.4.3 Analysis

**Updating the PCF** In both algorithms, the PCFs should be updated after each operation of point insertion or movement. The effect of a point  $\mathbf{x}_m$  on the PCF can be computed using the following formula

$$\delta_k(\mathbf{x}_m) = \frac{|V|}{|\partial V_d| r_k^{d-1} n^2} \sum_{i \neq m} k_\sigma(r_k - d_{im}) + k_\sigma(r_k - d_{mi}). \quad (5.11)$$

As an example, when moving a point,  $\delta_k(\mathbf{x}_{new}) - \delta_k(\mathbf{x}_{old})$  should be added to the PCF.



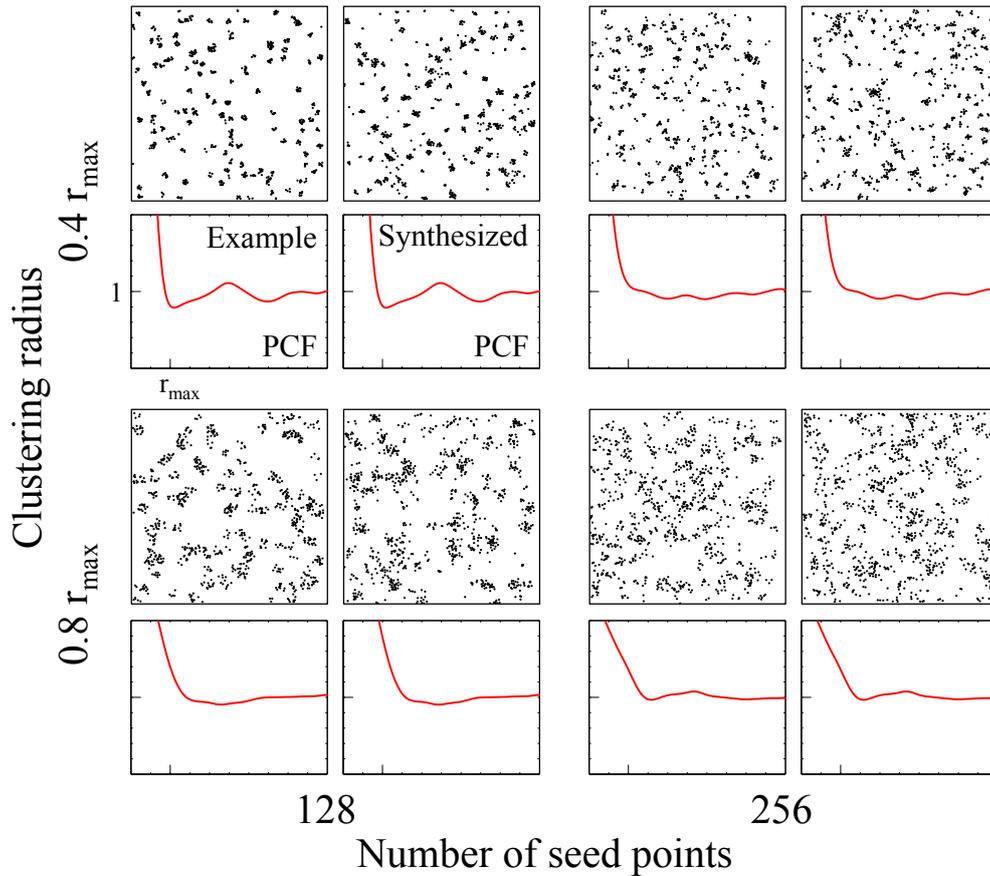
**Figure 5.7:** (a) Running times of the algorithms as a function of the number of points. (b) Convergence of the gradient descent algorithm for (left) hard-core distributions with different number of output points, and (right) for the distributions in Figure 5.8.

**Time Complexity** The time complexity of the algorithms depends on the number of iterations to convergence, number  $n_s$  of  $r_k$  samples, and the neighborhoods used when computing the gradient of the PCF and the PCF itself. The maximum neighborhood size depends on two factors,  $\sigma$  and  $r_b$ , and can be precisely given as  $r_{max}(r_b + \zeta\sigma)$  where  $\zeta$  is the cutoff factor used for the Gaussian kernels. For isotropic point sets,  $r_{max} = c_d\lambda^{-1/d}$  for a constant  $c_d$  that depends only on the dimension. The number of points in a hypersphere of radius  $r$  in  $d$  dimensions is given by  $r^d|V_d|\lambda$  with  $|V_d|$  the volume of a unit sphere. Substituting the expression of the maximum neighborhood size, we get that number of points in the neighborhoods is  $\alpha_d^d|V_d|$  with  $\alpha_d = c_d(r_b + \zeta\sigma)$ , which only depends on the dimension  $d$ . Hence, updating the PCF or computing the gradient of the PCF for a single point takes constant time with respect to the number of points, resulting in  $O(n)$  complexity for all algorithms. We verify the linear complexity of our algorithms and illustrate the convergence of the gradient descent algorithm in Figure 5.7 (a) and (b), respectively. Convergence of the gradient descent algorithm is independent of the number of points and 5 iterations for hard-core and 10 for clustering distributions are sufficient to get accurate characteristics.

#### 5.4.4 Extensions

**Multi-class Sampling** In point process statistics, marked point processes are used to describe point sets where the points have associated properties in addition to locations [Illian et al., 2008]. Each point can have a qualitative, discrete, or continuous mark. When discrete marks are used to indicate different classes of points, the pair correlation functions are extended to include correlations between classes such that one has the interclass pair correlations  $g_{ij}(r)$  for all classes  $i$  and  $j$ . The exact form of  $g_{ij}(r)$  depends on the mark correlation functions chosen. Since interclass correlations are coupled, minimization of an energy that involves all  $g_{ij}(r)$ 's will be unnecessarily complex. Instead of using all interclass correlations, we include only the intraclass pair correlations in the energy function to be minimized such that  $E_{total} = E + \sum_i E_i$ , where  $E$  is defined in Section 5.4.2 and  $E_i$  is computed using only the points in class  $i$ . In our experiments, this produced comparably accurate reconstructions and also reduced the time complexity and number of iterations of the algorithms.

**Adaptive Sampling** By adjusting the distance measure used, one can easily extend the algorithm to generate adaptive isotropic or anisotropic samplings. Here, we concentrate on adaptive isotropic samplings where the distance is given by  $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|/s(\mathbf{x})$  for a scaling function  $s$ . Ignoring the change in the scaling function [Fattal, 2011], the expressions for the PCF and its gradient change trivially. Given an arbitrary importance function  $f(\mathbf{x})$ , the scaling function can be set as  $s(\mathbf{x}) \propto f(\mathbf{x})^{-1/d}$ .

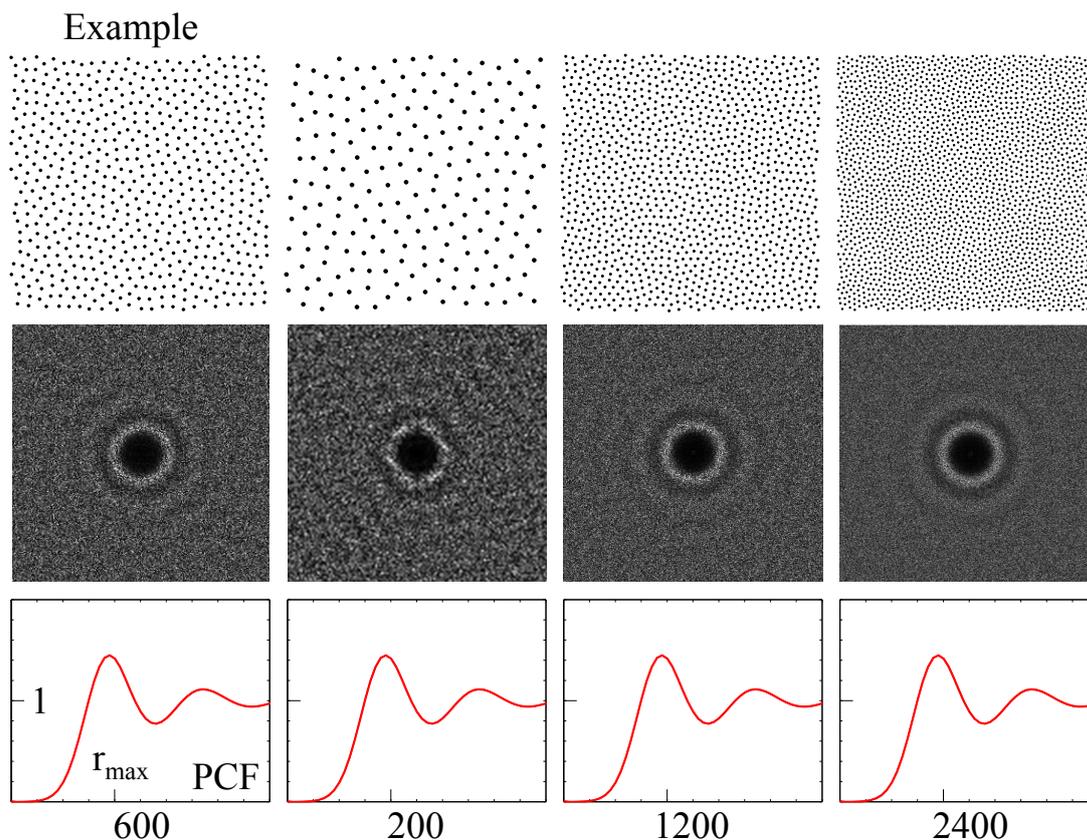


**Figure 5.8:** Reconstruction of clustering patterns generated by the Matérn clustering process with different number of seeds and clustering radiuses.

## 5.5 Results

As the first experiment, we tested if our algorithms can reproduce the characteristics of distributions generated by existing algorithms. In Figure 5.6, top, we show averaged spectra of 10 distributions generated by dart throwing, Balzer et al.'s algorithm [2009], and Matérn clustering process. For each algorithm, we used a single distribution generated as the example (shown in the figure), and ran our algorithm 10 times using the same example. The spectra of the generated distributions (Figure 5.6, bottom) almost exactly match those of the distributions by the original algorithms. As a further test, we generated points from the Matérn clustering process with different number of seed points and clustering radiuses in Figure 5.8. For all cases, our algorithm could accurately reproduce the same characteristics. We used  $r_b = 5$  for the clustering processes to get more accurate results.

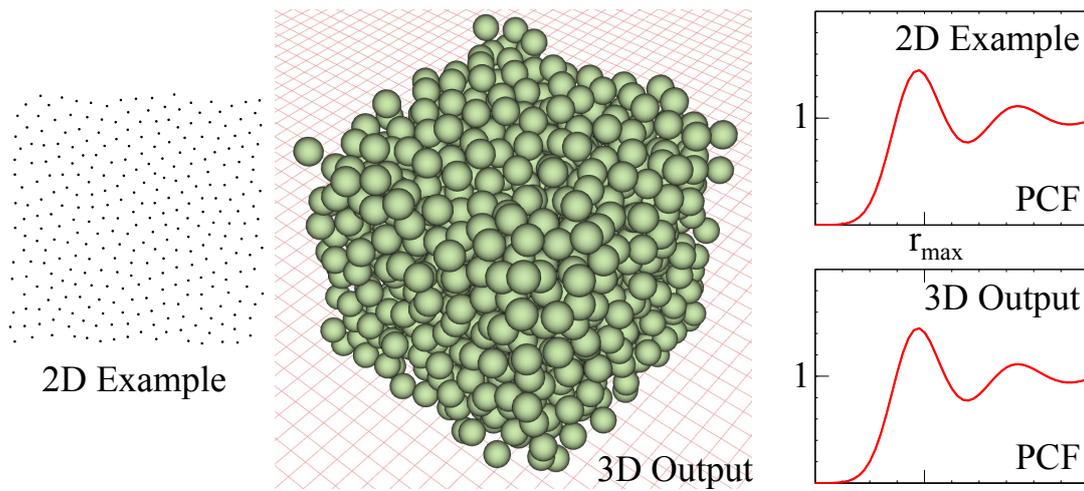
Multi-class distributions can also be accurately reconstructed by our method.



