Novel Spectral Representations and Sparsity-Driven Algorithms for Shape Modeling and Analysis

A Dissertation presented

by

Ming Zhong

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

Spring 2016

Stony Brook University

The Graduate School

Ming Zhong

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation

Hong Qin - Dissertation Advisor
Professor, Department of Computer Science

Joseph Mitchell - Chairperson of Defense
Professor, Department of Applied Mathematics and Statistics

Xianfeng David Gu
Associate Professor, Department of Computer Science

Xin Li
Associate Professor, School of Electrical Engineering & Computer Science
Louisiana State University

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Novel Spectral Representations and Sparsity-Driven Algorithms for Shape Modeling and Analysis

by

Ming Zhong

Doctor of Philosophy

in

Computer Science

Stony Brook University

2016

In this dissertation, we focus on extending classical spectral shape analysis by incorporating spectral graph wavelets and sparsity-seeking algorithms. Defined with the graph Laplacian eigenbasis, the spectral graph wavelets are localized both in the vertex domain and graph spectral domain, and thus are very effective in describing local geometry. With a rich dictionary of elementary vectors and forcing certain sparsity constraints, a real life signal can often be well approximated by a very sparse coefficient representation. The many successful applications of sparse signal representation in computer vision and image processing inspire us to explore the idea of employing sparse modeling techniques with dictionary of spectral basis to solve various shape modeling problems.

Conventional spectral mesh compression uses the eigenfunctions of mesh Laplacian as shape bases, which are highly inefficient in representing local geometry. To ameliorate, we advocate an innovative approach to 3D mesh compression using spectral graph wavelets as dictionary to encode mesh geometry. The spectral graph wavelets are locally defined at individual vertices and can better capture local shape information than Laplacian eigenbasis. The multi-scale SGWs form a redundant dictionary as shape basis, so we formulate the compression of 3D shape as a sparse approximation problem that can be readily handled by greedy pursuit algorithms.

Surface inpainting refers to the completion or recovery of missing shape geometry based on the shape information that is currently available. We devise a new surface inpainting algorithm founded upon the theory and techniques of sparse signal recovery. Instead of estimating the missing geometry directly, our novel method is to find this low-dimensional representation which describes the entire original shape. More specifically, we find that, for many shapes, the vertex coordinate function can be well approximated by a very sparse coefficient representation with respect to the dictionary comprising its Laplacian eigenbasis, and it is then possible to recover this sparse representation from partial measurements of the original shape. Taking advantage of the sparsity cue, we advocate a novel variational approach for surface inpainting, integrating data fidelity constraints on the shape domain with coefficient sparsity constraints on the transformed domain. Because of the powerful properties of Laplacian eigenbasis, the inpainting results of our method tend to be globally coherent with the remaining shape.

Informative and discriminative feature descriptors are vital in qualitative and quantitative shape analysis for a large variety of graphics applications. We advocate novel strategies to define generalized, user-specified features on shapes. Our new region descriptors are primarily built upon the coefficients of spectral graph wavelets that are both multi-scale and multi-level in nature, consisting of both local and global information. Based on our novel spectral feature descriptor, we developed a user-specified feature detection framework and a tensor-based shape matching algorithm.

Through various experiments, we demonstrate the competitive performance of our proposed methods and the great potential of spectral basis and sparsity-driven methods for shape modeling.

Contents

List of Figures **viii**

Acknowledgements **xi**

Publications **xii**

1 Introduction **1**

1.1 Problem Statement 1

1.2 Contributions 4

1.3 Dissertation Organization 8

2 Background Review **9**

2.1 Laplacian and Graph Fourier Transform 9

2.1.1 Manifold Fourier Transform 10

2.1.2 Mesh Laplacian and Spectral Mesh Processing 11

2.2 Kernels and Spectral Graph Wavelets 16

2.2.1 Kernel Functions 16

2.2.2 Wavelets on Graphs 18

2.2.3 Spectral Graph Wavelets 21

2.3 Sparse Representation Modeling 28

2.3.1 Sparse Modeling Problems 28

2.3.2 Computational Methods 31

2.3.3 Applications 33

3 Mesh Approximation and Compression 35

3.1 Introduction 35

3.1.1 Background 36

3.2 Spectral Mesh Compression 37

3.3 Sparse Shape Approximation via SGW 40

3.3.1 Sparse Approximation of Mesh Coordinates 41

3.3.2 SGW and Dictionary Design Strategies 42

3.3.3 Compression Ratio and Analysis 48

3.3.4 Mesh Partitioning 49

3.4 Experimental Results 51

3.4.1 Evaluation Method 51

3.4.2 Compression Performance 51

3.5 Chapter Summary 57

4 Surface Inpainting 58

4.1 Introduction 58

4.2 Related Work 60

4.2.1 Surface Inpainting 60

4.2.2 Sparse Signal Recovery and Inpainting 62

4.3 Variational Inpainting Model 63

4.4 Sparsity-Based Surface Inpainting 65

4.4.1 Laplacian Eigenbasis 65

4.4.2 Surface Inpainting 68

4.4.3 Filling Surface Holes 70

4.4.4 Remarks on Dictionary 71

4.5 Experiments 75

4.5.1 Geometry Recovery 75

4.5.2 Geometry Repair 78

4.5.3 Hole Filling 79

4.6 Chapter Summary 82

5 Generalized Feature Description and Detection 83

5.1 Introduction 83

5.2 Related Work 87

5.3 Local-to-global Shape Feature Definition 88

5.4 Multi-level And Multi-scale Shape Description 90

5.4.1 SGW-based Description 91

5.4.2 Contour-based Multi-scale Statistics 94

5.4.3 Informative Region-based Descriptor 96

5.5 Feature Detection Framework 96

5.5.1 Constructing Descriptors over Shapes 96

5.5.2 Feature Detection and Framework Properties 98

5.6 Experimental Results and Discussions 103

5.6.1 Parameter Evaluation 103

5.6.2 Repetitive Feature Detection within Certain Model 104

5.6.3 Feature Detection in Database 107

5.6.4 Comparisons and Discussion 107

5.6.5 Applications 113

5.7 Chapter Summary 117

6 Feature-Driven Shape Correspondence and Retrieval 118

6.1 High-Order Shape Matching 118

6.2 Hierarchical Shape Registration 123

6.3 Bag-of-Feature-Graphs Shape Retrieval 128

6.4 Chapter Summary 134

7 Conclusion and Future Work 135

7.1 Conclusion 135

7.2 Future Work 136

Bibliography 140

List of Figures

1.1 Hierarchy of this dissertation 6

2.1 Angles in cotangent weights 13
2.2 Spectral graph wavelets on the wolf model. 22
2.3 1D Mexican-hat Wavelet 24
2.4 Visualizations of the spectral Mexican-hat wavelets. 24
2.5 Transfer function of the Mexican-hat wavelet. 25

3.1 Visualization of MHBs. 39
3.2 Spectral approximation of the wolf model. 40
3.3 Spectral graph wavelets with different scales. 45
3.4 Comparison of three approximation methods. 47
3.5 Relationships between approximation performance and partition size. 50
3.6 Models used in the approximation experiments and their partitionings. 50
3.7 Mesh compression performance on the cow model. 52
3.8 Mesh compression performance on the fandisk model. 53
3.9 Mesh compression performance on the centaur model. 54
3.10 Mesh compression performance on the armadillo model. 55

4.1 Approximation of the double-torus model with Laplacian eigenbasis. 67
4.2 Estimating Laplacian eigenbasis coefficients with partial cube model. 72

4.3 Recovery of the bunny model with random missing vertices. . . . 73
4.4 Recovery of the horse model with random missing vertices. 73
4.5 Recovery of the fandisk model with random missing vertices. . . . 73
4.6 Recovery of the centaur model with random missing vertices. . . . 74
4.7 Geometry repair of the damaged cube model. 74
4.8 Geometry repair of the damaged epcot model. 75
4.9 Geometry repair of the damaged wolf model. 76
4.10 Inpainting errors given different ratios of randomly missing vertices. 77
4.11 Comparison of hole filling results using sparsity-regularized method and geometry-regularized method. 80
4.12 Hole inpainting on the bunny and hand models. 81

5.1 Pipeline of our generalized feature detection framework. 85
5.2 Illustration of the steps of our feature detection framework. 89
5.3 Statistics on wavelet decomposition coefficients. 93
5.4 Contour-based distance distribution. 95
5.5 Point sampling on the horse and Santa models. 97
5.6 Detection of repetitive features with different scales. 99
5.7 Illustration of the isometry-invariance property. 101
5.8 Coarse-to-fine recognition on elk and kettle models. 102
5.9 Retrieval results on the SHREC database. 105
5.10 Repetitive feature detection within one model. 106
5.11 Feature detection on the SHREC 2007 watertight retrieval database.108
5.12 Precision plots of different detection methods. 110
5.13 Comparisons of different detection methods for partial matching. . 111
5.14 Comparisons of detecting complex feature regions. 113
5.15 Partial shape matching and restoration. 115
5.16 Model recognition by identifying multiple components. 116

6.1 HKT for feature matching. 120
6.2 Matching hierarchy. 121

6.3 Matching deforming shapes with scale changes. 122
6.4 Matching with noise and topology changes. 123
6.5 Selected frames of matching objects with large deformations. . . . 123
6.6 Pipeline overview of our hierarchical registration framework. . . . 124
6.7 Steps of hierarchical registration algorithm. 125
6.8 Registration results using multi-resolution method. 127
6.9 Different representations of a given shape. 129
6.10 Nonrigid shapes and their BoFG descriptors. 132
6.11 Precision-recall curves of evaluated methods, with categories of (from *Left* to *Right*) null, scale change, and hole. 133

Acknowledgements

First of all, I would like to thank my advisor, Professor Hong Qin, for his persistent support and guidance throughout my PhD study. Without his inspirations and encouragement this dissertation would not be possible. I feel really fortunate to have the opportunity to work with him.

I am deeply grateful to all my committee members. Professor Joseph Mitchell and Professor Xianfeng Gu also served in my thesis proposal committee. And I would like to thank Professor Xiangming Jiao and Professor Alex Berg for serving in my research proficiency exam. Specially, I would thank Professor Xin Li for taking the time to serve as the external examiner of my defense committee.

I would like to thank my research collaborators including Dr. Tingbo Hou, Dr. Shengfa Wang, Xiaohua Hou and Nannan Li. I feel privileged that I could work with these wonderful colleagues and I am indebted to their help and insights.

Many of my other colleagues at Stony Brook and the two companies where I did internship, DreamWorks Animation and Google, also offered me lots of help and advise. In particular, I want to express my gratitude to Dr. Yanqing Chen, Dr. Rui Shi, Dr. Deepak Tolani, Dr. Jianping Hu, Dr. Xiaochao Wang, Ning Zhang and Dr. Vitalii Li.

Last but not least, I would like to give my special thanks to my parents and my wife. Without their support and encouragement, I would never have had this long but rewarding journey.

Ming Zhong

Publications

1. **Ming Zhong** and Hong Qin. "Surface Inpainting with Sparsity Constraints." *Computer Aided Geometric Design*, 41:23-35, 2016

2. Nannan Li, Shengfa Wang, **Ming Zhong**, Zhixun Su, and Hong Qin. "Generalized Local-to-global Shape Feature Detection based on Graph Wavelets." *IEEE Transaction on Visualization and Computer Graphics*, 2015

3. **Ming Zhong** and Hong Qin. "Sparse approximation of 3D shapes via spectral graph wavelets." *The Visual Computer*, 30(6-8): 751-761, 2014.

4. **Ming Zhong**. "Harmonic Shape Analysis: From Fourier to Wavelets." *Technical Report, Stony Brook University*, 2012.

5. Tingbo Hou, Xiaohua Hou, **Ming Zhong**, and Hong Qin. "Bag-of-feature-graphs: A new paradigm for non-rigid shape retrieval." In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1513-1516. IEEE, 2012.

6. **Ming Zhong**, Tingbo Hou, and Hong Qin. "A hierarchical approach to high-quality partial shape registration." In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 113-116. IEEE, 2012.

7. Tingbo Hou, **Ming Zhong**, and Hong Qin. "Diffusion-driven high-order matching of partial deformable shapes." In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 137-140. IEEE, 2012.

Chapter 1

Introduction

With the advent of ever more advanced and affordable 3D data acquisition technologies, digitalized 3D shapes have become ubiquitous in our daily life and played indispensable roles in numerous fields and applications, including computer-aided design, entertainment industry, medical research, etc. Although it is straightforward to represent and manipulate shape geometry directly in the spatial domain, in recent years there has been a growing trend towards spectral shape analysis, leveraging the eigen-structures of various mesh operators, especially the Laplacian operator. In this chapter, we give an overview of the main theme of this dissertation: spectral representations and sparsity-driven algorithms for shape modeling and analysis.

1.1 Problem Statement

In order to effectively encode 3D geometric data and facilitate different shape analysis and processing tasks, many different shape representations have been developed and adopted, including point clouds, polygon meshes, spline surfaces, voxels, etc. Choosing the most appropriate representation generally depends on the specific data source and task at handbut of all these representations, polygon meshes, especially triangle meshes, are perhaps the most widely used. Thanks to its conceptual simplicity, the discrete mesh representation is flexible, versatile, and highly efficient for batch processing and rendering.

From the point of view of signal processing, the vertex coordinate function of a mesh is essentially a vector-valued discrete signal defined on the domain of mesh vertices, just like an image is a color-valued signal defined on 2D grids. Naturally, the countless successful applications of digital signal processing with conventional types of signals, such as audio and images, have long motivated people to employ and adapt signal processing techniques to solve shape modeling problems in the mesh domain. For instance, in one of the pioneering works [147] on geometric signal processing, Taubin showed that the operation of surface smoothing can be carried out by applying the graph Laplacian operator to the mesh coordinates, which is equivalent to low-pass filtering of discrete signals defined on vertices.

One key aspect of signal processing is to represent signals in some transform domain by decomposing the signal as the linear combination of a suitable choice of basis vectors. Analyzing the coefficient representation with respect to the transform basis can often reveal important properties of the original signal, and many operations that are difficult to perform directly on the spatial domain can be easily accomplished in the transform domain by manipulating the coefficients. In Euclidean space, the most fundamental transform is the well known Fourier transform, which converts signals from time/space domain to frequency domain with multi-scale sinusoids as basis.

The classic Fourier basis functions are actually the eigenfunctions of the second-order differential operator in Euclidean space, corresponding to the spectra of 1D Laplacian. In general, spectral analysis refers to the analysis in terms of the eigenvalues and eigenfunctions of certain linear operator.

In the context of geometry processing and analysis, the natural equivalent to the Fourier basis is the set of eigenfunctions of the mesh's Laplacian operator. According to the spectral graph theory, the Laplacian eigenfunctions form an orthonormal and complete basis of functions defined on the mesh vertices, thus inducing the graph Fourier transform which affords space-frequency decomposition of functions defined on the surface, analogous to the Fourier transform in regular domains.

Unlike classic Fourier basis functions which are simply fixed sinusoids, the manifold harmonic basis differ with the connectivity, geometry, and the type of Laplacian operator that is adopted [163]. As a result, the mesh Laplacian eigenvectors and eigenvalues actually encode substantial topological and geometric information and can help characterize the global shape property and reveal intrinsic structure of the original mesh. This lends to the popularity of spectral methods for shape analysis and processing, including compression [67], segmentation [90], deformation [129], remeshing [35], paramterization [166], shape indexing [128], and retrieval [80], to name just a few.

Myriads of successful applications notwithstanding, there are certain limitations in employing the Laplacian eigenpairs directly for shape analysis. Primarily, the Laplacian eigenvalues and eigenvectors are determined by the global Laplacian matrix and thus encode information of the entire shape. Hence, they are more suitable for representing the overall shape but are not very effective in encoding local details. In addition, the Laplacian eigenpairs are not very stable across shapes and direct comparisons become unreliable after the first few eigenpairs. To more effectively describing the properties of local regions and the pair-wise relations between regions in a shape, people have developed a series of sophisticated spectral representations building upon the Laplacian eigenpairs, including various forms of kernels and distances.

In recent years, the focus of harmonic analysis has been moving from orthogonal basis with minimum size like the Fourier basis to richer, more expressive dictionaries with many redundant atoms such as wavelets, and from simple determinant time-frequency transform to coefficient decomposition based on sparsity-seeking optimization. Comparing with Fourier-based signal processing, richer dictionaries allows more flexibility in basis design and affords greater expressive power. With suitable dictionary and appropriate sparse optimization algorithm, the obtained representation can provide a very concise description or approximation of the original signal, help manifest a signal's essential components, and potentially facilitate a diversity of applications such as signal compression, pattern recognition, noise reduction, source separation and signal restoration.

The concepts of redundant dictionary and sparse representation have gained great momentum in the signal processing community [98], but most existing study focused on regular domains such as images and audio rather than the more challenging and universal problem of signal processing on graphs. In this dissertation, we will present our research work in shape analysis and processing by incorporating spectral representations and sparse modeling techniques.

1.2 Contributions

The basic idea of sparse modeling is to use as few as possible elementary functions chosen in a dictionary to decompose or approximate a signal, based on the intuition that a meaningful high-dimensional signal probably possesses a low-dimensional intrinsic structure, which can be captured by a sparse coefficient representation with respect to a suitable dictionary. Given a redundant dictionary, we can compute a sparse decomposition or approximation of the original signal by using various sparse optimization algorithms. The obtained sparse representation provides not only a more concise description of the original signal, which can be utilized for signal compression, but also, in many cases, a more precise and structure-revealing one.

The efficacy of sparsity-driven algorithms highly depends on the selection of dictionaries. Generally, the more "expressive" the dictionary is, the fewer elementary functions are needed to faithfully reconstruct the original signal, as there are more "words" available to express the information. Hence, instead of selecting a complete and orthogonal basis like the Fourier basis as the dictionary, it is often desirable to construct a redundant, overcomplete dictionary.

A natural choice for constructing redundant dictionary is to use wavelets. A family of wavelets can be generated by scaling and translating a single mother wavelet function, forming a rich dictionary of elementary signals of different frequencies and centered at different locations. However, defining wavelets and wavelet transform on mesh or manifold domain have always been a challenging problem, since there is no intuitive way to define scaling on irregular mesh grids. Recently, Hammond et al. proposed the spectral graph wavelets (SGWs) and spectral graph wavelet transform (SGWT) [59] , in which the wavelet functions are defined with the spectral graph basis and scaling is carried out in the Fourier domain. The SGWs constitute an overcomplete wavelet frame, whose properties such as multi-scale and spatial-locality of SGWs have proved to be valuable in a variety of data analysis applications.

In addition to acting as overcomplete basis for shape signals, the spectral graph wavelets also captures valuable intrinsic geometric information of the original shape in a multi-scale and spatially-localized way, and thus become a promising choice to serve as the building blocks for constructing local shape descriptors.

In this dissertation, we present solutions combining graph-based spectral representations, especially spectral graph wavelets, and sparse modeling methods to a series of fundamental problems in shape analysis and geometry processing, including mesh compression, surface inpainting, feature description, shape correspondence and shape retrieval. Through various experiments, we demonstrate the competitive performance of our proposed methods and the great potential of spectral representations and sparse modeling in shape analysis and processing. Fig. 1.1 highlights the hierarchy of our methodology. Specifically, the contributions of this dissertations are summarized as follows:

Sparse Approximation of 3D Shapes (Chapter 3)

We present an innovative approach to 3D mesh compression using spectral graph wavelets as dictionary to encode mesh geometry. In contrast to Laplacian eigenbasis, the spectral graph wavelets are locally defined at individual vertices and can better capture local shape information in a more accurate way. Nonetheless, the multi-scale spectral graph wavelets form a redundant

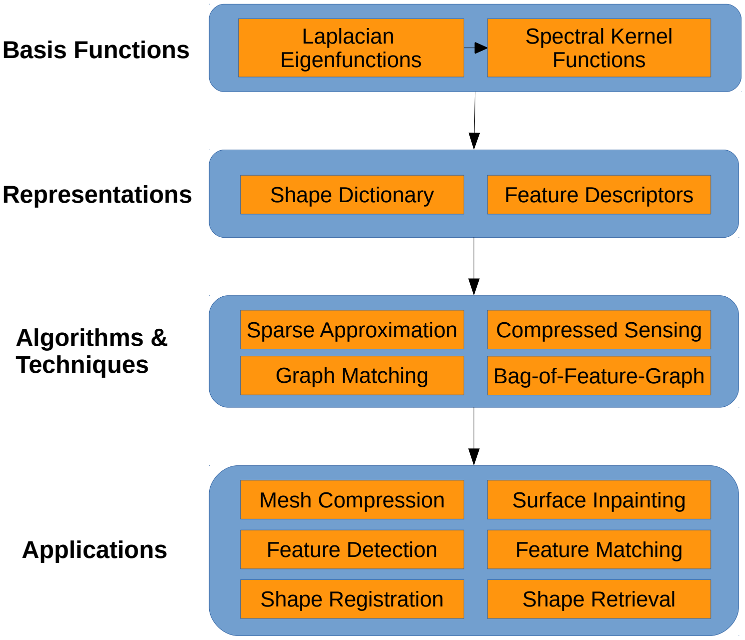


Figure 1.1: Conceptal hierarchy of this dissertation.

dictionary as shape bases, so we formulate the compression of 3D shape as a sparse approximation problem that can be readily handled by algorithms such as orthogonal matching pursuit. Various experiments demonstrate that our method are superior to the existing spectral mesh compression methods.

Sparsity-Driven Surface Inpainting (Chapter 4)

We devise a new algorithm for completing surface with missing geometry and topology founded upon the theory and techniques of sparse signal recovery. We find that for many shapes the vertex coordinate function can be well approximated by a very sparse coefficient representation with respect to the dictionary comprising its Laplacian eigenbasis, and it is then possible to recover this sparse representation from partial measurements of the original shape. Taking advantage of the sparsity cue, we advocate a novel variational approach for surface inpainting, integrating data fidelity constraints on the shape domain with coefficient sparsity constraints on the transformed domain. Because of the powerful properties of Laplacian eigenbasis, the inpainting results of our method tend to be smooth and globally coherent with the remaining shape. We demonstrate the performance of our new method via various examples in geometry restoration, shape repair, and hole filling.

Shape Feature Description based on Spectral Wavelets (Chapter 5)

We propose a new kind of shape feature descriptors built upon the coefficients of spectral graph wavelets and biharmonic distance fields. Our novel descriptors are both multi-scale and multi-level in nature, effectively encoding both local and global information for the characterization of user-specified feature regions. Via extensive experiments and comprehensive comparisons with the state-of-the-art, our descriptor has exhibited many attractive advantages such as being geometry-aware, versatile, robust, discriminative, and isometry-invariant.

Feature-Driven Shape Correspondence and Retrieval (Chapter 6)

We develop effective feature-driven articulated shape correspondence and retrieval algorithms based on spectral descriptors. For coarse matching, we adopt tensor-based high-order graph matching to maximizes the geometric compatibility between features tuples, and for dense matching, we present a hierarchical shape registration algorithm, generating correspondence in multiples levels in a coarse-to-fine manner. We also propose the novel Bag-of-Feature-Graph (BoFG) descriptor for shape retrieval. For each geometric word in the vocabulary, BoFG constructs a graph that records spatial relations of all feature pairs in the shape, weighted by their similarities to this word. The BoFG descriptor significantly reducing the number of points required for computing distributions in comparisons with more traditional Bag-of-Words descriptors.

1.3 Dissertation Organization

The remainder of this proposal is organized as follows. In Chapter 2, we briefly review harmonic analysis in the manifold and mesh domain, including the theories and applications of manifold harmonic basis and spectral graph wavelets; we also give an overview of the concept of sparse and redundant representation as well as related computational methods. In Chapter 3, we present an innovative approach to 3D mesh compression using spectral graph wavelets as dictionary to encode mesh geometry; in contrast to Laplacian eigenbasis, the spectral graph wavelets are locally defined at individual vertices and can better capture local shape information in a more accurate way. In Chapter 4, we devise a new algorithm for completing surface with missing geometry and topology founded upon the theory and techniques of sparse signal recovery using the manifold harmonic basis as dictionary. In Chapter 5, we propose a new kind of shape descriptors built upon spectral graph wavelets for the characterization of user-specified feature regions and develop a generalized feature detection framework. In Chapter 6, we present our new algorithms for feature-driven shape correspondence and retrieval based on spectral shape descriptors. Finally, we conclude this dissertation with discussions and outline a few future research directions in Chapter 7.

Chapter 2

Background Review

In this chapter, we first briefly review spectral analysis in the context of shape modeling, including the theories and applications of graph Fourier transform and spectral graph wavelets. Then we give an overview of the concept of sparse representation modeling as well as its related computational methods and applications in signal processing.

2.1 Laplacian and Graph Fourier Transform

Fourier transform is perhaps the most fundamental tool for classical time-frequency (or space-frequency) signal analysis. The key idea is to decompose a temporal/spatial signal into the linear combinations of a set of sinusoid functions (i.e., the Fourier basis) of different frequencies. The collection of Fourier basis functions $\{\phi_n; n \in \mathbb{Z}\}$ form a complete orthogonal basis [55] of the underlying function space $\mathcal{F}$, and the transform coefficients gives a frequency-domain representation of the original signal. In general, this decomposition can be written as

$$f = \sum_{k=-\infty}^{\infty} \langle f, \phi_k \rangle \phi_k \tag{2.1.1}$$

Consider the classical Fourier analysis in $\mathbb{R}^1$. From the perspective of differential equations, the kth Fourier basis function $\phi_k(x) = e^{-i\omega_k x}$ satisfies the following equation:

$$-\frac{\partial^2 \phi_k(x)}{\partial x^2} = \omega_k^2 \phi_k(x). \tag{2.1.2}$$

That is, in 1D Euclidean space, the Fourier basis functions are actually the eigenfunctions of the second-order derivative operator $\frac{\partial^2}{\partial x^2}$, which can be regarded as 1D Laplace operator. The eigenvalue ω_k^2 is the square of the frequency of associated eigenfunction ϕ_k.

The concept of Fourier transform and Fourier basis can be generalized to the manifold and mesh/graph domains, as will be described in this section.

2.1.1 Manifold Fourier Transform

Given a manifold $\mathcal{M}$ with Riemannian metric g. The goal is to find a family of functions $\{\phi_k(x)\}$ that form a orthonormal and complete basis of the Hilbert space $Ł^2(\mathcal{M}, \}) : \mathcal{M} \to \mathbb{R}$ with properties similar to the classical Fourier basis. In manifold space, the equivalent to the Laplace operator is the *Laplace-Beltrami operator* which acts on scalar functions defined on the manifold. We denote by $\Delta_{\mathcal{M}}$ the Laplace-Beltrami operator of $\mathcal{M}$. The Laplace-Beltrami operator is a self-adjoint and semi-positive definite operator [130], hence $\Delta_{\mathcal{M}}$ admits an orthonormal eigensystem. By solving the following Dirichlet problem for the Laplacian

$$\begin{cases} \Delta_{\mathcal{M}} f(x) = -\lambda f(x), \quad x \in \mathcal{M} \\ f|_{\partial\mathcal{M}} = 0 \end{cases} \tag{2.1.3}$$

we obtain the eigenvalues $\{\lambda_k\}_{k=0}^{\infty}$ and eigenfunctions $\{\phi_k(x)\}_{k=0}^{\infty}$ of the Laplace-Beltrami operator. According to the spectral theorem, the eigenvalues constitute a real diverging sequence

$$0 \leq \lambda_0 \leq \lambda_1 \leq \cdots \leq +\infty$$

and the eigenfunctions $\{\phi_k\}_{k=0}^{\infty}$ form a complete and orthonormal basis of the Hilbert space $L^2(\mathcal{M})$ [83], i.e., the space of integrable functions defined on $\mathcal{M}$.

The eigenvalues $\{\lambda_k\}_{k=0}^{\infty}$ are sometimes referred to as the Laplace-Beltrami spectra [128], which are analogous to $\{\omega_n^2\}$ in Eq. 2.1.2 in classical Fourier transform; their square roots can be deemed as the global shape frequencies. The eigenfunctions, also known as the manifold harmonics or shape harmonics [154], have global periodic oscillations on the manifold, behaving similarly to sine and cosine functions over the real line.

A function $f(x) \in L^2(\mathcal{M})$ then can be uniquely expanded on the manifold harmonics

$$f(x) = \sum_{k=0}^{\infty} \hat{f}(k)\phi_k(x), \tag{2.1.4}$$

where

$$\hat{f}(k) = \langle f(x), \phi_k(x) \rangle. \tag{2.1.5}$$

We may call $\{\hat{f}(k)\}$ the *manifold harmonics transform* of function $f(x)$.

2.1.2 Mesh Laplacian and Spectral Mesh Processing

Now we consider the discretization of Laplacian-Beltrami operator and MHB. In geometry processing, we often represent or approximate surfaces with discrete meshes. Consider a manifold $\mathcal{M}$ approximated by triangular mesh M with vertex set $V := \{v_i, i = 1, \ldots, N\}$, edge set E, and face set F. $|V| = N$ is the size of M. In addition, we define $N(i) = \{j|(v_i, v_j) \in E\}$ and $d_i = |N(i)|$. $N(i)$ denotes the index set of the 1-ring neighborhood of the vertex v_i, and d_i is the valence of v_i.

In principle, M can be viewed as the geometric embedding of a graph structure G into $\mathbb{R}^3$. In another word, M can be decomposed into the topological component, namely the underlying graph structure G, and the geometric component, i.e., the vertex coordinate function $\mathbf{P}$ which maps each vertex v_i of the graph to a position in the 3D Euclidean space $p_i \in \mathbb{R}^3$.

The discrete Laplace-Beltrami operator of M is a $N \times N$ matrix $\Delta_M = (\delta_{ij})$, i.e., the Laplace matrix, defined by the result of applying it to a function f defined on V:

$$\Delta_M f(v_i) = \sum_{j=0}^{N-1} \delta_{ij} f(v_j). \tag{2.1.6}$$

The Laplacian is a local operator and its effect only impacts each vertex's immediate neighbors. Hence, the Laplacian matrix has the general form

$$\Delta_M f(v_i) = \frac{1}{a_i} \sum_{j \in N(i)} w_{ij}(f(v_i) - f(v_j)), \tag{2.1.7}$$

where $N(i)$ denotes the index set of the 1-ring neighbors of v_i, and a_i are the masses associated with each vertex and w_{ij} represents the weight of each edge.

Depending on the choice of a_i and w_{ij}, mesh Laplacian may have many different forms and can be classified as either *combinatorial* or *geometric* [163].

One simple discretization of the smooth Laplace-Beltrami operator is to define $\Delta f(p_i)$ as the average difference between the function values at v_i and its 1-ring neighborhood

$$\Delta f(v_i) = \frac{1}{d_i} \sum_{j \in N(i)} [f(v_i) - f(v_j)]. \tag{2.1.8}$$

The corresponding Laplacian matrix is

$$\Delta(i,j) = \begin{cases} d_i & i = j \\ -1 & (p_i, p_j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.1.9}$$

Δ and some of its variations are called the *combinatorial Laplacian* or *graph Laplacian*, since they only take into account the connectivity of the underlying

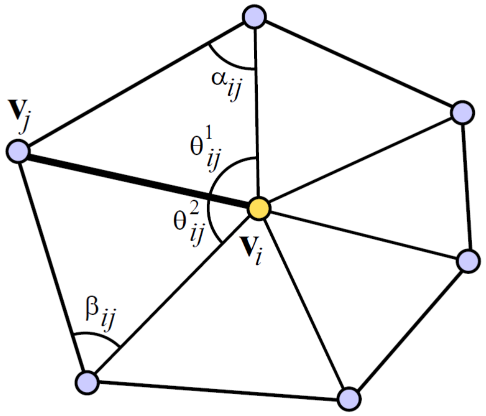


Figure 2.1: Angles in cotangent weights (courtesy of [143]).

graph topology while ignoring the geometric properties. Graph Laplacian can serve as a good approximation of the smooth Laplacian only when the mesh vertices are uniformly distributed.