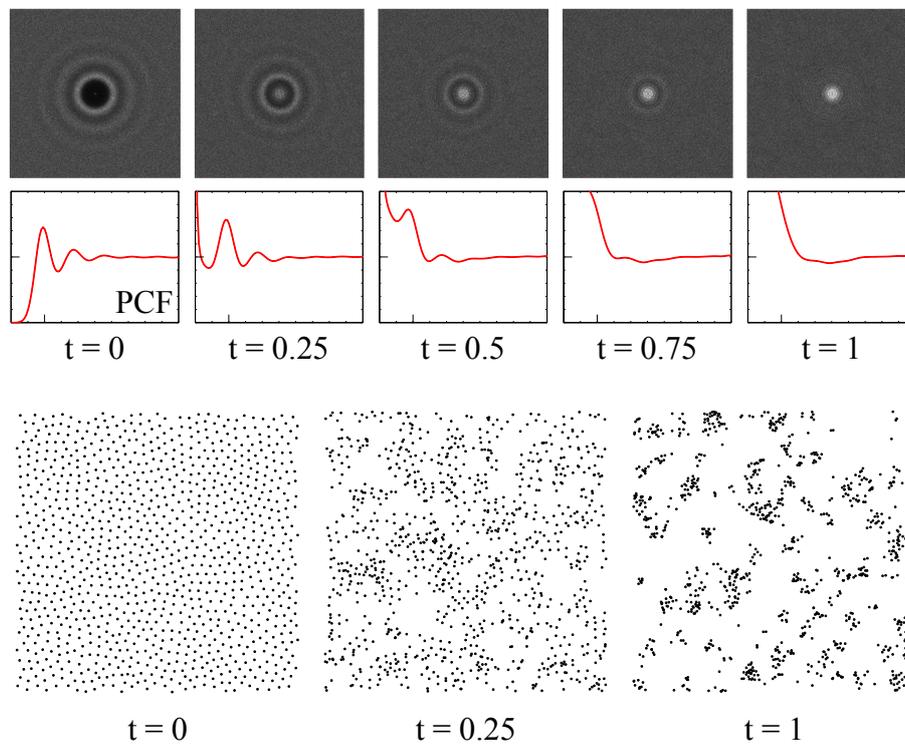
**Figure 5.9:** Given an example point set with 600 points, our algorithm can generate point sets with any number of points.

This is illustrated in Figures 5.14, 5.13, and 5.15 (a, b). Intra-class as well as overall PCFs are well-preserved. The distributions of points in different classes can have very different characteristics. In Figure 5.14, although a small number of points from each class are used as examples, accurate results are obtained for all four classes.

As illustrated in Figure 5.9, our algorithm does not need the example sample size to be the same as the output size. An example point set with 600 points is sufficient to generate outputs of various sizes with identical characteristics. Yet, since the PCF depends on a density estimation of distances, for extremely small point sets, there can be unwanted fluctuations. We experimentally found out that for  $r_b = 2.5$ , example sets of around a hundred points are sufficient to produce accurate PCFs for 2 dimensions. Dimensionality of the example and output point sets can also be set arbitrarily since the PCF is defined for any dimensions. We show an example in 3 dimensions in Figure 5.10.

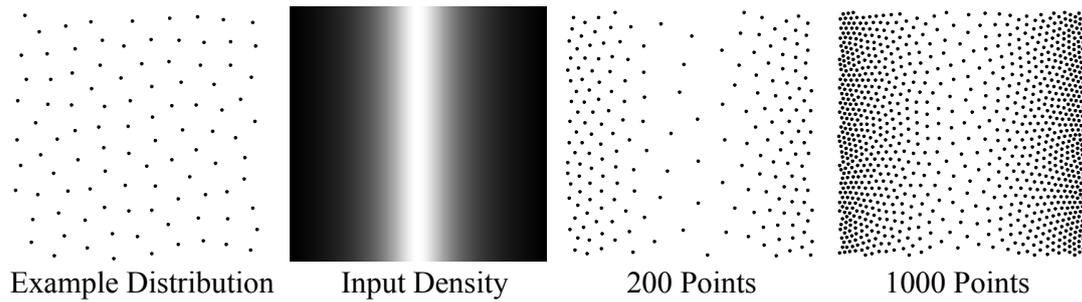


**Figure 5.10:** A 3D point set generated using a given 2D example and their PCFs.



**Figure 5.11:** By interpolating PCFs of point sets generated by Balzer et al.'s algorithm [2009] and Matérn clustering process, distributions with novel characteristics can be obtained. Average spectra of 10 distributions are shown.

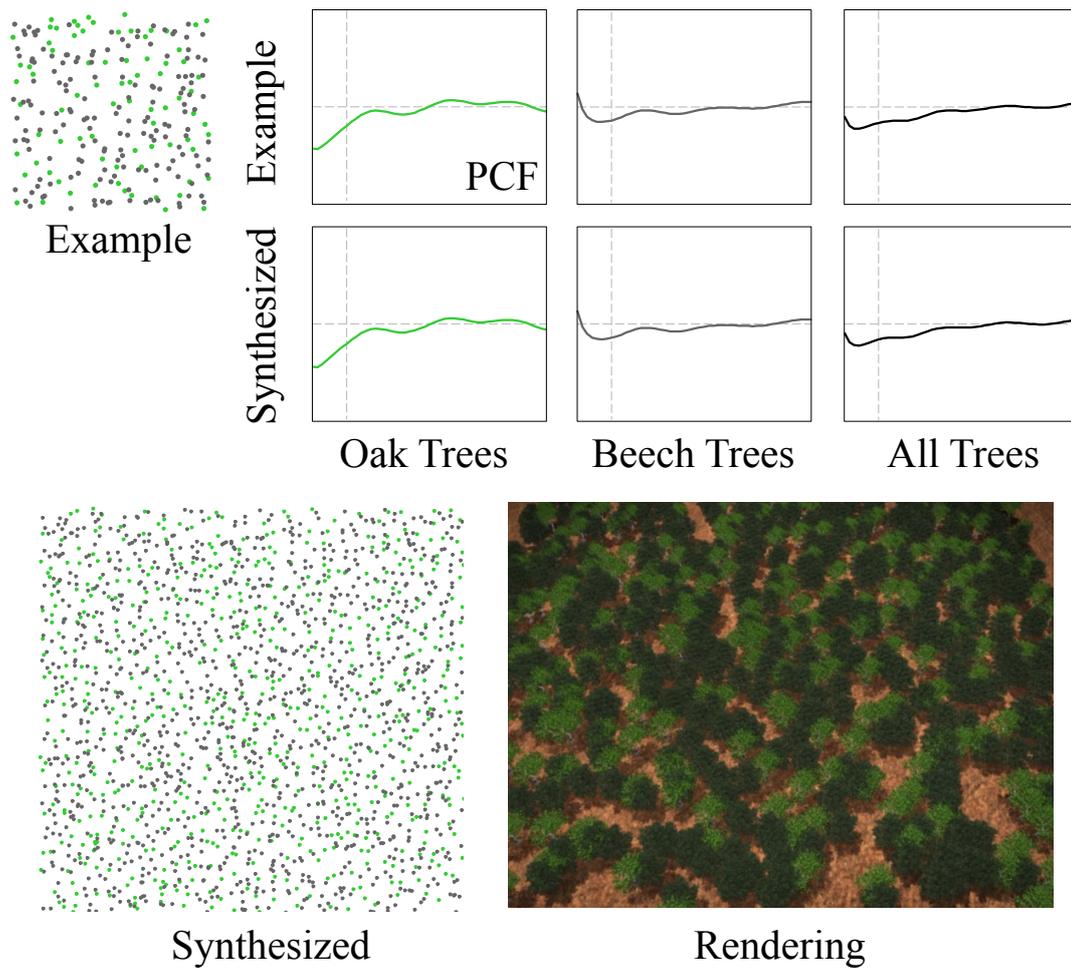
By interpolating the PCFs, a family of distributions with novel characteristics can be obtained. Since the PCFs effectively lie on a line as shown in



**Figure 5.12:** *An example distribution and input density can be combined for adaptive sampling.*

Figure 5.5, simple linear interpolation can generate valid PCFs from which distributions can be synthesized. An interesting application of this interpolation is to combine hard-core and clustering distributions. Results of such an experiment are shown in Figure 5.11. The PCF  $\bar{\phi}_0$  of a hard-core (Balzer et al.'s algorithm [2009]) and that  $\bar{\phi}_1$  of a clustering (Matérn clustering process) distribution are interpolated as  $\bar{\phi}_0(1 - t) + \bar{\phi}_1 t$  and new distributions having the interpolated PCFs are synthesized (with  $r_b = 5$ ). As the parameter  $t$  is changed, the evolution of the spectra and point distributions can be clearly seen.

Finally, density adaptation of our algorithm is illustrated in Figure 5.12 for different number of points. Given an example sample distribution and an input density, our algorithm can generate adaptive samplings with characteristics matching the example, and density matching the input density.



**Figure 5.13:** Given a small example distribution of two types of trees, our algorithm can generate a bigger forest with the same characteristics. For the PCF graphs,  $r \in [\sigma, 5] r_{max}$ .

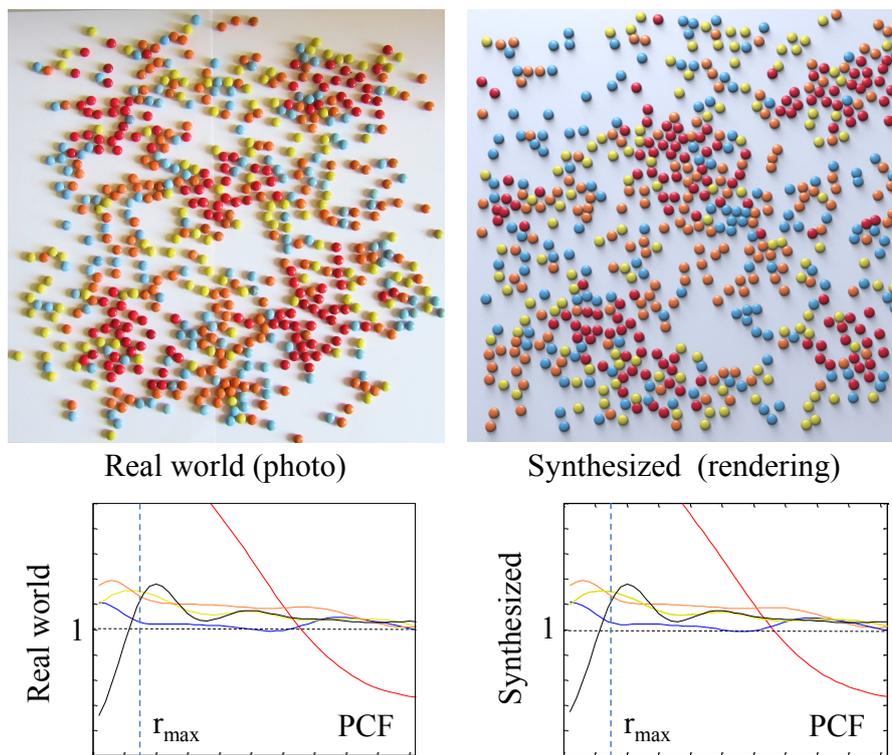
## 5.6 Applications

### 5.6.1 Synthesizing Real World Structures

Many entities in the real world have distributions with particular characteristics. Often, these distributions have various scales of clustering and repulsion, and interclass as well as intraclass interactions, leading to interesting patterns. We applied our algorithms to learn PCFs of real world distributions and generate new distributions with the same characteristics.

An example synthesis result is shown in Figure 5.14, where a photo of candies with four different colors is used to generate the example PCFs (the black

curve in each PCF graph is the PCF of all points). As shown in the PCF graphs, our algorithms are able to precisely match the characteristics of all the distributions. This is also apparent in the rendering, where red candies are in clusters, and all candies as well as candies of the same colors are distributed with a hard-core distance between them. A further example is shown in Figure 5.13, where data gathered from a real distribution of oak and beech trees [Pommerening, 2002] is used as the input example. Our synthesis algorithm can precisely reproduce the characteristics, which allows us to construct and render a bigger forest following the real world distributions.

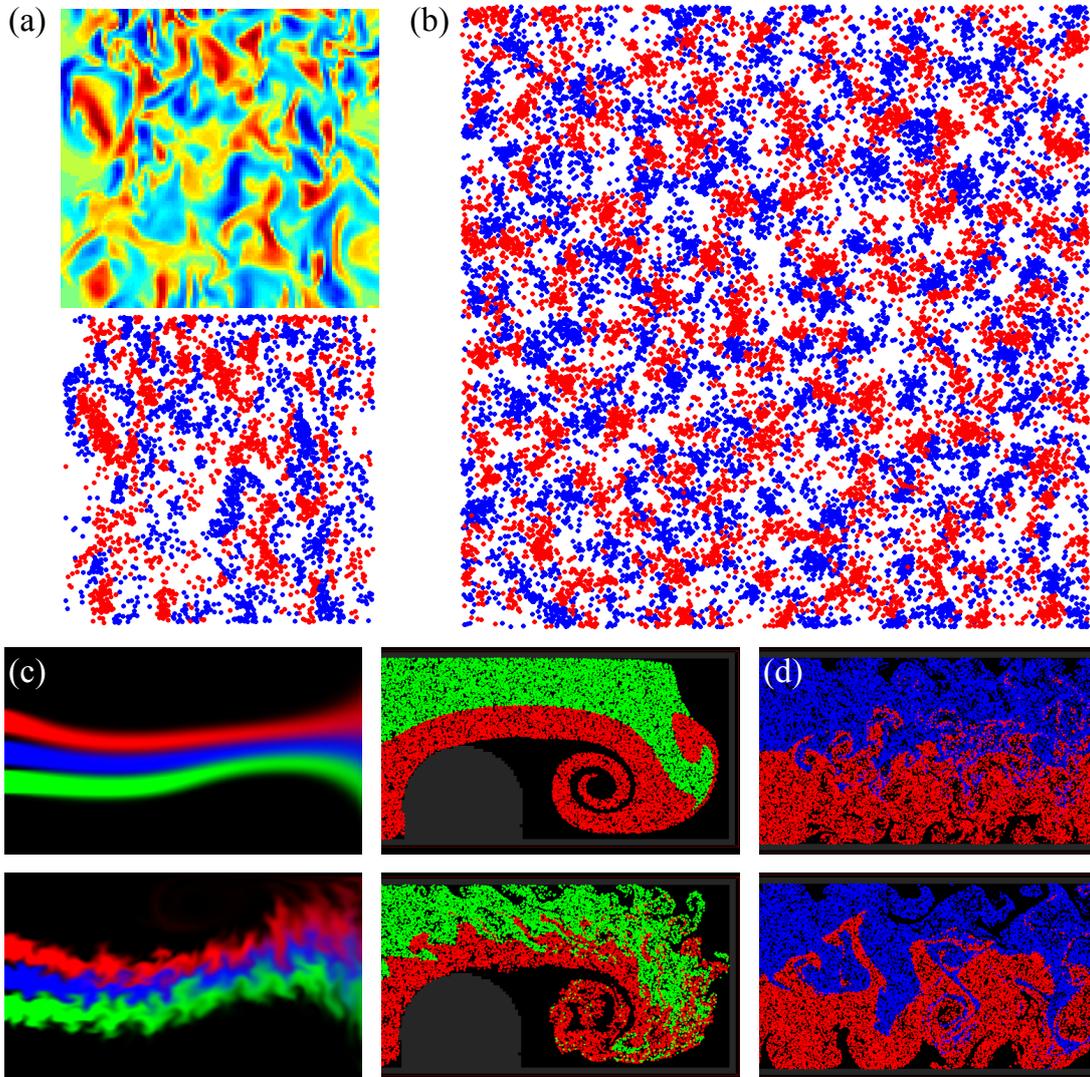


**Figure 5.14:** A new distribution of candies is generated based on a real world distribution extracted from a photograph.

### 5.6.2 Turbulent Fluid Simulations

In fluid simulations, methods for detail enhancement have become popular in recent years. These methods augment a low-resolution base simulation with synthetic sub-grid detail, by e.g. applying a curl noise turbulence texture [Kim et al., 2008]. While the detail structure is known for the special case of homogeneous, fully-developed turbulence in which Kolmogorov’s law holds, turbulent details in real flows tend to be more complex, due to anisotropic

effects and transition processes. It would therefore be interesting to extract the fingerprint of a set of given reference turbulent flow simulations, and use this as a basis for detail synthesis on top of other arbitrary flows.



**Figure 5.15:** (a) Top: a vorticity field extracted from a reference high-resolution simulation, and bottom: an example distribution with negative (blue) and positive (red) particles generated according to the field. (b) Synthesized particle distribution. (c) Top: base simulations, and bottom: base simulations with the synthesized vorticity. (d) Using different example fields leads to different behavior.

Our method allows us to achieve this transfer directly in a Lagrangian manner, using a very compact fingerprint. A small vorticity field from a real high-resolution simulation is taken as the input (Figure 5.15 (a), top). This field

is positive (red) in some regions and negative (blue) in others. The sign determines the direction of rotation (clockwise or counterclockwise), and the absolute value determines how strong the vorticity is, i.e. how much the fluid is rotating in that region. The field is separated into a negative and a positive field, and for each, a set of vortex particles is randomly placed according to the absolute value of the density field (Figure 5.15 (a), bottom). Next, we compute the PCFs for this two-class distribution of negative and positive particles, and use our multi-class synthesis algorithm to generate a large distribution of arbitrary size and resolution (Figure 5.15 (b)). Finally, these synthesized vortex particles act as our turbulence representation [Selle et al., 2005], which means they induce small-scale rotations to the flow. Figure 5.15 (c) illustrates the base simulations and the synthesized flows using the generated vortex particles. Using fingerprints recorded from different reference simulations, we can also obtain different turbulence strength and behavior as illustrated in Figure 5.15 (d).

## 5.7 Discussion and Outlook

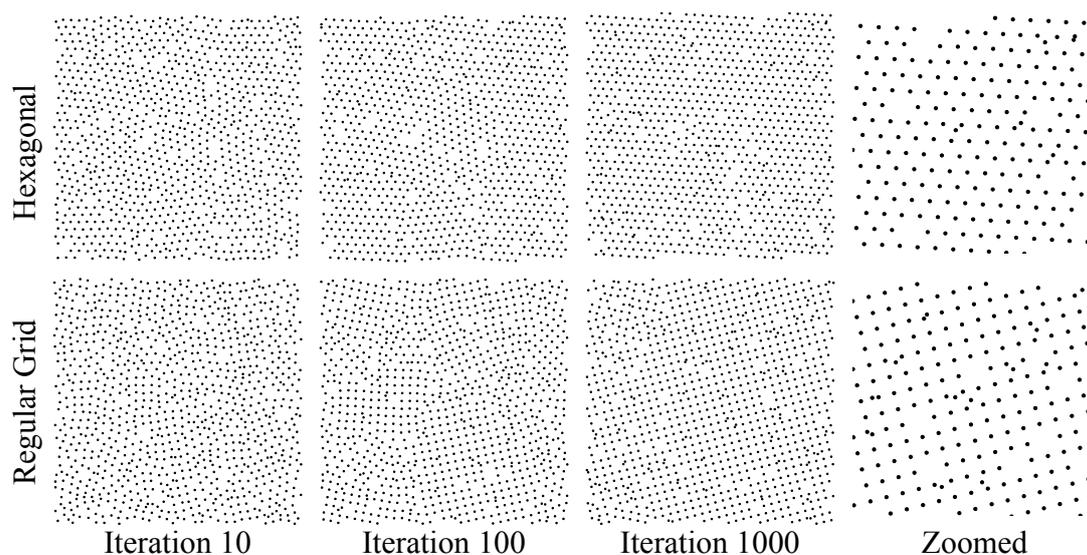
In this chapter, we introduced novel analysis and synthesis techniques for point distributions with general characteristics. We also presented several experiments and example applications where our techniques can be useful.

A fundamental limitation of utilizing the PCF is that it is only a second-order statistic depending on pairs of points and hence cannot uniquely characterize a given point pattern. However, as explained in Section 5.2.3, there is strong empirical and theoretical results stating that the PCF is sufficient to characterize most of the point processes uniquely. Nevertheless, the approach we developed can also be generalized to higher order statistics.

On a more practical level, we observed that point distributions with highly regular structures are harder to synthesize, as illustrated for the hexagonal and regular grids in Figure 5.16. Furthermore, the Poisson disk radiuses [Lagae and Dutré, 2008] of the generated blue-noise distributions are slightly lower than expected. For example, if dart throwing or Balzer et al.'s algorithm [2009] is used to produce an example distribution, the Poisson disk radiuses are in the range  $[0.67, 0.75]$  and  $[0.73, 0.74]$  with an average of 0.7031 and 0.7352, respectively. In comparison, the example distributions we used have 0.7137 and 0.7765 as Poisson disk radiuses. Due to the global nature of the PCF, our algorithms cannot precisely see the minimum of the distances, in other words, summing many small distances reduces the effect of a single distance. In addition, setting a lower limit  $r_a$  due to numerical instability causes the algorithms to not see some of the low distances. To solve this problem, a minimum distance constraint can be imposed, and estimation methods for lower  $r$  values [Illian et al., 2008] can be utilized.

We believe the proposed analysis and synthesis methods can be extended in several ways and be utilized in many other interesting applications. Although we presented results based on discrete marks, i.e. classes, the synthesis algorithms can also be extended to reconstruct distributions of points with continuous marks. This will allow to reconstruct various quantitative properties of objects such as length, size or age along with their locations. Another important direction is using space-time processes and associated statistics [Illian et al., 2008] to learn, categorize, and synthesize realistic movements of objects such as humans or animals along with marks for further properties such as gazing direction. Space-time processes can be further used to generate point samplings for dynamic data such as videos or simulations.

By approximating the distance metric of a manifold [Wei and Wang, 2011; Li et al., 2010b], our methods can also be extended to manifolds. Since the



**Figure 5.16:** *Distributions with highly regular structures are harder to synthesize.*

algorithms only operate on the lower values of  $r$ , i.e. shorter distances, local approximations of geodesic distances can be used. An interesting direction is deriving an expression that can be used in a generalization of the gradient-descent-and-projection algorithm we presented in Chapter 4, by appropriate approximations of the metric of a surface defined as in Chapter 3. This will lead to a unified treatment of general point distributions on point-sampled surfaces.

Running times of our algorithms can be significantly improved by integrating parallelization [Wei, 2008; Schmaltz et al., 2010] or tiling [Ostromoukhov et al., 2004; Kopf et al., 2006] approaches. For the generalized dart throwing algorithm, the third type of the Matérn hard-core process can be adapted for increased parallelism. This process allows parallel updates of all points at once at each iteration of the simulation. The gradient descent algorithm resembles force based synthesis algorithms and hence can be parallelized with similar techniques [Schmaltz et al., 2010].

Finally, we believe that many ideas from the point process statistics literature can be extended and adapted for various applications in computer graphics to help us better understand and computationally mimic the nature.

---

# C H A P T E R

# 6

## Conclusions

In this thesis, we proposed novel methods for processing manifolds and point patterns. We presented a manifold reconstruction method that offers a sharp feature preserving and continuous MLS based surface definition, a manifold sampling technique that results in accurate reconstructions and high quality blue-noise samplings, and point pattern analysis-synthesis/reconstruction algorithms that are the first to provide a unified treatment of general point distributions. In addition to our extensive experiments, the utility of our techniques has been tested and observed by many researchers worldwide, resulting in high recognition and number of citations. We also published the source codes resulting from our projects online.

On the way of developing our algorithms, we introduced several theoretical ideas to computer graphics. We believe that the link between MLS surfaces and LKR can be used to further improve reconstructions by integrating various statistical methods into reconstruction algorithms. We also anticipate that kernel based techniques can be utilized for improved processing of manifolds, following the idea of operating in the RKHS of the heat kernel. For point patterns, our analysis opens up a new way of looking at distributions and underlying processes. We believe that such unified techniques can lead to better understanding of possibly marked distributions in a diverse selection of fields in computer graphics.

During the course of this thesis, we tried to unify concepts and methods by

## Conclusions

establishing relations among different techniques. This unification served two purposes: it enabled us to better understand and compare different techniques, and to see new ways of improving existing methods. This effort forced us to look beyond the computer graphics literature, especially into statistics, machine learning, data analysis, spectral analysis, and physics. We observed that many fundamental ideas have been created from scratch in different fields, emphasizing different aspects and consequences. Hence, it turned out to be crucial to investigate various related fields to fully grasp the properties of particular methods. We believe that we contributed to the inter-disciplinary understanding of some concepts and methods in manifold/pattern sampling and reconstruction, and that widening of the communication channels among different fields is essential for further advancement.