To faithfully approximate smooth Laplacian on arbitrary meshes, we need to take into account geometric information such as the distances between neighboring vertices and the angles between contiguous edges. Such defined Laplacian is called the *geometric Laplacian*.

Constructing discrete Laplace-Beltrami operator on general meshes is not a trivial task. In fact, it is impossible to make discrete Laplacian to simultaneously converge to smooth Laplacian and be symmetric on general meshes [132]. Many different versions of geometric discrete Laplacian have been proposed [5, 33, 83, 122, 154, 160]. One of the most popular scheme was proposed by Meyer et al. [101]. It uses the cotangents of the two angles opposite to an edge to weight the edge, and the area of the Voronoi cell size surrounding a vertex to weight the vertex. Its action on vertex-based function f on mesh M is

$$\Delta f(v_i) = \frac{1}{a_i} \sum_{j \in N(i)} w_{ij} \frac{f(v_i) - f(v_j)}{d_i}. \tag{2.1.10}$$

Here a_i is the area of the Voronoi cells around vertex v_i, and the weights

$$w_{ij} := \frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2}, \tag{2.1.11}$$

where α_{ij} and β_{ij} denote the two angles opposite to the edge (v_i, v_j) (See Fig. 2.1).

Let us define the area matrix $A = diag(a_i)$ and weight matrix W as

$$W(i,j) = \begin{cases} \sum_{k\in N(i)} w_{ik} & i = j \\ -w_{ij} & (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

The geometric Laplacian matrix then can be written as $L = A^{-1}W$.

Generally, such defined L is not symmetric. However, we can rewrite the the equation $L\phi = \lambda\phi$ as the generalized eigenvalue problem

$$W\phi = \lambda A\phi. \tag{2.1.12}$$

Since W is symmetric and A is symmetric positive-definite, the generalized eigenvectors $\phi_{\mathbf{i}}$ corresponding to different generalized eigenvalues λ_i are orthogonal, and all of the generalized eigenvalues/eigenvectors are real. We should note that the orthogonality is with respect to the inner product induced by A

$$\langle\phi_{\mathbf{i}}, \phi_{\mathbf{j}}\rangle = {\phi_{\mathbf{i}}}^T A\phi_{\mathbf{j}} = 0, \quad i \neq j. \tag{2.1.13}$$

If the mesh vertices are evenly distributed, i.e., each vertex has the same Voronoi cell size, we can make $A = I$ by proper normalization. In this case, the A-inner product becomes the standard dot product (I-inner product). This is unfortunately not valid for general meshes whose vertices are not distributed uniformly over the surface area. We may adopt the symmetric version of mesh Laplacian $L_s = A^{-1/2}WA^{-1/2}$, which will give the same eigenvalues [154]. The original eigenvectors can be obtained by $\phi_{\mathbf{i}} = W^{-1/2}\phi_{\mathbf{i}}^{\mathbf{s}}$. However, this symmetrization is not preferable since solving the generalized eigenvalue problem is more stable than inverting W [127].

Let $\{\lambda_i\}_{i=0}^{N-1}$ be the set of generalized eigenvalues of $\Delta_M = A^{-1}W$, and $\{\phi_{\mathbf{i}} \in \mathbb{R}^N\}$ their corresponding eigenvectors. A square-integrable scalar function $\mathbf{f}$ defined on V can be expanded as the linear combination of the eigenvectors

$$\mathbf{f}(p) = \sum_{k=0}^{N-1} \langle \mathbf{f}, \phi_{\mathbf{k}} \rangle_A \phi_{\mathbf{k}}(p), \tag{2.1.14}$$

where the inner-product is the A-induced scalar product

$$\langle \mathbf{f}, \mathbf{g} \rangle_A = \mathbf{f}^T A \mathbf{g} = \sum_{i=0}^{N-1} a_i \mathbf{f}(i) \mathbf{g}(i). \tag{2.1.15}$$

Unlike classic Fourier basis functions which are simply fixed sinusoids, the manifold harmonic basis differ with the connectivity, geometry, and the type of Laplacian operator that is adopted [163]. As a result, the mesh Laplacian eigenvectors and eigenvalues actually encode substantial topological and geometric information and can help characterize the global shape property and reveal intrinsic structure of the original mesh. In addition, Laplace-Beltrami operator is globally defined and is completely determined by the metric tensor, which is itself an isometry invariant. Hence, the Laplacian eigenvalues and eigenfunctions encode meaningful global intrinsic information about the shape and they are invariant under isometric deformations up to a change in sign [132, 146]. This lends to the popularity of spectral methods in the area of geometry processing and analysis.

There has been a great number of methods in literature directly employing the Laplacian eigenstructures to tackle various shape modeling problems, including compression [67], segmentation [90], deformation [129], remeshing [35], paramterization [166], shape indexing [128, 132], and retrieval [80]. We refer readers to [163] for a more thorough review of the theories and applications of spectral methods for shape analysis and processing.

2.2 Kernels and Spectral Graph Wavelets

2.2.1 Kernel Functions

On manifold $\mathcal{M}$, consider a bivariate kernel $\Theta = \{\theta(x,y)\}$ which shares the same eigenvectors, $\{\phi_i\}_{i=0}^{\infty}$, with the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$. Suppose Θ admits an eigendecomposition $\Theta\phi_i = \alpha_i\phi_i$ with the eigenvalues $\{\alpha_i\}_{i=0}^{\infty}$, then the kernel function can be represented as

$$\theta(x,y) = \sum_{i=0}^{\infty} \alpha_i \phi_i(x)\phi_i(y), \tag{2.2.1}$$

or

$$\Theta = \Phi diag(\{\alpha_i\})\Phi^t. \tag{2.2.2}$$

This is to say, Θ can be diagonalized by the eigenbasis of $\Delta_{\mathcal{M}}$.

Suppose the eigenvalues of Θ is some functions of the eigenvalues of $\Delta_{\mathcal{M}}$, i.e., there is some function $g(\cdot)$ such that $\alpha_i = g(\lambda_i)$, then the kernel can be expressed as

$$\theta(x,y) = \sum_{i=0}^{\infty} g(\lambda_i)\phi_i(x)\phi_i(y). \tag{2.2.3}$$

Since the Laplacian eigenvalues $\{\lambda_i\}$ can be regarded as the frequency of the shape, $g(\cdot)$ is essentially a transfer function and Θ is a filtered version of $\Delta_{\mathcal{M}}$

The concept of manifold/graph Fourier transform can be extended to the bivariate case to define kernels. Suppose we have a bivariate kernel $\theta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ which corresponds to a self-adjoint operator Θ. The bivariate kernel θ can be expanded on the manifold Fourier basis

$$\theta(x,y) = \sum_{k=0}^{\infty} \hat{\theta}(k)\phi_k(x)\phi_k(y), \tag{2.2.4}$$

where

$$\hat{\theta}(k) = \langle\langle\theta(x,y), \phi_k(x)\rangle, \phi_k(y)\rangle. \tag{2.2.5}$$

$\hat{\theta}(k)$ can be deemed as the Fourier transform of the bivariate kernel with a slight abuse of language.

For example, the Laplace-Beltrami operator itself can be expanded as

$$\Delta_{\mathcal{M}}(x,y) = \sum_{i=0}^{\infty} \lambda_k \phi_k(x)\phi_k(y). \tag{2.2.6}$$

Its Fourier transform therefore is $\widehat{\Delta_{\mathcal{M}}}(k) = \lambda_k$.

The kernel function $\theta(x,y)$ describes the relations between each pair of points on the manifold, and its diagonal $\theta(x,x)$ amounts to a signature function for the characterization of individual points.

One well known kernel function is the heat kernel $K(t,x,y)$, which is the fundamental solution to the heat equation with appropriate boundary condition. For example, the heat kernel for the Dirichlet problem is solution to the equation

$$\begin{cases} \frac{\partial K}{\partial t}(t,x,y) = \Delta_{\mathcal{M}} K(t,x,y) & \forall t > 0 \text{ and } \forall x, y \in \mathcal{M} \\ \lim_{t\to 0} K(t,x,y) = \delta_x(y) & \forall x, y \in \mathcal{M} \\ K(t,x,y) = 0. & x \in \partial\mathcal{M} \text{ or } y \in \partial\mathcal{M} \end{cases} \tag{2.2.7}$$

Expanded on the manifold harmonic functions, the heat kernel with time parameter t has the following expression:

$$K(t,x,y) = \sum_{k=0}^{\infty} e^{-\lambda_k t} \phi_k(x)\phi_k(y). \tag{2.2.8}$$

On the mesh domain, the heat kernel affords a multi-scale, stable and intrinsic characterization of the geometric shape [146], and have been the foundation for many shape analysis applications including shape matching [114] and shape retrieval [14].

2.2.2 Wavelets on Graphs

Wavelet is a powerful analytical tool in signal processing. Intuitively speaking, a wavelet is just a wave-like pulse in time (or space). By scaling and translating a single mother wavelet, we may obtain a family of wavelet functions that cover the entire domain in question. Similar to Fourier analysis in which a function f can be decomposed into a series of component harmonics, in wavelet analysis f can be expanded by a family of component wavelets. Nevertheless, there are some fundamental differences between the Fourier and wavelet analysis:

- In Fourier analysis, each component harmonic is globally defined in space/time. In wavelet analysis, the component wavelet are all locally defined at different locations.

- In Fourier analysis, each component harmonic has an exclusive frequency. In wavelet analysis, we use multiple wavelets localized at different locations to represent the information of a single frequency. Generally, for large scale information (low frequency), we use fewer wavelets; whereas for small scale information (high frequency), we use more wavelets.

- The component harmonics in Fourier analysis are all orthogonal to each other. In fact, the Fourier basis functions form an orthonormal basis of the space of square-integrable functions. This is not necessarily true for wavelet functions.

In a nutshell, wavelet functions can be simultaneously localized in both time/space and frequency domain, in contrast to the Fourier transform in which the basis harmonic functions are all globally defined in time/space. For signals whose primary information lies in localized singularities, such as edges in images or step discontinuities in time series signals, wavelet transform affords a more compact representations than a transform with global basis such as the Fourier transform.

There have been many efforts to introduce wavelet methods to the field of visual computing. Representative applications include image segmentation [44], image-based rendering [112], volume rendering [89], scientific visualization [30], spectral rendering [28], animation compression [119], etc.

Classical wavelets are constructed by translating and scaling a mother wavelet in Euclidean space, however, transplanting wavelets to graphs (specifically, triangular meshes) is not straightforward due in part to the fact that it is unclear how to apply the scaling operation on a signal that is defined on the mesh vertices, so early studies using wavelet mostly relied on the spherical parameterization [92, 159].

One popular scheme to imitate scaling on meshed surfaces is achieved via explicit subdivision, which iteratively refines the mesh geometry, and at the same time, also refines the functions defined on the mesh. The constructed wavelets are biorthogonal and locally supported. The subdivision wavelets rely on the subdivision connectivity of the mesh, which restricts the application scope to data compression and level-of-detail rendering. The idea of subdivision wavelets was first proposed by Schröder and Sweldens [136], in which the lifting scheme was used to construct wavelets on sphere. Lounsbery et al. [93] studied MRA of wavelets constructed on surfaces of arbitrary topology type. In [8], Bertram et al. utilized bicubic B-spline subdivision to construct wavelet transform that affords boundary curves and sharp features. As a drawback, the subdivision wavelet requires the meshes to have subdivision connectivity, where remeshing process is frequently needed. To avoid remeshing, Valette and Prost [153] extended the subdivision wavelet for triangular meshes using irregular subdivision scheme that can be directly computed on irregular meshes. On spherical domains, Haar wavelets [11, 108] were constructed over nested triangular grids generated by subdivision. Recently, the spherical Haar wavelet basis was improved to the SOHO wavelet basis [82] that is both orthogonal and symmetric.

Another method to construct graph and mainfold wavelets is through diffusion [29, 62]. In sharp contrast to the aforementioned subdivision wavelets, the diffusion wavelets adopt a bottom-up philosophy starting from the fine input data. The *Diffusion wavelets* [29] use a diffusion operator and its powers to expand the nested subspaces, where scaling functions and wavelet functions are obtained by orthogonalization and rank-revealing compression. This diffusion-driven methodology naturally dilates the functions associated with the underlying heat diffusion process, which solely depends on manifold geometry. It allows flexible construction directly from data. However, the constructed scaling and wavelet functions are not locally-supported, which limits the functionality of space localization. In fact, it is impossible to construct wavelets that are simultaneously fully orthogonal, locally supported, and symmetric [93]. As an improvement, the biorthogonal diffusion wavelets (BDW) [95] were introduced, relieving the excessively-strict orthogonality property of scaling functions. In [97], diffusion wavelets were adopted to approximate scalar-valued functions based on analyzing the structure and topology of the state space. Rustamov [133] studies the relation between mesh editing and diffusion wavelets by introducing the generalized linear editing (GLE). However, neither the DW nor the BDW have achieved localization in both manifold and frequency domain.

In [62], an admissible diffusion wavelets (ADW) on meshed surfaces and point clouds is proposed. The ADW are constructed in a bottom-up manner that starts from a local operator in a high frequency, and dilates by its dyadic powers to low frequencies. By relieving the orthogonality and enforcing normalization, the wavelets are locally-supported and admissible.

It is attractive to be able to define wavelet transform directly on 3D shapes without the need of parameterization. Various schemes of manifold wavelets have been proposed via different approaches [2]. Diffusion wavelets, introduced by Maggioni and Coifman [29], use diffusion as a scaling tool to achieve multiscale analysis. Wavelet and scaling functions are constructed by repeatedly applying a diffusion operator T on the graph or manifold space. After applying dyadic powers of T at each scale, a localized orthogonalization procedure is performed to yield nested approximation spaces, and then wavelets are produced by locally orthogonalizing vectors spanning the difference of these approximation spaces. The derived diffusion wavelets are orthogonal, compact, and multiscale in nature, and have been employed in 3D mesh compression in [96]. In [53,125], tree-based, data-adaptive wavelet transforms are developed for high-dimensional Euclidean data sets and weighted graphs, under the assumption that the data have a rich geometrical structure that can be captured by a hierarchical tree.

2.2.3 Spectral Graph Wavelets

The primary reason that classical wavelet transforms cannot be directly adapted to graph or manifold is that for a mother function $\psi(x)$ defined on a manifold, there is no obvious definition for $\psi(sx)$. One approach to solve this problem is appealing to the Fourier domain, with the help of aforementioned manifold harmonics. Although scaling cannot be explicitly expressed on manifold domain, it can be easily defined on the frequency domain. The idea of spectral wavelet transform was introduced in [59] on the graph domain, denoted as the Spectral Graph Wavelet Transform (SGWT). Here we extend the concept to general manifold, denoted as the Spectral Graph Wavelet Transform (SGWT)

Given manifold $\mathcal{M}$ with appropriate boundary condition. Assume its Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ has the eigen-decomposition $\{\lambda_k, \phi_k\}$. The eigenvectors $\{\phi_k\}$ form a complete and orthonormal basis of $L^2(\mathcal{M})$, commonly known as the manifold harmonics. The corresponding eigenvalues $\{\lambda_k\}$ satisfy

$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots \tag{2.2.9}$$

For any function f defined on $\mathcal{M}$, its generalized Fourier transform $\hat{f}$ is defined as

$$\hat{f}(k) = \langle \phi_k, f \rangle = \sum_{k=0}^{\infty} \phi_k(x) f(x) \tag{2.2.10}$$

And the inverse Fourier transform is

$$f(x) = \sum_{k=0}^{\infty} \hat{f}(k) \phi_k(x) \tag{2.2.11}$$

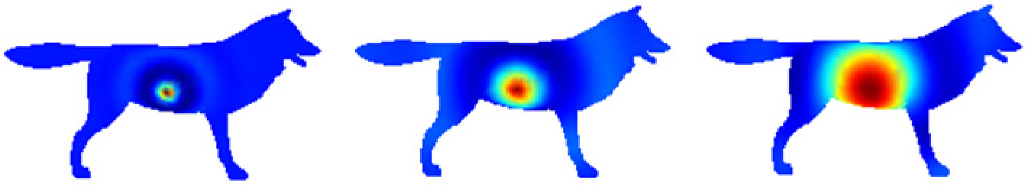


Figure 2.2: Spectral Graph Wavelets centered at one vertex on the wolf model. From left to right are wavelets from high frequency to low frequency with scale 1, 3 and 5.

The Parseval relations holds for the manifold harmonics transform

$$\langle f, g \rangle = \langle \hat{f}, \hat{g} \rangle \tag{2.2.12}$$

We generate SGWT from a special wavelet operator that acts on functions defined on the manifold. Given a real-valued transfer function g, the wavelet operator T_g is defined by how it modulate on $f : \mathcal{M} \to \mathbb{R}$ on Fourier domain

$$\widehat{T_g f}(k) = g(\lambda_k)\hat{f}(k) \tag{2.2.13}$$

Employing the inverse Fourier transform, we obtain the spectral representation of $T_g f$

$$(T_g f)(\cdot) = \sum_{k=0}^{\infty} g(\lambda_k)\hat{f}(k)\phi_k(\cdot) \tag{2.2.14}$$

To obtain spectral wavelets, we need localized and scaled versions of $T_g f$. The scaling is defined by dilating the transfer function as $g(t\lambda_k)$. The localization at point $x \in \mathcal{M}$ is realized by applying wavelet operators to unit impulse at x, represented by the Dirac delta function $\delta_x(\cdot)$

Since

$$\delta_x(\cdot) = \sum_{k=0}^{\infty} \phi_k(x)\phi_k(\cdot) \tag{2.2.15}$$

We have the Fourier transform of δ_x

$$\hat{\delta_x}(k) = \phi_k(x) \tag{2.2.16}$$

Set $f = \delta_x$ in (2.2.14), we have the spectral wavelet at scale t and localized at point x

$$\psi_{t,x}(\cdot) = (T_g^t \delta_x)(\cdot) = \sum_{k=0}^{\infty} g(t\lambda_k)\phi_k(x)\phi_k(\cdot) \tag{2.2.17}$$

The spectral wavelet can also be represented as a bivariate kernel

$$\Psi_t(x, y) = \psi_{t,x}(y) = \sum_{k=0}^{\infty} g(t\lambda_k)\phi_k(x)\phi_k(y) \tag{2.2.18}$$

For a real-valued function f defined on $\mathcal{M}$, the spectral wavelet transform is

$$\mathcal{W}_f^{\psi}(x, t) = \langle \psi_{x,t}, f \rangle \tag{2.2.19}$$

Applying the Parseval relation (2.2.12), we obtain the spectral representation of continuous spectral wavelet transform

$$\mathcal{W}_f^{\psi}(x, t) = \langle \hat{\psi_{x,t}}, \hat{f} \rangle = \sum_{k=0}^{\infty} \hat{\psi_{x,t}}(k)\hat{f}(k) = \sum_{k=0}^{\infty} g(t\lambda_k)\hat{f}(k)\phi_k(x) \tag{2.2.20}$$

If seen as a function of x, the Fourier transform of the above spectral wavelet transform is

$$\widehat{\mathcal{W}_f^{\psi}}(k) = g(t\lambda_k)\hat{f}(k) \tag{2.2.21}$$

Similar to classical wavelet transform, the spectral wavelet transform is invertible only if the transfer function g satisfies the admissibility condition

$$C_{\psi} = \int_0^{\infty} \frac{g^2(a)}{a} da < \infty \tag{2.2.22}$$

and the zero-mean condition $g(0) = 0$.

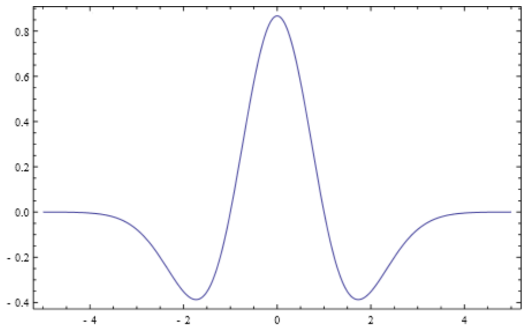
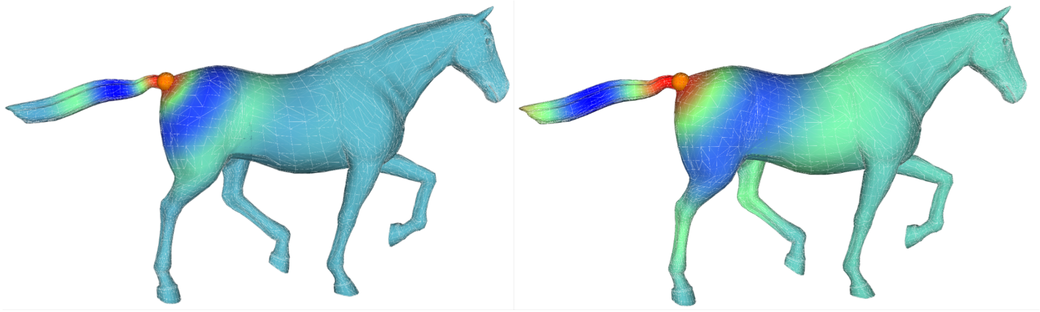


Figure 2.3: 1D Mexican-hat Wavelet



$t = 10$ $t = 30$

Figure 2.4: Color plots of the spectral Mexican-hat wavelet with scales of 10 and 30. The reference point is denoted by the orange ball. We can see that the function values oscillate around the reference point. When t is larger, the spread of oscillation also increases.

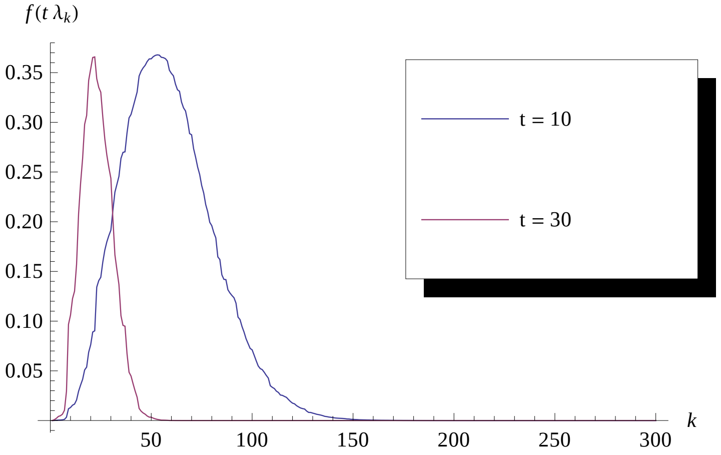


Figure 2.5: The Mexican-hat wavelet transfer function in the frequency domain.

In classical wavelets defined on real line, the space localization is apparent. If the mother wavelet $\psi(x)$ is localized in the interval $[-\epsilon, \epsilon]$, then the wavelet $\psi_{a,b}(x)$ will be localized with $[b - a\epsilon, b + a\epsilon]$. in the limit as $b \to 0$, $\psi_{a,b}(x) \to 0$ for $x \neq b$.

For spectral wavelets, the localization property is less straightforward since the scaling is defined implicitly in the Fourier domain. For g sufficiently regular, the normalized spectral wavelet $\psi_{t,x}/\|\psi_{t,x}\|$ will vanish on vertices sufficiently far from x in the limit of fine scales, i.e. as $t \to 0$. We should expect $\psi_{t,x}(y)$ to be small if x and y are separated and t is small.

If two transfer functions g and g' are close to each other, then the derived spectral wavelets should be close to each other in the manifold domain.

As an example, we consider the Mexican hat wavelet. In 1D Euclidean space, the Mexican hat wavelet is defined as

$$\psi(t) = \frac{2}{\sqrt{3\sigma}\pi^{\frac{1}{4}}}(1 - \frac{t^2}{\sigma^2})e^{\frac{-t^2}{2\sigma^2}}. \qquad (2.2.23)$$

Its graph is shown in Fig. 2.3, which exhibits clear localization in space. In manifold space, we may analogously define the spectral Mexican hat wavelet as

$$\psi_{t,x}(\cdot) = \sum_{k=0}^{\infty} t^2 \lambda_k^2 e^{-t^2\lambda_k^2} \phi_k(x)\phi_k(\cdot), \tag{2.2.24}$$

with the transfer function $g(t\lambda) = t^2\lambda^2 e^{-t^2\lambda^2}$.

Fig. 2.4 visualizes the value of the wavelet functions over the surface, with the scale $t = 10$ and $t = 30$. Fig. 2.5 shows the Fourier transform of the wavelet functions in frequency domain. It is easy to see that

- The spectral wavelet function is localized in both manifold and frequency domains.
- On manifold, the values of the spectral wavelet functions attenuate and oscillate as the distance from the reference point increases.
- For a larger scale, the spectral wavelet has a wider windows in space, but a narrower window in frequency.

By construction, the spectral wavelets $\psi_{t,x}$ are all orthogonal to the the null eigenvector ϕ_0, and nearly orthogonal to ϕ_l for λ_l near zero [59]. For more effective representation of low-frequency signals, it is necessary to introduce the spectral scaling functions, which are also defined as a bivariate kernel by a single real-valued generator function $h : \mathbb{R}^+ \to \mathbb{R}$, which acts as a low-pass filter and satisfies

$$\begin{cases} h(0) > 0 \\ h(x) \to 0|_{x\to\infty}. \end{cases} \tag{2.2.25}$$

Introducing the scaling functions helps ensure stable recovery of the original signal f from the wavelet coefficients when the scale parameter t is sampled at discrete values t_j. Stable recovery will be assured if $G(\lambda) = h(\lambda)^2 + \sum_{j=1}^{J} g(t_j\lambda)^2$

is bounded away from zero. The design of the scaling function generator h is uncoupled from the choice g.

The spectral wavelets depend on the continuous scale parameter t. For practical computation, t must be sampled at a finite number of scales $\{t_j\}_{j=1}^{J}$, which generates NJ wavelets $\psi_{t_j,n}$ along with N scaling functions s_n. It can be proven [59] that the set $\Gamma = \{\phi_n, n = 0, \ldots, N-1\} \cup \{\psi_{t_j,n}, j = 1, \ldots, J, n = 1, \ldots, N\}$ form a frame with bounds

$$A = \min_{\lambda \in [0, \lambda_{N-1}]} G(\lambda)$$

and

$$B = \max_{\lambda \in [0, \lambda_{N-1}]} G(\lambda),$$

where $G(\lambda) = h^2(\lambda) + \sum_j g(t_j\lambda)^2$. That is to say, for all f defined on the manifold, the following inequality holds:

$$A\|f\|^2 \leq \sum_k |\langle f, \Gamma_k \rangle|^2 \leq B\|f\|^2, \tag{2.2.26}$$

where $\Gamma = \{\Gamma_k\}$.

The spectral wavelet transform is an overcomplete transform, mapping an input vector f of size N to $N(J+1)$ coefficients $c = Wf$. Given a set of coefficients c, the synthesis/reconstruction of f can be given by solving the matrix equation

$$(W^*W)f = W^*c. \tag{2.2.27}$$

Because of its attractive and powerful properties such as spatial localization, multiscale, and geometry awareness, SGW has already been adopted as a descriptor for a handful of shape analysis applications. Kim et al. [72, 73] introduced a wavelet-based multi-scale descriptor for the analysis of cortical surface signals using the SGWT and Li et al. [84] proposed a SGWT-based descriptor and utilized the intrinsic spatial pyramid matching (ISPM) for global shape retrieval. Though these researches discover the potentials of SGWs, they all concentrate on global shape analysis based on point signatures, ignoring the SGWs' power in integrating the local-to-global geometrical and topological information.

2.3 Sparse Representation Modeling

In recent years, sparse modeling techniques have become increasingly popular in various fields of signal processing and analysis, especially in image processing and computer vision. Its widespread applications include data compression, signal denoising, pattern recognition, etc.

The fundamental idea of sparse modeling is to decompose or approximate the signal in question as the linear combination of a very small subset of vectors selected from a large number of candidate elementary vectors. These candidate elementary vectors, also called *atoms*, constitute a set called the *dictionary*. With a given dictionary, the signal is encoded by the coefficients w.r.t. the selected atoms and can be easily reconstructed. The rationale of sparse modeling is that most meaningful high-dimensional signals probably have some intrinsic structures or patterns, which can be exploited for efficient representation in a subspace of much lower dimension. It is often desirable for the dictionary to be redundant or overcomplete such that it can accommodate a sparse set of atoms that can better capture an input signal's intrinsic characteristics.

2.3.1 Sparse Modeling Problems

Consider signal $\mathbf{b} \in \mathbb{R}^n$ and dictionary matrix $\mathbf{D} \in \mathbb{R}^{n \times m}$ with $n < m$. If the dictionary constitutes an overcomplete basis of $\mathbb{R}^n$, the linear systems of equations $\mathbf{D}\mathbf{x} = \mathbf{b}$ is underdetermined and has infinite solutions.

To make the solution $\mathbf{x}$ unique, we can introduce an objective function $J(\mathbf{x})$ to govern the desired properties of the solution vector $\mathbf{x}$. The general optimization problem subject to linear equality constraints is as follows:

$$\min_{\mathbf{x}} J(\mathbf{x}) \quad \text{subject to} \quad \mathbf{b} = \mathbf{D}\mathbf{x}. \tag{2.3.1}$$

Alternatively, the objective function can be used as a regularization term, giving rise to the following regularized least square problem:

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda J(\mathbf{x}). \tag{2.3.2}$$

The regularization term can be viewed as imposing certain priors distributions on the solution $\mathbf{x}$. Comparing with Eq. 2.3.1, regularization allows users to make a tradeoff between the reconstruction fidelity and the desired property (e.g., smoothness) of the solution.

Eq. 2.3.1 and Eq. 2.3.2 are commonly encountered in different areas, including signal processing, machine learning and statistics. One common choice for $J(\mathbf{x})$ is the function of squared l_2-norm $\|\mathbf{x}\|_2^2$, which aims to minimize the "energy", i.e., the Euclidean norm of the solution vector. Eq. 2.3.1 then has the closed-form solution

$$\hat{\mathbf{x}} = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{b} = \mathbf{D}^+\mathbf{b}, \tag{2.3.3}$$

where $\mathbf{D}^+ = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}$ is the pseudo-inverse of $\mathbf{D}$. Let $J(\mathbf{x}) = \mathbf{\Gamma}\mathbf{x}$, where $\mathbf{\Gamma} \in \mathbb{R}^{m\times m}$, Eq. 2.3.2 then becomes the famous Tikhonov regularization [54], which has the explicit solution

$$\hat{\mathbf{x}} = (\mathbf{D}\mathbf{D}^T + \lambda\mathbf{\Gamma}\mathbf{\Gamma}^T)^+\mathbf{D}^T\mathbf{b}. \tag{2.3.4}$$

Due to its computational simplicity, the l_2 norm is commonly used as the or penalty term. However, for many applications, minimizing the total energy of solutions is not very meaningful.

If we know that the signal $\mathbf{b}$ has a very sparse coefficient representation $\mathbf{x}'$ with respect to $\mathbf{D}$, i.e., $\mathbf{b} \approx \mathbf{D}\mathbf{x}'$, $\|\mathbf{x}'\|_0 \ll m$, then we can specify $J(\mathbf{x})$ such that the objective to minimize the number of non-zero coefficients in solution $\mathbf{x}$.

The resultant optimization problem is called sparse decomposition and can be formulated as follows:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{b} = \mathbf{D}\mathbf{x}. \tag{2.3.5}$$

Here $\|\mathbf{x}\|_0$ denotes the pseudo norm of $\mathbf{x}$ which counts the number of non-zero elements in $\mathbf{x}$.

For real life signals, exact sparsity w.r.t. a fixed dictionary is elusive; instead, natural signals tend to be *compressible*, meaning that the representation coefficients decay rapidly when sorted in order of decreasing magnitude. Hence, a more practical formulation is to allow a bounded error of the sparsely reconstructed signal $\mathbf{D}\mathbf{x}$, giving rise to the best subset selection problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{b} - \mathbf{D}\mathbf{x}\|_2^2 < \epsilon. \tag{2.3.6}$$

or written as a regularization problem,

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0 \tag{2.3.7}$$

Clearly, the approximation quality and the sparsity of the coefficient vector depend both on the signal itself and the dictionary $\mathbf{D}$. We can expect more concise and accurate coefficient representations if the atoms in $\mathbf{D}$ can better capture properties of concerned signal. Take the 2D image domain as example. The global Fourier basis vectors are suitable for representing global signal trends, while wavelets, with its local support, can better represent isotropic features of different scales. To efficiently encode anisotropic image feature such as lines and curves, various extensions to wavelets have been proposed including contourlet [34], ridgelets [22], curvelets [20], etc. By incorporating new design parameters such as directions, these new shape basis afford more effective characterization of images dominated with different features.

2.3.2 Computational Methods

The l_0 optimization problems (Eq. 2.3.5 and Eq. 2.3.6) are NP-hard, so searching through all possible support sets by brute force is intractable except for problems of very small sizes. A variety of methods have been developed for finding near-optimal solution to l_0 optimization, and the most important two classes are *greedy pursuit* and *convex relaxation* [152].

Greedy Pursuit Methods

One of the most commonly-used approaches to sparse approximation is the greedy pursuit method. The central idea is to iteratively refine a sparse solution in a greedy manner. More specifically, in each iteration one or more atoms of the dictionary are chosen and the corresponding coefficients are modified such that the greatest improvement in approximation quality can be achieved. Representative greedy pursuit algorithms include matching pursuit (MP) [99], orthogonal matching pursuit (OMP) [116], and simultaneous orthogonal matching pursuit (S-OMP) [151]. The latter is suitable for solving the simultaneous sparse approximation problem where the input signal have multiple correlated channels and the same subset of atoms is to be used for every channel.

The basic idea of greedy pursuit methods is to iterative refine the estimation of the coefficient vector $\mathbf{x}$. In each step, one or several coefficients are modified to yield biggest possible improvement in approximating the signal. The simplest greedy pursuit algorithm is *matching pursuit (MP)* [99], described as follows

1. Set the index set $\Omega = \emptyset$, the residual $\mathbf{r_0} = \mathbf{b}$, and the counter $k = 1$.

2. Find an index n_k such that atom α_{n_k} is most correlated with the residual

$$n_k = \arg\max_n |\langle \mathbf{r}_{k-1}, \alpha_n \rangle|,$$

   and add n_k to Ω. The coefficient corresponding to α_{n_k} is denoted as x_{n_k}.

3. Update the residual $\mathbf{r}_k = \mathbf{r}_{k-1} - x_{n_k}\alpha_{n_k}$.

4. Increment k. Repeat (2)-(4) until stopping criterion is met.

The possible stopping criteria can be that a fixed number of atoms have been selected, the magnitude of residual is smaller than a threshold, or no remaining atoms have strong correlation with the residual.

A much improved version of MP is the algorithm known as *orthogonal matching pursuit (OMP)* [116]. The major difference with MP is an additional step of coefficients update. After a new index (that is most strongly correlated with the residual) is identified and added to the index set Ω, the coefficients calculated in previous steps are replaced with new coefficients which approximate the original signal in the least square sense. Contemporary greedy pursuit methods have more sophisticated mechanism for selecting and pruning the set of active atoms, including the stagewise orthogonal matching pursuit (StOMP) [37], regularized orthogonal matching pursuit [150], and compressive sampling matching pursuit (CoSaMP) [105]. Apart from greedy selection, another method to achieve greedy pursuit is iterative thresholding, or *shrinkage* [10, 31]. The basic idea is to iterative updating the coefficients with thresholding applied in each step to enforce sparsity.

Theoretically, it has been proved that greedy pursuit methods can produce near-optimal sparse approximations if the dictionary is sufficiently incoherent [149]. If the dictionary is sufficient random and the signal is sparse enough, simple pursuit methods can provably recover the sparse representation with high probability [150].

Convex Relaxation Methods

Another approach to sparse approximation problems is to replace the highly discontinuous l_0 norm with l_1 norm, yielding l_1 optimization problems which are convex and tractable. This convex relaxation is generally reasonable, as it has been proved that for most large underdetermined system, l_1 and l_0 minimization will produce the same unique solution, provided the signal is sufficiently sparse [36].

The two most common l_1 optimization problems are *basis pursuit*

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{b} = \mathbf{D}\mathbf{x}, \tag{2.3.8}$$

and *basis pursuit denoising (BPDN)*

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{b} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{2.3.9}$$

where the parameter λ is a regularization parameter which governs the sparsity of the solution.

Practical methods for solving l_1 optimization include *interior point methods* [19, 26, 71], iteratively reweighted least squares (IRLS) algorithm [25, 32, 126], and stepwise algorithms [39, 111, 123]. We refer readers to [17] and [152] for more detailed review of the algorithms for sparse solution of underdetermined linear system.

2.3.3 Applications

Sparsity-drive signal processing have numerous applications, and the following are the most commonly encountered.

- **Analysis.** Given signal $\mathbf{y}$ which is generated from a sparse coefficient vector $\mathbf{x}_0$ with respect to dictionary $\mathbf{A}$, i.e., $\mathbf{y} = \mathbf{A}\mathbf{x}_0$, we can compute a sparse approximation of $\mathbf{y}$ by solving Eq. 2.3.7. Under certain incoherent and sparsity conditions, the sparse solution $\mathbf{x}$ can well recover the true underlying vector $\mathbf{x}_0$.

- **Compression.** With greedy pursuit algorithms such as OMP, a nonlinear approximation of the original signal can be easily computed, yielding a compressive coefficient representation that can best approximate the original signal.

- **Denoising.** Suppose the observation of the original sparse signal $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ is the noisy version $\tilde{\mathbf{y}} = \mathbf{y} + \mathbf{v}$. When $\mathbf{x}_0$ is sufficiently sparse, it can often be reliably recovered by solving the sparse approximation problem, which subsequently yields a denoised signal.

- **Compressed sensing.** Let $P \in \mathbb{R}^{j \times n}$ be a measurement matrix of the original sparse signal $\mathbf{y} = \mathbf{A}\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $j < n$. Given measure $\mathbf{c} = P\mathbf{y}$, the sparse coefficient vector $\mathbf{x}$ can be recovered by solving

  $$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{c} - \mathbf{P}\mathbf{A}\mathbf{x}\|_2 \leq \epsilon, \tag{2.3.10}$$

  as long as the original signal is sufficiently sparse and the sensing matrix $\mathbf{P}$ conforms to restricted isometry property. The original signal then can be recovered from the coefficient representation.

- **Source separation.** Suppose the observed signal $\mathbf{y}$ is the superposition of two different sub-signals $\mathbf{y_1}$, $\mathbf{y_2}$, which are sparsely generated with dictionaries $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively. The sparse coefficient vectors of the two sub-signals can be estimated by solving the following optimization problem

  $$\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2 = \min_{\mathbf{x}_1, \mathbf{x}_2} \|\mathbf{x}_1\|_0 + \|\mathbf{x}_2\|_0 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A_1}\mathbf{x_1} - \mathbf{A_2}\mathbf{x_2}\|_2^2 \leq \epsilon_1^2 + \epsilon_2^2, \tag{2.3.11}$$

  and the separate sub-signals can be reconstructed as $\hat{\mathbf{y}}_1 = \mathbf{A}_1\hat{\mathbf{x}}_1$ and $\hat{\mathbf{y}}_2 = \mathbf{A}_2\hat{\mathbf{x}}_2$.

Chapter 3

Mesh Approximation and Compression

Sparse approximation have gained great success on the regular domain of 2D images. However, the topics of approximating or compressing signals defined over meshes are much less investigated due to the irregularity of underlying domains. In this chapter, we takes an initiative to explore the challenging problem of sparse approximation of discrete 3D shapes for compressive shape representation.

3.1 Introduction

Conventional Fourier analysis decomposes a signal into mutually independent components with the multiscale and orthogonal Fourier bases. Compression is achieved by discarding certain number of high-frequency Fourier coefficients. This scheme has been transplanted to mesh compression, using the eigenbases of mesh Laplacian, i.e, the manifold harmonic basis (MHB), as the Fourier bases [67]. The key disadvantage of Fourier compression is that the Fourier bases are only localized in the frequency domain yet having global support in the spatial domain, and thus are not efficient in encoding local signal information. A popular and powerful solution is to use wavelet bases, which are functions localized in both vertex domain and frequency domain and can capture local signal information in a more compact and efficient way.

In this chapter, we propose to use the spectral graph wavelets (SGW), pioneered by Hammond et al. [59], for mesh approximation and compression. To the best of our knowledge, we believe that it is the first attempt to exploit the SGW in sparse representation, with a unique application in 3D geometric compression. The SGW has many attractive properties such as spatial localization, being smooth, multi-scale, and shape-aware, and being flexible and versatile for 3D shapes of arbitrary topology and complicated geometry, hence is well suited for encoding shapes with many local details. We employ the SGW as shape bases to construct redundant dictionary with multiscale wavelets centered around each vertex, and employ the simultaneous orthogonal matching pursuit (S-OMP) algorithm to find a sparse coding of the original shape geometry. The primary contributions of this work are hinging upon the unique integration of the spectral graph wavelets (SGW) and sparse representation and its powerful application in 3D shape compression. To the best extent of our knowledge, our current work is the first attempt to employ the SGW in the task of 3D mesh compression.

Through our extensive experiments, we wish to demonstrate that our compression method outperforms the MHB-based Fourier compression in terms of compression quality at different compression ratio settings. Since our sparse shape approximation framework is independent of any data-specific dictionary design, other formulations of bases or dictionaries, as well as other powerful sparse approximation algorithms, can all be migrated into our system with very little extra workload. So we are expecting further computational improvement in compression performance in the near future.

3.1.1 Background

Harmonic analysis techniques such as Fourier transform and wavelet transform are fundamental tools for compact representations of images, audio, and video signals. The most prominent applications include the JPEG [155] and JPEG2000 [141] image compression standards, which are based on 2D discrete cosine transform and discrete wavelet transform, respectively. The key idea of harmonic compression is to decompose the original signal into a set of harmonic basis and reduce the representation size by discarding coefficients that correspond to much less noticeable signal components.

While traditional harmonic analysis oftentimes uses orthogonal basis, such as the Fourier basis, recent years have witnessed the increasing popularity of sparse approximation methods, which enable the utility of redundant or over-complete dictionaries for signal representation. From the dictionary of elementary signals, a small subset that can best capture the structure of the input signal is selected, and the input signal is approximated by a linear combination of the selected signals.

A 3D mesh can be expressed as the connectivity information of the mesh topology plus the 3D mesh coordinates. The mesh connectivity defines the domain of coordinate functions and have several efficient, lossless coding [58, 131]. To compress the mesh coordinates, traditional Fourier and wavelet compression techniques for images can not be directly applied, since 3D meshes generally do not have a fixed regular graph structure. Consequently, there is no universally feasible Fourier basis and the dictionary should be derived from specific object's graph topology. In [67], Karni and Gotsman employed the mesh Laplacian eigenbases to encode the mesh geometry, and the compression is achieved by discarding high-frequency coefficients. Later, Karni and Gotsman extended the spectral compression method by using fixed eigenbases derived from a 6-regular mesh to approximate the eigenbases of the non-regular input meshes, avoiding the cost of Laplacian decomposition on the decoder side [68].

3.2 Spectral Mesh Compression

Consider a 3D mesh $M = (V, E)$ with vertices V and edges E, where $V = \{v_1, v_2, \ldots, v_n\}$. A vector-valued function $\mathbf{f} : V \to \mathbb{R}^c$ defined on V can be represented as an $n \times c$ matrix, where the ith row represents the function value at

v_i.

As we mentioned in Chapter 2, the Laplacian matrix of M admits an orthogonal and complete basis of $L^2(M)$, i.e., the space of square-integrable functions defined on M, and enables Fourier-like harmonic analysis on the vertex domain.

There are many different formations of discrete Laplacian. A combinatorial Laplacian matrix is determined solely by the connectivity of the mesh. A geometric Laplacian, on the other hand, takes into account both the topological and geometric information.

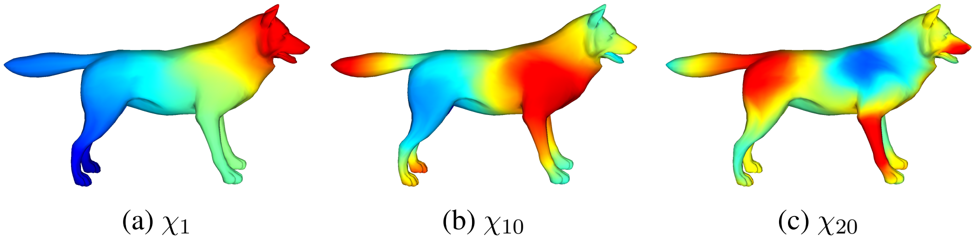
Although a geometric Laplacian affords much more precise description of the mesh geometry, it is not a feasible choice in mesh compression applications because the geometric information is unknown on the decoder side. On the other hand, a combinatorial Laplacian can be easily reconstructed in the decoder size since the mesh connectivity can be efficiently encoded and transmitted independent of the geometric coordinates. In this work, we use the graph Laplacian defined as

$$L_{ij} = \begin{cases} 1 & \text{if } j \in N(i), \\ -d(i) & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{3.2.1}$$

where $d(i)$ represents the valence of v_i.

The graph Laplacian admits an eigensystem $\{\lambda_k, \chi_k\}_{k=0}^{n-1}$, where $\{\lambda_k, \chi_k\}$ denotes the kth eigenvalue and eigenfunction. According to the spectral theorem, the eigenfunctions $\{\chi_k\}$ form an complete and orthonormal basis, called the *Laplacian eigenbasis*, or, in the context of shape analysis, the manifold harmonic basis (MHB). Fig. 3.1 visualizes the MHB on an example mesh. It is straightforward to see that the values of MHB oscillate between negative and positive on the surface, and the larger the associated eigenvalues, the more frequent the oscillation becomes, similar to the behaviors of regular Fourier basis functions in the Euclidean domain.

As mentioned earlier, the MHB can be employed to define the graph Fourier transform, also known as the *manifold harmonic transform* (MHT), which con-



(a) χ_1 (b) χ_{10} (c) χ_{20}

Figure 3.1: Visualization of the Laplacian eigenfunctions (MHB). From left to right, the first, tenth, and twentieth eigenfunctions are highlighted.

verts a function between spatial domain and frequency domain. Any $f \in L^2(M)$ can be expanded by MHB as

$$f = \sum_{k=0}^{n-1} \widehat{f}_k \phi_k = \sum_{k=0}^{n-1} \langle f, \phi_k \rangle \phi_k, \tag{3.2.2}$$

in which $\widehat{f}_k$ is the k-th MHT coefficient of f.

The MHT is the theoretical foundation of the spectral mesh compression proposed by Karni and Gotsman [67]. Viewing the Euclidean mesh coordinates $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ as functions defined on vertices, the basic idea of spectral compression is to compute the MHT of the coordinate function and then truncate out certain number of high-frequency coefficients. Take the x-coordinate function $\mathbf{x}$ as an example. The original coordinates can be perfectly recovered as in Eq. (3.2.2) if all n MHT coefficients $\{\widehat{x}_0, \ldots, \widehat{x}_{n-1}\}$ are used. If we only retain the first $n' < n$ coefficients, the reconstructed x-coordinate function $\mathbf{x}' = \sum_{k=0}^{n'-1} \widehat{x}_k \phi_k$ is a low-pass-filtered version of $\mathbf{x}$ and can be regarded as an acceptable approximation. The reconstructed mesh is smooth and the overall appearance tends to be very similar to the original mesh, since low-frequency components, which correspond to large-scale shape information, are prioritized to be preserved, and human's visual system tends to be more forgiving to the loss of high-frequency information.

Fig. 3.2 shows an example of spectral mesh approximation with different number of MHB.

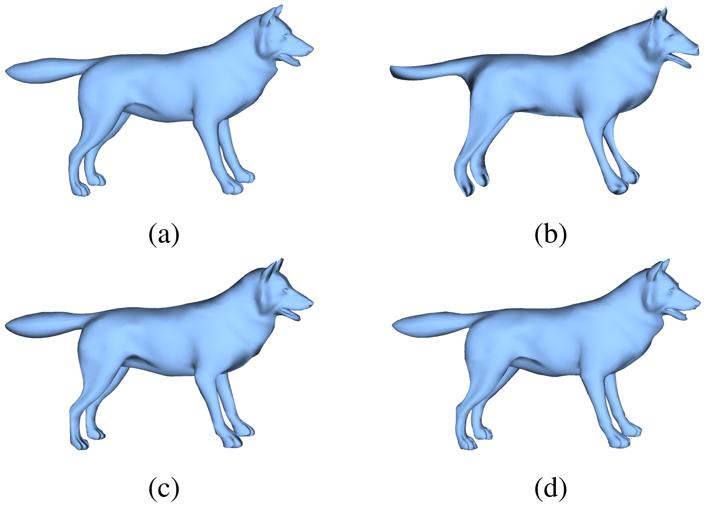


Figure 3.2: Spectral approximation [67] of a 3D wolf model containing 4,344 vertices. (a) The original mesh. (b) Reconstruction using 100 eigenbases. (c) Reconstruction using 300 eigenbases. (d) Reconstruction using 1000 eigenbases.

3.3 Sparse Shape Approximation via SGW

In Sec. 3.2 we have shown that the mesh coordinates can be transformed into the frequency domain via MHT using the Laplacian eigenbasis, and compression can be achieved by trimming out a user-specified number of high-frequency coefficients. The main drawbacks of this naive and simple low-pass spectral compression method are: (1) It innately favors low-frequency information while most high-frequency geometric details are compromised; (2) The Laplacian eigenbasis, which serve as the compression dictionary, all have global support and therefore are not efficient in encoding local geometry.

In this section, we propose to use SGW to construct a redundant dictionary. Because of its powerful property of spatial localization, the multiscale SGW functions are much more efficient in representing local mesh geometry around individual vertices than the Laplacian eigenbasis. Since the size of SGW dictionary is much larger than the number of mesh vertices, we employ powerful sparse approximation algorithms to find a compact representation which selects the most appropriate basis in the procedure.

3.3.1 Sparse Approximation of Mesh Coordinates

For mesh M, let $\mathbf{D}$ be a dictionary of $L^2(M)$ containing m normalized basis vectors. The dictionary can be written as a $n \times m$ matrix $\mathbf{D} = \begin{pmatrix} \mathbf{a_1} & \mathbf{a_2} & \dots & \mathbf{a_m} \end{pmatrix}$, where $\mathbf{a_i} \in \mathbb{R}^{n\times 1}$. Our aim is to approximate function $\mathbf{y} \in L_2(M)$ with a linear combination of the atoms in $\mathbf{D}$, expressed in the matrix form as $\widehat{\mathbf{y}} = \mathbf{D}\mathbf{x} = \sum_{i=1}^{m} x_i \mathbf{a_i}$. Here the vector $\mathbf{x} \in \mathbb{R}^m$ is the coefficient representation of the input signal $\mathbf{y}$ w.r.t. the dictionary $\mathbf{D}$.

An effective compression of the original signal $\mathbf{y}$ requires the number of elementary signals that participate in the linear combination to be small and the reconstructed result $\widehat{\mathbf{y}}$ to be as close to $\mathbf{y}$ as possible. In principle, the number of non-zero elements of the coefficient vector $\mathbf{x}$, denoted by the pseudo-norm $\|\mathbf{x}\|_0$, should satisfy $\|\mathbf{x}\|_0 \ll n$ in order to achieve significant reduction in storage. Fixing the number of participating atoms in the sparse approximation to be n', the problem to produce the optimal sparse representation $\mathbf{x}$ can be formulated as

$$\min_{\mathbf{x}} \|\mathbf{y} - D\mathbf{x}\|_2^2 \text{ subject to } \|\mathbf{x}\|_0 = n' \qquad (3.3.1)$$

which is called the *Best Basis Selection* problem

If the input signal has c channels, denoted as an $n\times c$ matrix $\mathbf{Y} = (\mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_c})$, the coefficient representation should be a $m \times c$ matrix $\mathbf{X} = (\mathbf{x_1}, \dots, \mathbf{x_c})$ satisfying $\mathbf{Y} \approx \mathbf{DX}$. We may either treat each channel (column) of $\mathbf{Y}$ independently and select different subsets of participating atoms for each channel, or enforce to select the same subset for all the channels and minimize the combined approximation errors. The latter one is called the *simultaneous sparse approximation* problem

$$\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F^2$$

subject to

$$\begin{cases} support(\mathbf{x_1}) = \cdots = support(\mathbf{x_c}) \\ \|\mathbf{x_1}\|_0 = \cdots = \|\mathbf{x_c}\|_0 = n', \end{cases} \tag{3.3.2}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $support(\mathbf{x_i})$ denotes the index set of non-zero elements in $\mathbf{x_i}$.

The vertex coordinates of a mesh can be treated as a 3-channel signal $p(\mathbf{v}) = (\mathbf{v_x}, \mathbf{v_y}, \mathbf{v_z})$. Since the three coordinate functions are correlated, it is preferable to formulate the mesh compression as the simultaneous sparse approximation problem. Determining the optimal solution to Eq. (3.3.2) is NP-hard with the time complexity of $O(nm^{n'}n'^2$, but we can find approximate solutions using greedy pursuit algorithms such as *simultaneous orthogonal matching pursuit* (S-OMP) [151]. The key idea is to iteratively select from the dictionary a new atom that has the best correlation with the residual shape, and then project the original mesh onto the space spanned by the selected atoms to obtain a new approximate shape. Please refer to Algorithm 1 for details.

We may also adopt the *simultaneous matching pursuit* (S-MP) algorithm which can be viewed as a simplification of S-OMP. The main difference from S-OMP is the omission of the step to update all existing coefficients by orthogonal projection. If all atoms in $\mathbf{D}$ are mutually orthogonal (e.g., Fourier dictionary), S-MP and S-OMP will produce exactly the same result.

3.3.2 SGW and Dictionary Design Strategies

The key to effective sparse approximation is the selection of elementary functions that form the dictionary. A natural choice is the MHB dictionary composed entirely of Laplacian eigenbasis $\{\chi_0, \chi_2, \ldots, \chi_{n-1}\}$. The MHB functions have global support and multiple frequencies, making the MHB dictionary a good choice for encoding a shape when global, periodical, and symmetric information is prioritized to be preserved. In addition, since MHB are orthogonal basis, we can replace orthogonal matching pursuit (OMP) with the much faster matching

Algorithm 1 S-OMP on 3D mesh coordinates.

```
Input:
    • 3D mesh coordinates $\mathbf{S} \in \mathbb{R}^{n \times 3}$.
    • Dictionary $\mathbf{D} = \{\mathbf{a_1}, \mathbf{a_2}, \dots, \mathbf{a_m}\}, \mathbf{a_i} \in \mathbb{R}^3$.
    • The number of atoms to be selected $n'$.
Initialization:
    • The initial index set of selected atoms $\Lambda_0 = \emptyset$.
    • The initial residual $\mathbf{R_0} = \mathbf{S}$;
    • The iteration counter $t = 1$.
Procedure:
(1) Find an index $i_t$ of $\mathbf{D}$ that satisfies
        $i_t = \arg\max_{j \notin \Lambda_{t-1}} \sum_{k=1}^{3} |\langle \mathbf{R_{t-1} e_k}, \mathbf{a_j} \rangle|,$
    where $e_k$ denotes the $k$th canonical basis vector in $\mathbb{R}^3$.
(2) Set $\Lambda_t = \Lambda_{t-1} \bigcup \{i_t\}$.
(3) Compute the coefficient matrix $\mathbf{C_t}$ by solving the least-square problem
        $\mathbf{C_t} = \arg\min_{\mathbf{X}} = \|\mathbf{S} - \mathbf{DX}\|_2^2$
    subject to $support(X) = \Lambda_t$.
(4) Calculate the new approximation and residual:
        $\widehat{\mathbf{S}}_\mathbf{t} = \mathbf{DC_t},$
        $\mathbf{R_t} = \mathbf{S} - \widehat{\mathbf{S}}_\mathbf{t}.$
(5) Stop if $t = n'$. Otherwise, increment $t$ and go to (1).
Output:
    • The index set of selected atoms $\Lambda_{n'}$.
    • Final coefficient matrix $\mathbf{C_{n'}}$.
```

pursuit (MP) and the approximation results will be the same.

However, MHB dictionary is very inefficient in capturing non-periodical, local details due to the global support. It is desirable to have a dictionary with a class of functions that have local support but are still smooth and multiscale. In this work, we propose to use normalized multiscale spectral graph wavelets (SGW) as atoms to construct the dictionary for sparse approximation.

As we mentioned in Chapter 2, the spectral graph wavelets (SGW) are expressed as bivariate kernel functions expanded on the Laplacian eigenbasis

$$\boldsymbol{\Psi}_{\mathbf{t}}(i,j) = \sum_{k=0}^{n-1} g(t\lambda_k)\chi_k(i)\chi_k(j), \tag{3.3.3}$$

where g is the real-valued wavelet generator function and t is the scale parameter. The ith row of Ψ_t

$$\psi_{t,i}(\cdot) = \sum_{k=0}^{n-1} g(t\lambda_k)\chi_k(i)\chi_k(\cdot) \tag{3.3.4}$$

is the spectral wavelet localized at v_i in the vertex domain and at scale t in the frequency domain.

The spectral scaling functions is defined with the scaling generator function h which satisfy

$$\begin{cases} h(0) > 0 \\ h(x) \to 0|_{x\to\infty}. \end{cases} \tag{3.3.5}$$

In practice, the scale parameter t also need to be discretized. The spectral graph wavelets $\psi_{t,i}$ are near orthogonal to χ_k for λ_k near 0, i.e., low-frequency eigenbasis, for any discrete scale t. Hence, to better capture low-frequency signal information, [59] also introduced the *spectral scaling functions* which have similar constructions with SGW but act like low-pass filters

$$\boldsymbol{\Phi}_t(i,j) = \sum_{k=0}^{n-1} h(t\lambda_k)\chi_k(i)\chi_k(j), \tag{3.3.6}$$

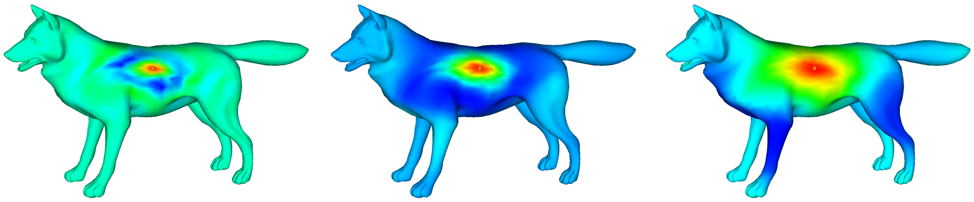


Figure 3.3: Visualizations of spectral graph wavelets localized at the same point but with different scales. From left to right, the spectral wavelets at scale 1, 3 and 5 are plotted.

in which

Suppose we compute the spectral wavelets at J different scales $\{t_1, t_2, \dots, t_J\}$, the constructed SGW then comprises $(J+1) \times n$ functions in $\mathbb{R}^n$. In this chapter, we adopt the same formulation of wavelet and scaling functions used in [59] with the generator function

$$g(x) = \begin{cases} x^2 & \text{if } x < 1 \\ -5 + 11x - 6x^2 + x^3 & \text{if } 1 \leq x \leq 2 \\ 4x^{-2} & \text{if } x > 2. \end{cases} \tag{3.3.7}$$

The J scales are selected to be logarithmically equally spaced between the minimum scale $t_J = 2/\lambda_{max}$ and the maximum scale $t_1 = 40/\lambda_{max}$, where λ_{max} is the upper bound of the Laplacian eigenvalues. For the scaling function, the generator is $h(x) = \gamma \exp(-(\frac{20x}{0.6\lambda_{max}})^4)$, in which $\gamma = h(0)$ equals the maximum value of g.

Fig. 3.3 visualizes multiscale spectral graph wavelets on a 3D mesh. It may be noted that, the values of wavelets are attenuated and oscillating on the mesh, and wavelets with a larger scale have a wider oscillating window.

The SGW dictionary has several advantages:

- The SGW atoms are compact and localized at vertices, suitable for encoding local geometric features.

- The SGW atoms can cover multiple scales, enabling the efficient representation of both small-scale and large-scale shape information in the vincinity of each vertex.

- The computation of SGW from MHB is straightforward, and can be done on the decoder side provided the mesh connectivity is known.

On the flip side, the SGW are less efficient than MHB for encoding global shape structures. Moreover, since SGW functions always have extreme values at their origin vertices, a mesh reconstructed from SGW atoms may exhibit unpleasant protrudes at vertices where selected SGW are centered, which can be further ameliorated by constructing a dictionary that contains both MHB and SGW. The mixed dictionary potentially inherits the advantages of both waveforms, at the cost of increased dictionary size.

The SGW or SGW+MHB dictionary are non-orthogonal and redundant, forcing us to use costly sparse approximation methods such as S-OMP. The enlarged dictionary also increase the storage requirements for the dictionary themselves and for the sparse coefficient representation (see Sec. 3.3.3).

Fig. 3.4 shows an example of sparse approximation results using three methods: (1) Spectral mesh compression via MHB, as described in Sec. 3.2; (2) S-MP with the MHB dictionary; (3) S-OMP with the SGW dictionary. In this example, the S-MP method produces higher-quality shape approximation than the naive low-pass spectral approximation using the same MHB dictionary. Adopting the multiscale SGW dictionary in place of MHB further improves the approximation results. In particular, the SGW dictionary are more effective in preserving local geometric features, while MHB-based approximations have the tendency to smooth out some body parts such as the wolf's legs when the number of participating bases is small.

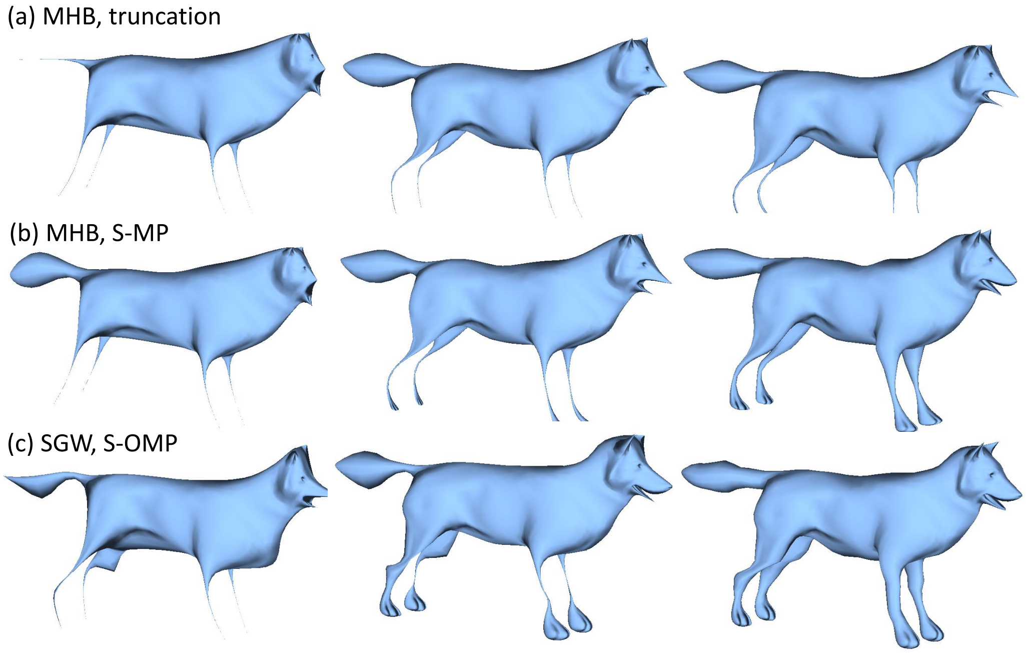


Figure 3.4: Comparison of the approximation results of three different approximation methods. *Top row*: Spectral compression by truncating MHB coefficients. *Second row*: S-MP approximation with MHB bases. *Bottom row*: S-OMP approximation with SGW bases. For each method, from left to right, the number of participating bases are 20, 50, and 100, respectively.