### 6.1 Wider Outlook

In addition to the future directions and extensions we have mentioned, we envision several other research directions in the scope of geometry and pattern processing.

Due to the availability of vast sources of data, fusing and utilizing the diverse variety of information for manifold processing is increasingly becoming a necessity. The kernel based methods we have presented are particularly suited for such tasks, by enhancing the kernels with different types of data. Indeed, this has been the focus of *kernel engineering* in other fields, which amounts to designing kernels and the associated feature spaces such that the information at hand is used optimally.

Another fruitful idea that has been started to be explored recently is abandoning the local approximation methods for a global and structure-aware reconstruction. This development follows the non-local filtering approaches in, for example, image processing, and depends on the assumption that repetitions are indispensable in almost all geometry around us (see Section 2.1.4 for some examples of these methods). This idea can be further extended to propose *glocal* methods, that is, methods that integrate global structure preservation into local approximations to achieve continuous, analytic, efficient, and flexible representations. We believe that kernels provide a unifying framework for such approaches. Furthermore, by treating the repetitions in a stochastic framework, synthesis by using point pattern techniques can also be utilized.

The complexity of the geometric data has started to force the researchers to leave the manifoldness assumption. Processing non-manifold geometry can significantly widen the scope of the current methods and impact other fields

that have traditionally worked under the manifoldness assumption such as image processing and data analysis. Although there have been recent efforts for identifying non-manifold structures and their types [Dey et al., 2012], a principled approach for handling manifold mixtures and singularities is missing.

Learning manifold mixtures can also be coupled with point pattern methods for better preservation of stochastic structures. Our pattern reconstruction methods can work in arbitrary metric spaces. By extending them such that the resulting distributions of points also respect the singularities, sampling of manifold mixtures with given characteristics can be obtained. Immediate benefiting applications from such an approach will be image stippling and halftoning, where the current approaches typically assume some smooth density field, although the image itself can contain non-smooth intensity.

In general, we believe that many methods that deal with stochastic point distributions can be improved and enhanced by incorporating concepts from spatial statistics, physics, and point processes. As mentioned in Section 5.7, space-time processes and marked processes are two important subjects of study that promises a principled treatment of a wide range of problems.

Finally, we believe that perception still plays a minor role in the manifold and pattern reconstruction literature, unlike other fields such as image processing. Utilizing perceptual metrics instead of the common practice of formulating problems based on researchers' intuition can have a prominent impact on many current methods used.

## *Conclusions*

---

# A P P E N D I X



## Appendix

### A.1 Iteratively Reweighted Least Squares

Iteratively Reweighted Least Squares (IRLS) iteratively solves a non-linear problem by solving a linear system at each iteration, weighted by the residuals of the previous iteration. This allows to use arbitrary error functions generalizing the usual least squares error. By making the error function less sensitive to the outliers, robust approximation methods can be obtained.

Here, we derive the general formula for IRLS. Assume that the data is given in the form  $\{\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}$ , where  $\mathbf{x}_i$ 's reside in the domain of the function we would like to estimate robustly, and  $y_i$ 's are the corresponding values of the function. Then, we would like to solve the following minimization problem

$$\min_{\mathbf{s}} \sum \rho(\mathbf{x}_i, y_i, \mathbf{s}). \quad (\text{A.1})$$

This in fact corresponds to the general form of an M-Estimator. To make it more tractable, IRLS uses the following form:

$$\min_{\mathbf{s}} \sum \rho(f(\mathbf{x}_i, \mathbf{s}) - y_i) = \min_{\mathbf{s}} \sum \rho(r_i). \quad (\text{A.2})$$

## Appendix

Here, the  $r_i$ 's denote the residuals, that is, the deviation of the estimated function value from the expected value. Note that a least squares problem is a special case of this problem with the quadratic error,  $\rho(r_i) = r_i^2$ . For the  $\psi$ -type M-Estimators, the function  $\rho$  is differentiable and hence this minimization problem can be solved by taking derivative with respect to the parameter vector  $\mathbf{s}$  and setting it to zero:

$$\frac{\partial \sum \rho(r_i)}{\partial \mathbf{s}} = \sum \psi(r_i) \frac{\partial r_i}{\partial \mathbf{s}} = \sum w(r_i) r_i \frac{\partial r_i}{\partial \mathbf{s}} = 0. \quad (\text{A.3})$$

Here,  $\psi(x) = \frac{\partial \rho}{\partial x}$  is called the *influence function*, and  $w(x) = \psi(x)/x$ . The reason for writing the problem this way becomes apparent if we consider the weighted least squares problem:  $\min_{\mathbf{s}} \sum w(r_i) r_i^2$ . If we assume that the weights  $w(r_i)$  are constant, the solution of this weighted least squares problem corresponds exactly to that in equation A.3. This suggests that one can get the weights using the residuals from the previous iteration and solve the following least squares system  $\min_{\mathbf{s}} \sum w(r_i^{k-1}) r_i^2$ , and then update the weights according to the new residuals, and continue iteratively.

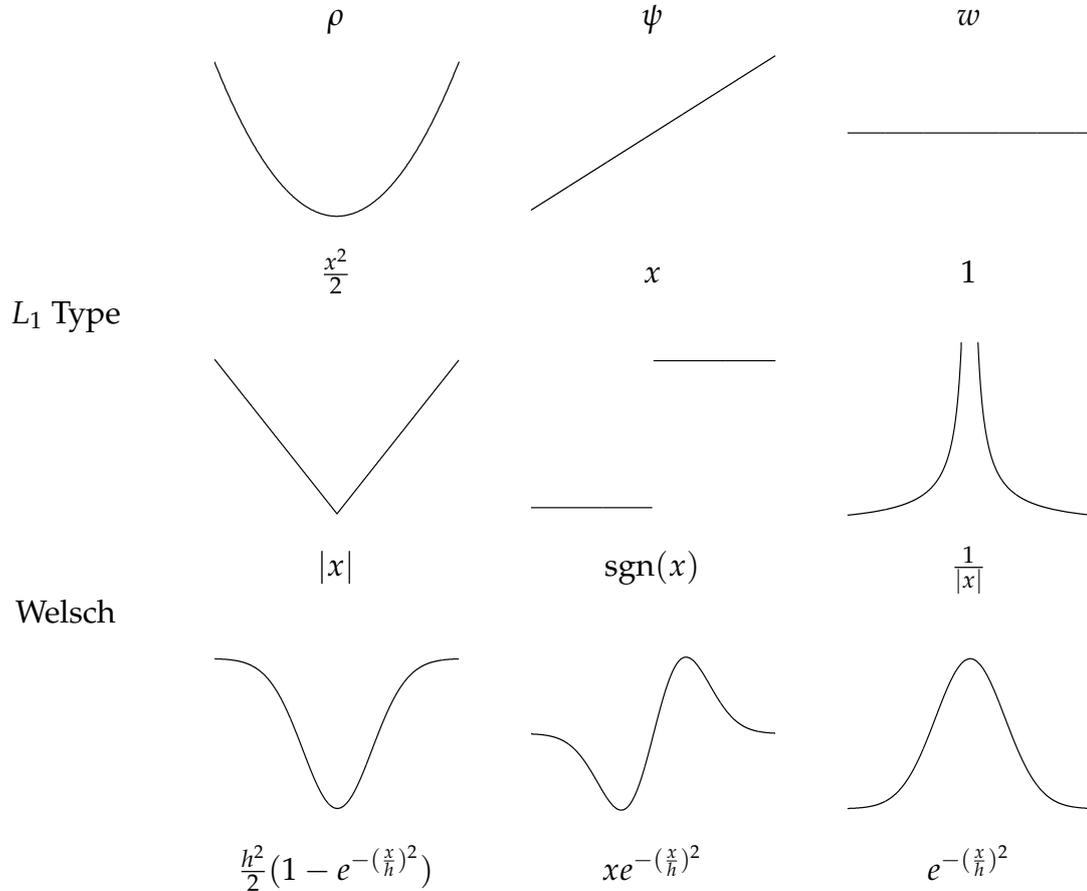
For a simple illustration of this general solution method, consider the linear function  $f(x) = ax$ ,  $f(x) : \mathbb{R} \rightarrow \mathbb{R}$ . We would like to estimate  $a$  with the IRLS. The residuals are given by  $r_i = ax_i - y_i$ , and their derivatives are  $\frac{\partial r_i^2}{\partial a} = 2(ax_i - y_i)x_i$ . Hence, the ordinary least squares solution is  $a = \sum y_i x_i / \sum x_i x_i$ . To generalize this solution for another error function with IRLS, this expression is simply replaced by  $a^k = \sum y_i x_i w(r_i^{k-1}) / \sum x_i x_i w(r_i^{k-1})$ , where  $r_i^{k-1} = a^{k-1} x_i - y_i$ . Note that all information about the error function is buried into the weight function  $w$  and thus it is trivial to implement this new minimization once the weight function is decided upon. So the main decision to make is how to choose the weight functions.

**Initial Weights.** Since this is an iterative procedure, the initial condition, i.e. the initial weight functions, should be chosen. A trivial choice is to select  $w_i^0 = 1$ , which means the system to solve for the first iteration is just an ordinary least squares system. Although it is known that this choice is not optimal [Huber, 2004; Hoseinnezhad and Bab-Hadiashar, 2007], it provides an easy and general way to handle the initial weights and performs well in practice. Hence, we use this simple choice of the initial weights.

**Weight Functions.** A more fundamental question is how to choose the weight functions  $w$ . As defined above, the weights are derived from the *objective function*  $\rho$  and the *influence function*  $\psi$ . The most important property of  $\rho(r)$  is that it should be robust to outliers, meaning that it should not grow if the residual  $r > r_0$ , where  $r_0$  depends on the expected residuals of

## A.1 Iteratively Reweighted Least Squares

Least Squares



**Figure A.1:** Some objective, influence and weight functions.

outliers. This causes that the outliers have less effect on the solution of the minimization problem. Apart from this, the following properties are often required for stable evaluation: bounded  $\psi$ , symmetric, positive-definite, and convex  $\rho$  in the parameter  $\mathbf{s}$  with a unique minimum at zero. We plot some objective, influence, and weight functions in Figure A.1.

## A.2 Pseudocode for Projecting a Point onto the Surface

```
repeat
  i = 0;
  repeat
    sumW = sumGw = sumF = sumGF = sumN = 0;
    for p in neighbors(x) do
      px = x - p.position;
      fx = dot(px, p.normal);

      if i>0 then alpha = exp(-((fx-f)/sigma_r)^2)
                    * exp(-(norm(p.normal-grad_f)/sigma_n)^2);
        else alpha = 1;

      w      = alpha * phi(norm(px)^2);
      grad_w = alpha * 2 * px * dphi(norm(px)^2);

      sumW += w;
      sumGw += grad_w;
      sumF += w * fx; sumGF += grad_w * fx;
      sumN += w * p.normal;
    end
    f      = sumF / sumW;
    grad_f = (sumGF - f * sumGw + sumN) / sumW;
    until ++i>max_iters || convergence();
    x = x - f * grad_f;
  until norm(f * grad_f) < threshold;
```

### A.3 Distribution of Effect of a Point on the Spectrum

Suppose that the eigenvectors  $\mathbf{u}_i$  of the  $n \times n$  matrix  $\mathbf{H}$  sample the eigenfunctions  $u_i$  such that  $(\mathbf{u}_i)_j = u_i(x_j)$ . Then eigenvalues of  $\mathbf{H}$  also well approximate those of  $h$  [Braun, 2006], thus we can approximately set  $\lambda_i(\mathbf{H}) = e^{-\lambda_i t}$ . If we define the eigenvector matrix as  $\mathbf{U}_{ij}^T = (\mathbf{u}_i)_j$ , it can be shown [Schölkopf et al., 1998] that there is the following relation between  $\mathbf{v}_i$  and  $\mathbf{U}$ :

$$\mathbf{v}_i = \frac{1}{\sqrt{\lambda_i(\mathbf{H})}} \sum_{d=1}^n \mathbf{U}_{id}^T \phi_d$$

Hence, we can write

$$\begin{aligned} \delta_i = (\phi^T \mathbf{v}_i)^2 &= \frac{1}{\lambda_i(\mathbf{H})} \left( \sum_{d=1}^n \mathbf{U}_{id}^T (\phi^T \phi_d) \right)^2 \\ &= \frac{1}{e^{-\lambda_i t}} \left( \sum_{d=1}^n \mathbf{U}_{id}^T h(x, x_d) \right)^2 \\ &= \frac{1}{e^{-\lambda_i t}} \left( \sum_{d=1}^n \mathbf{U}_{id}^T \sum_{k=0}^{\infty} e^{-\lambda_k t} u_k(x) u_k(x_d) \right)^2 \\ &= \frac{1}{e^{-\lambda_i t}} \left( \sum_{k=0}^{\infty} e^{-\lambda_k t} u_k(x) \sum_{d=1}^n \mathbf{U}_{id}^T \mathbf{U}_{dk} \right)^2 \\ &= \frac{1}{e^{-\lambda_i t}} \left( e^{-\lambda_i t} u_i(x) \right)^2 \\ &= e^{-\lambda_i t} u_i^2(x) \end{aligned}$$

In the fourth step, we used the fact that  $\mathbf{U}$  is a unitary matrix. This means that as long as  $u_i(x)$ 's are not simultaneously zero and for short time, each  $\delta_i$  will get a non-zero value. The zero set of  $u_i(x)$  is called the nodal set (for 2-manifolds, the nodal line) with dimension one less than the manifold and can have at most  $i + 1$  nodal domains, where its sign does not change, for  $\lambda_i$  in increasing order [Cheng, 1976]. This implies that those points where more than one  $u_i(x)$  is zero should be at the intersection of many nodal sets. Hence, for most points on the manifold, most  $\delta_i$ 's will be non-zero.

## A.4 Projection onto a Span in a Reproducing Kernel Hilbert Space

Suppose we have vectors  $\phi_i = \phi(\mathbf{x}_i)$  that form the span, and the vector  $\phi = \phi(\mathbf{x})$  to be projected onto the span. We also assume that dot product in this vector space is given by the kernel  $k$  such that  $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$ . Although it is possible to find a basis of the span and compute the norm of the projection, an easier way is using the fact that projection of  $\phi$  is the closest point of  $\phi$  on the span. Thus it is sufficient to minimize the distance of  $\phi$  to the span to compute the norm of the orthogonal component  $\|\mathbf{o}\|$ , from which norm of the projection follows as  $\|\mathbf{d}\|^2 = \|\phi\|^2 - \|\mathbf{o}\|^2$ .

We want to minimize the distance of  $\phi$  to the span, which is given by  $\|\phi - \sum a_i \phi_i\|^2$  with respect to  $a_i$ 's. Denoting the kernel matrix as  $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ , and defining the vectors  $(\mathbf{a})_i = a_i$  and  $(\mathbf{k})_i = k(\mathbf{x}, \mathbf{x}_i)$ , this minimization can be expressed as

$$\|\phi\|^2 + \|\sum a_i \phi_i\|^2 - 2\phi^T \sum a_i \phi_i = k(\mathbf{x}, \mathbf{x}) + \mathbf{a}^T \mathbf{K} \mathbf{a} - 2\mathbf{k}^T \mathbf{a}$$

which is minimized for  $\mathbf{a} = \mathbf{K}^{-1} \mathbf{k}$ . Substituting this expression into  $\|\phi\|^2 - \|\phi - \sum a_i \phi_i\|^2$  we get  $\|\mathbf{d}\|^2 = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$ .

## A.5 Notation and Glossary

### A.6 Notation

#### A.6.1 Spaces

$\Omega$ .....	General space
$\emptyset$ .....	The empty set
$\mathbb{R}$ .....	Real numbers
$\mathbb{N}$ .....	Natural numbers
$\mathbb{R}^n$ .....	The $n$ -dimensional real space
$\mathcal{H}$ .....	A Hilbert space
$\mathcal{F}$ .....	A feature space
$M$ .....	A manifold
$L^2_\mu(\Omega)$ .....	The space of square-integrable functions over $\Omega$ with measure $\mu$

#### A.6.2 Operators

$\int_\Omega f(x)dx$ .....	Integral of $f$ over a space $\Omega$ , $x \in S$
$\frac{\delta f(x)}{\delta x}$ .....	Partial differentiation of $f$ with respect to $x$
$\langle \cdot, \cdot \rangle_S$ .....	Dot product in space $S$
$\mathbf{x}^T \mathbf{y}$ .....	Dot product of the vectors $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^d$
$\tilde{f}$ .....	Approximation of a function or operator $f$
$\hat{f}$ .....	Estimator of a function or operator $f$
$\bar{x}$ .....	Empirical mean of a set of numbers $x_i$
$\ x\ _{\mathcal{I}}$ .....	The norm of $x$ in an inner product space $\mathcal{I}$
$\ \mathbf{x}\ $ .....	The vector norm of $\mathbf{x}$ in $\mathbb{R}^d$
$\mathbf{E}[x]$ .....	Expected value of $x$
$Df_{\mathbf{x}}$ .....	Differential of $f$ at $\mathbf{x}$
$\nabla$ .....	Gradient operator
$\Delta$ .....	The Laplace operator
$\Delta_M$ .....	The Laplace-Beltrami operator for manifold $M$

### A.6.3 Other Definitions

$\mu$ .....	Measure
$\lambda_i(\mathbf{M})$ .....	The $i^{\text{th}}$ eigenvalue of the matrix $\mathbf{M}$
$\sigma$ .....	Scale parameter of a kernel
$d(\mathbf{x}, \mathbf{x})$ .....	A distance function
$b_i(\mathbf{x})$ .....	The $i^{\text{th}}$ basis function of a space
$\mathbf{K}$ .....	Kernel matrix with $K_{ij} = k(x_i, x_j)$ for a kernel $k$

### A.6.4 Background and Related Work

$f$ .....	Implicit function of a surface
$k$ .....	Radial basis function
$\alpha$ .....	Coefficient in RBF based approximations
$w$ .....	Blending weight function in partition of unity
$g$ .....	Locally approximating function in partition of unity
$\mathbf{u}$ .....	Vector of coefficients in MLS
$\mathbf{b}$ .....	Vector of basis functions in MLS
$\phi$ .....	Weighting function in MLS

### A.6.5 Meshless Reconstruction of Manifolds

$f$ .....	Implicit function of a surface
$\mathbf{v}$ .....	Regular value of a function
$\mathbf{n}$ .....	Normal vector on a surface
$\mathbf{H}$ .....	Hessian matrix
$\phi$ .....	Radially decaying weight function
$\rho$ .....	Error function to be minimized
$g_{\mathbf{s}}$ .....	Local approximation parametrized by $\mathbf{s}$
$w$ .....	Weight function for robust approximation
$r$ .....	Residual in kernel regression
$\sigma_r$ .....	Scaling factor for spatial robustness terms
$\sigma_n$ .....	Scaling factor for normal robustness terms
$h$ .....	Kernel scaling factor
$k$ .....	Radial basis (kernel) function
$\mathbf{d}$ .....	Vector of known depth values

### A.6.6 Spectral Sampling of Manifolds

$u$ .....	Eigenfunction of the Laplace-Beltrami operator
$\lambda$ .....	Eigenvalue of the Laplace-Beltrami operator
$H_t$ .....	The heat operator of a manifold
$h_t$ .....	The heat kernel of a manifold
$\phi(x)$ .....	Feature vector/map for $x$
$\mathbf{L}$ .....	Normalized graph Laplacian matrix of a graph
$\mathbf{H}_t$ or $\mathbf{H}$ .....	Heat kernel matrix
$\mathbf{C}$ .....	Covariance matrix of points in a feature space
$\mathbf{v}_i$ .....	The $i^{\text{th}}$ eigenvector of $\mathbf{C}$
$\mathbf{o}$ .....	Orthogonal component of a vector to a subspace
$\mathbf{r}$ .....	Component of a vector in a linear subspace
$s(x)$ .....	The measure that quantifies the change in the Laplace-Beltrami spectrum due to the point $x$
$\mathbf{h}$ .....	Vector with entries $\mathbf{h}_i = h_t(x, x_i)$
$\mathbf{k}$ .....	Vector with entries $\mathbf{k}_i = k(x, x_i)$ for a kernel $k$
$\mathbf{D}$ .....	Diagonal normalization matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{K}_{ij}$
$\varphi(x)$ .....	Feature vector/map for $x$

**A.6.7 Reconstruction of Point Patterns**

$\mathbf{x}_s$ .....	Seed point
$V$ .....	Borel set
$N(V)$ .....	Number of points in the Borel set $V$
$ V $ .....	Volume of the set $V$
$V_d$ .....	Unit sphere in $d$ dimensions
$\delta V_d$ .....	Boundary of a unit sphere in $d$ dimensions
$P(n)$ .....	Probability mass function of $n$
$p(x)$ .....	Probability density function of $x$
$\mathbf{o}$ .....	Typical point of an isotropic point process
$b(\mathbf{o}, r)$ .....	Ball of radius $r$ centered at $\mathbf{o}$
$\Lambda(\mathbf{x})$ .....	Stochastic density
$\lambda(\mathbf{x})$ .....	Density of a point distribution
$\varrho$ .....	Product density
$dV$ .....	Volume measure
$g(\mathbf{x}, \mathbf{y})$ .....	The pair correlation function
$g(r)$ .....	The PCF for isotropic point distributions
$r_a$ .....	Lower limit for a PCF estimate
$r_b$ .....	Upper limit for a PCF estimate
$\phi$ .....	Vector in the pair correlation space
$\iota$ .....	The irregularity measure

## A.7 Glossary

APSS	.....	The reconstruction method of [Guennebaud and Gross, 2007]
AS	.....	The sampling method of [Alexa et al., 2001]
IMLS	.....	The reconstruction method of [Kolluri, 2005]
IRLS	.....	Iteratively reweighted least squares
KS	.....	The sampling method of [Kitago and Gopi, 2006]
LKR	.....	Local kernel regression
MLS	.....	Moving least squares
PCA	.....	Principle component analysis
PCF	.....	The pair correlation function
PCS	.....	Pair correlation space
PSR	.....	The reconstruction method of [Kazhdan et al., 2006]
RIMLS	.....	Robust IMLS
RKHS	.....	Reproducing kernel Hilbert space
RLKR	.....	Robust local kernel regression
RMS	.....	Root mean square
RSA	.....	Random sequential adsorption
SPSS	.....	The reconstruction method of [Adamson and Alexa, 2003]
VR	.....	The sampling method of [Valette et al., 2008]
YR	.....	The sampling method of [Yan et al., 2009]

## A.8 Curriculum Vitae

A. Cengiz Öztireli | Probsteistr. 117a 8051 Zürich | +41 76 204 27 58 | [a.cengiz.oztireli@gmail.com](mailto:a.cengiz.oztireli@gmail.com) |

### A. CENGİZ ÖZTİRELİ

#### Research Statement

My research is devoted to understanding, modelling, recreating, augmenting, and extending complex phenomena in the form of geometry, motion, and patterns; and developing tools and methods for intuitive and effective control.