3.3.3 Compression Ratio and Analysis

Now we analyze the compression ratio of the simultaneous sparse representation using a simple coding scheme. Assume the dictionary D are known in advance on both the encoder and decoder sides. The sparse $m \times 3$ coefficient matrix $\mathbf{X}$ contains n' non-zero rows, and can be conveniently expressed by $3n'$ non-zero values and a vector of size n' specifying the indices of corresponding atoms. For a dictionary containing m atoms, each index occupies $\lceil \log_2 m \rceil$ bits, and the total cost to store the index vector is $n' \lceil \log_2 m \rceil$. If the sparsity of $\mathbf{X}$, namely the ratio of non-zero elements, is greater than $1/\lceil \log_2 m \rceil$, it would be more efficient to represent the non-zero positions with a bit-vector of size m. Assuming each signal element in $\mathbf{Y}$ takes up k bits, and each coefficient in $\mathbf{X}$ requires k' bits to store, the storage size of the original 3D coordinates is then $3nk$ bits. The effective compression ratio is

$$\Gamma = \frac{3n'k' + \min(m, n' \lceil \log_2 m \rceil)}{3nk}. \tag{3.3.8}$$

Assume that the dictionary contains $m = \alpha n$ atoms, and both the coordinates and coefficients are stored in single-precision $k = k' = 32$, the compression ratio is then

$$\Gamma = \frac{n'}{n} + \min(\frac{\alpha}{96}, \frac{n' \lceil \log_2 \alpha n \rceil}{96n}). \tag{3.3.9}$$

In comparison, the compression ratio of the coefficient truncation method introduced in Sec. 3.2 is simply n'/n with n' coefficients, since there is no need to store the indices of non-zeros.

From Eq. (3.3.9) we can easily see that enlarging the dictionary (larger α) increases the overhead ratio for a given mesh. In addition, when the coefficient matrix is very sparse, the overhead ratio becomes smaller for larger meshes, since $log_2 \alpha n$ increases slower than n.

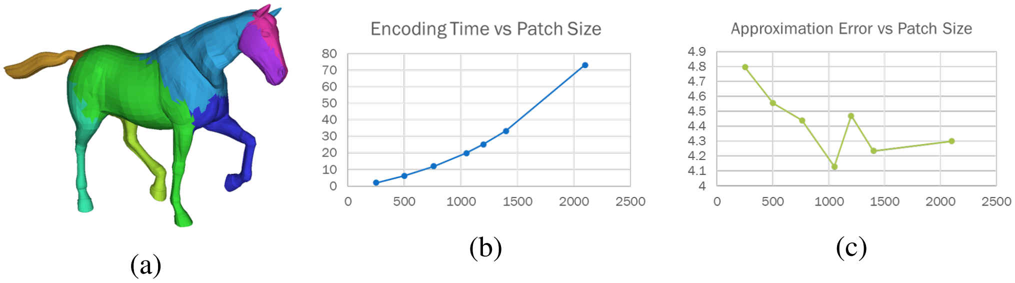
3.3.4 Mesh Partitioning

The most time-consuming part of S-OMP is the computation of the maximum inner product between the residual and available atoms, which costs $O(mn)$ in each iteration. If the required number of iterations, i.e., the sparsity of the solution is l, the total time complexity of S-OMP will be $O(lmn)$. Generally, the required number of iterations l and the size of dictionary m are linearly proportional to the mesh size n, hence the total time complexity is $O(n^3)$, which is unacceptable for very large meshes.

In addition, all the dictionaries we use are constructed from the eigenvectors of mesh Laplacian, but the full Laplacian eigendecomposition of a large mesh is very time consuming and can be numerically instable. Hence, when the input mesh is very large, it is necessary to perform graph partitioning and perform the compression algorithm on each individual sub-mesh. As suggested in [67], we use the METIS package [69] for fast graph partitioning. If the average size of each submesh is v, the time complexity of S-OMP will decrease to $O(nv^2)$.

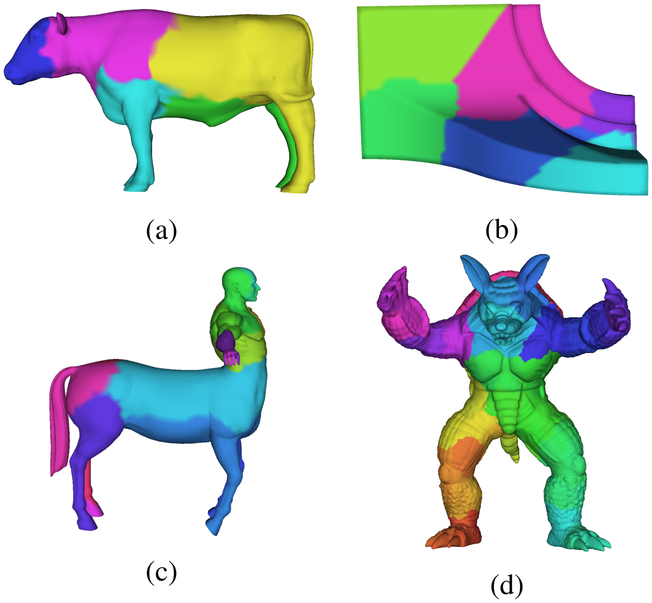
As mentioned in Sec. 3.3.3, the overhead ratio of storing the indices of selected atoms is smaller when the mesh size becomes larger. Moreover, increasing the number of sub-meshes also increases the occurrences of unpleasant artifacts along sub-mesh boundaries, known as the edge effects. On the flip side, partitioning a large mesh to smaller ones may result in better local fitting.

Taking into account these factors, it is not immediately clear what partition size should be optimal. In our implementation, a large mesh is decomposed into patches containing approximately equal number of vertices, with the maximum patch size empirically set to be $1,000$, which seems to strike a good balance between time cost and approximation error. Fig. 3.5 shows an example of how the approximation error and encoding time change with the partition size.



(a) (b) (c)

Figure 3.5: Relationships between approximation performance and partition size. (a) The *Horse* model partitioned into patches of about $1,000$ vertices. (b) Time performance *vs* patch size. (c) Approximation errors *vs* patch size.



(a) (b)

(c) (d)

Figure 3.6: Models used in our approximation experiments and their partitionings. (a) *Cow*: 4,315 vertices. (b) *Fandisk*: 6,475 vertices. (c) *Centaur*: 15,768 vertices. (d) *Armadillo*: 172,974 vertices.

3.4 Experimental Results

3.4.1 Evaluation Method

To evaluate the effectiveness of lossy mesh compression methods, we adopt the mesh comparison metric proposed in [67] to measure the errors between the original mesh geometry and approximate ones. Let M_1 and M_2 be two meshes to be compared, both containing n vertices, and $\mathbf{v_i^1}$ and $\mathbf{v_i^2}$ denote the 3D coordinates of the i-th vertex in M_1 and M_2, respectively. The *geometric error* between M_1 and M_2 is

$$\|M_1 - M_2\|_g = \sum_{i=1}^{n} \frac{1}{n} \|\mathbf{v_i^1} - \mathbf{v_i^2}\|_2. \tag{3.4.1}$$

To better capture visual closeness such as smoothness, [67] introduces another metric which measures the errors after applying the geometric Laplacian to mesh coordinates, i.e., transforming the absolute coordinates to differential coordinates

$$GL(\mathbf{v_i}) = \mathbf{v_i} - \frac{\sum_{j \in N(i)} l_{ij}^{-1} \mathbf{v_j}}{\sum_{j \in N(i)} l_{ij}^{-1}}, \tag{3.4.2}$$

where l_{ij} represents the edge length between v_i and v_j. The *differential error* between M_1 and M_2 is then defined as

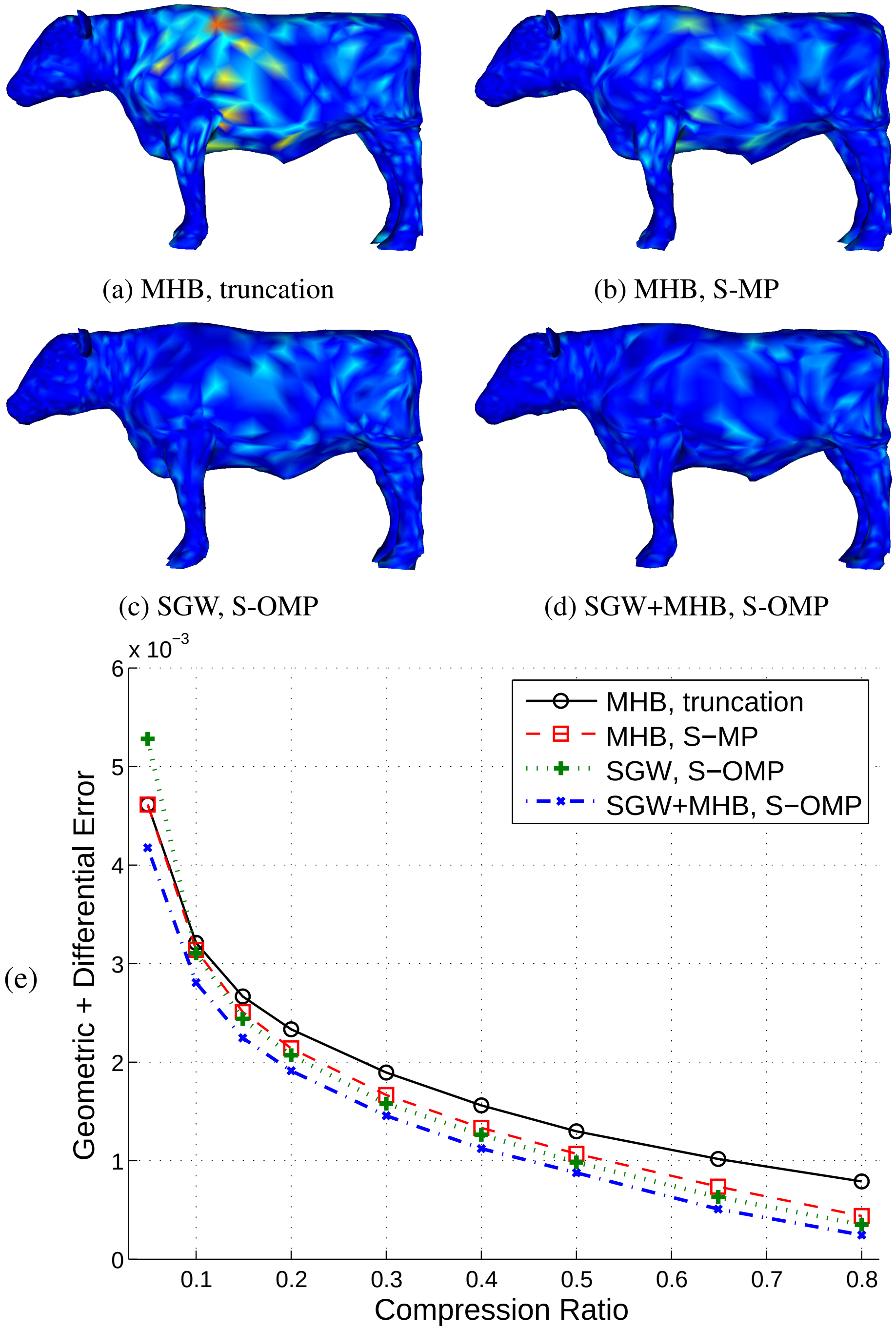
$$\|M_1 - M_2\|_d = \sum_{i=1}^{n} \frac{1}{n} \|GL(\mathbf{v_i^1}) - GL(\mathbf{v_i^2})\|_2. \tag{3.4.3}$$

The final error metric is the average of geometric error and differential error

$$\|M_1 - M_2\| = \frac{1}{2}(\|M_1 - M_2\|_g + \|M_1 - M_2\|_d). \tag{3.4.4}$$

3.4.2 Compression Performance

In all our tests, we compare the compression performance between the classical spectral compression method based on MHB coefficient truncation [67] and our



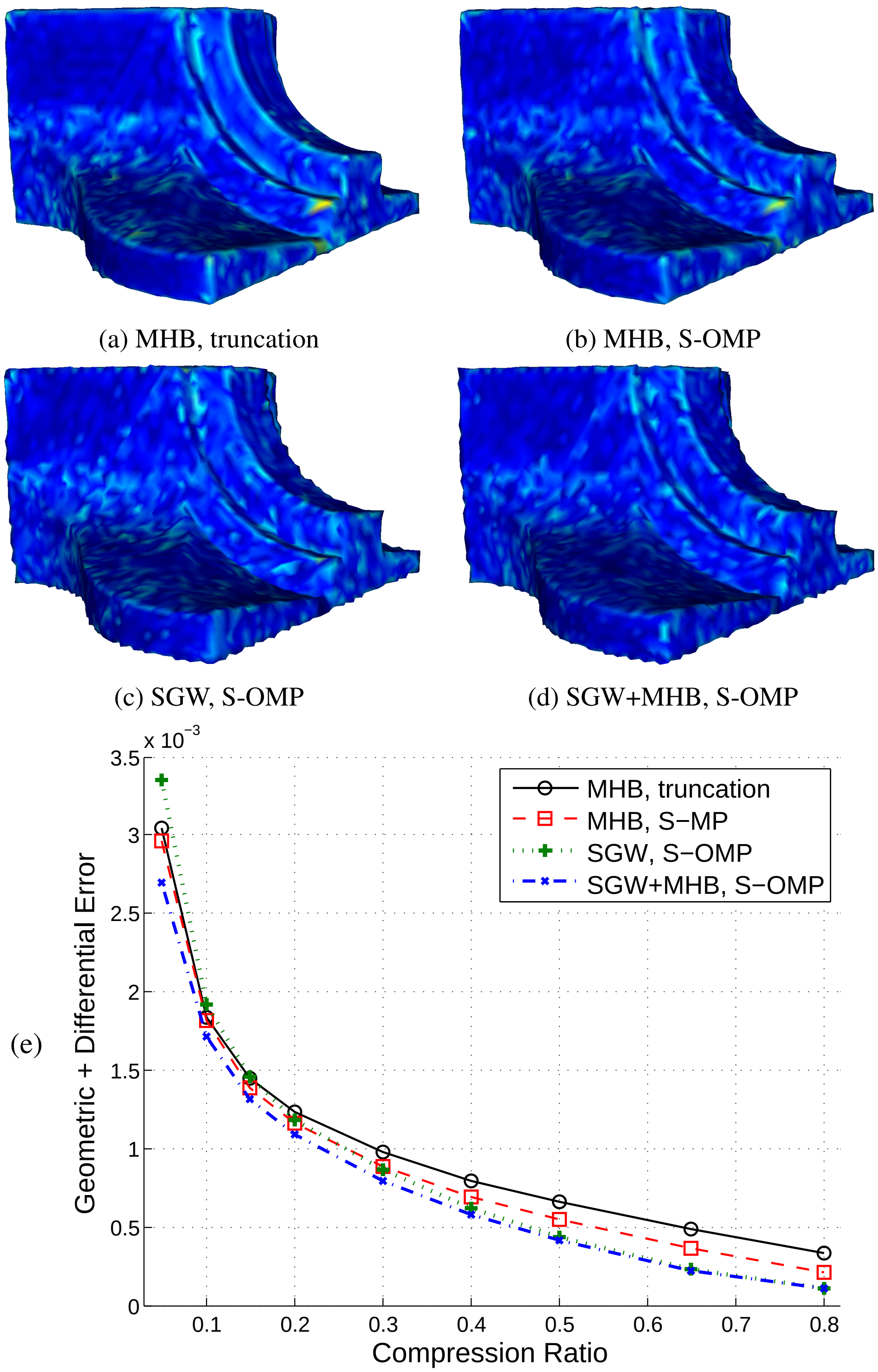
(a) MHB, truncation

(b) MHB, S-MP

(c) SGW, S-OMP

(d) SGW+MHB, S-OMP

Figure 3.7: Comparison of mesh compression performance for the cow model. (a-d) show the reconstructed meshes at 20% compression ratio and visualize each vertex's positional error comparing with the original model. (e) shows how the compression errors change with the target compression ratios.



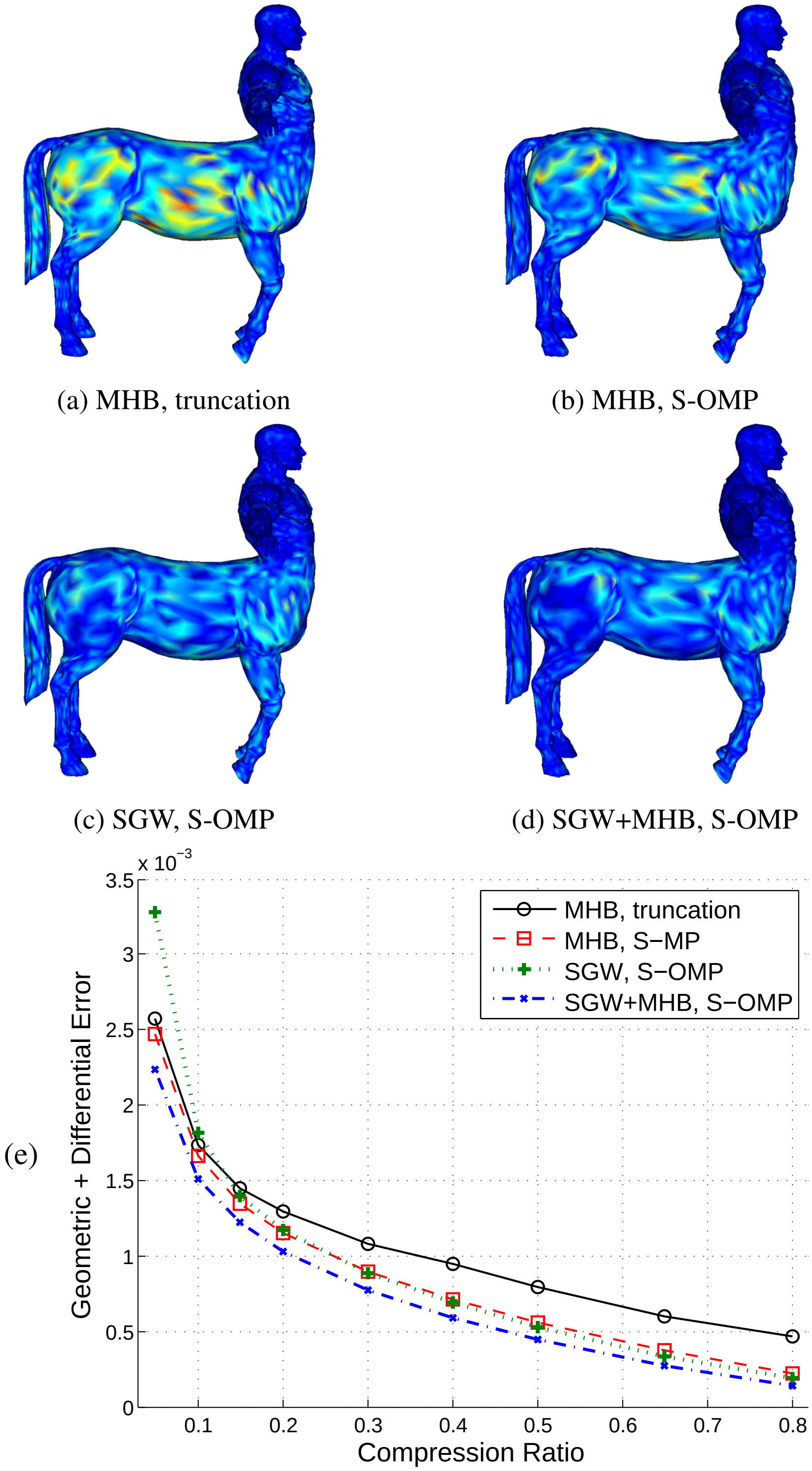
(a) MHB, truncation

(b) MHB, S-OMP

(c) SGW, S-OMP

(d) SGW+MHB, S-OMP

Figure 3.8: Comparison of mesh compression performance for the fandisk model. (a-d) show the reconstructed meshes at 20% compression ratio and visualize each vertex's positional error comparing with the original model. (e) shows how the compression errors change with the target compression ratios.


(a) MHB, truncation

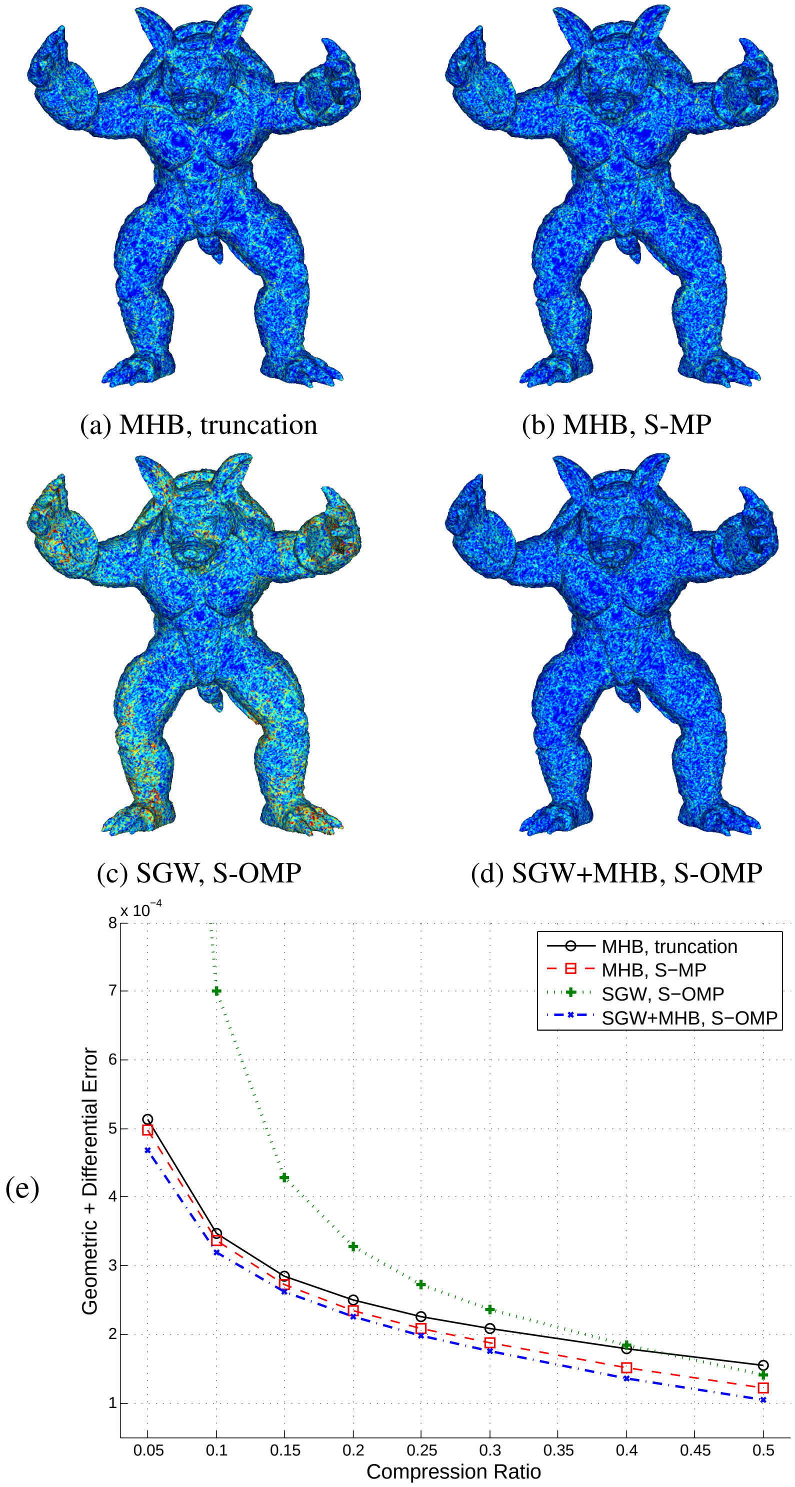
(b) MHB, S-OMP

(c) SGW, S-OMP

(d) SGW+MHB, S-OMP

(e)

Figure 3.9: Comparison of mesh compression performance for the centaur model. (a-d) show the reconstructed meshes at 20% compression ratio and visualize each vertex's positional error comparing with the original model. (e) shows how the compression errors change with the target compression ratios.



(a) MHB, truncation

(b) MHB, S-MP

(c) SGW, S-OMP

(d) SGW+MHB, S-OMP

(e)

Figure 3.10: Comparison of mesh compression performance for the armadillo model. (a-d) show the reconstructed meshes at 15% compression ratio and visualize each vertex's positional error. Comparing (c) and (d), it is obvious that the SGW+MHB dictionary produces much smaller positional error than the SGW-only dictionary. (e) shows how the compression errors change with the target compression ratios.

sparse approximation compression method employing three different dictionaries: (1) S-MP with the MHB-only dictionary; (2) S-OMP with the SGW-only dictionary; (3) S-OMP with the SGW+MHB dictionary.

Model (#vertices)	Ratio	Error	Error Reduction	Timing (s)
Cow (4,315)	20%	1.91e-3	18.0%	3.4
	40%	1.12e-3	27.5%	10.5
Fandisk (6,475)	20%	1.09e-3	11.6%	7.2
	40%	5.74e-4	28.1%	18.9
Centaur (15,768)	20%	1.03e-3	20.3%	19.6
	40%	5.89e-3	37.7%	50.3
Armadillo (172,974)	20%	2.26e-3	9.4%	178.0
	40%	1.76e-3	15.7%	354.5

Table 3.1: Statistics of compression errors and running times using the S-OMP algorithm with SGW+MHB dictionary (on a machine with quad-core 2.4GHz processor and 16G RAM). The "Error Reduction" column denotes how much the compression error is reduced comparing with the MHB truncation method.

Fig. 3.6 shows the original meshes and their partitioning used in our experiments, including a "cow", a "fandisk", a "centaur", and a "armadillo" model. All meshes are scaled to have unit surface area. The evaluation results are shown in Fig. 3.7, Fig. 3.8, Fig. 3.9, and Fig. 3.10, respectively. For each 3D mesh, we compute the approximation at specified compression ratios in the ranges between 5% and 80%. The overall compression quality is measured by the combined geometric and differential error (see Eq. (3.4.4)) w.r.t. the original mesh. Table 3.1 documents the compression errors and timing of S-OMP with SGW+MHB dictionaries, and compares the errors with the MHB coefficient truncation method.

From the experimental results, we see that, with a properly chosen dictionary, simultaneous sparse approximation can generate higher-fidelity mesh compression than the MHB truncation method at the same compression ratio. The SGW functions are a viable choice for efficient mesh approximation, but the performance of a SGW-only dictionary degenerates significantly when the required compression ratio is small or the mesh is large, which is especially evident in the armadillo model (Fig. 3.10). A dictionary combining SGW and MHB overcomes this deficiency, and its performance in mesh approximation is consistently superior to the MHB truncation method.

3.5 Chapter Summary

In this chapter, we have developed an algorithm for sparse approximation of 3D shapes. We employed the spectral graph wavelets to construct the redundant dictionary of shape bases, and used simultaneous orthogonal matching pursuit to seek a sparse representation of the input mesh. The use of spatially-localized wavelets makes our algorithm very suitable and powerful for better approximating shapes with many local and fine geometric features. Through comprehensive experiments we have demonstrated the superiority of our algorithm for approximating complex 3D objects at different compression ratio settings towards sparse representation.

As for the future work, we plan to investigate other improved formulations of harmonic basis to enhance the expressive power of dictionary. For example, it is potentially desirable to have data-dependent, anisotropic wavelets that are adaptive to shape features such as sharp corners and edges to attain more efficient and sparse representation of shape geometry. We also plan to explore faster sparse approximation algorithms such as stagewise orthogonal matching pursuit (StOMP) [37] to arrive at better time performance.

Chapter 4

Surface Inpainting

4.1 Introduction

In principle, surface inpainting refers to the completion or recovery of missing shape geometry based on the shape information that is currently available. The most prominent application of surface inpainting is mesh repair. Due to factors such as occlusions, low reflectance, and quality limitations of scanning equipments, 3D models generated from range scanners often contain holes that need to be filled; sometimes the source model itself has missing pieces and requires digital repair to attain a complete model. Another common application of surface inpainting is to remove geometric features because of shape editing needs. This is achieved by replacing the unwanted shape regions with inpainting patches.

From the statistical point of view, surface inpainting can be viewed as an estimation problem which infers the missing geometry from the observable shape, and the inpainting result is determined by the statistical model we have adopted. Generally speaking, there is no universally acceptable "correct" estimation; selecting the best inpainting is usually subjective or dependent on the requirement of downstream applications.

Most existing surface inpainting methods tackle the problem only in the mesh domain. These methods typically employ some geometric constraints as heuristics to ensure that the obtained inpainting patch is visually pleasing and blend naturally with its neighboring geometry. The primary issue of geometry-constrained inpainting is that these methods only utilize the shape information in the vicinity of missing regions rather than consider the model in question in its entirety.

Our new surface inpainting method documented is inspired by the theory of sparse signal recovery and compressed sensing. The intuition is that for a meaningful 3D model, even though its global geometry is a high-dimensional signal, it most likely has a low-dimensional intrinsic structure. That is to say, the high-dimensional shape signal actually lives in a low-dimensional subspace, which can be captured by a sparse coefficient representation in some transformed (e.g., Fourier) domains. In another word, the coordinate function of a shape with N vertices can be decomposed as or well approximated by the linear combination of $k \ll N$ basis signals. According to the compressed sensing theory, the sparse coefficient representation can be recovered from partial measurements as long as the signal is sufficiently sparse and the sensing matrix satisfies certain properties [21].

The critical idea of our new inpainting algorithm is to estimate the spectral coefficient representation of the shape geometry from partial observations by imposing sparsity constraints on the reconstructed coefficients, exploiting the fact that most 3D models are highly compressible with respect to their Laplacian eigenfunctions [67]. To the best of our knowledge, the utility of Laplacian eigenbasis towards the shape inpainting application has not yet been explored in the past. The estimation problem can be formulated with a data term strongly emphasizing fidelity to the observations and a penalty term constraining sparsity of the representation. Thus, the surface inpainting could be transformed to a sparse signal recovery problem and can be solved by either l_0 or l_1 optimization techniques. Such effort represents our first attempt towards technical innovation.

The primary advantage of our method is that the inpainting takes into account the information of all the remaining shape instead of only the vicinity of missing regions. Rather than directly estimating the missing geometry, we actually estimate the reconstruction coefficients of the whole original shape. Since the mesh

Laplacian basis functions are smooth and have global support, the reconstructed inpainting shape is naturally smooth and globally coherent with a simple intrinsic structure.

The main contributions of this work are:

- We introduce a new surface inpainting framework based on representations in the transformed domain and sparsity constraints on reconstruction coefficients. This framework can make use of the information of the whole remaining shape and inpainting results tend to be simple and globally coherent.

- We study the sparsity of 3D shapes with respect to their Laplacian eigenbases and show their effectiveness in surface inpainting.

- We demonstrate the high performance of our inpainting method with several examples in hole filling and mesh editing.

4.2 Related Work

4.2.1 Surface Inpainting

Many methods have been proposed in the research literature dealing with the general problem of surface inpainting, bearing different names such as hole filling, mesh completion, and surface restoration. We refer readers to [3] for a recent survey of popular algorithms for hole filling and mesh completion.

One simple approach for surface inpainting is by filling the missing region with an inpainting patch that interpolates the surrounding geometry. The interpolating patch may be generated with simple polynomial functions [156], triangular B-splines [121], or radial basis functions [13], and are generally smooth and continuous across the boundaries. The interpolation-based approaches, however, only work well with disk-like holes and are not suitable for filling regions with complex boundaries.