#### EDUCATION

---

- |           |  |
|-----------|--|
| 2008-     | <b>ETH Zürich</b> , Switzerland<br>Ph.D. in Computer Science<br>Specialization: Computer Graphics  |
| 2006-2008 | <b>ETH Zürich</b> , Switzerland<br>M.S. in Computer Science<br>Specialization: Visual Computing<br><i>Fast-track to Ph.D.</i>  |
| 2002-2006 | <b>Koc University</b> , Turkey<br>B.S. in Computer Engineering and Electronics Engineering, <i>Double Major</i><br><i>Summa cum laude, ranked 1<sup>st</sup> in the university</i> |
| 1996-2002 | <b>AMAL High School</b> , Istanbul, Turkey<br>Specialization: Science and Mathematics<br><i>Summa cum laude, ranked 1<sup>st</sup> in the school</i>                               |

#### AWARDS & HONORS

---

- |           |   |
|-----------|---|
| 2010      | <b>Research Project Grant</b> – SNSF (Swiss National Science Foundation)<br>Granted 100,000 CHF for a research project      |
| 2009      | <b>Best Student Paper Award</b> – EUROGRAPHICS 2009   |
| 2008      | <b>Fast Track to Ph.D.</b> – ETH Zürich<br>Offered to students with excellent grades and interest in research               |
| 2006-2008 | <b>Merit based Scholarship</b> – Solidarity Fund for Foreign Students, ETH Zürich   |
| 2006      | <b>Fulbright Science and Technology Award</b><br>Granted to 25 students in the world based on academic excellence each year |
| 2006      | <b>Best Student in School</b> – Koc University  |
| 2002-2006 | <b>Vehbi Koc Scholar Awards</b> – Koc University<br>For exceptional academic performance in all semesters                   |
| 2002-2006 | <b>Full Scholarship</b> – Koc University<br>Merit based scholarship for undergraduate education, tuition and expenses       |

## Appendix

A. Cengiz Öztireli | Probsteistr. 117a 8051 Zürich | +41 76 204 27 58 | [a.cengiz.oztireli@gmail.com](mailto:a.cengiz.oztireli@gmail.com) |

- 2002 **Ranked 227<sup>th</sup>** – In the Turkish National University Entrance Exam  
Among about 1.5 million participants
- 2002 **Best Student in School** – AMAL High School

### RESEARCH & WORK EXPERIENCE

---

- 2008-2010 EU Immersence Project – **Researcher**  
Reconstruction and rendering part
- 2008- ETH Zurich – **Research and Teaching Assistant**  
Computer Graphics Laboratory – Prof. Markus Gross
- 2006-2008 ETH Zurich – **Research Assistant**  
Applied Geometry Group – Prof. Mark Pauly
- 2003 Oxygen Computer Istanbul - **Software Developer**  
Internship
- 2002-2006 Koc University – **Research Assistant**  
Robotics and Mechatronics Laboratory – Prof. Cagatay Basdogan

### SUPERVISION

---

- 2012 Marko Bencun – Master Student  
Laplace-Beltrami Operator on Point Set Surfaces
- 2011 Raphael Das Gupta – Master Student  
Sparse Representations for Geometry
- 2011 Umut Uyumaz – Master Student  
3D Modeling with a Symmetric Sketch

### TEACHING

---

- 2012 Visual computing
- 2011 Visual computing, Surface representations and geometric modelling
- 2010 Computer graphics, Surface representations and geometric modelling
- 2009 Visual computing, Physically-based simulation
- 2008 Visual computing

### JOURNAL & CONFERENCE PUBLICATIONS

---

- **ANALYSIS AND SYNTHESIS OF POINT DISTRIBUTIONS BASED ON PAIR CORRELATION**  
A. C. Öztireli, M. Gross.  
Proceedings of *ACM SIGGRAPH Asia* (Singapore, November 28-December 1, 2012),  
*ACM Transactions on Graphics*, vol. 31, no. 6, pp. 170:1-170:10, 2012.
- **ADAPTIVE SURFACE SPLATTING FOR FACIAL RENDERING**  
H. Kim, A. C. Öztireli, M. Gross, S. Choi  
*Computer Animation and Virtual Worlds*, vol. 23, no. 3-4, pp. 363-373, 2012.
- **3D MODELING WITH A SYMMETRIC SKETCH**

A. Cengiz Öztireli | Probsteistr. 117a 8051 Zürich | +41 76 204 27 58 | [a.cengiz.oztireli@gmail.com](mailto:a.cengiz.oztireli@gmail.com) |

A. C. Öztireli, U. Uyumaz, T. Popa, A. Sheffer, M. Gross  
Proceedings of *CAE/SBIM/NPAR 2011* (Vancouver, Canada, 5-7 August, 2011)

- **SPECTRAL SAMPLING OF MANIFOLDS**  
A. C. Öztireli, M. Alexa, M. Gross.  
Proceedings of *ACM SIGGRAPH Asia* (Seoul, Korea, December 15-18 December, 2010),  
*ACM Transactions on Graphics*, vol. 29, no. 5, pp. 168:1-168:8, 2010.
- **FEATURE PRESERVING POINT SET SURFACES BASED ON NON-LINEAR KERNEL REGRESSION**  
A. C. Öztireli, C. Guennebaud, M. Gross  
Proceedings of *Eurographics 2009* (Munich, Germany, March 30 - April 3, 2009),  
Computer Graphics Forum, vol. 28, no. 2, pp. 493-501, 2009.  
(Best Student Paper Award)
- **A NEW FEATURE-BASED METHOD FOR ROBUST AND EFFICIENT RIGID-BODY REGISTRATION OF OVERLAPPING POINT CLOUDS**  
A. C. Öztireli, C. Basdogan.  
*The Visual Computer: International Journal of Computer Graphics*, Springer-Verlag New York, Inc., vol. 24, issue 7 (2008), pp. 679-688, 2008.

### PROFESSIONAL ACTIVITIES

---

**Reviewer** ACM SIGGRAPH, ACM SIGGRAPH Asia, ACM Transactions on Graphics,  
Computer Graphics Forum, Elsevier Computers & Graphics

### POSITIONS OF RESPONSIBILITY

---

2010 **Founding member** – ETH Turkish Society

### LANGUAGE SKILLS

---

Turkish	Mother tongue
English	Fluent
German	Intermediate

### INTERESTS

---

Music	Guitar, piano, accordion, harmonica, saz
Modelling	Plastic models of planes, cars, and buildings
Sport	Football, tennis, skiing, bodybuilding

### REFERENCES

---

*Available upon request.*

## *Appendix*

## Bibliography

- [Achlioptas et al., 2001] Dimitris Achlioptas, Frank McSherry, and Bernhard Schölkopf. Sampling techniques for kernel methods. In *NIPS*, pages 335–342, 2001.
- [Adamson and Alexa, 2003] Anders Adamson and Marc Alexa. Approximating and intersecting surfaces from points. In *Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 230–239. Eurographics Association, 2003.
- [Adamson and Alexa, 2006a] Anders Adamson and Marc Alexa. Anisotropic point set surfaces. In *Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 7–13. ACM Press, 2006.
- [Adamson and Alexa, 2006b] Anders Adamson and Marc Alexa. Point-sampled cell complexes. *ACM Trans. Graph. (SIGGRAPH 2006)*, 25(3):671–680, 2006.
- [Alexa and Adamson, 2004] Marc Alexa and Anders Adamson. On normals and projection operators for surfaces defined by point sets. In *Eurographics Symposium on Point-Based Graphics*, pages 149–156, 2004.
- [Alexa and Adamson, 2007] Marc Alexa and Anders Adamson. Interpolatory point set surfaces - convexity and hermite data. *ACM Trans. Graph., to appear*, 2007.

## Bibliography

- [Alexa et al., 2001] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Point set surfaces. In *VIS '01*, pages 21–28, Washington, DC, USA, 2001. IEEE Computer Society.
- [Alexa et al., 2003] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Cláudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Computer Graphics and Visualization*, 9(1):3–15, 2003.
- [Alliez et al., 2002] Pierre Alliez, Mark Meyer, and Mathieu Desbrun. Interactive geometry remeshing. *ACM Trans. Graph.*, 21(3):347–354, 2002.
- [Alliez et al., 2003] Pierre Alliez, Éric Colin de Verdière, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In *SMI '03*, page 49, Washington, DC, USA, 2003. IEEE Computer Society.
- [Alliez et al., 2007] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Proceedings of the fifth Eurographics symposium on Geometry processing, SGP '07*, pages 39–48, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [Amenta and Bern, 1998] Nina Amenta and Marshall Bern. Surface reconstruction by voronoi filtering. In *Proceedings of the fourteenth annual symposium on Computational geometry, SCG '98*, pages 39–48, New York, NY, USA, 1998. ACM.
- [Amenta and Kil, 2004a] Nina Amenta and Yong Kil. Defining point-set surfaces. *ACM Trans. Graph. (SIGGRAPH 2004)*, 23(3):264–270, 2004.
- [Amenta and Kil, 2004b] Nina Amenta and Yong Kil. The domain of a point set surface. In *Eurographics Symposium on Point-Based Graphics 2004*, pages 139–147, 2004.
- [Amenta et al., 2000] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proceedings of the sixteenth annual symposium on Computational geometry, SCG '00*, pages 213–222, New York, NY, USA, 2000. ACM.
- [Amenta et al., 2001] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(23):127 – 153, 2001. ;ce:title;Combinatorial Curves and Surfaces;/ce:title;.
- [Avron et al., 2010] Haim Avron, Andrei Sharf, Chen Greif, and Daniel Cohen-Or. 11-sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph.*, 29(5):135:1–135:12, November 2010.

- [Balzer et al., 2009] Michael Balzer, Thomas Schlömer, and Oliver Deussen. Capacity-constrained point distributions: A variant of Lloyd’s method. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2009)*, 28(3):86:1–8, 2009.
- [Bauer et al., 2009] Ulrich Bauer, Konrad Polthier, and Max Wardetzky. Uniform convergence of discrete curvatures from nets of curvature lines, December 14 2009.
- [Belkin and Niyogi, 2006] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In *NIPS*, pages 129–136, 2006.
- [Belkin et al., 2009] Mikhail Belkin, Jian Sun, and Yusu Wang. Constructing laplace operator from point clouds in rd. In *SODA ’09*, pages 1031–1040, Philadelphia, PA, USA, 2009.
- [Bezrukov et al., 2002] Alexander Bezrukov, Monika Bargie, and Dietrich Stoyan. Statistical analysis of simulated random packings of spheres. *Particle and Particle Systems Characterization*, 19(2):111–118, 2002.
- [Bobenko and Springborn, 2007] A. Bobenko and B. Springborn. A discrete laplacebeltrami operator for simplicial surfaces. *Discr. Comp. Geom*, 38(4):740–756, 2007.
- [Boissonnat and Oudot, 2003] J. D. Boissonnat and S. Oudot. Provably good surface sampling and approximation. In *SGP ’03*, pages 9–18, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [Boissonnat and Oudot, 2005] Jean-Daniel Boissonnat and Steve Oudot. Provably good sampling and meshing of surfaces. *Graph. Models*, 67(5):405–451, September 2005.
- [Boris Mederos and de Figueiredo, 2003] Luiz Velho Boris Mederos and Luiz Henrique de Figueiredo. Robust smoothing of noisy point clouds. In *SIAM Geometric Design and Computing*, page 13. Nashboro Press, 2003.
- [Boulch and Marlet, 2012] Alexandre Boulch and Renaud Marlet. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum*, 31(5):1765–1774, 2012.
- [Boutin and Kemper, 2007] Mireille Boutin and Gregor Kemper. Which point configurations are determined by the distribution of their pairwise distances? *International Journal of Computational Geometry & Applications*, 17:31–43, 2007.
- [Boutin et al., 2004] Mireille Boutin, Mireille Boutin, Mireille Boutin, Gregor Kemper, and Gregor Kemper. On reconstructing n-point configurations from the distribution of distances or areas. *Adv. Appl. Math.*, 32:709–735, 2004.

## Bibliography

- [Braun, 2006] Mikio L. Braun. Accurate error bounds for the eigenvalues of the kernel matrix. *J. Mach. Learn. Res.*, 7:2303–2328, 2006.
- [Calakli and Taubin, 2011] F. Calakli and Gabriel Taubin. Ssd: Smooth signed distance surface reconstruction. *Comput. Graph. Forum*, pages 1993–2002, 2011.
- [Carmo, 1976] Manfredo Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.
- [Carr et al., 2001] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH 2001: Proc. of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, New York, NY, USA, 2001. ACM.
- [Celniker and Gossard, 1991] George Celniker and Dave Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques, SIGGRAPH '91*, pages 257–266, New York, NY, USA, 1991. ACM.
- [Cheng, 1976] Shiu-Yuen Cheng. Eigenfunctions and nodal sets. *Commentarii Mathematici Helvetici*, 51(1):43–55, 1976.
- [Choudhury and Tumblin, 2003] Prasun Choudhury and Jack Tumblin. The trilateral filter for high contrast images and meshes. In *The 14th Eurographics workshop on Rendering*, pages 186–196, 2003.
- [Cignoni et al., 2001] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 2001.
- [Cleveland, 1979] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:859–836, 1979.
- [Cohen-Steiner and Morvan, 2006] David Cohen-Steiner and Jean-Marie Morvan. Second fundamental measure of geometric sets and local approximation of curvatures. *J. Differential Geom.*, 73(3):363–394, 2006.
- [Coifman and Lafon, 2006] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.
- [Coifman et al., 2005] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, F. Warner, and S. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *Proceedings of the National Academy of Sciences*, pages 7426–7431, 2005.
- [Cook, 1986] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72, 1986.