Typical mesh-based hole filling algorithms have two steps: (1) Find an initial triangulation of the missing region defined by the hole boundary; (2) Optimize the inserted mesh to improve its fairness and coherence with surrounding shapes. In [87], Liepa performed hole triangulation with a dynamic programming technique taking into account the dihedral angles and areas of the created triangles. The inserted mesh is then optimized with Laplacian smoothing to improve fairness. In [165], surface holes are patched by an advancing-front mesh generation method, and the vertex positions are optimized by solving a Poisson equation based on the desirable triangle normals computed from boundary vertices. In [4], the coordinates of the inserted vertices are optimized by minimizing the discrete thin-plate energy. In [86], [157] and [107], complex holes are first partitioned into sub-holes by feature curves extended from the existing parts; typical hole-filling then can be performed on these sub-holes which are much more planar.

Another class of inpainting algorithms are based on variational methods. The basic idea is to iteratively evolve the inpainting shape by optimizing a functional that constrains certain geometric properties of the inserted mesh, e.g., positions, areas, tangency, and curvatures. In [120], Pernot et al. developed a hole filling algorithm which minimizes the variational involving curvature between the surrounding and inpainted geometry. In [23], the completing surface is chosen such that a power of the mean curvature is minimized. In [27], Clarenz et al. proposed a shape restoration algorithm by computing the l_2-gradient flow of the Willmore energy which ensures the continuity of the normal field.

Finally, a large number of mesh inpainting methods can be classified as exemplar-based or template-based. The basic idea is to find a known patch or a template model that has similar context with the damaged or missing region and then adapt the selected patch for inpainting. Symmetry-guided methods [85, 115, 118, 138] take advantage of intrinsic shape symmetry to repair damaged regions by transplanting similar regions from the object itself; the pitfall is that it only works when the damaged region's corresponding symmetric counterpart exists and is not missing or damaged. Template-guided methods [50, 76, 117, 158, 164] conduct mesh completion by matching the incomplete region with a template model which may be deformed to adapt to the context. The main challenges are: (1) Suitable template model may not be available in the template database; (2) Inpainting results can be affected by cross-shape parameterization.

4.2.2 Sparse Signal Recovery and Inpainting

Recent years have witnessed a surge in the research of sparsity-based signal recovery. The fundamental idea is that a sufficiently sparse signal can be reliably reconstructed from partial measurements by exploiting the sparsity cue.

Sparse signal recovery has seen most success in compressed sensing applications, where the measurement/sensing matrix is typically chosen as a normalized random matrix which satisfies the restricted isometry property with high probability. For the inpainting problem, the measurement is expressed as a mask matrix, which is not strictly a valid compressed sensing process. Nonetheless, we can still take advantage of the sparsity constraints to recover the original signals in many situations.

For image inpainting and restoration tasks, many algorithms based on sparse representation have been published. In [56] and [57], Guleryuz proposed an algorithm for image recovery based on adaptive sparse representation. In [40], images are decomposed into texture and cartoon components, each of which is sparse with respect to a particular dictionary; the missing parts then can be easily reconstructed. In [42] and [43], Fadili et al. formulated image inpainting as a maximum-likelihood estimation problem with a sparsity-promoting prior penalty imposed on the reconstructed coefficients. A similar formulation is proposed in [18] where images have sparse framelet representations and the incomplete image can be restored via an iterative shrinkage algorithm. This formulation balances the sparsity of coefficients, fidelity to the existing data, as well as the smoothness of the solution. In [109], Ogawa et al. proposed an image recovery algorithm based on sparse representation, in which the low-dimensional subspaces optimal for targeted missing textures are adaptively selected.

There have been very few works on the sparsity-induced recovery of signals defined on graphs. In [167], Zhu and Rabbat proposed to use the dictionary of graph Laplacian eigenfunctions to recover smooth and sparse graph signals and applied them to the reconstruction of wireless sensor networks data from partial node readings. To the best of our knowledge, our method is the first of such attempts to tackle the problem of geometry inpainting/completion via sparse signal recovery.

4.3 Variational Inpainting Model

The problem of mesh signal inpainting can be stated as follows. Consider a triangle mesh $M = \{V, E\}$ with n vertices, where V and E denote the set of vertices and edges, respectively. Let $\mathbf{f} \in \mathbb{R}^n$ be a vector signal defined on the mesh vertices. Assume the signal values at a subset of vertices $V' \subset V$ are already known, the goal of inpainting is to compute a reasonable estimate of the remaining signal values at $V - V'$. For the problem of inpainting surface geometry, the mesh signal is the coordinate function and the unknown parts correspond to surface holes.

Assume the number of known vertices is $|V'| = n'$. We can define the $n' \times n$ projection matrix P as

$$P(i,j) = \begin{cases} 1 & \text{if } v_j \text{ is the } i\text{th element of } V' \\ 0 & \text{otherwise.} \end{cases} \tag{4.3.1}$$

Denote the observable parts of $\mathbf{f}$ to be $\mathbf{f}' \in \mathbb{R}^{n'}$, which should satisfy $\mathbf{f}' = P\mathbf{f}$, the general inpainting problem can be formulated as a constrained optimization problem

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \Pr(\mathbf{f}) \quad \text{s.t.} \quad \|P\mathbf{f} - \mathbf{f}'\|_2^2 < \epsilon, \tag{4.3.2}$$

or equivalently as a penalized maximum-likelihood estimation problem

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \Pr(\mathbf{f}) + \lambda\|P\mathbf{f} - \mathbf{f}'\|_2^2, \tag{4.3.3}$$

where the data term $\|P\mathbf{f} - \mathbf{f}'\|_2^2$ emphasizes fidelity to the available observations, while $\Pr(\hat{f})$ is a prior regularizing certain properties of the reconstructed signal.

Traditionally, priors are chosen to optimize the fairness of the inserted mesh or its coherence with the surrounding geometry. For example, a commonly-adopted prior for surface optimization is $\Pr(\mathbf{f}) = \|L\mathbf{f}\|_2^2$ which aims to maximize the smoothness of the estimated signal, generating the so-called *least-squares meshes* [144]. Here L is the Laplace operator of the shape.

Instead of computing the approximate signal $\hat{\mathbf{f}}$ in the mesh domain directly, we may first estimate the original signal's representation in some transformed domains. Consider a dictionary D of m atoms, where each atom is an elementary signal defined on the mesh; written in the matrix form, $D = (\mathbf{d_1}, \ldots, \mathbf{d_m}), \mathbf{d_i} \in \mathbb{R}^{n\times 1}$. The original signal f may be represented as the linear combination of columns in D

$$\mathbf{f} = D\alpha = \sum_{i=1}^{m} \alpha_i \mathbf{d_i}, \tag{4.3.4}$$

where $\alpha = (\alpha_1, \ldots, \alpha_m)^T$ is the coefficient representation of $\mathbf{f}$ w.r.t. the dictionary D.

Obviously, if we can estimate the coefficient representation of the whole original signal from partial measurements $\mathbf{f}'$, then we also obtain an inpainting of the missing signal values. If we know in advance that the coefficients of representation of $\mathbf{f}$ satisfy certain statistical properties, we can estimate the coefficients by imposing a prior on α

$$\hat{\alpha} = \arg\min_{\alpha} \Pr(\alpha) \quad \text{s.t.} \quad \|PD\alpha - \mathbf{f}'\|_2^2 < \epsilon, \tag{4.3.5}$$

The complete original signal can then be estimated as $\hat{\mathbf{f}} = D\hat{\alpha}$.

4.4 Sparsity-Based Surface Inpainting

The fundamental idea of our sparsity-based surface inpainting method is that, for most natural shapes, although the surface geometry is a high-dimensional signal, it actually lives in a low-dimensional subspace and has a sparse representation in some transformed domains. Hence, we can set the sparsity of coefficients as the prior in Eq. 4.3.5 to estimate the coefficient representation of the global shape and recover the missing geometry. As long as the "complexity" of the original shape is much smaller than the number of available observations, we have a good chance to obtain a plausible restoration.

In this section, we first discuss the sparsity of shape geometry w.r.t. the mesh Laplacian eigenbasis, demonstrating the potentials of Laplacian eigenfunctions for sparsity-based geometry processing. Then we propose a sparsity-constrained formulation for the problem of surface inpainting with known connectivity. Finally, we extend our inpainting method to hole filling-in where mesh connectivity is nonexistent in the missing regions in the first place.

4.4.1 Laplacian Eigenbasis

For a discrete mesh, its graph Laplacian matrix L is typically defined as

$$L(i,j) = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{4.4.1}$$

The set of eigenfunctions of L, $\Phi = \{\phi_i\}_{i=1}^{n}$, are commonly referred to as Laplacian eigenbasis or manifold harmonic basis (MHB) [154]. The Laplacian eigenfunctions are analogous to the classic Fourier basis in Euclidean space and have the following similar properties:

- Functions in $\{\phi_i\}$ all have global support on the mesh.
- Functions in $\{\phi_i\}$ exhibit wave-like periodical oscillations on the mesh with different frequencies corresponding to the eigenvalues $\{\lambda_i\}$.

- $\{\phi_i\}$ form a complete, orthonormal basis of the square-integrable function space $L^2(M)$ defined on the mesh.
- $\{\phi_i\}$ induce a spectral transform: Any signal $f \in L^2(M)$ have a unique decomposition w.r.t. $\{\phi_i\}$

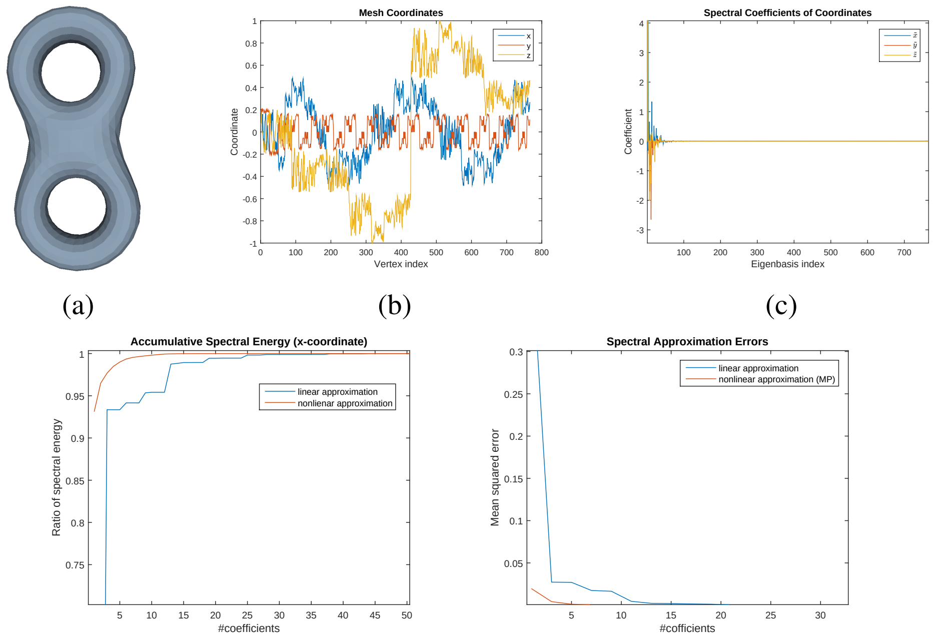
  $$f = \sum_{k=1}^{n} \tilde{f}(k)\phi_k = \sum_{k=1}^{n} \langle f, \phi_k \rangle \phi_k,$$

  in which $\tilde{f}(k)$ denotes the kth spectral/Fourier coefficient.

The aforementioned attractive properties make Laplacian eigenfunctions potentially efficient for representing shape signals defined on meshes. In [67], Karni and Gotsman utilized the truncated spectral coefficients for compressed representation of mesh geometry, which is very similar to the JPEG format for image compression. In [6], Ben-chen and Gotsman further proved that the Laplacian eigenbasis is the optimal basis for mesh compression in the mean square error (MSE) sense, provided that the distribution of the vertex coordinates satisfy certain natural assumptions.

The spectral mesh compression method introduced in [67] basically computes the linear approximation of mesh geometry expanded on its Laplacian eigenbasis. In linear approximation, spectral coefficients are always added from low-frequency to high-frequency, regardless of their respective contributions to the original signal. Better coefficient sparsity can be achieved through nonlinear approximation by prioritizing coefficients of larger magnitude.

As an example, Fig. 4.1 shows the power of Laplacian eigenbasis for shape approximation. Fig. 4.1(b)-(c) visualize the mesh coordinate functions of the example mesh and their spectral transform coefficients, respectively. We can easily see that the coordinate functions have very dense support in the natural graph basis, but can be sparsely represented in the spectral/Fourier domain in the sense that the majority of spectral coefficients are almost 0. Moreover, the few significant coefficients are mostly concentrated in the low-frequency end, especially for



(a) (b) (c)

(d) (e)

Figure 4.1: Approximation of the double-torus model with the Laplacian eigenbasis. (a) Original shape; (b) Vertex coordinate functions; (c) Spectral coefficients of the coordinate functions w.r.t. the Laplacian eigenbasis; (d) Ratios of spectral energy contained in the first k coefficients; (e) Approximation error of the mesh geometry using the first k coefficients.

a smooth shape model.

Fig. 4.1(d) shows the spectral energies contained in the first k coefficients (linear approximation) and the first k most significant coefficients (nonlinear approximation). We see that the vast majority of spectral energy are captured by the first few significant coefficients.

Fig. 4.1(e) shows how the mesh reconstruction error changes with the number of coefficients being used. In this example, we see that the approximation error becomes negligible using only about 20 non-zero coefficients.

4.4.2 Surface Inpainting

In the previous section, we have shown that the geometry of a 3D shape generally has a sparse representation w.r.t. its Laplacian eigenbasis. Hence, we can set the Laplacian eigenvector as the reconstruction dictionary and use the sparsity of coefficients as a prior to estimate the representation of missing shape geometry.

Following the formulation in Sec. 4.3, surface inpainting can be rewritten as the following sparse approximation problem

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad \|P\Phi\alpha - \mathbf{x}'\|_2^2 < \epsilon, \tag{4.4.2}$$

$$\hat{\mathbf{x}} = \Phi\alpha. \tag{4.4.3}$$

Here the pseudo-norm $\|\alpha\|_0 = \#\{i : \alpha_i \neq 0\}$ denotes the support of α, which counts the number of non-zero components of α, Φ denotes the dictionary matrix comprising the Laplacian eigenfunctions, and $\hat{\mathbf{x}}$ and $\mathbf{x}'$ represent the estimated and observable coordinate functions, respectively.

We should note that, since the Laplacian eigenbasis constitute a complete dictionary, Eq. 4.4.2 is solvable even if we set $\epsilon = 0$, in which case the reconstructed shape will exactly match the known geometry. However, strictly sparse signals are rare in real life. It is much more likely that the unknown shape geometry is *compressible* or *weakly sparse* w.r.t. the dictionary of Laplacian eigenvectors, i.e., the nonlinear approximation errors observe a power law decay as the number of participating basis vectors increases [145]. In practice, we set $\epsilon > 0$ to trade off exact reproduction for a sparser α, i.e., allowing the reconstructed signal to have small discrepancies with the observation.

Solving l_0 optimization is an NP-hard problem in nature. Fortunately, under certain conditions, greedy algorithms such as orthogonal matching pursuit (OMP) and its variants can generate the exact sparse solution or a good enough approximation [106, 150].

Another approach to find an approximated solution to Eq. 4.4.2 is to relax the highly discontinuous l_0 norm with l_1 norm, i.e.

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_1 \quad \text{s.t.} \quad \|P\Phi\alpha - \mathbf{x}'\|_2^2 < \epsilon, \tag{4.4.4}$$

or equivalently,

$$\hat{\alpha} = \arg\min_{\alpha} \|P\Phi\alpha - \mathbf{x}'\|_2^2 + \lambda\|\alpha\|_1. \tag{4.4.5}$$

The estimation problem then becomes convex and solvable. There are several readily available algorithms for solving l_1 optimization, e.g., interior point method [71], iteratively reweighted least squares (IRLS) [60], least angle regression (LARS) [144], and iterative shrinkage-thresholding [31].

For the task of surface inpainting, we find that l_1 optimization algorithms tend to be more robust and generally produce better inpainting results than greedy algorithms. In this work, we use the l1_ls solver introduced in [71] which implements a fast interior-point method for solving l_1-regularized least-square problems like Eq. 4.4.5.

As an example, Fig. 4.2 demonstrates the potentials of our sparsity-based inpainting method. We randomly label 40% of vertices of the original cube model as missing vertices, and use the coordinates of the remaining vertices to estimate the spectral coefficient representation of the original shape by solving Eq. 4.4.4. Fig. 4.2(b) shows the shape reconstructed from the estimated spectral coefficients. Fig. 4.2(c)-(d) shows the spectral coefficients computed from the original x-coordinate function and the coefficients estimated by our sparsity-based method, respectively.

In this example, our method recovers the sparse coefficient representation in a very precise way.

4.4.3 Filling Surface Holes

One of the most important technical elements of our surface inpainting method is the dictionary of global shape basis, which are determined by the global mesh connectivity. For some applications such as repairing damaged surface regions, the mesh connectivity of the region to be repaired is already known in advance before reconstruction and we may not need to modify it. For hole filling applications, however, the inpainting regions are completely blank without any inside information. It is imperative to establish interior mesh connectivity, by way of vertex insertion and patch triangulation, before our surface inpainting method can be applied.

Obviously, how a patch (to be used to cover the hole region) is triangulated directly influences the final inpainting result in our framework. In general, a good patch triangulation should ensure the vertex density of the inserted mesh to be consistent with the remaining mesh. In this work, we adopt the algorithms proposed by Liepa in [87] for hole triangulation and refinement. Algorithm 2 summarizes the pipeline of our sparsity-based hole filling method.

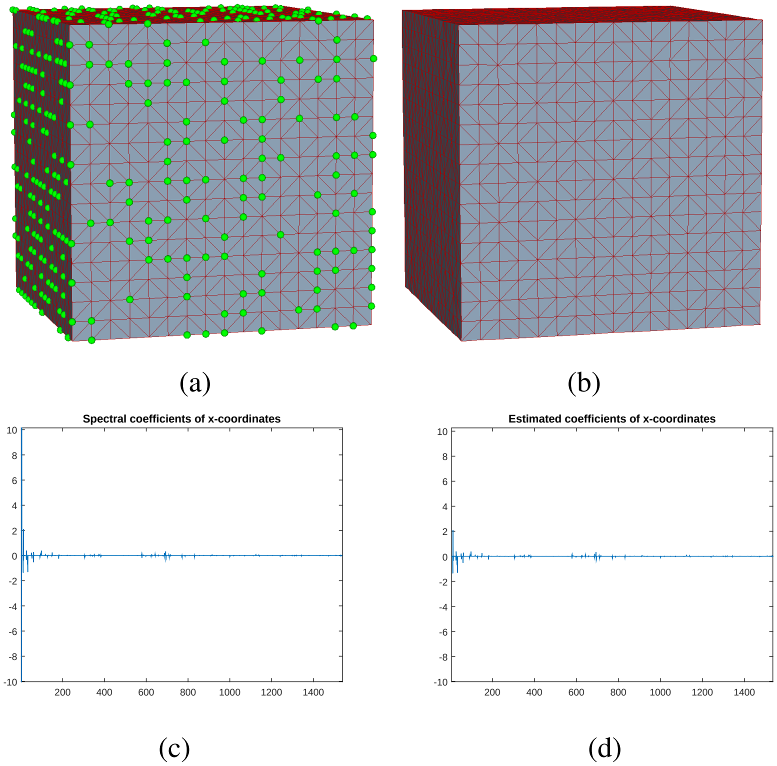
Algorithm 2 Sparsity-Based Hole Filling

```
Require: Input mesh $M$
Identify surface holes,
Triangulate and refine holes using the algorithms described in [87],
Compute the mesh Laplace matrix $L$ and the Laplacian eigenbasis dictionary $\Phi$,
for coordinate $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ do
    Compute the spectral representation $\alpha$ of the global coordinates by solving Eq. 4.4.4,
    Reconstruct the coordinates of the inserted mesh with $\Phi\alpha$,
end for
```

4.4.4 Remarks on Dictionary

Although the dictionary of Laplacian eigenvectors in general has strong compressive power for encoding shape geometry, it also has some limitations. Similar to Fourier basis, the Laplacian eigenvectors are most suitable for representing smooth signals or globally repetitive features, but are generally not optimized for encoding shapes with many local sharp features. In the image domain, other than 2D Fourier basis, people have developed various types of harmonic basis (e.g., wavelet, curvelet, ridgelet, etc) for efficient encoding of images of different properties. For example, the ridgelets are especially efficient in representing piecewise smooth images with global straight edges [41]. In the mesh domain, however, we do not have such diverse harmonic basis to choose from, which for now limits the power of sparsity-based methods.

Another issue is related to the ratio of Laplacian eigen-decomposition. Computing the full set of Laplacian eigenvectors of a large mesh is extremely time consuming, generally infeasible for meshes with more than a few thousand vertices on a regular PC. Fortunately, for our surface inpainting applications, it is actually not necessary or even desirable to compute the full set of eigenvectors. On the one hand, for smooth shapes, the spectral energy is overwhelmingly concentrated on the low-frequency end, and a dictionary composed of only low-frequency eigenvectors can well approximate the shape geometry with very little error. On the other hand, the high-frequency Laplacian eigenvectors are less stable than the low-frequency ones, and including them in the dictionary may cause overfitting and result in worse inpainting results, since the high-frequency eigenvectors are more correlated with local geometric details than with the overall structure of the shape. In our experiments, we find that the best inpainting results are usually achieved with a dictionary of 20% to 50% total eigenvectors.



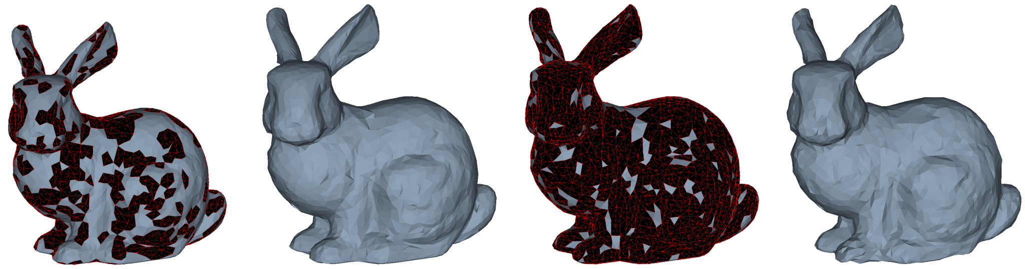
(a)

(b)

(c)

(d)

Figure 4.2: Estimating the Laplacian eigenbasis coefficients of the cube model with 40% random missing vertices. (a) Original shape model; green dots denote vertices that are labelled as missing. (b) The reconstructed shape using our inpainting method. (c) The coefficient representation of the original shape's x-coordinates. (d) Estimated coefficient representation of the x-coordinates inferred from the information of available vertices.



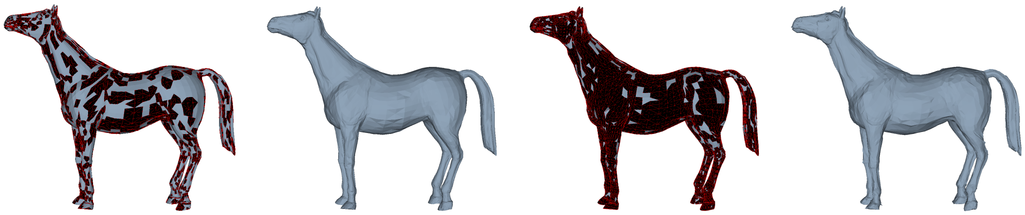
(a) 20% vertices missing.

(b) Shape recovered from (a).

(c) 50% vertices missing.

(d) Shape recovered from (c).

Figure 4.3: Recovery of the bunny model with 20% and 50% random missing vertices.



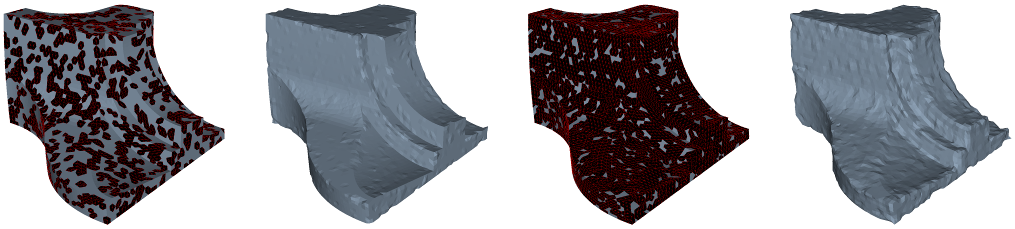
(a) 20% vertices missing.

(b) Shape recovered from (a).

(c) 50% vertices missing.

(d) Shape recovered from (c).

Figure 4.4: Recovery of the horse model with 20% and 50% random missing vertices.



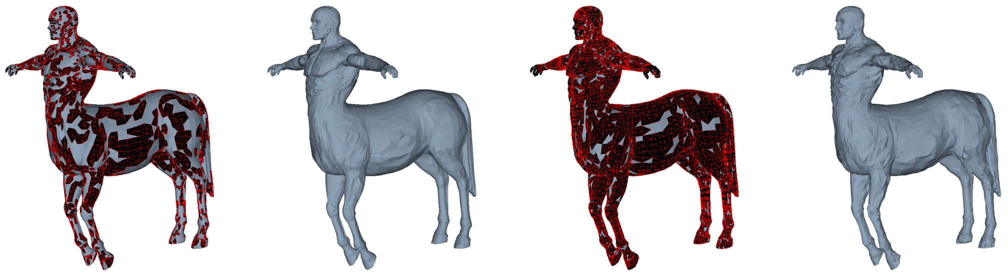
(a) 20% vertices missing.

(b) Shape recovered from (a).

(c) 50% vertices missing.

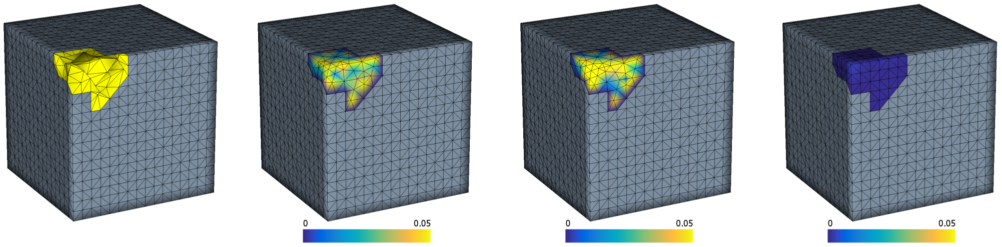
(d) Shape recovered from (c).

Figure 4.5: Recovery of the fandisk model with 20% and 50% random missing vertices.



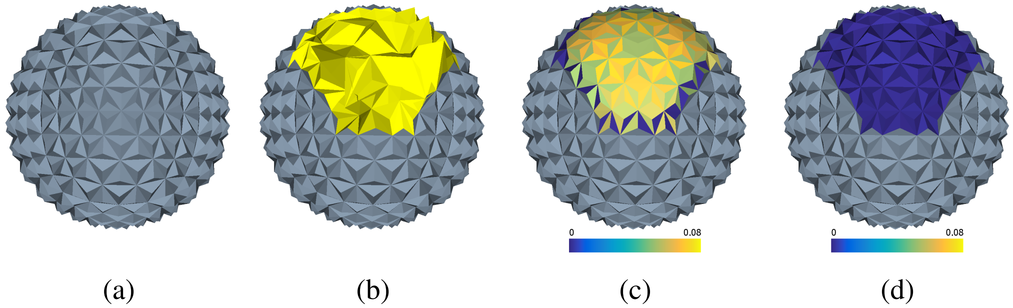
(a) 20% vertices missing.
(b) Shape recovered from (a).
(c) 50% vertices missing.
(d) Shape recovered from (c).

Figure 4.6: Recovery of the centaur model with 20% and 50% random missing vertices.



(a) (b) (c) (d)

Figure 4.7: Geometry repair of the cube model by replacing the selected damaged regions (marked in yellow) with an inpainting patch. (a) The damaged model; (b) Repaired with Laplacian regularized least square smoothing [104]; (c) Repaired with thin-plate energy minimization [4]; (d) Repaired with our inpainting method. In (b)-(d), the per-vertex error (compared with the ground truth) is color-coded.



(a) (b) (c) (d)

Figure 4.8: Geometry repair of the damaged epcot model. (a) The original epcot model; (b) The damaged model (damaged region is marked in yellow); (c) Repaired with thin-plate energy minimization [4]; (d) Repaired with our inpainting method. In (c) and (d), the per-vertex error (compared with the ground truth) is color-coded.

4.5 Experiments

In this section, we first evaluate the performance of our sparsity-based inpainting algorithms on recovering missing geometry from partial observations. Then we demonstrate how our method can be applied to repairing damaged geometry and filling surface holes.

4.5.1 Geometry Recovery

To evaluate the performance of geometry recovery, for each testing model, we randomly label 20%-50% vertices as missing and use our sparsity-based inpainting method to estimate the original geometry based on the coordinates of the still available vertices. The estimated coordinates are then compared with the original coordinates.

All the testing models have been translated and scaled to be contained inside the unit cube. The recovery error is measured as the root-mean-square error (RMSE) of the coordinates of the missing vertices.

Fig. 4.3, Fig. 4.4, Fig. 4.5, and Fig. 4.6 show some examples of geometry recovery with 20% and 50% missing vertices. Fig. 4.10 shows two examples

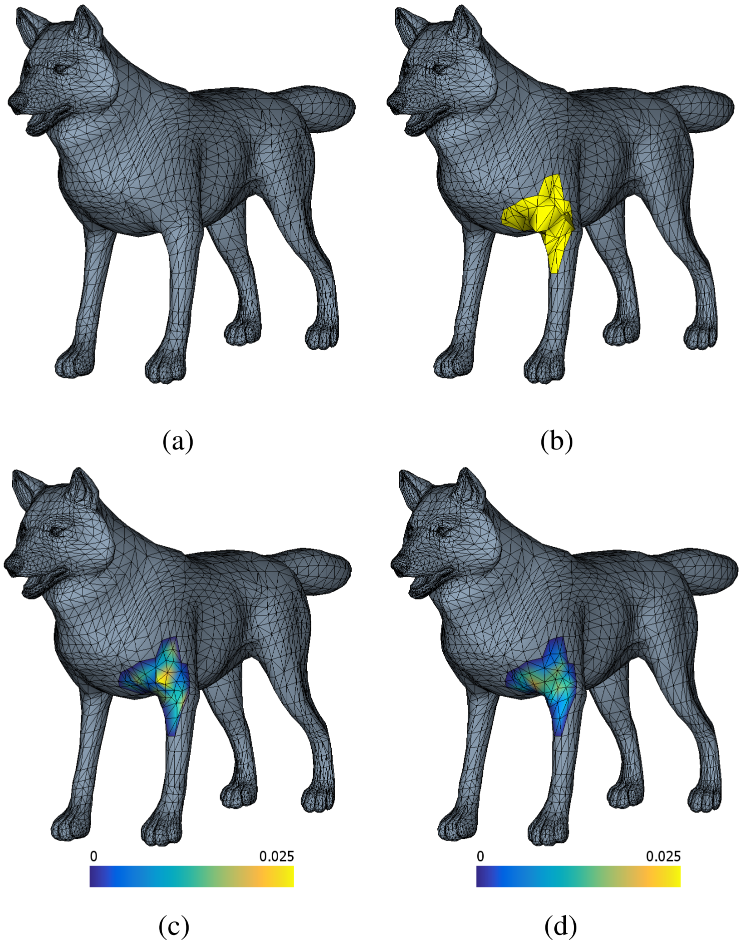
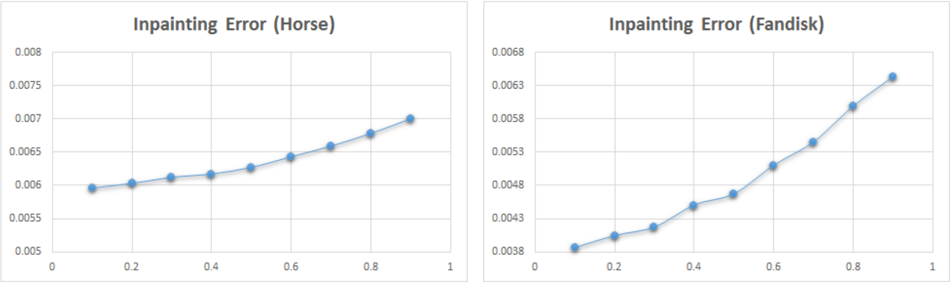


Figure 4.9: Geometry repair of the damaged wolf model. (a) The original wolf model; (b) Damaged model with significant noise in the region marked in yellow; (c) Repaired with Laplacian regularized least square smoothing [104]; (d) Repaired with our inpainting method. In (c) and (d), the per-vertex inpainting error (compared with the ground truth) is color-coded.



(a) Horse model

(b) Fandisk model

Figure 4.10: Inpainting errors given different ratios of randomly missing vertices using our method. The horse model uses 2,000 eigenvectors and the fandisk model uses 3,000 eigenvectors to construct dictionaries, respectively.

Mesh	#vertices	#eigenvectors	Decomposition time (s)	Missing ratio	Error	l_1 time (s)
bunny	2.5k	1000	19.6	0.2	1.9e-2	2.8
				0.5	2.2e-2	2.1
horse	8.4k	4000	1015.4	0.2	8.6e-3	19.7
				0.5	1.0e-2	59.0
fandisk	6.5k	3000	445.2	0.2	8.8e-3	17.4
				0.5	1.2e-2	33.9
centaur	15.8k	2000	397.7	0.2	7.1e-3	12.2
				0.5	8.0e-3	15.0

Table 4.1: Geometry recovery errors and time performance. For each model, we test the recovery performance with 20% and 50% randomly selected vertices labelled as missing. Each experiment has been repeated three times and averaged on a system with quad-core 2.4GHz CPU and 16GB RAM.

of how the inpainting error (MSRE) changes with the ratio of vertices randomly labeled as missing.

Table 4.1 documents the recovery errors and time performance of our tests. From the experimental results, we have the following observations:

- When the missing vertices are randomly dispersed on the shape, our sparsity-based method can reliably recover the missing coordinates with great precision, even when the ratios of missing vertices are as high as 50%.
- The l_1 estimation generally becomes more time consuming when the ratio of missing vertices increases.
- As noted in Sec. 4.4.4, using the truncated Laplacian eigenbasis dictionary is acceptable for restoring smooth shapes. However, for shapes with many edges and corners, such as the fandisk model (see Fig. 4.5), our inpainting method cannot well preserve local discontinuities, since the high-frequency basis are simply not present in the truncated dictionary.

4.5.2 Geometry Repair

Our sparsity-based inpainting method is very suitable for repairing partially damaged geometry. After manually selecting the damaged regions, we can apply our inpainting method to estimate the original whole shape with the same connectivity based on the remaining parts of the shape. The corrupted regions can then be substituted by the inpainting patch.

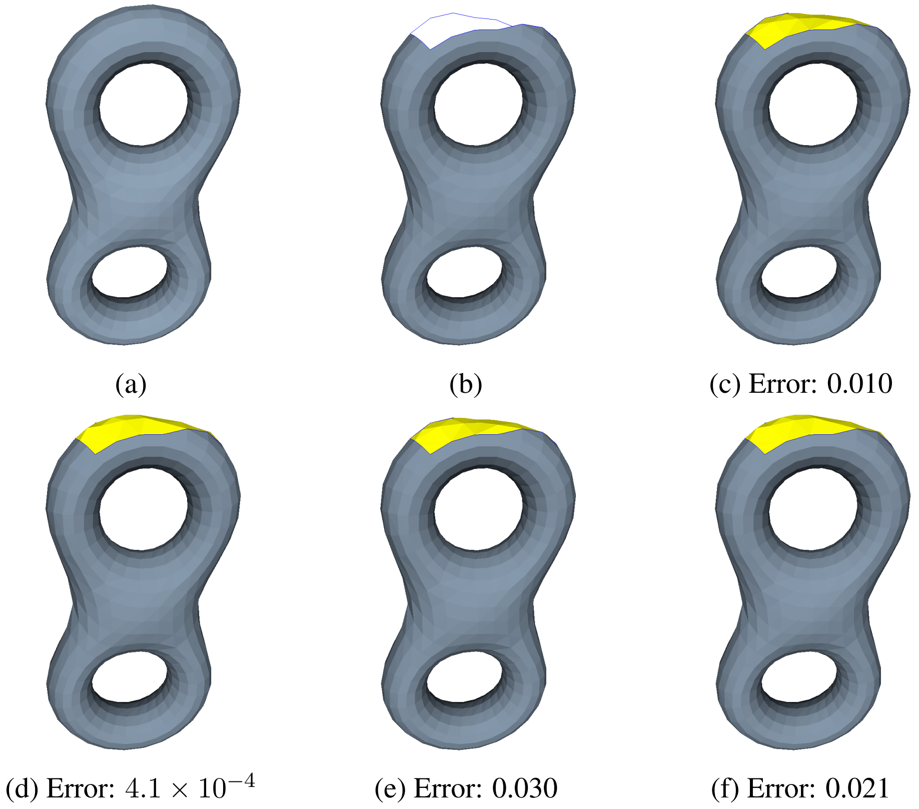
Fig. 4.7, 4.8, and 4.9 demonstrate repairing damaged local geometry using our sparsity-regularized inpainting method. The results are compared with two geometry-regularized mesh optimization methods: Laplacian regularized least square smoothing [104] and thin-plate energy minimization [4]. We can see that, although geometry-regularized methods can generate patches that are smooth and blend well with the surroundings, they fail to recognize the intrinsic structures of the original shapes; consequently, important geometric features are simply smoothed out. In contrast, our sparsity-regularized inpainting method takes into account the global shape structures, and almost perfectly recovers the edges and corners in the cube model (Fig. 4.7) and the geometric textures of the epcot model (Fig. 4.8) from partial observations.

4.5.3 Hole Filling

As introduced in Sec. 4.4.3, for general hole filling tasks, the mesh connectivity information inside holes are probably unknown. We must first triangulate holes in a proper way and then apply our geometry inpainting method to optimize the newly inserted mesh. How the hole is triangulated directly impacts the global Laplacian eigenbasis which subsequently determine the estimated recovery.

As an example, Fig. 4.11 compares the results of filling the holes of a double torus model with and without original connectivity information, using our sparsity-regularized method and the geometry-regularized method proposed in [4]. We can see that estimating with a different connectivity significantly alters the final hole fairing results. In this example, our method generate shapes that are more approximate to the original shape both with the original connectivity and with the new connectivity.

In general cases, we cannot expect the hole filling result using our sparsity-regularized inpainting method to precisely match the original shape when the number of vertices and connectivity of the patching mesh, generated from hole triangulation and refinement, are different from the original mesh. Nonetheless, the resulting patching meshes tend to be coherent with the whole remaining shape, thanks to the global shape awareness of our method. Fig. 4.12 shows two examples of filling holes utilizing our inpainting method. The results are comparable to the geometry-regularized surface restoration method in [4].



(a)

(b)

(c) Error: 0.010

(d) Error: 4.1×10^{-4}

(e) Error: 0.030

(f) Error: 0.021

Figure 4.11: Compare hole filling results using our sparsity-regularized method and the geometry-regularized method introduced in [4]. The error is measured as the root-mean-squared deviation from the estimated vertices to the original shape. (a) The original double torus model. (b) The model with a hole. (c) Inpainted using the method in [4] with the original mesh connectivity. (d) Inpainted using our method with the original mesh connectivity. (e) Inpainted using the method in [4] with the mesh connectivity generated from hole triangulation and refinement. (f) Inpainted using our method with the same mesh connectivity as (e).

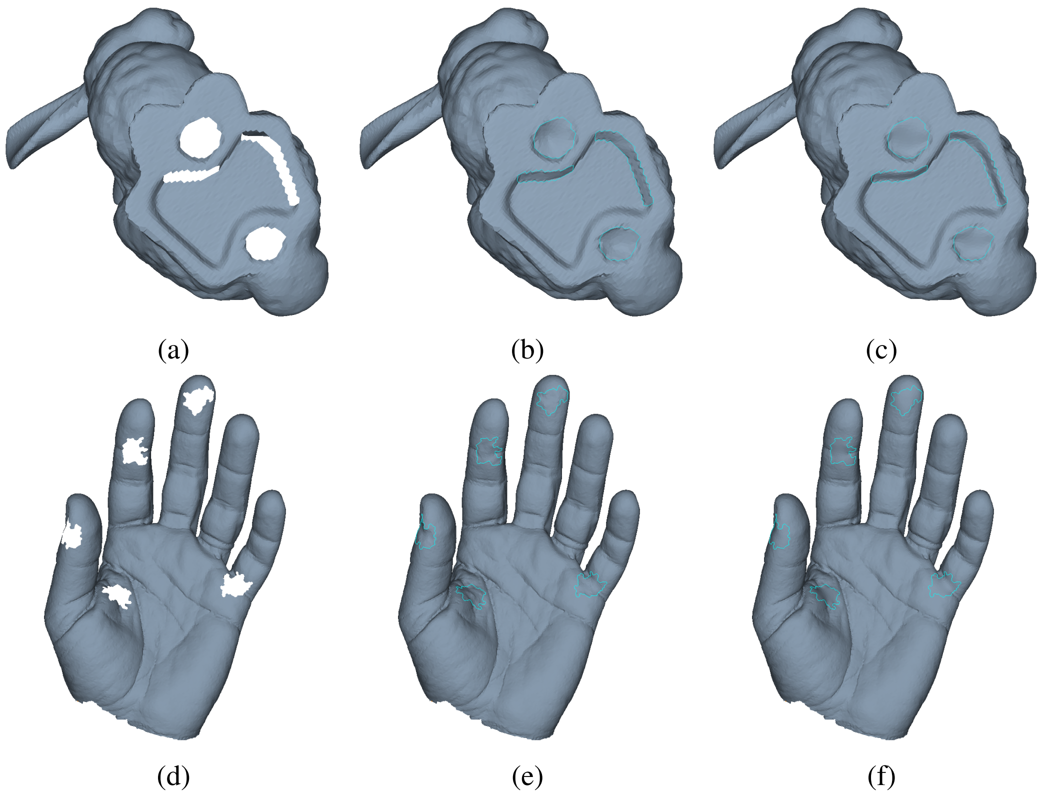


Figure 4.12: Inpainting existing holes on the bunny and hand models. (a)(d) Original models with holes; (b)(e) Hole-filling result using our method; (c)(f) Hole-filling result using the method of [4].

4.6 Chapter Summary

In this chapter, we have proposed a novel surface inpainting algorithm based on sparse signal recovery. Instead of directly estimating the local missing geometry, our new inpainting framework is designed to discover the coefficient representation of the entire original shape in a transform domain. When the shape geometry is sufficiently sparse with respect to the dictionary of transform basis, chances are we can accurately recover this sparse representation by imposing sparsity constraints on the coefficients given partial observations. In our method, we adopt the mesh Laplacian eigenbasis as dictionary, and formulate surface inpainting as a sparse signal recovery problem. Leveraging standard l_1 optimization techniques, we can obtain an estimated shape which agrees with the observable parts and are globally coherent. For shapes that are highly compressible w.r.t. the Laplacian eigenbasis, we have experimentally demonstrated the great potential of our method for geometry restoration, geometry repair, and hole filling.

For the future work, we plan to extend our sparsity-based inpainting framework by integrating geometric constraints such as curvatures and normals, which should improve the geometric consistency of the inpainting result. We are also interested in designing new types of shape basis and exploring more sophisticated strategies for constructing dictionaries, e.g., dictionaries that are adaptive to the input shape.

Chapter 5

Generalized Feature Description and Detection

5.1 Introduction

Studies in feature abstraction and analysis have been gaining momentum because features are essential for numerous downstream graphics tasks and applications such as shape recognition, segmentation, analysis, understanding, etc [14, 74, 142]. Influenced by the trending concept of high-level representations in computer vision, which are based on object-wise components, more attention has now been directed towards region-wise feature analysis in geometry modeling. In this work, we advocate a new region-based and user-specified type of feature as well as a novel wavelet-inspired multi-scale and multi-level descriptor, and they jointly enable our feature detection framework that can further facilitate various applications.

Conventional feature descriptors are usually constructed by considering the discontinuities of certain differential attributes of different orders (e.g., the second-order attribute like surface curvature) that naturally afford their discriminative power in characterizing point features, line/curve features, small patch-based features with regular boundaries, etc. Such descriptions have been employed in point/patch-based recognition, point-wise correspondence, and curvature-based saliency detection with great success. However, for more complicated applications such as modeling by example [46], model composition [77], and key component analysis [140], the aforementioned conventional features are usually too localized to capture the multi-scale neighboring information, and it is desirable to have a flexible, region-induced feature description. Furthermore, in many real-world settings, shape data may be degraded due to acquisition imperfections and noises, necessitating the use of region descriptors which tend to be much more robust.

Existing works related to region-wise analysis include partial matching, shape correspondence, saliency extraction, etc. Boundaries of the regions in question are usually confined to regular but non-adaptive shapes [52, 70, 75, 103], thus neighboring and in-between geometry information may not be fully captured. As for region description, trending measures include the distributions of various types of point descriptors [7, 91, 110] and the global analysis of the regions based on spectral decomposition [63, 80]. Some regional measures are not discriminative enough to solely characterize the regions in question, for which post-processing like geometric hashing [78] or random sample consensus [45] is required. Nevertheless, these settings are not hierarchical enough to characterize the focal regions. On the other hand, multi-scale shape analysis methods [134, 146], in spite of their great descriptive power, have not yet been employed to construct regional descriptions. These insights inspire us to propose a more comprehensive and stable type of shape description that can encode the region of interest with high discriminative power and efficiency.

In this chapter, we propose a local-to-global shape feature via user specification, introduce an informative region descriptor, and then present a shape feature detection framework to facilitate a host of graphics applications. The proposed shape feature extends the definition of conventional features to a region-wise manner in a user-specified way (as highlighted in Fig. 5.1). To encode user-specified features, we proactively seek an informative regional descriptor con-

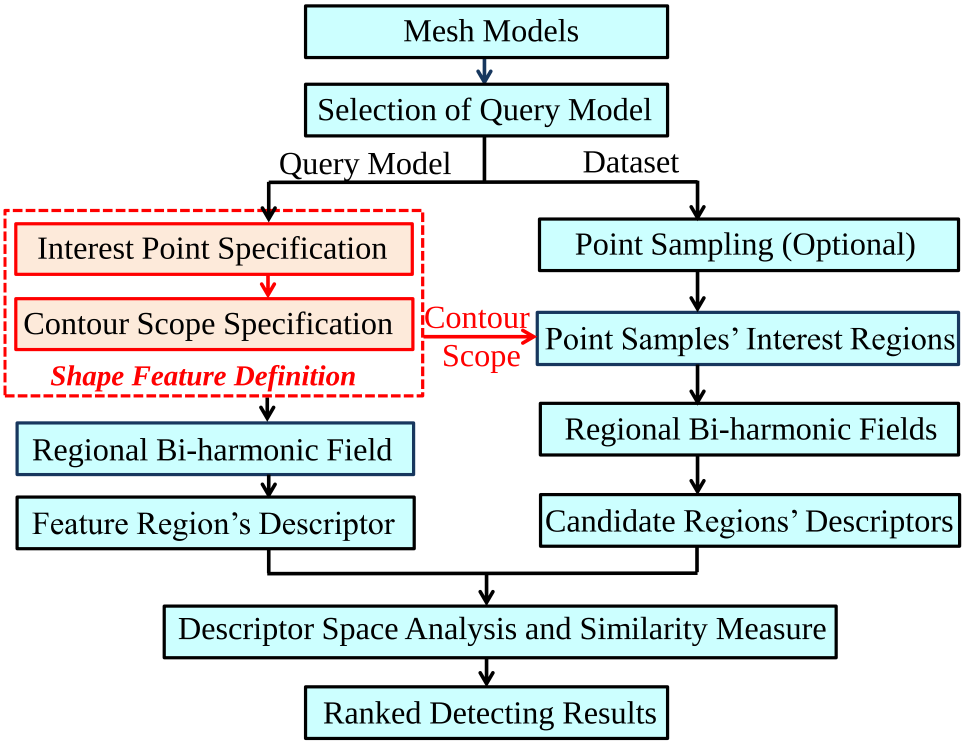

Mesh Models
Selection of Query Model
Query Model
Dataset
Interest Point Specification
Contour Scope Specification
Shape Feature Definition
Contour
Scope
Point Sampling (Optional)
Point Samples' Interest Regions
Regional Bi-harmonic Field
Regional Bi-harmonic Fields
Feature Region's Descriptor
Candidate Regions' Descriptors
Descriptor Space Analysis and Similarity Measure
Ranked Detecting Results

Figure 5.1: Pipeline of our feature detection and description framework.

structed in a multi-scale and multi-level way. Our descriptor takes advantage of the bi-harmonic distance field and the SGWs. The stability and robustness of bi-harmonic distance field guarantee the technically-sound foundation for the whole descriptor. SGWs naturally accommodate local and global geometry with a multi-scale solution, and such solution is consistent across multiple levels. We devise a new statistical method based on the decomposition coefficients of the shape signal, enabling the joint analysis of the underlying geometry together with different shape signals. In order to comprehensively characterize the shape features, we also incorporate the contour-centered geometric statistics into our descriptor. All of these enable our feature detection framework as shown in Fig. 5.1. We quantify each model's regions of interest around central points (or point samples) according to the user-specified feature scope on the query model. Then descriptors are constructed on candidate regions across different models, which is equivalent to transform each region into the high-dimensional feature space. After the region-wise feature space is constructed, various analytical tasks can be performed. The primary contributions of this work can be summarized as follows:

- We propose to define a generalized shape feature type via user specification, which is geometry-aware and is the organic coupling of local and global description. Also, it is a fundamental tool that can help unite different types of graphics applications.

- Our region-based descriptor is primarily built upon the SGWs that are both multi-scale and multi-level in nature, elegantly integrating both local (differential) and global (integral) information. We also introduce the contour-centered geometric statistics to enhance the descriptor's discriminative power.

- We develop a feature detection framework, which can integrate different types of state-of-the-art region descriptors and further facilitate widespread graphics applications including partial matching, coarse-to-fine recognition, model recognition, etc.

5.2 Related Work

This section will briefly review prior research related to region analysis and the latest progresses on SGWs.

Region-wise Description and Detection. We first review how current methods define boundaries for given regions. Broadly speaking, they characterize a local neighborhood around a central point in two ways. The first is based on Euclidean distance, such as spheres [70], blowing bubbles [103], rings [52], shape context [75] or priori decompositions [49, 65]. The second is to use geodesic distance like geodesic fans [162] and spiral pathway [79]. The shapes of region boundaries defined by the above methods are mostly restricted to regular formats, and such nonadaptive neighborhoods cannot precisely reflect the local geometrical or topological distortions. Region-wise descriptors can be roughly divided into two categories: point-based and region-based methods. Point-based methods provide the quantitative measure by organizing single-valued point signatures into certain kinds of distributions. Various kinds of point descriptors have been incorporated in this manner, e.g., the shape index (SI) [148], shape diameter function (SDF) [137], heat kernel signature (HKS) [124], Zernike moments [100], etc. Region-based methods analyze the entire focal region through spectral decomposition [63, 66], which can robustly depict the intrinsic geometry. However, due to the instability of local Laplacian decomposition, these methods usually cannot well handle small and complex regions.

Region-based Partial Matching and Correspondence. In literature, region analysis has been discussed mainly in the research of partial matching problems, and several categories of techniques have been employed. Skeletal-graph-based approaches such as [9] couple geometry and structure in a single skeletal descriptor based on the theory of Reeb graph. The main drawback is that sub-parts cannot be recognized automatically. Multi-criterion optimization approaches [24, 79, 94] try to match subparts by striking balance between significance and similarity criteria. This type of methods require the knowledge of correspondence between shapes, otherwise, it can only be solved by alternating between correspondence and part area, which is time-consuming. Bag-of-words (BoWs) technique has been adopted [16, 161] to represent a shape or a subpart as a collection of local feature signatures quantized in some vocabularies of “geometric words”. If the geometric vocabulary is sufficient and the shapes have significant common parts, it is possible to compare partially-similar shapes, otherwise, these methods oftentimes fail to function properly. Furthermore, improper additions of the spatial information and imprecise binning process may lead to the averaging-off effects of geometric information. The concept of non-point-wise correspondence was first proposed in [124] by using region-wise local descriptors and optimizing over the integration domain upon which the integral descriptors of the two parts match. This method can exactly match fragments to entire shapes, however, since it utilizes the absolute values in calculation like integration, it cannot deal with the partially similar correspondence. Besides, many of the above approaches rely heavily on exact or meaningful shape decomposition process as in [65, 80], which is computationally expensive and significantly influences the final correspondence results. Also, in order to achieve meaningful results, many approaches utilize time-consuming post-processing like in [49] and [65]. So it requires more effective and generalized region detection techniques to help speed up and improve the precision of the partial matching and correspondence processes.

5.3 Local-to-global Shape Feature Definition

In this section, we introduce the definition process of the proposed novel shape feature, which generalizes conventional features (e.g., point, line, or patch features) to a local-to-global level via user specification. Our shape feature is extracted via the bi-harmonic distance field [88], which is robust, globally “shape-aware”, parameter-free, and widely used in geometry processing. Among these attractive properties, the consecutive depicting power inspires us to incorporate it for integrating multi-scale regional information that is required for subsequent analysis. In addition, the cross-sections of contours in the bi-harmonic distance

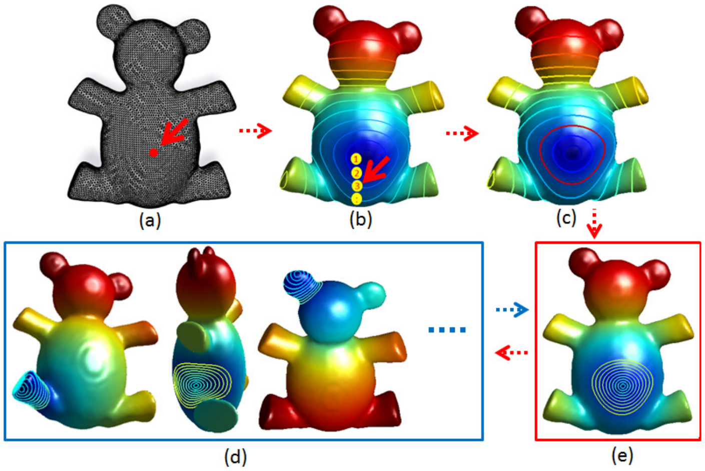


Figure 5.2: The functional pipeline of our feature detection framework. (a)-(c) show the user's inputs in the specification process, red arrows in (a) and (b) denote the specified point of interest and contour scope, respectively. The bottom row shows the specified shape feature (e) and analogous feature regions (d) (we only display three cases here).

field naturally form boundaries for user-specified features, thus avoiding the shape decomposition process.

Our shape feature is a user-specified partial region with two parameters: the point of interest and the contour scope, both of which can be determined using a simple user-interactive process. The point of interest is the relative center of the feature, it can either be picked directly on the mesh, as shown in Fig. 5.2(a), or be automatically initialized as the extreme point of a function defined on the surface. In order to specify the contour scope of the point of interest, we should first introduce the metric with which we set the scope, namely, the bi-harmonic distance field.

Let us consider a 3D mesh represented as a graph $M = (V; E)$ with vertices V and edges E, where $V = \{v_1, v_2, ..., v_n\}$ and n is the number of the vertices. A vector-valued function $f\colon \mathbf{V} \to \mathbb{R}^q$ defined on V can be represented as an $n \times q$ matrix, where the i-th row represents the function value at v_i, we denote it as

$f(i)$. According to [88], the bi-harmonic distance between vertex v_i and v_j can be expressed as

$$D_{bh}(i,j)^2 = \sum_{k=1}^{m} \frac{(\chi_k(i) - \chi_k(j))^2}{\lambda_k^2}, \tag{5.3.1}$$

where $\{\lambda_k\}$ and $\{\chi_k(\cdot)\}$ are, respectively, the first m non-zero eigenvalues and the corresponding eigenfunctions of the Laplacian-Beltrami operator with "cotangent formula" discretization [101].

For each vertex, Eq. (5.3.1) defines a diffusion field around it. We compute the diffusion field of the specified interest point (denoted as v_s) and then construct a set of contours (Fig. 5.2(b)) across the entire model with contour points located on the edges of the mesh model. These contours can very well reflect the changes locally around the central point and characterize the corresponding global structures, and these will be discussed later. In order to make the setting of parameters more stable, we normalize the original models using a unit box. Then the total number of the contours distributed in the diffusion field can be set empirically to the integer nearest to $max(D_{bh}(s,\cdot))/0.05$, which is dense enough to depict the diffusion field. Then the user can choose one of the contours to set the scope of the feature (as shown in Fig. 5.2(b)) and denote it as S_s. Fig. 5.2(c) illustrates the feature defined with the red boundary.

Our shape features can be located anywhere and vary spatially in scale depending on specific applications. They integrate the local and global geometry, which affords their great potential in bridging differential and integral geometry information.

5.4 Multi-level And Multi-scale Shape Description

Our novel descriptor primarily make use of the statistics of SGWs coefficients. As a supplement, contour-based statistics are also included the descriptor to achieve better discriminative power.

5.4.1 SGW-based Description

As mentioned in Chapter 2, spectral graph wavelet transform (SGWT) defines a new type of wavelet analysis on graphs by performing scaling of mother wavelet in the Fourier domain instead of the spatial domain. The SGWs are expressed as bivariate kernel functions expanded on the manifold harmonic basis

$$\Psi_t(i, j) = \sum_{k=0}^{n-1} g(t\lambda_k)\chi_k(i)\chi_k(j), \tag{5.4.1}$$

where g is the real-valued wavelet generating kernel and t is the scale parameter. The ith row of $\Psi_t(\cdot, \cdot)$

$$\psi_{t,i}(\cdot) = \Psi_t(i, \cdot) = \sum_{k=0}^{n-1} g(t\lambda_k)\chi_k(i)\chi_k(\cdot), \tag{5.4.2}$$

is the spectral wavelet spatially-localized at v_i, and in the frequency domain, localized at scale t. It should be noted that we choose to use the geometric mesh Laplacian for more geometric-aware description, instead of the combinatorial Laplacian originally used in [59].

Multi-scale SGWs can well represent both the high frequency and low frequency geometric information around the index point. Suppose we compute the spectral wavelets at J different scales $\{t_1, t_2, ..., t_J\}$, and adopt the same formulation of generating kernel functions used in [59], given by

$$g(x) = \begin{cases} x^2 & if\ x < 1 \\ -5 + 11x - 6x^2 + x^3 & if\ 1 \leq x \leq 2 \\ 4x^{-2} & if\ x > 2 \end{cases}, \tag{5.4.3}$$

and the J scales are selected to be logarithmically equally spaced between the minimum scale $t_J = 2/\lambda_{max}$ and the maximum scale $t_1 = 40/\lambda_{max}$, where λ_{max} is the upper bound of the Laplacian eigenvalues. The settings of t_1 and t_J guarantee that $g(t_1 x)$ has power-law decay for $x > \lambda_{min}$ and $g(t_J x)$ has monotonic polynomial behavior for $x < \lambda_{max}$.

Using the above formulations, we can easily obtain the wavelet coefficients of a given function centered on a specific vertex v_i as

$$W_f(t, i) = \langle \psi_{t,i}, f \rangle = \sum_{l=0}^{n-1} g(t\lambda_l)\hat{f}(l)\chi_l(i), \tag{5.4.4}$$

where

$$\hat{f}(l) = \langle \chi_l, f \rangle. \tag{5.4.5}$$

Here, the signal function f can be any kind of surface signal depending on the downstream application. For example, mean curvature, characterizing detailed local distortions, is a good choice for the recognition of repetitive features within individual models and the detection of similar features across models with different poses. For coarse-to-fine recognition, the HKS is more favorable thanks to its robustness to noise. The coefficients W_f obtained from the spectral wavelet transform is the inner product of the signal function and the corresponding wavelet at scale t and location i. It is an encoding of the signal at that particular scale, i.e., it describes the original signal in certain frequency with respect to the local geometry and topology. Repeating this process for J scales (as shown in Fig. 5.3(a)-(d) with 4 scales), the collection of coefficients obtained comprises our multi-level and multi-scale descriptor.

Instead of using W_f directly as descriptors like in [84], we incorporate it into our descriptor by taking advantage of the consecutive depicting power of the biharmonic distance field. In order to comprehensively describe the geometric information contained within the feature, we divide the feature region into thinner bands with more contours as shown in Fig. 5.2(c)-(e). The number of dense contours may be set automatically as the integer nearest to $S_s/0.025$, and such dense contours can elaborately depict and organize the inner geometrical information within the feature region. Fig. 5.3(e) illustrates the setup of our SGW-based statistical bands based on W_f, where different layers convey multi-level (from high

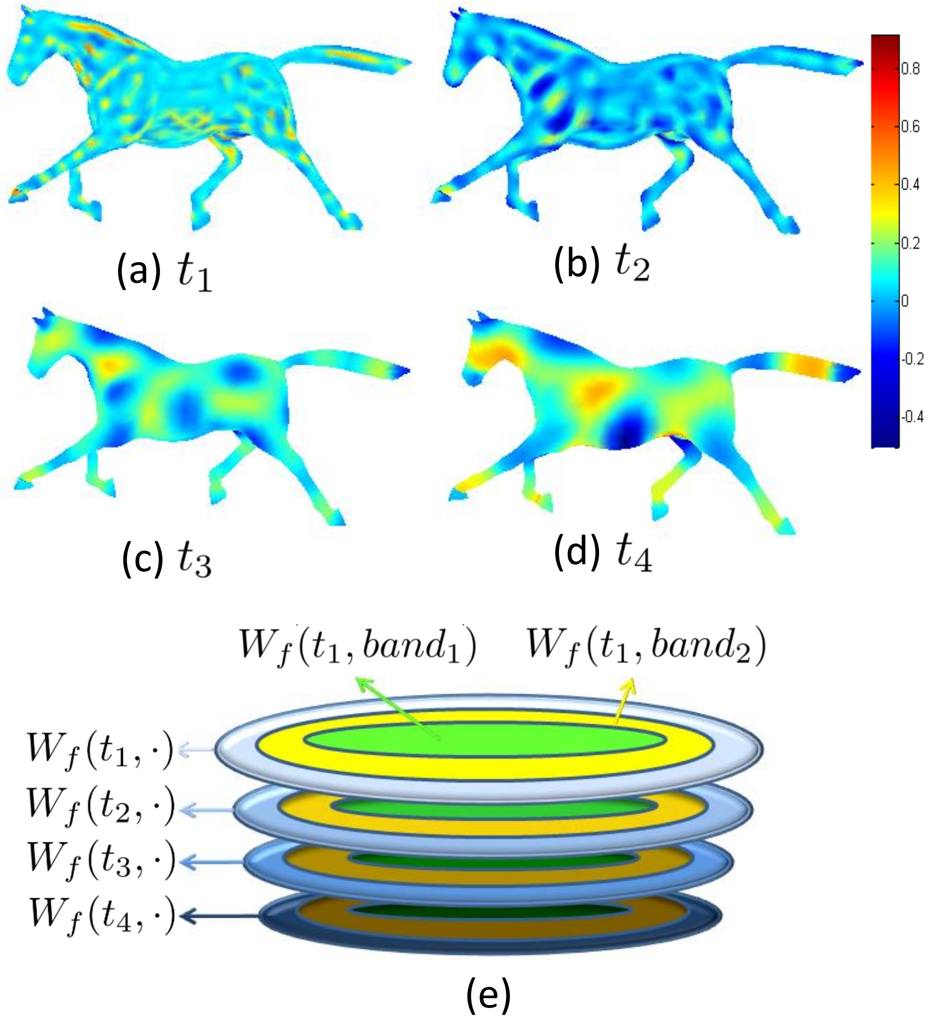


Figure 5.3: Statistics on signal's wavelet decomposition coefficients ($W_f(\cdot, t)$). (a)-(d) list t from small (t_1) to large (t_4) value, corresponding to the decomposed signals varying from high to low frequencies. (e) illustrates the composition of statistics based on W_f within one feature region.

to low frequency) information and different bands (denoted in yellow and green) in-between contours encode multi-scale information. By stretching the "matrix-like" statistic (as in Fig. 5.3(e)) into a high dimensional vector, we obtain the descriptor of v_s, denoted as D_s, given by

$$D_s = [\underline{W_{t_1}^{b_1}, \dots, W_{t_1}^{b_L}}, \underline{W_{t_2}^{b_1}, \dots, W_{t_2}^{b_L}}, \underline{\dots}, \underline{W_{t_J}^{b_1}, \dots, W_{t_J}^{b_L}}], \tag{5.4.6}$$

where $W_{t_i}^{b_j}$ denotes the statistic of W_f with scale t_i on the j-th band, and L is the number of contours. Here $W_{t_i}^{b_j}$ can be expressed as the 1-norm of $W_f(t_i, \cdot)$ over the j-th band

$$W_{t_i}^{b_j} = \sum_{p \in b_j} |W_f(t_i, p)|, \tag{5.4.7}$$

where p is the vertex index, $p \in b_j$ denotes p is a vertex located on the j-th band.