- [Daniels et al., 2007] Joel II Daniels, Linh K. Ha, Tilo Ochotta, and Claudio T. Silva. Robust smooth feature extraction from point clouds. In *SMI '07*, pages 123–136, Washington, DC, USA, 2007. IEEE Computer Society.
- [Deussen et al., 1998] Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomir Mech, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques, SIGGRAPH '98*, pages 275–286, New York, NY, USA, 1998. ACM.
- [Dey and Goswami, 2003] Tamal K. Dey and Samrat Goswami. Tight cocone: a water-tight surface reconstructor. In *SM '03*, pages 127–134, New York, NY, USA, 2003. ACM.
- [Dey and Goswami, 2004] Tamal K. Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. In *Proceedings of the twentieth annual symposium on Computational geometry, SCG '04*, pages 330–339, New York, NY, USA, 2004. ACM.
- [Dey and Sun., 2005] T. K. Dey and J. Sun. Normal and feature estimations from noisy point clouds. Technical Report OSU-CISRC-7/50-TR50, Ohio State University, 2005.
- [Dey et al., 2010] T. K. Dey, P. Rajan, and Y. Wang. Convergence, stability, and discrete approximation of laplace spectra. In *SODA '10*. ACM, 2010. to appear.
- [Dey et al., 2012] T. K. Dey, X. Ge, Q. Que, I. Safa, L. Wang, and Y. Wang. Feature-preserving reconstruction of singular surfaces. *Computer Graphics Forum*, 31(5):1787–1796, 2012.
- [Dey, 2006] Tamal K. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis (Cambridge Monographs on Applied and Computational Mathematics)*. Cambridge University Press, New York, NY, USA, 2006.
- [Digne et al., 2012] Julie Digne, David Cohen-Steiner, Pierre Alliez, Mathieu Desbrun, and Fernando De Goes. Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets. Rapport de recherche RR-7991, INRIA, June 2012.
- [Digne, 2012] Julie Digne. Similarity based filtering of point clouds. In *CVPR Workshops*, pages 73–79, 2012.
- [Dippé and Wold, 1985] Mark A. Z. Dippé and Erling Henry Wold. Antialiasing through stochastic sampling. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques, SIGGRAPH '85*, pages 69–78, New York, NY, USA, 1985. ACM.

## Bibliography

- [Drineas and Mahoney, 2005] Petros Drineas and Michael W. Mahoney. Approximating a gram matrix for improved kernel-based learning. In *Learning Theory*, volume 3559, pages 323–337. Springer Berlin / Heidelberg, 2005.
- [Du et al., 1999] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [Dunbar and Humphreys, 2006] Daniel Dunbar and Greg Humphreys. A spatial data structure for fast poisson-disk sample generation. *ACM Trans. Graph.*, 25:503–508, July 2006.
- [Ebeida et al., 2011] Mohamed S. Ebeida, Anjul Patney, Scott A. Mitchell, Andrew Davidson, Patrick M. Knupp, and John D. Owens. Efficient maximal poisson-disk sampling. *ACM Transactions on Graphics*, 30(4), 2011.
- [Edelsbrunner and Mücke, 1994] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13(1):43–72, January 1994.
- [Fattal, 2011] Raanan Fattal. Blue-noise point sampling using kernel density model. *ACM SIGGRAPH 2011 papers*, 28(3):1–10, 2011.
- [Fleishman et al., 2005] Shachar Fleishman, Daniel Cohen-Or, and Cláudio T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, 2005.
- [Friedman and Stamos, 2012] S. Friedman and I. Stamos. Online facade reconstruction from dominant frequencies in structured point clouds. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8, june 2012.
- [Fu and Zhou, 2008] Yan Fu and Bingfeng Zhou. Direct sampling on surfaces for high quality remeshing. In *SPM '08*, pages 115–124, New York, NY, USA, 2008. ACM.
- [Gal et al., 2007] Ran Gal, Ariel Shamir, Tal Hassner, Mark Pauly, and Daniel Cohen-Or. Surface reconstruction using local shape priors. In *Proceedings of the fifth Eurographics symposium on Geometry processing, SGP '07*, pages 253–262, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [Gamito and Maddock, 2009] Manuel N. Gamito and Steve C. Maddock. Accurate multidimensional poisson-disk sampling. *ACM Trans. Graph.*, 29:8:1–8:19, December 2009.
- [Garland and Heckbert, 1997] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97: Proc. of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

- [Grigor'yan, 1998] A. Grigor'yan. Estimates of heat kernels on riemannian manifolds. In *Spectral Theory and Geometry. ICMS Instructional Conference*, pages 140–225. Cambridge Univ. Press, 1998.
- [Guennebaud and Gross, 2007] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. *ACM Trans. Graph. (SIGGRAPH 2007)*, 26(3):23.1–23.9, 2007.
- [Guennebaud et al., 2008] Gael Guennebaud, Marcel Germann, , and Markus Gross. Dynamic sampling and rendering of algebraic point set surfaces. *Computer Graphics Forum (Eurographics 2008)*, 27(3), 2008.
- [Guillemot et al., 2012] Thierry Guillemot, Andrs Almansa, and Tamy Boubekeur. Non local point set surfaces. In *Proceedings of the International Conference on 3D Imaging, Moldeing, Processing, Visualization and Transmission (3DIMPVT)*, 2012.
- [Ham et al., 2004] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *ICML '04*, page 47, New York, NY, USA, 2004. ACM.
- [Hofmann et al., 2008] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):pp. 1171–1220, 2008.
- [Hoppe et al., 1992] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH 92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, New York, NY, USA, 1992. ACM.
- [Hoseinnezhad and Bab-Hadiashar, 2007] R. Hoseinnezhad and A. Bab-Hadiashar. A novel high breakdown m-estimator for visual data segmentation. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–6, Oct. 2007.
- [Huang et al., 2009] Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph.*, 28(5):176:1–176:7, December 2009.
- [Huang et al., 2012] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang. Edge-aware point set resampling. *ACM Transactions on Graphics, to appear*, pages 1–12, 2012.
- [Huber, 2004] Peter J. Huber. *Robust Statistics*. Wiley, 2004.
- [Illian et al., 2008] Janine Illian, Antti Penttinen, Helga Stoyan, and Dietrich Stoyan, editors. *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley and Sons, Ltd., 2008.

## Bibliography

- [Ipsen and Nadler, 2009] I. Ipsen and B. Nadler. Refined perturbation bounds for eigenvalues of hermitian and non-hermitian matrices. *SIAM Journal of Matrix Analysis*, 31(1):40–53, 2009.
- [Jodrey and Tory, 1985] W. S. Jodrey and E. M. Tory. Computer simulation of close random packing of equal spheres. *Phys. Rev. A*, 32:2347–2351, Oct 1985.
- [Jones et al., 2003] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. In *SIGGRAPH 2003: ACM SIGGRAPH 2003 Papers*, pages 943–949, New York, NY, USA, 2003. ACM.
- [Jones, 2006] Thouis R. Jones. Efficient generation of poisson-disk sampling patterns. *journal of graphics, gpu, and game tools*, 11(2):27–36, 2006.
- [Karoui and d’Aspremont, 2009] Noureddine El Karoui and Alexandre d’Aspremont. Approximating eigenvectors by subsampling. 2009.
- [Kazhdan et al., 2006] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing 2006*, pages 43–52, 2006.
- [Kazhdan, 2005] Michael Kazhdan. Reconstruction of solid models from oriented point sets. In *Proceedings of the third Eurographics symposium on Geometry processing, SGP ’05, Aire-la-Ville, Switzerland, Switzerland, 2005*. Eurographics Association.
- [Kesavan, 1998] S. Kesavan. Listening to the shape of a drum. *Resonance*, 3:49–58, 1998. 10.1007/BF02841422.
- [Kim et al., 2008] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. *ACM Trans. Graph.*, 27(3):50:1–50:6, August 2008.
- [Kitago and Gopi, 2006] Masaki Kitago and Meenakshisundaram Gopi. Efficient and prioritized point subsampling for csrbf compression. In *Symp. on Point-based Graphics, Eurographics*, 2006.
- [Kolluri et al., 2004] Ravikrishna Kolluri, Jonathan Richard Shewchuk, and James F. O’Brien. Spectral surface reconstruction from noisy point clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, SGP ’04*, pages 11–21, New York, NY, USA, 2004. ACM.
- [Kolluri, 2005] Ravikrishna Kolluri. Provably good moving least squares. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1008–1018, January 2005.
- [Kondor and Lafferty, 2002] Risi I. Kondor and John Lafferty. Diffusion Kernels on Graphs and Other Discrete Structures. In *In Proceedings of the ICML*, pages 315–322, 2002.

- [Kopf et al., 2006] Johannes Kopf, Daniel Cohen-Or, Oliver Deussen, and Dani Lischinski. Recursive wang tiles for real-time blue noise. *ACM Trans. Graph.*, 25:509–518, July 2006.
- [Kuo and Yau, 2006] Chuan-Chu Kuo and Hong-Tzong Yau. A new combinatorial approach to surface reconstruction with sharp features. *Visualization and Computer Graphics, IEEE Transactions on*, 12(1):73–82, jan.-feb. 2006.
- [Lafon and Lee, 2006] S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1393–1403, Sept. 2006.
- [Lagae and Dutré, 2006] Ares Lagae and Philip Dutré. Poisson sphere distributions. In L. Kobbelt, T. Kuhlen, T. Aach, and R. Westermann, editors, *Vision, Modeling, and Visualization 2006*, pages 373–379, Berlin, November 2006. Akademische Verlagsgesellschaft Aka GmbH.
- [Lagae and Dutré, 2008] Ares Lagae and Philip Dutré. A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum*, 27(1):114–129, March 2008.
- [Lai et al., 2007] Yu-Kun Lai, Qian-Yi Zhou, Shi-Min Hu, Johannes Wallner, and Helmut Pottmann. Robust feature classification and editing. *IEEE Trans. Vis. Comp. Graphics*, 13(1):34–45, 2007.
- [Levin, 2003] David Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization*, pages 37–49, 2003.
- [Lévy, 2006] Bruno Lévy. Laplace-beltrami eigenfunctions towards an algorithm that “understands” geometry. In *Shape Modeling International*, page 13. IEEE Computer Society, 2006.
- [Lewis, 1989] J. P. Lewis. Algorithms for solid noise synthesis. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques, SIGGRAPH '89*, pages 263–270, New York, NY, USA, 1989. ACM.
- [Li et al., 2010a] Bao Li, Ruwen Schnabel, Reinhard Klein, Zhiquan Cheng, Gang Dang, and Shiyao Jin. Robust normal estimation for point clouds with sharp features. *Computers & Graphics*, 34(2):94–106, 2010.
- [Li et al., 2010b] Hongwei Li, Li-Yi Wei, Pedro V. Sander, and Chi-Wing Fu. Anisotropic blue noise sampling. In *ACM SIGGRAPH Asia 2010 papers, SIGGRAPH ASIA '10*, pages 167:1–167:12, New York, NY, USA, 2010. ACM.
- [Li et al., 2011] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf,

## Bibliography

- Daniel Cohen-Or, and Niloy J. Mitra. Globfit: consistently fitting primitives by discovering global relations. *ACM Trans. Graph.*, 30(4):52:1–52:12, July 2011.
- [Lipman et al., 2007] Yaron Lipman, Daniel Cohen-Or, and David Levin. Data-dependent MLS for faithful surface approximation. In *Symposium on Geometry Processing*, pages 59–67, 2007.
- [Liu et al., 2006] Rong Liu, Varun Jain, and Hao Zhang. Subsampling for efficient spectral mesh processing. In *Computer Graphics International*, pages 172–184, 2006.
- [Lloyd, 1982] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129 – 137, mar 1982.
- [Ma et al., 2011] Chongyang Ma, Li-Yi Wei, and Xin Tong. Discrete element textures. *ACM Trans. Graph.*, 30(4):62:1–62:10, August 2011.
- [Manson et al., 2008] J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. In *Proceedings of the Symposium on Geometry Processing, SGP '08*, pages 1411–1420, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.
- [Matern, 1960] B. Matern. Spatial variation. stochastic models and their application to some problems in forest surveys and other sampling investigations. *Meddelanden fran Statens Skogsforskningsinstitut*, 49(5), 1960.
- [Matheron, 1968] G. Matheron. Schema booleen sequentiel de partition aleatoire. *N-83 CMM, Paris School of Mines Publication*, 1968.
- [McCool and Eugene, 1992] Michael McCool and Fiu Eugene. Hierarchical poisson disk sampling distributions. In *Proceedings of the conference on Graphics interface '92*, pages 94–105, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
- [Medvedev and Naberukhin, 1987] N.N. Medvedev and Yu.I. Naberukhin. Shape of the delaunay simplices in dense random packings of hard and soft spheres. *Journal of Non-Crystalline Solids*, 94(3):402 – 406, 1987.
- [Minh et al., 2006] HaQuang Minh, Partha Niyogi, and Yuan Yao. Mercers theorem, feature maps, and smoothing. In Gbor Lugosi and HansUlrich Simon, editors, *Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 154–168. Springer Berlin Heidelberg, 2006.
- [Mitchell, 1987] Don P. Mitchell. Generating antialiased images at low sampling densities. In *SIGGRAPH 87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 65–72, New York, NY, USA, 1987. ACM.