In Fig. 5.3, we observe that information contained in W_f of different time scales corresponds to the fine-to-coarse multi-level information, and the statistics on the bands convey the near-to-far multi-scale knowledge. These collectively make use of SGWs' power in integrating geometric information. In addition, we notice that descriptors based on W_f are scale-invariant as long as SGWs are normalized.

5.4.2 Contour-based Multi-scale Statistics

The contours of bi-harmonic distance field encode rich information of local-to-global geometric variation. So we further introduce the perimeters of contours and the distance distribution of contour points to help characterize the focal region's shape in an orderly and quantitative manner.

For a specified feature and its corresponding contours, we first calculate contours' perimeters and concatenate them as $\{p^{c_1}, p^{c_2}, ..., p^{c_L}\}$, where c_i denotes the index of the i-th contour. Then, for each contour, we compute the Euclidean distances between the contour points and their barycenter (as shown in Fig. 5.4), and further evaluate the probability distribution of the distances. We denote the

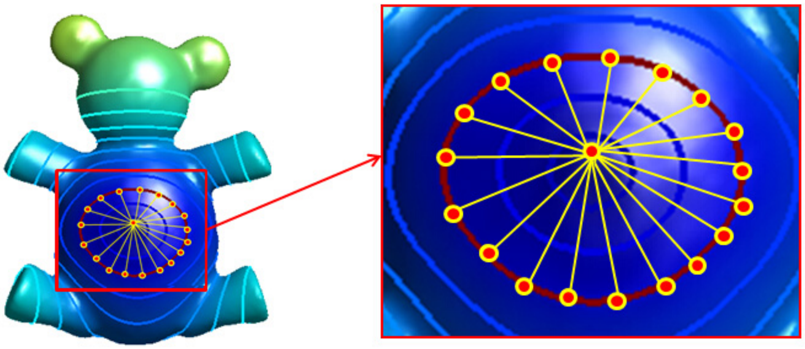


Figure 5.4: Contour-based distance distribution. The zoomed-in part shows the distances under current consideration when performing statistical calculation.

distance-related statistics as $\{ds^{c_1}, ds^{c_2}, ..., ds^{c_L}\}$. Here, ds^{c_i} is a vector stacking up the probability distribution of the distances concerning the i-th contour. We uniformly separate the distance values into M bins, ranging from zero to the maximum value after removing the top and bottom 5% to rule out possible outliers. Then its stack pattern is

$$ds^{c_i} = [\frac{num(b_1)}{num(c_i)}, \frac{num(b_2)}{num(c_i)}, ..., \frac{num(b_M)}{num(c_i)}], \tag{5.4.8}$$

where $num(b_j)$ is the number of points with distance values falling in the j-th bin and $num(c_i)$ is the number of points on the i-th contour. For multiple contours with the same value, they should be considered as a whole when performing statistical calculation.

These two measurements help describe the shape of the bi-harmonic distance field completely and identify the details of shape's distortion. The purpose of introducing the distance distribution is to distinguish between different contours with the same perimeters.

5.4.3 Informative Region-based Descriptor

So far, the separate parts of our descriptor have all been introduced. Now, it sets the stage for us to integrate them together to form our final informative hi-dimensional descriptor as

$$\begin{aligned} D_s = [&\omega_s * (W_{t_1}^{b_1}, \dots, W_{t_J}^{b_L}), \\ &\omega_p * (p^{c_1}, \dots, p^{c_L}), \\ &(ds^{c_1}, \dots, ds^{c_L})], \end{aligned} \tag{5.4.9}$$

where ω_s and ω_p are weights to adjust the contributions of the three parts in the descriptor. The settings of these weights will be detailed in Section 5.6.

Our descriptor integrates the attractive properties of both SGWs and contour-based measurements. From the viewpoint of feature mapping, SGWs establish a powerful foundation for hierarchical representation of the geometrical and topological details. Our design of the regional description encodes the shape feature in the aspects of both "breadth" and "depth", paving the way for our feature detection framework.

5.5 Feature Detection Framework

Using the same way to define feature regions around candidate points and formulate the corresponding descriptions, descriptors concerning candidate regions on the same model or different models in the database can be easily computed for analytical purposes.

5.5.1 Constructing Descriptors over Shapes

To construct descriptors on candidate regions on one or more models, the central points and contour scopes should also be determined first. As for the central points, we implement the farthest-point-sampling strategy [102] to uniformly ex-

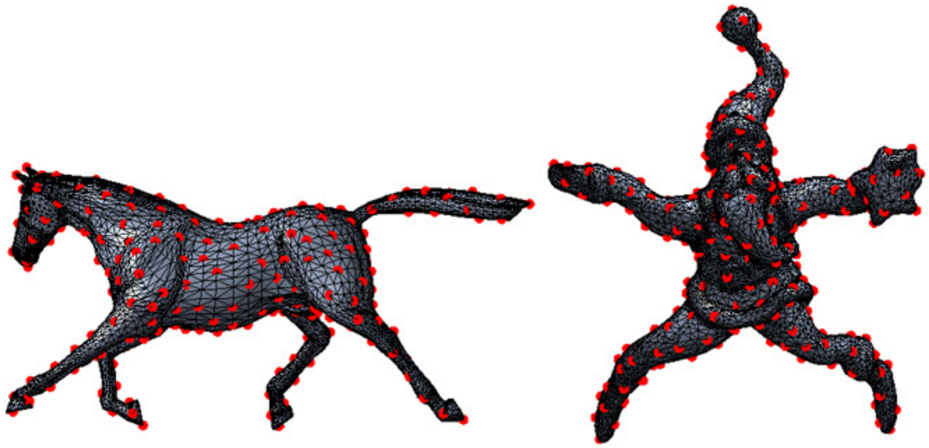


Figure 5.5: Sampling results on horse and Santa models using the farthest-point-sampling strategy. Only part of the sampling points are shown for visual clarity.

tract points on the mesh model (as shown in Fig. 5.5). This strategy ensures not only the uniform distribution of the candidate points, but also the inclusion of the end points, which are interesting alternatives for user's selection. However, we want to mention that the sampling process is optional, and users could either pick the desired sampling methods or simply use all the vertices as candidates according to specific applications.

Then the construction of descriptors across versatile shapes is based on the knowledge of the shape feature defined. That is, candidate regions are determined automatically according to the contour scope of the shape feature specified. Suppose that v_s and v_i are respectively the interest point and one of the candidate points. Then the scope of v_i is set as $S_s * max(D_{bh}(i, \cdot))/max(D_{bh}(s, \cdot))$ and this can help ensure the robustness of our method for deformable models. With the region scope determined, the construction of the corresponding descriptor is conducted in the same way as the specified feature (in Section 5.4), which is detailed in Algorithm 3.

With each candidate region equipped with a high-dimensional descriptor, our key task is to analyze the similarity in the descriptor space. We shall first introduce the proper measurements for this new space. It has been found that both $L1$ and $L2$ norms are discriminative enough to measure the distance between any two descriptors, representing the corresponding regions. As alternatives, the covariance distance and $\chi 2$ distance are also tested to be good choices for comparing the distributions' similarities.

5.5.2 Feature Detection and Framework Properties

Here, we detail the effectiveness of our feature detection framework together with several of its attractive properties and more results will be shown in Section 5.6. It shall first be emphasized that high sampling rates always lead to dense distribution of the candidate points, thus several neighboring points may have similar diffusion regions, and this will lead to multiple detected results that are in the vicinity of each other. Therefore we empirically reject candidate regions that have more than 50% overlaying with the afore-ranked regions. This strategy can ensure the uniqueness of the detected features as well as the broader coverage of all relevant feature regions, and also make our approach robust to different sampling processes.

We first show a simple feature detection result within the bear model as displayed in Fig. 5.6 (here, mean curvature is chosen as the signal function). The top row shows the features specified with different scales and the second row shows the detected results. It can be observed that the detected similar feature regions are affected by the specification of the feature scope (the selected contour indices here are 2, 3, and 6, respectively). Though small-scale query leads to rather trivial outcomes, the most similar parts are still among the top-ranked results. Furthermore, it is obvious that larger scales can lead to more accurate results thanks to added information. The analogous feature regions defined (proportional to the specified feature) and the corresponding descriptors jointly ensure the accuracy of the detection of the repetitive feature regions.

Our shape feature and its description possess many attractive properties like being concise to store, fast to compute, and efficient to match, etc. Here, we

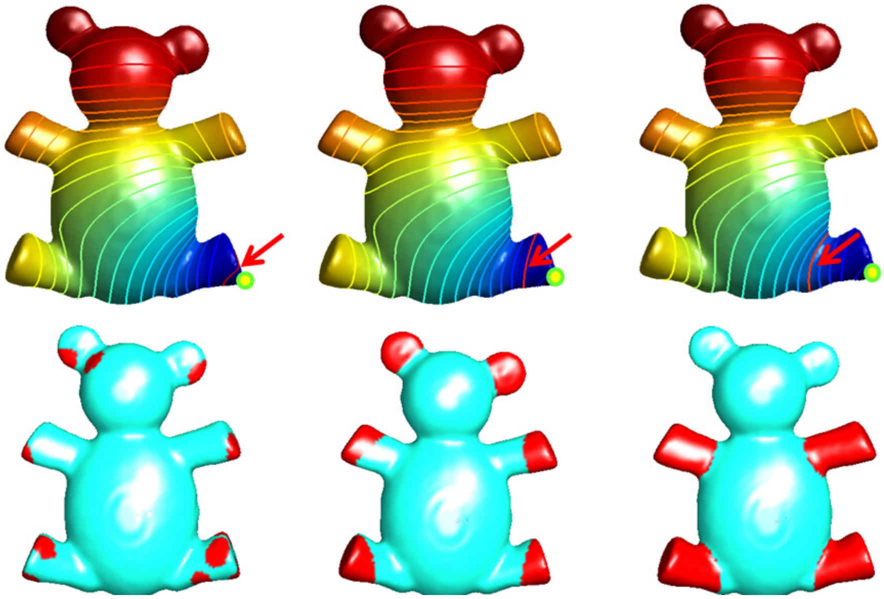


Figure 5.6: Detection of repetitive features with different scales. The first row shows the specified features and the second row shows the detected results within the bear model. Yellow points on query models denote the specified points of interest.

Algorithm 3 Construction of Descriptors over Shapes.

```
Require: User-specified interest point $v_s$ and contour index $n_s$;
Ensure: Descriptors of the specified feature ($D_s$) and all the other candidate regions ($\{D_i\}$);
Conduct the eigen-decomposition on all models;
Compute the $W_f$ of each point using Eq. (5.4.4);
Compute the bi-harmonic distance of $v_s$, denoted as $D_{bh}\{s, \cdot\}$;
Uniformly construct $max(D_{bh}(s, \cdot))/0.05$ contours across the entire model;
With user's specification of the contour number ($n_s$), compute the corresponding value as $S_s$;
For any candidate point $v_i$, compute its bi-harmonic distance $D_{bh}\{s, \cdot\}$, and set its region scope as $S_s * max(D_{bh}\{i, \cdot\})/max(D_{bh}\{s, \cdot\})$;
Reconstruct $S_s/0.025$ contours on the specified feature and all the candidate feature regions;
Compute the SGWs-based and contour-related measurements using Eq. (5.4.9);
Return $D_s$ and $\{D_i\}$.
```

demonstrate two more desirable properties that can facilitate various practical tasks.

Isometry-invariance. We verify the property of isometry-invariance through tests carried on three categories of models in different poses. These models are chosen from the SCAPE and TOSCA databases. As visualized in Fig. 5.7, we deliberately specify features on the human arm containing the elbow, dog leg with elbow and finger with knuckle to validate the property. The models in Fig. 5.7(b) are the deformed ones showing the top-2 most similar feature regions detected on each of them (red and orange highlight the first and second one, respectively). The top-2 detected results are shown here since our approach is capable of identifying similar regions, but distinguishing between symmetric parts is beyond the technical scope of this work. The retrieved results empirically prove that our region-based descriptor is isometry-invariant, which is inherited from the properties of graph wavelets and bi-harmonic distance field.

Robustness to Noise. We add 0.5 (of the mesh's mean edge length) noise to

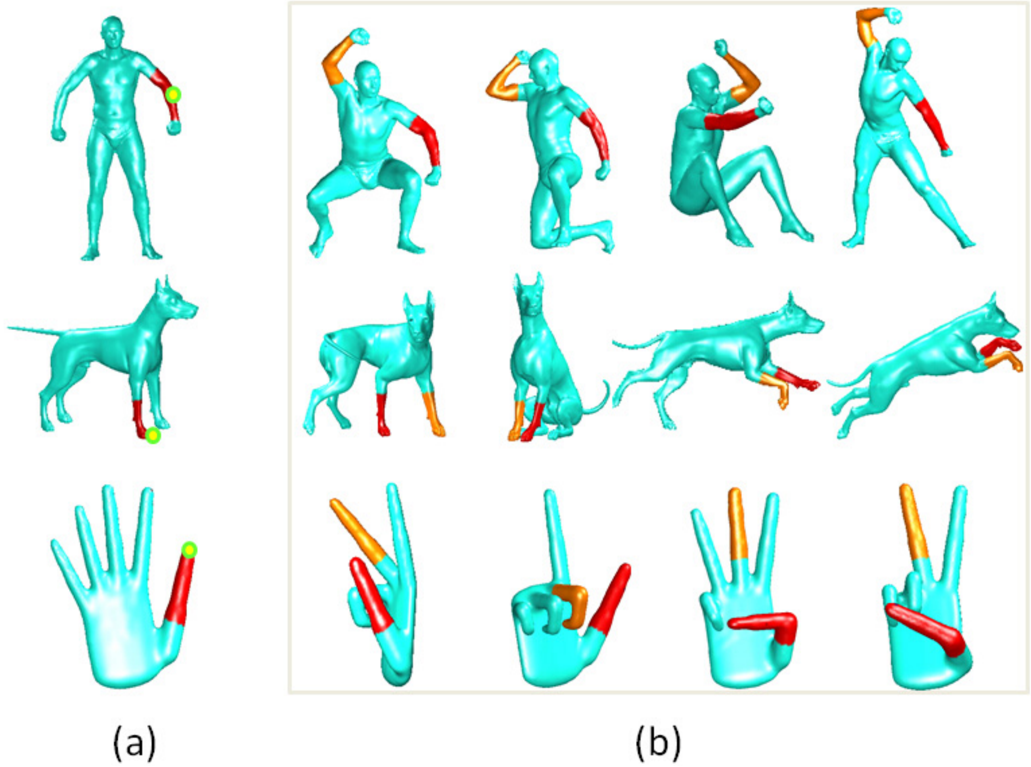


Figure 5.7: Illustration of isometry-invariance. (a) Query. (b) Detected results on deformed models with red and orange denoting the top-2 similar regions.

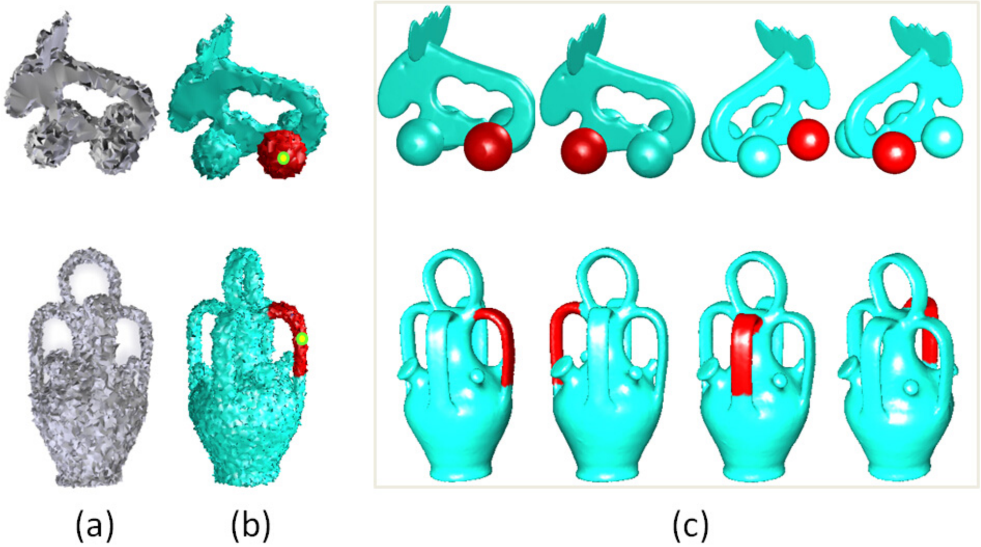


Figure 5.8: Coarse-to-fine recognition on elk and kettle models with 0.5 (mean edge length) noise. (a) Noisy model. (b) Specified features. (c) Recognized regions.

the query models as shown in Fig. 5.8(a) and set the signal as HKS. We selectively enlarge the SGWs-related part of our descriptor to combat noise. The features are specified on these noisy models as shown in Fig. 5.8(b). From the detection results displayed in Fig. 5.8(c), it can be observed that even with large noise, our approach can still recognize the intrinsic geometric characteristics of the features and identify the corresponding similar features, including the ball shape on the elk model and the bended handle of the kettle model from the shape database, thanks to the robustness and stability of bi-harmonic distance field and HKS. The robustness of our approach is of great significance in practical applications as will be demonstrated in Section 5.6.

Table 5.1: Run Time for Constructing Descriptors

Model	# vertices	Timing (min)		
		Eigen	W_f	Total
Dinosaur	14K	0.17	0.14	0.33
Dog	26K	0.30	0.35	0.69
Armadillo	34K	0.38	0.48	0.97
Santa	75K	0.93	1.23	2.32
Dragon	430K	5.51	8.16	14.15

5.6 Experimental Results and Discussions

In this section, we demonstrate the performance of our approach via experiments in various aspects. All the experiments were conducted on a 3.5GHz Intel(R) Core(TM) i7 computer with 16GB memory. We used the cusparse and cublas libraries in CUDA to help reduce the computational time for wavelet transform significantly. For instance, for the SHREC 2007 partial retrieval dataset, in which the average model size is 18K vertices, the whole process of constructing descriptors on one model takes an average of 0.38 minutes, and the whole database costs 114 minutes with 20% sampling rate for each model. More timing details concerning versatile models are shown in Table 5.1.

5.6.1 Parameter Evaluation

There are several parameters in our approach, most of them can be set automatically or empirically set as constants. Two parameters, the sampling rate and the signal function, could be tuned according to specific applications' requirements for better results.

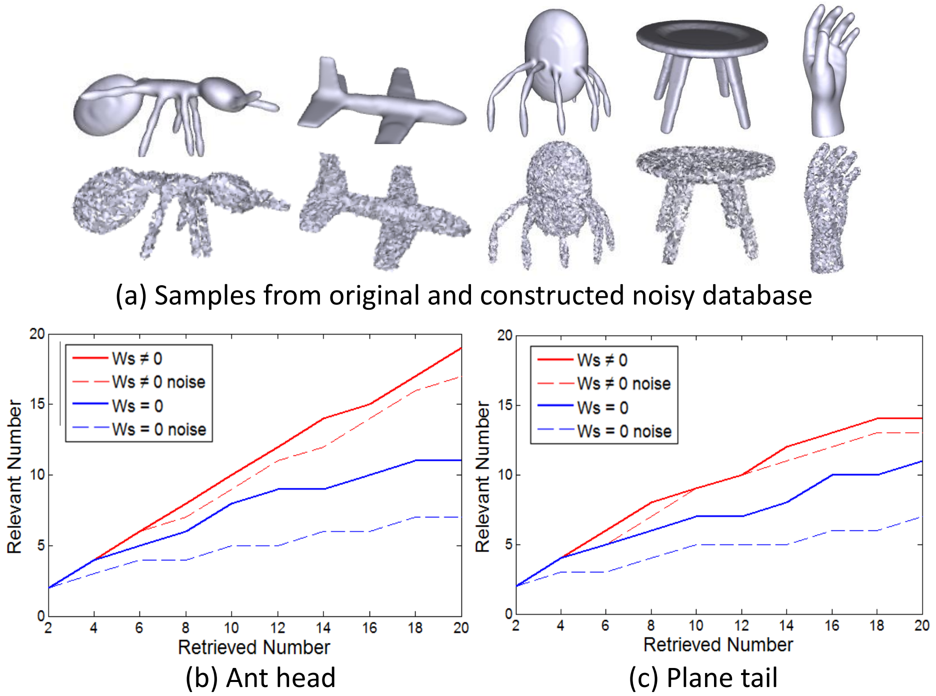
Parameters in the Construction of Descriptors. For the SGWs-related part of the descriptor, we calculate bi-harmonic distances and graph wavelets using the first 300 eigenvalues of each model, and they only need to be computed once. The number of graph wavelets' time scales is set to be 5, which can well represent the frequency information. We empirically set the number of the bins in distance distribution to be 10, and it has been tested to be sufficient to characterize the structure features. The weights ω_s and ω_p are automatically set (based on the mean scale of each part) to balance the three parts that comprise the descriptor. Fig. 5.9 demonstrates the function of graph wavelets by comparing the retrieval results with $ws = 0$ and $w_s \neq 0$ on SHREC database and the corresponding constructed database (note that each model is added with 0.5 mean-edge-length noise as shown in Fig. 5.9(a)). The relevant number is the number of retrieved models that contain partial regions that match the query region (here, ant head and plane tail). It clearly shows that graph wavelets holds the power to characterize the focus regions discriminatively and robustly. Therefore, we selectively enlarge the weight of SGWs-related part (w_s) to combat the disadvantages of the other parts in specific applications.

Sampling Rate. The setting of sampling rate depends on specific applications. Even when the sampling rate is decreased to 5%, our sampling strategy still ensure the inclusion of endpoints. It should be noted that if the application requires high-precision detection, all the vertices of the model could be taken into consideration. For the feature detection framework, the sampling rate of 20% can meet almost all the needs in our experiments.

Signal Selection. The signal function influences SGW-related statistics directly. Therefore, signal selection is among the key problems that should be considered. In principle, the selection depends on specific applications as analyzed in Section 5.4. Furthermore, some applications require the signal to be intrinsic, for which signals like Gaussian curvature, thickness, etc. are expected to perform better.

5.6.2 Repetitive Feature Detection within Certain Model

Repetitive feature detection is of great importance to applications, such as self-symmetry detection [49] and non-local processing propagation [100]. We ran-


(a) Samples from original and constructed noisy database

(b) Ant head

(c) Plane tail

Figure 5.9: Retrieval results (on the SHREC database) with $ws = 0$ (red line) and $w_s \neq 0$ (blue line), and dash lines show the results on noisy models.

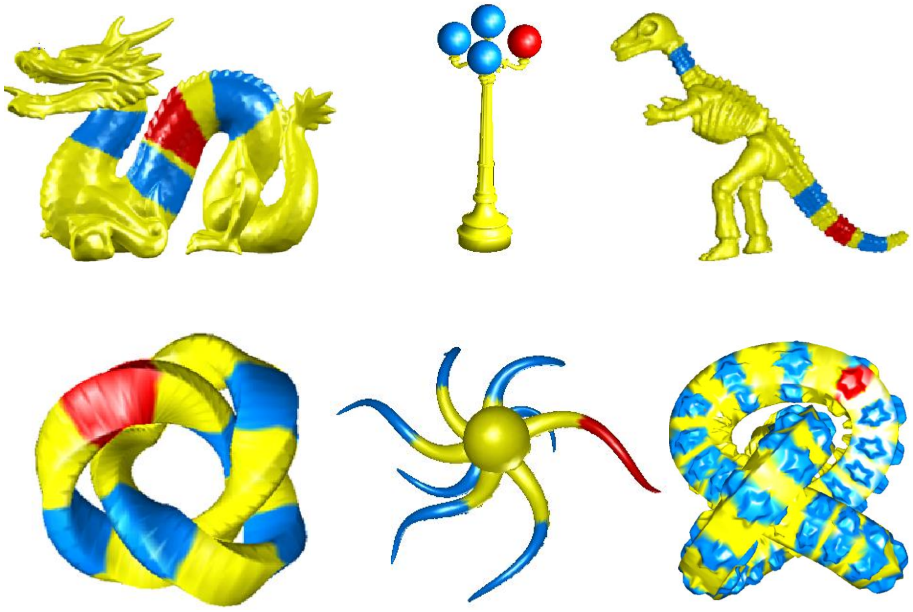


Figure 5.10: Repetitive feature detection within one model. Red denotes query feature and blue denotes some detected results.

domly select some regions of interest to be the shape features as illustrated in Fig. 5.10 (in red). The specified features show that our diffusion-manner demarcation can well cover the interest regions of any kind of shapes if only the interest points and scopes are chosen properly. The detection results show that our descriptor can reliably locate the repetitive feature regions within the dragon model even with large deformations, and it can effectively distinguish between the lumpy local shape of dinosaur's tail from its cylinder-like legs, etc. It very well demonstrates that our SGW-centered descriptor can depict the local details and reflect multi-scale geometric distortions thanks to its gradational construction.

5.6.3 Feature Detection in Database

The performance of our feature detection framework is evaluated on the SHREC 2007 watertight retrieval benchmark, which contains 20 categories and each consists of 20 meshes. In order to demonstrate that our framework is not restricted to the segmentation-based regions like four-legs, we test three kinds of artificial models and each of the query model is unique without any transformed equivalent in the dataset.

We deliberately specify features on kettle, glasses, and chair as shown in Fig. 5.11(a). The slab of the chair is a vivid example, and the existing methods based on segmentation cannot achieve such trans-boundary shapes. It's obvious that the top-4 similar parts in Fig. 5.11(b) resemble the query region very well, which validates that our descriptor based on the continuously distributed contours could characterize the focal region thoroughly. The feature detection process has the ability to correctly identify the most similar parts with equal-proportional scales thanks to the stable characteristic of bi-harmonic distance. By examing which models in the database contain the similar shapes such as kettle base, eyeglasses or slab of chair, our framework can facilitate co-analysis across models and enable search-based shape modeling.

5.6.4 Comparisons and Discussion

Due to the unique technical strategy of our feature description and detection, there is no existing work that is directly comparable with ours. From the perspective of applications, partial matching appears to be the most relevant one. Therefore, we first compare our approach with five popular existing methods in partial matching or shape retrieval based on region description. These methods can all properly fit into our framework and make effective comparisons. They are

- D2 Shape Distribution (D2): statistics of distances between any random pair of points on the region [110].

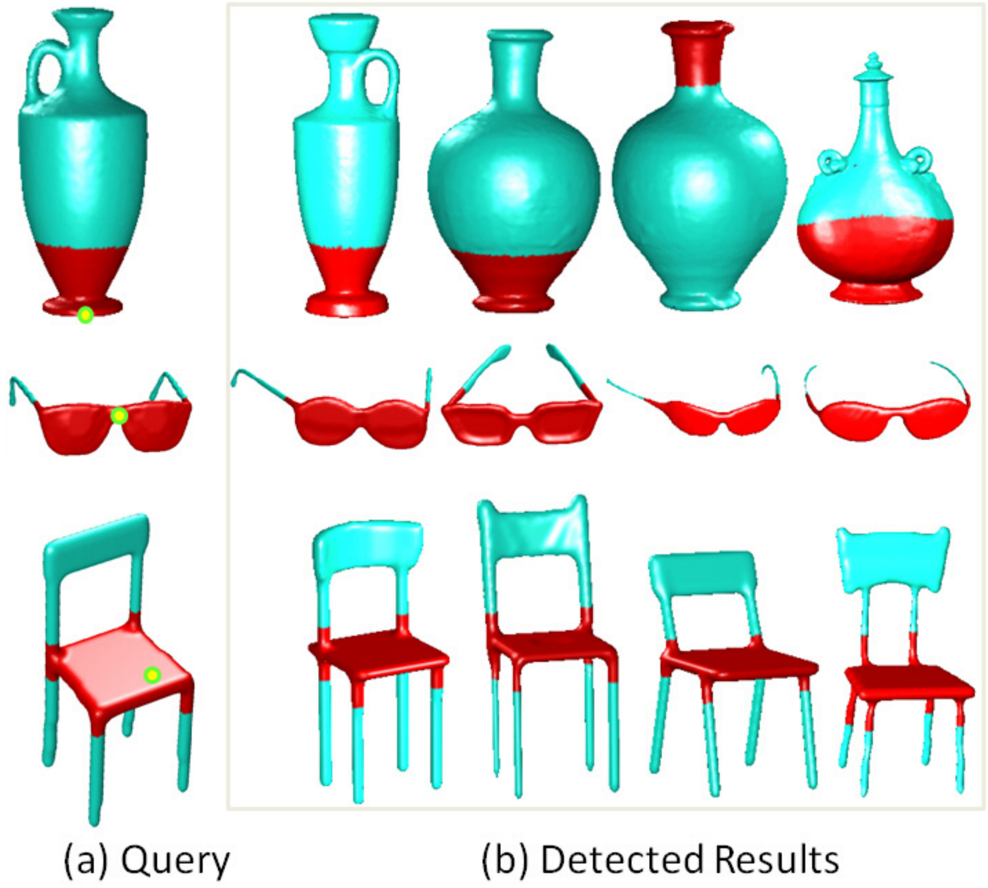


Figure 5.11: Feature detection on the SHREC 2007 watertight retrieval database.

Table 5.2: Precisions of different detection methods.

Method	Ant Head		Plane Tail		Human Leg	
	15	30	15	30	30	60
Ideal	100%	66.7%	100%	66.7%	100%	66.7%
D2	26.7%	30.0%	26.7%	23.3%	40.0%	41.7%
CF	33.3%	30.0%	53.3%	56.7%	46.7%	48.3%
ZM	53.3%	43.3%	53.3%	53.3%	46.7%	50.0%
SDF	46.7%	56.7%	46.7%	50.0%	50.0%	58.3%
PS	80.0%	60.0%	46.7%	43.3%	73.3%	60.0%
Ours	**93.3%**	**63.3%**	**80.0%**	**60.0%**	**86.7%**	**63.3%**

- Conformal Factors (CF): statistics of conformal geometric factors of points on the region [7].