- [Mitra and Nguyen, 2003] Niloy J. Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the nineteenth annual symposium on Computational geometry, SCG '03*, pages 322–328, New York, NY, USA, 2003. ACM.
- [Moghaddam et al., 2008] Baback Moghaddam, Amit Gruber, Yair Weiss, and Shai Avidan. Sparse regression as a sparse eigenvalue problem. In *Information Theory and Applications Workshop, 2008*, pages 121–127, 27 2008-Feb. 1 2008.
- [Morgenthaler, 2007] Stephan Morgenthaler. A survey of robust statistics. *Statistical Methods and Applications*, 15(3):271–293, February 2007.
- [Moscinski et al., 1989] J. Moscinski, M. Bargie, Z. A. Rycerz, and P. W. M. Jacobs. The force-biased algorithm for the irregular close packing of equal hard spheres. *Molecular Simulation*, 3(4):201–212, 1989.
- [Mullen et al., 2010] Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Computer Graphics Forum*, 29(5):1733–1741, 2010.
- [Naberukhin et al., 1991] Yu. I. Naberukhin, V.P. Voloshin, and N.N. Medvedev. Geometrical analysis of the structure of simple liquids: percolation approach. *Molecular Physics*, 73(4):917–936, 1991.
- [Nagai et al., 2009] Yukie Nagai, Yutaka Ohtake, and Hiromasa Suzuki. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. In *Proceedings of the Symposium on Geometry Processing, SGP '09*, pages 1339–1348, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.
- [Neyman and Scott, 1952] J. Neyman and E.L. Scott. A theory for the spatial distribution of galaxis. *Astrophys. J.*, 116:144–163, 1952.
- [Ng et al., 2010] Heung-Sun Ng, Tai-Pang Wu, and Chi-Keung Tang. Surface-from-gradients without discrete integrability enforcement: A gaussian kernel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:2085–2099, 2010.
- [Ohser and Mcklich, 2000] Joachim Ohser and Frank Mcklich, editors. *Statistical Analysis of Microstructures in Materials Science*. John Wiley and Sons, Ltd., 2000.
- [Ohtake et al., 2003] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, July 2003.
- [Ohtake et al., 2004] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 3d scattered data

## Bibliography

- approximation with adaptive compactly supported radial basis functions. In *Shape Modeling Applications*, pages 31–39, June 2004.
- [Ohtake et al., 2005] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. 3d scattered data interpolation and approximation with multilevel compactly supported rbfs. *Graphical Models*, 67(3):150 – 165, 2005. ;ce:title;Special Issue on SMI 2003;/ce:title;.
- [Ostromoukhov et al., 2004] Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23:488–495, August 2004.
- [Öztireli and Gross, 2012] A. Cengiz Öztireli and Markus Gross. Analysis and synthesis of point distributions based on pair correlation. *ACM Trans. Graph.*, 31(6):170:1–170:10, November 2012.
- [Öztireli et al., 2009] Cengiz Öztireli, Gaël Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Eurographics 2009*, pages 493–501, 2009.
- [Öztireli et al., 2010] A. Cengiz Öztireli, Marc Alexa, and Markus Gross. Spectral sampling of manifolds. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA '10, pages 168:1–168:8, New York, NY, USA, 2010. ACM.
- [Öztireli et al., 2011] A. Cengiz Öztireli, Umut Uyumaz, Tiberiu Popa, Alla Sheffer, and Markus Gross. 3d modeling with a symmetric sketch. *EG*, August 2011.
- [Pauly and Gross, 2001] Mark Pauly and Markus Gross. Spectral processing of point-sampled geometry. In *SIGGRAPH 2001: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 379–386, New York, NY, USA, 2001. ACM.
- [Pauly et al., 2002] Mark Pauly, Markus Gross, and Leif P. Kobbelt. Efficient simplification of point-sampled surfaces. In *VIS '02*, pages 163–170, Washington, DC, USA, 2002. IEEE Computer Society.
- [Pauly et al., 2003] Mark Pauly, Richard Keiser, Leif P. Kobbelt, and Markus Gross. Shape modeling with point-sampled geometry. *ACM Trans. Graph. (SIGGRAPH 2003)*, 22(3):641–650, 2003.
- [Peyré and Cohen, 2006] Gabriel Peyré and Laurent D. Cohen. Geodesic remeshing using front propagation. *Int. J. Comput. Vision*, 69(1):145–156, 2006.
- [Pinkall and Polthier, 1993] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

- [Pommerening, 2002] A. Pommerening. Approaches to quantifying forest structures. *Forestry*, 75:305–324, 2002.
- [Pottmann et al., 2004] H. Pottmann, T. Steiner, M. Hofer, C. Haider, and A. Hanbury. The isophotic metric and its application to feature sensitive morphology on surfaces. In T. Pajdla and J. Matas, editors, *Computer Vision — ECCV 2004, Part IV*, volume 3024 of *Lecture Notes in Computer Science*, pages 560–572. Springer, 2004.
- [Reuter et al., 2005] Patrick Reuter, Pierre Joyot, Jean Trunzler, Tamy Boubekour, and Christophe Schlick. Point set surfaces with sharp features. Technical report, LaBRI, 2005.
- [Reuter et al., 2006] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace-beltrami spectra as “shape-dna” of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.
- [Rustamov, 2007] Raif M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In Alexander Belyaev and Michael Garland, editors, *SGP07*, pages 225–233, Barcelona, Spain, 2007. Eurographics Association.
- [Salman et al., 2010] Nader Salman, Mariette Yvinec, and Quentin Merigot. Feature preserving mesh generation from 3d point clouds. *Computer Graphics Forum*, 29(5):1623–1632, 2010.
- [Saucan et al., 2008] Emil Saucan, Eli Appleboim, and Yehoshua Y. Zeevi. Sampling and reconstruction of surfaces and higher dimensional manifolds. *J. Math. Imaging Vis.*, 30(1):105–123, 2008.
- [Schaback and Wendland, 2006] Robert Schaback and Holger Wendland. Kernel techniques: From machine learning to meshless methods. *Acta Numerica*, 15(1):543–639, 2006.
- [Schlömer et al., 2011] Thomas Schlömer, Daniel Heck, and Oliver Deussen. Farthest-point optimized point sets with maximized minimum distance. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics, HPG ’11*, pages 135–142, New York, NY, USA, 2011. ACM.
- [Schmaltz et al., 2010] Christian Schmaltz, Pascal Gwosdek, Andrs Bruhn, and Joachim Weickert. Electrostatic halftoning. *Computer Graphics Forum*, 29(8):2313–2327, 2010.
- [Schnabel et al., 2007] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

## Bibliography

- [Scholkopf and Smola, 2001] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [Schölkopf et al., 1998] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.
- [Schreiner et al., 2006] J. Schreiner, C.E. Scheidegger, S. Fleishman, and C.T. Silva. Direct (re)meshing for efficient surface processing. *Computer Graphics Forum (Eurographics 2006)*, 25(3):527–536, 2006.
- [Selle et al., 2005] Andrew Selle, Nick Rasmussen, and Ronald Fedkiw. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.*, 24(3):910–914, July 2005.
- [Seversky et al., 2011] Lee M. Seversky, Matt S. Berger, and Lijun Yin. Harmonic point cloud orientation. *Computers and Graphics*, 35(3):492 – 499, 2011. [;ce:title;Shape Modeling International \(SMI\) Conference 2011;/ce:title;.](#)
- [Shapiro, 1993] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, 41(12):3445–3462, 1993.
- [Shen et al., 2004] Chen Shen, James F. O’Brien, and Jonathan R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph. (SIGGRAPH 2004)*, pages 896–904, 2004.
- [Shepard, 1968] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *ACM national conference*, pages 517–524, 1968.
- [Sun et al., 2009] Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Comput. Graph. Forum*, 28(5):1383–1392, 2009.
- [Taubin, 1995] Gabriel Taubin. A signal processing approach to fair surface design. In *SIGGRAPH 95: Proc. of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, New York, NY, USA, 1995. ACM.
- [Terzopoulos et al., 1987] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21:205–214, August 1987.
- [Thomas, 1949] M. Thomas. A generalization of Poisson’s binomial limit for use in ecology. *Biometrika*, 36:18–25, 1949.
- [Torquato, 2002] S. Torquato, editor. *Random Heterogeneous Materials. Microstructure and Macroscopic Properties*. Springer-Verlag, New York, 2002.

- [Turk and O'brien, 2002] Greg Turk and James F. O'brien. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.*, 21(4):855–873, October 2002.
- [Turk, 1992] Greg Turk. Re-tiling polygonal surfaces. *SIGGRAPH Comput. Graph.*, 26(2):55–64, 1992.
- [Ulichney, 1987] Robert Ulichney. *Digital halftoning*. MIT Press, Cambridge, MA, USA, 1987.
- [Valette et al., 2008] Sebastien Valette, Jean Marc Chassery, and Remy Prost. Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 14:369–381, 2008.
- [Vallet and Lévy, 2008] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Eurographics)*, 2008.
- [von Luxburg, 2004] Ulrike von Luxburg. *Statistical Learning with Similarity and Dissimilarity Functions*. PhD thesis, Technischen Universität Berlin, 2004.
- [Wardetzky et al., 2007] Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. Discrete laplace operators: no free lunch. In *SGP '07*, pages 33–37, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [Wei and Wang, 2011] Li-Yi Wei and Rui Wang. Differential domain analysis for non-uniform sampling. In *ACM SIGGRAPH 2011 papers, SIGGRAPH '11*, pages 50:1–50:10, New York, NY, USA, 2011. ACM.
- [Wei, 2008] Li-Yi Wei. Parallel poisson disk sampling. *ACM Trans. Graph.*, 27:20:1–20:9, August 2008.
- [Wei, 2010] Li-Yi Wei. Multi-class blue noise sampling. *ACM Trans. Graph.*, 29:79:1–79:8, July 2010.
- [White et al., 2007] K.B. White, D. Cline, and P.K. Egbert. Poisson disk point sets by hierarchical dart throwing. In *Interactive Ray Tracing, 2007. RT '07. IEEE Symposium on*, pages 129–132, sept. 2007.
- [Witkin and Heckbert, 1994] Andrew Witkin and Paul S. Heckbert. Using particles to sample and control implicit surfaces. In *21st annual conference on Computer graphics and interactive techniques*, pages 269–278. ACM Press, 1994.
- [Yan et al., 2009] D.-M. Yan, B. Lévy, Y. Liu, F. Sun, and W. Wang. Isotropic remeshing with fast and exact computation of restricted voronoi diagram. *Comput. Graph. Forum (Symp. on Geometry Processing 2009)*, 28(5):1445–1455, 2009.
- [Yellott, 1983] J I Yellott. Spectral consequences of photoreceptor sampling in the rhesus retina. *Science*, 221(4608):382–385, 1983.

## *Bibliography*

- [Zhang et al., 2007] Hao Zhang, Oliver van Kaick, and Ramsay Dyer. Spectral methods for mesh processing and analysis. In *Eurographics State-of-the-art Report*, pages 1–22, 2007.
- [Zheng et al., 2010] Qian Zheng, Andrei Sharf, Guowei Wan, Yangyan Li, Niloy J. Mitra, Daniel Cohen-Or, and Baoquan Chen. Non-local scan consolidation for 3d urban scenes. *ACM Trans. Graph.*, 29(4):94:1–94:9, July 2010.