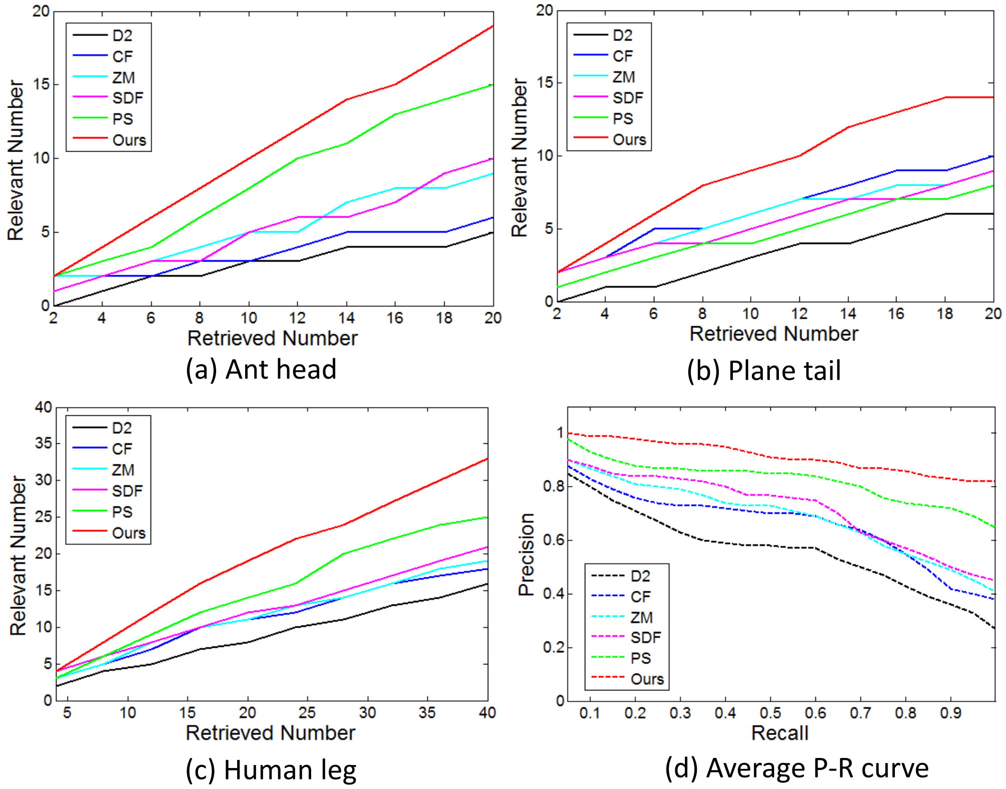
- Zernike Moments based signature (ZM): local shape signature based on transformation of Zernike Moments [100].

- Local SDF Signature (SDF): statistics of points' Shape Diameter Function [137].

- Patch Spectral Geometric Features (PS): normalized spectra of patch spectrum decomposition [63].

The above methods induce the direct geometric measurement (D2), intrinsic curvature-related measurement (CF), heightmap-based measurement (ZM), volume-based measurement (SDF), and spectral-analysis measurement (PS), respectively.

We conduct comparison tests on SHREC 2007 watertight database as well as the McGill database that contains 255 objects divided into ten classes and the intra-class variations consist of non-rigid transforms applied to models. By setting proper parameters in the above methods, effective results are achieved. Fig. 5.12 shows the detection results of all methods concerned. (a)-(c) are the detection



(a) Ant head

(b) Plane tail

(c) Human leg

(d) Average P-R curve

Figure 5.12: Precision plots of different detection methods. (a)-(c) show results of different queries. (d) shows the average P-R graph on SHREC watertight database.

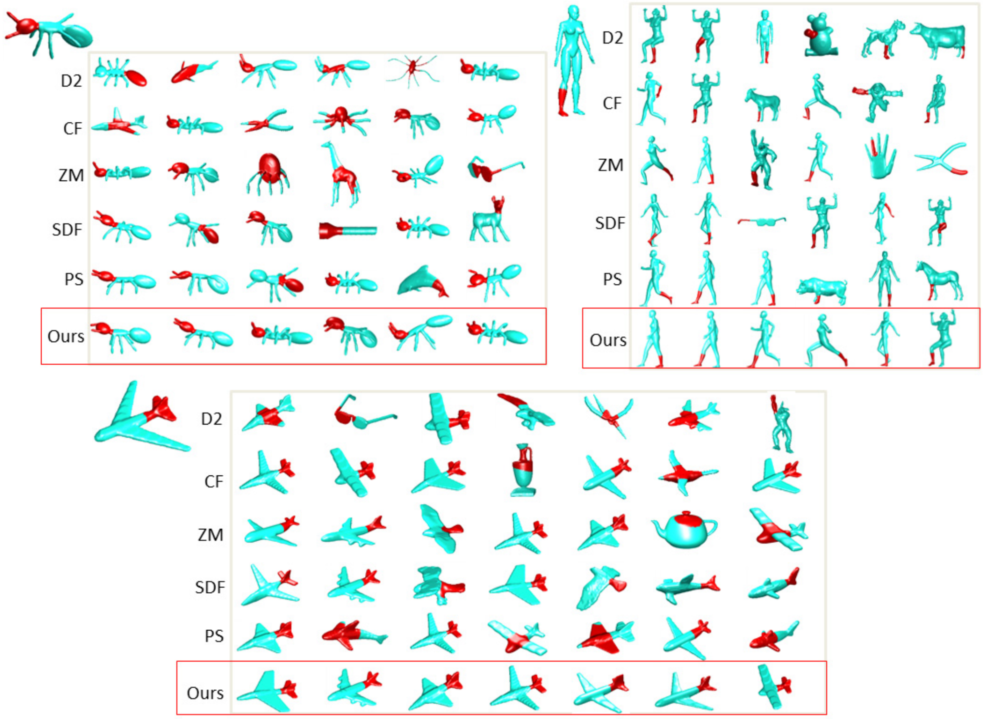


Figure 5.13: Comparisons of different detection methods for partial matching. The queries are ant head, human leg, and plane tail.

results with the queries of ant head, plane tail, and human leg. In the SHREC database, every category contains 20 different models, so we retrieve the top 20 results of ant head and 40 for human leg, since left and right legs of human are documented separately (we call it the multiple-region case). (d) shows the average PR graph based on all queries without multiple regions.

Table 5.2 details the precisions of different detection methods. The precision is computed as the ratio of relevant number to the retrieved number (the second line shows the retrieved number for each query). Table 5.2 and Fig. 5.13 collectively show that each method has its own strength in describing some specific kind of shape. D2 is excellent in describing regular shapes, such as spheres, because the histogram statistics sometimes have the averaging effect on the spatial information, and the ant head with two tentacles makes it difficult for D2 to depict. CF, which integrates the gaussian-curvature knowledge, performs well in characterizing highly-curved region such as the plane tails. ZM is suitable for depicting small patches, as for large feature regions it works better in detecting highly curved ones. SDF takes into account the volume-based information and the region's area ratio knowledge, so it performs well for cylinder-like shape, but not for local complex shapes. PS performs very well in most cases except for the relatively small and complex structure like plane tails due to the instability of regional Laplacian decomposition. In contrast, our method performs stably with high precision in detecting various types of feature regions across different models.

Moreover, we conduct the comparison with two most related works that cannot fit into our feature detection framework, namely, Gal's [49] and Itskovich's works [65]. Since these two works obtain the feature regions by decomposition process and clustering of the pre-divided patches, there inevitably exist over-segmentation phenomenon as shown in the zoomed-in part of Fig. 5.14. In comparison, our method can flexibly specify the interesting feature region and exactly detect the similar regions with the proper scales.

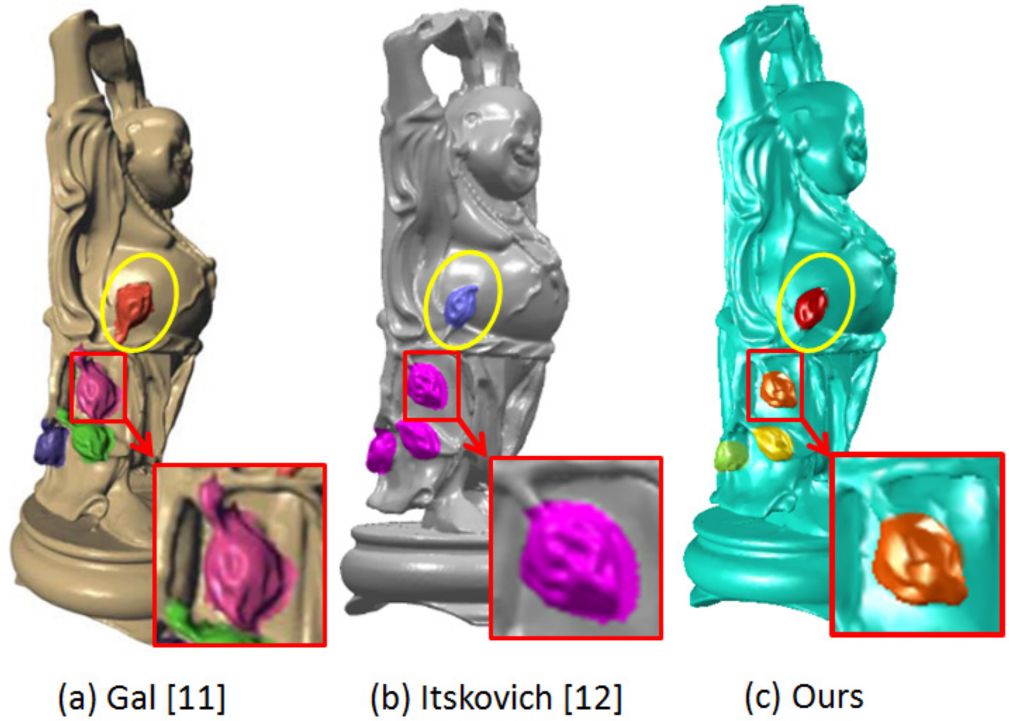

(a) Gal [11] (b) Itskovich [12] (c) Ours

Figure 5.14: Comparison on detection of complex feature regions. Figure (a) and (b) are cited from the corresponding works. On each model, the flower inside the yellow circle is the query region.

5.6.5 Applications

Our feature detection framework enjoys plenty of desirable properties as demonstrated in Section 5.5. They can further facilitate a host of applications as will be shown and analyzed.

Partial Matching and Restoration. We analyze the combined models from SHREC 2007's partial retrieval dataset, which comprises the SHREC 2007's watertight dataset and a query set of 30 models. Each combined model is obtained by merging or removing several subparts of models belonging to the watertight dataset. Existing algorithms concerning partial matching consider the retrieval of the query set models as a big challenge. We demonstrate that our framework provides a powerful tool to restore the combined models, thus can effectively aid the matching and recognition. As the three cases shown in Fig. 5.15, choosing proper interest points (yellow points in (a)) and contour scopes could give rise to diffusion regions that cover large scale of the searched models. After completing the detecting process in the watertight database, the original model can be recognized as shown in (b), and (c) shows another two detected models containing the top-ranked similar feature regions. Moreover, if conventional point-wise correspondence is required for downstream tasks, we can further achieve this goal easily. Since our feature is defined in a diffusion manner, after the similar region is matched, the exact point matching can be obtained by shrinking the diffusion region back to its source point, thus helping the conventional point-based correspondence and other complex partial matching tasks.

Model Recognition. Another immediate application is model recognition based on key components as suggested in [140]. The feature detection framework is much more powerful and does not need the complex process of finding key components as in [140]. Fig. 5.16(a) shows the query models, on which we selectively specify three or four features that are considered to be essential for characterizing the armadillo and horse model. The whole shape similarity is computed as the sum of the distances between the specified feature regions of the query model and the corresponding regions of the target model. Apparently, specifying more feature regions can achieve more precise results. The ranked retrieval results in Fig. 5.16(b) show that our method can correctly retrieve the relevant models from the database even when the models are somewhat incomplete (like the first recognized armadillo model).

Other Potential Applications. Many more practical applications could potentially benefit from the attractive properties of our approach. For example, thanks to the robustness demonstrated in Section 5.5, our framework can serve as the foundation for search-based modeling and coarse-to-fine part replacement that frequently relies upon the conventional denoising processes in the pre-processing stage. The reliability demonstrated in the test of repetitive feature detection shows that our detection results could potentially help with the recognition of repeated

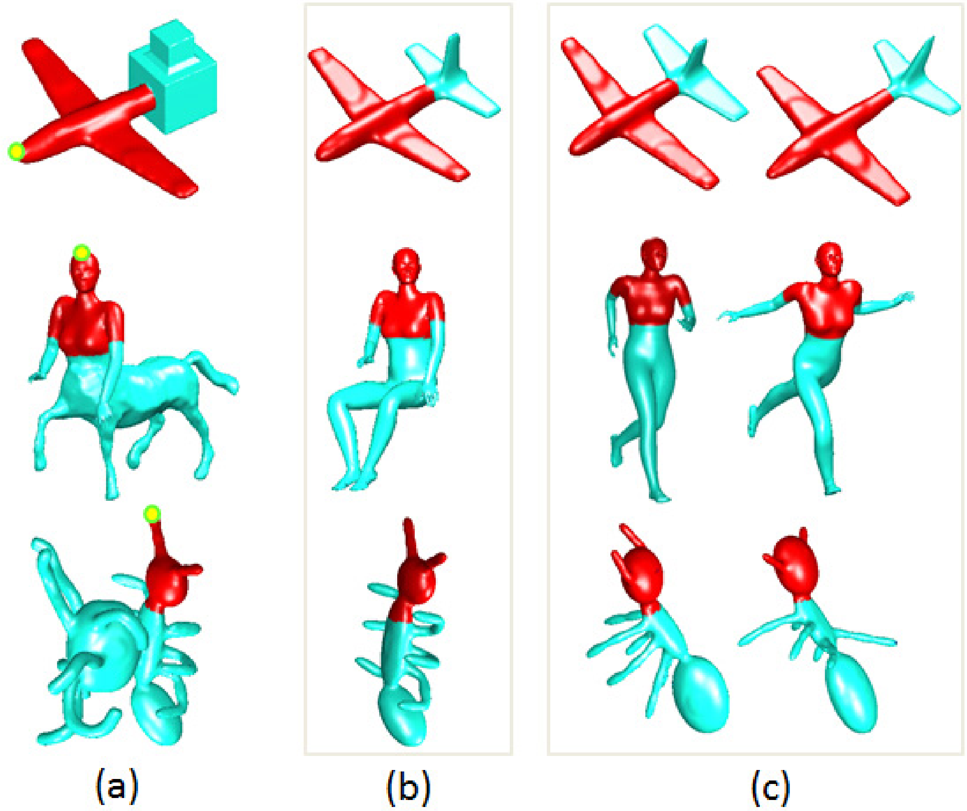


Figure 5.15: Partial matching and restoration on the SHREC's partial query dataset. (a) Combined models with query regions in red. (b) Restored models. (c) Another two models containing the top-ranked similar feature regions.

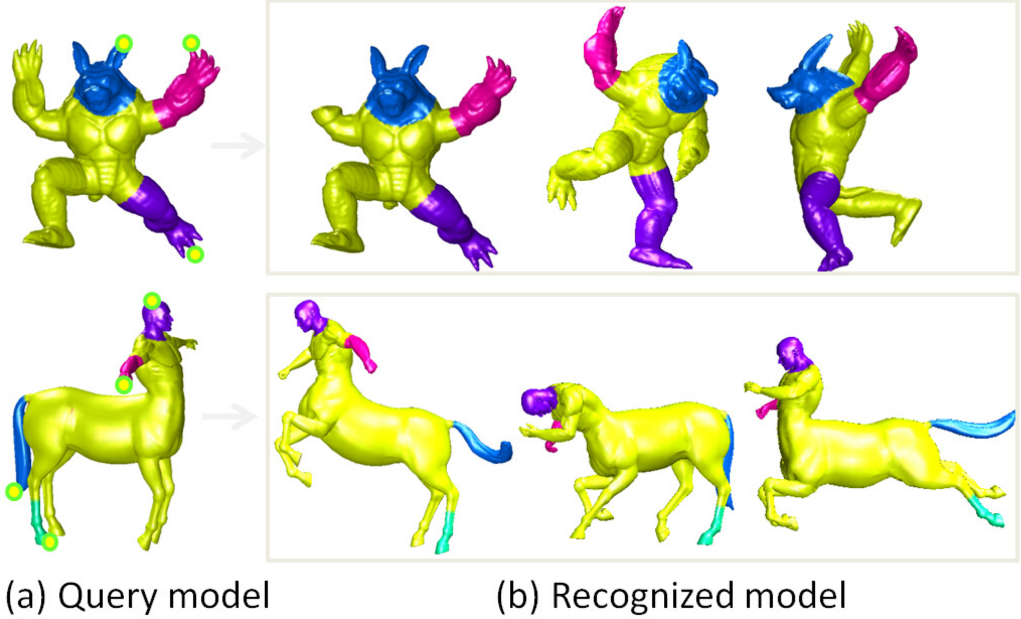


Figure 5.16: Model recognition by identifying multiple components of interest. (a) Query models with 3 or 4 specified features. (b) Top three recognized models.

patterns, cut-and-paste editing, and self-symmetry detection.

5.7 Chapter Summary

In this chapter, we have detailed the description and detection of our proposed generalized local-to-global features on 3D geometric models, which organically couple both local (differential) and global (integral) geometric attributes. The multi-scale and multi-level descriptor based on SGWs has exhibited its potential in depicting any user-specified feature region and distinguishing among descriptor vectors in the corresponding region-wise descriptor space. Furthermore, our novel descriptor is comprising many desirable properties which can facilitate a host of graphics applications, as showcased in our comprehensive experiments. Extensive comparisons with other state-of-the-art techniques/methods have demonstrated certain key advantages of our method in terms of geometry-awareness, reliability, robustness, etc.

One of our future work is to make the user interface and functionality more flexible and intuitive, e.g., allowing users to sketch the boundary of intended feature region. We also plan to to incorporate more powerful analytical tools into the descriptor space for improved performance. Finally, we intend to broaden the application scope of the proposed shape features to support feature-centric registration, structure-driven co-segmentation and high-fidelity model production.

Chapter 6

Feature-Driven Shape Correspondence and Retrieval

As we mentioned in Chapter 2, the eigenvalues and eigenvectors of mesh Laplacian matrix encode valuable intrinsic geometric information of the original mesh. The Laplacian spectrum on its own only affords very limited descriptive power of the global shape. However, the Laplacian eigenpairs can be integrated into more sophisticated spectral representations for highly discriminative and robust local feature descriptors and distance measures, which can facilitate various high-level shape analysis tasks. In this chapter, we present our contributions in feature-driven shape correspondence and retrieval based on robust spectral representations such as the heat kernels.

6.1 High-Order Shape Matching

Signatures and distances induced by the Laplacian eigenpairs are well suited for corresponding points on deformable shapes, since the Laplacian spectra encode intrinsic, global information of the topological and geometric information of the mesh in question. Among the many spectral representations, heat kernel is one of the most commonly used thanks to its robustness and multi-scale properties.

For better feature points correspondence, other than considering the similarity of point heat kernel signatures and pair-wise heat kernel distances, we additionally take into account the compatibility of heat kernels across more than two points by conducting high-order graph matching on the manifold. The heat kernel tensor (HKT) is a high-order potential of geometric compatibility of feature tuples on manifolds. To facilitate the matching process, we further build up a two-level hierarchy via feature clustering. This simple hierarchy greatly reduces the search space of HKT, and therefore the computation time.

Geometric relations among features are extremely important on deformable shapes, and collectively they are much more reliable than single feature point in shape matching. Therefore, we adopt the advanced tensor matching [38], and transplant it to manifolds via a diffusion-driven relation measure, given by

$$d_t(x, y) = \frac{1}{4}(-t \log h_t(x, y))^{1/2}. \tag{6.1.1}$$

Here, $h_t(x, y)$ denotes the heat kernel from point x to y at time t

$$h_t(x, y) = \sum_{l=0}^{\infty} e^{-\lambda_l t} \phi_l(x) \phi_l(y), \tag{6.1.2}$$

where λ_l and ϕ_l are the l-th eigenvalue and eigenfunction of the Laplace-Beltrami operator. When $t \to 0$, $d_t(x, y)$ is indeed a metric and converges to the geodesic between x and y.

We consider two partial shapes M_1 and M_2 with overlaps and boundary changes. Let N_1 be the number of features extracted on M_1, and N_2 be the one on M_2. A pair $i = (i_1, i_2)$ denotes a candidate match with a point i_1 from M_1 and i_2 from M_2. The problem of matching point sets is equivalent to finding an assignment matrix $X_{N_1 \times N_2}$, such that

$$X_{i_1,i_2} = \begin{cases} 1 & i_1 \text{ matches } i_2 \\ 0 & \text{otherwise} \end{cases}, \text{ with } \sum_{i_2} X_{i_1,i_2} \leq 1. \tag{6.1.3}$$

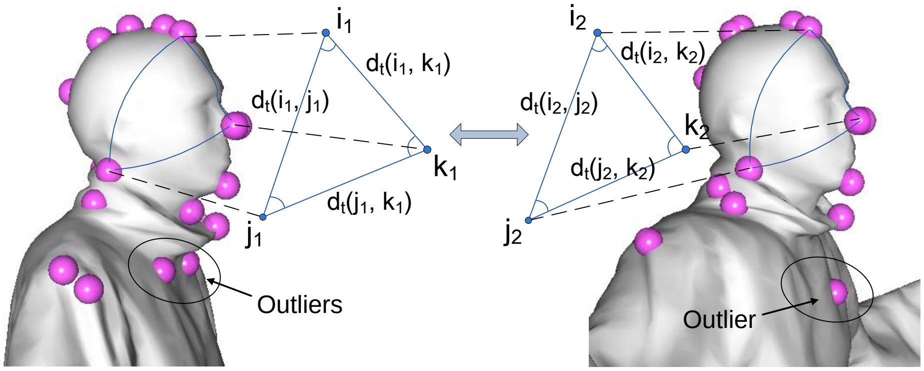


Figure 6.1: HKT for shape matching. Three candidate matches (i, j, k) form two "triangles". Some outlier features are circled.

Note that there may be outliers in the feature set. As shown in Fig. 6.1, some outliers are circled. For an outlier i_1, there is no match in the second feature set, i.e., $\sum_{i_2} X_{i_1,i_2} = 0$. We adopt the tensor formulation [38] for high-order graph matching on manifold. Specifically, we consider a tuple of three candidate matches (i, j, k) without conflicts, i.e., $i_1 \neq j_1 \neq k_1$ and $i_2 \neq j_2 \neq k_2$. They may form two "triangles" by connecting them with d_t, as shown in Fig. 6.1. Since small heat kernels are error-prone, we select large heat kernels with a threshold $\epsilon_h(t) = 10^{-6}$. In the case when the three points do not form a triangle, we simply drop this tuple.

The tuple of candidate matches is then embedded into a 3D space by three angles of this triangle. The distance in the embedded space is given by

$$d_\theta(i, j, k) = \|\theta_{i_1,j_1,k_1} - \theta_{i_2,j_2,k_2}\|_2, \tag{6.1.4}$$

where θ_{i_1,j_1,k_1} is a vector comprising three angles of the triangle formed by points i_1, j_1, k_1, and $\|.\|_2$ denotes the l^2-norm. The affinity of the tuple (i, j, k) without conflicts is defined as

$$\tau_{i,j,k} = e^{-d_\theta(i,j,k)^2/\sigma}, \tag{6.1.5}$$

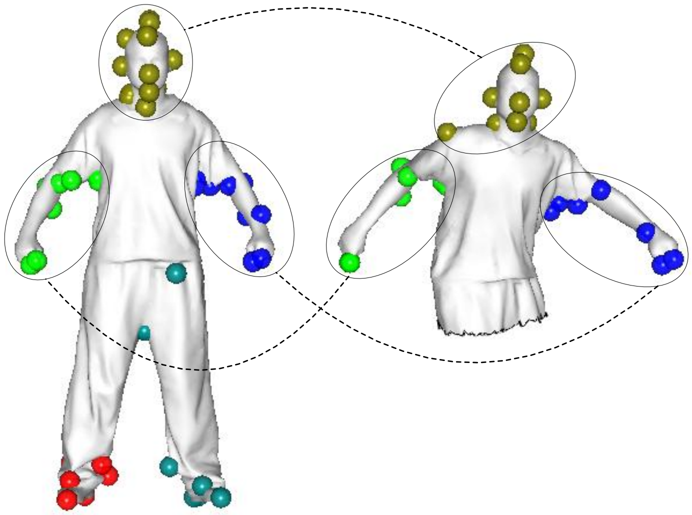


Figure 6.2: Matching hierarchy. Extracted features are clustered into sub-graphs.

where σ is a parameter, which can be set as $\sigma = \mathrm{mean}(d_\theta)$. For tuples with conflicts, we let their affinities equal to zero. The high-order score of assignment X is defined as

$$\mathrm{score}(X) = \sum_{i,j,k} \tau_{i,j,k} X_{i_1,i_2} X_{j_1,j_2} X_{k_1,k_2}. \tag{6.1.6}$$

We rewrite the score using tensor notation, given by

$$\mathrm{score}(X) = T \otimes_1 X \otimes_2 X \otimes_3 X, \tag{6.1.7}$$

where $\otimes_d$ denotes the tensor product in d dimension. We call T the *heat kernel tensor*, as it utilizes heat kernels to form the tensor. The HKT can be fused with different order of potentials. Here, the HKT is a $3rd$-order tensor with entries $\tau_{i,j,k}$ defined in Eq. (6.1.5). The final results are obtained according to their matching scores subject to conflict constraints in Eq. (6.1.3).

For articulated shapes, we design a two-level hierarchy to improve the time performance by reducing the searching space. Articulated shapes with long branches can be easily segmented using some low frequency eigenfunctions of the Laplace-

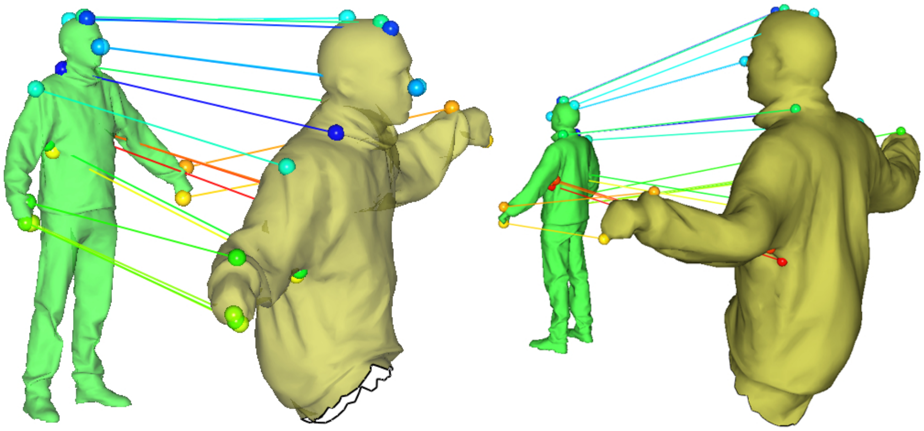


Figure 6.3: Matching deforming shapes with scale changes.

Beltrami operator. In the upper level, we find centers of clusters as the local extrema of the first two non-trivial Laplace-Beltrami eigenfunctions, and remove redundant ones that are very close to selected centers. In the lower level, extracted shape features are then clustered into sub-graphs based on their heat kernels to the cluster centers. The goal of the upper-level matching is to reduce the searching space, and it can be skipped whenever necessary.

The cluster centers comprise a hyper-graph in the upper level of the hierarchy, as shown in Fig. 6.2. In the hyper-graphs with hyper-nodes (cluster centers), we compute their HKT. We release conflicting constraints by allowing candidate matches that have matching scores greater than 80% of the maximal one. This will prune diverse sub-graphs, and reduce the search space of HKT. At the lower level, we run HKT in each cluster. For the high-order optimization in Eq. (6.1.7), we use the tensor power iteration with l^1-norms of columns. The complexity of one power iteration is $O(m)$, where m is the number of non-zero elements in the tensor. We restrict the number of triangles (i.e., non-zero elements) to $64N_1$ by randomly selecting tuples. As a result, the computation of HKT is very efficient.

We conduct various experiments to demonstrate the performance, including scale change (Fig. 6.3), noise/topology change (Fig. 6.4)), and large deformation (Fig. 6.5).

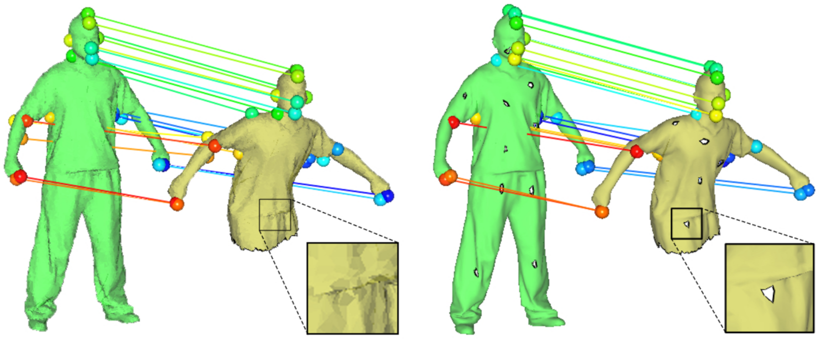


Figure 6.4: Matching with noise (*Left*) and topology changes (*Right*).

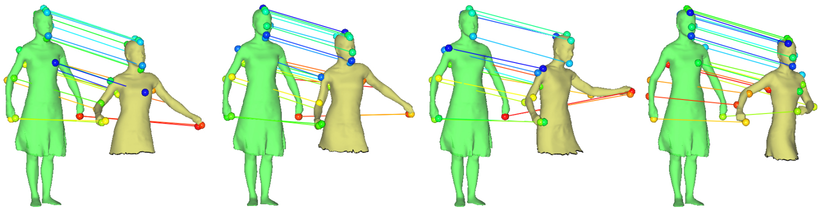


Figure 6.5: Selected frames of matching objects with large deformations.

6.2 Hierarchical Shape Registration

We improve the registration algorithm in [61] by implementing a hierarchical corresponding process. The central idea is to generate correspondences in multiple levels in a coarse-to-fine manner, with additional features incrementally inserted in each level. The registration starts from the coarsest resolution. Registration results obtained in one level serve as references for the registration in the next level. We adopt the heat kernel coordinates for local shape parameterization, giving rise to a complete solution capable of registering partial shapes undergoing isometric deformation with higher accuracy.

A common and effective approach to dense correspondence is first matching a small number of pre-selected feature points, and then using the matched features as references for dense correspondences. Features, encoding important information of shapes, can be used to parameterize the shapes and serve as anchors to bootstrap the matching of the rest points. In general, it is necessary to have a

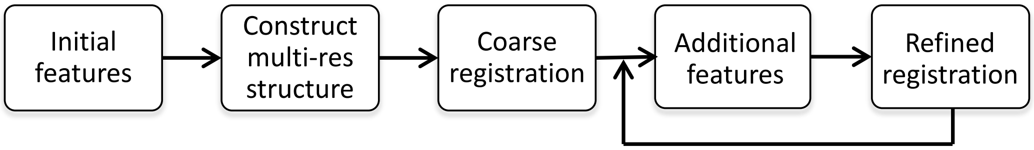


Figure 6.6: Pipeline overview of our hierarchical registration framework.

fairly large number of matched features to obtain dense correspondence of good quality. Otherwise, the feature-based parameterization may have difficulty in discriminating nearby elements, especially in parts of shapes that are far away from any features. However, automatically finding and matching a large number of features is very difficult and error-prone. Even in the case of user-assisted feature matching, one would prefer a small set of matched features, since manually corresponding many features is burdensome and time-consuming.

As illustrated in Figure 6.6, the main steps of our method are: (1) Detect and match features to get a small initial set of feature matches; (2) Construct hierarchical structures of input shapes; (3) Perform registration at the coarsest level using the initial feature set; (4) Select some newly registered points as additional features; (5) Perform registration at the next level using results from the previous level and the expanded set of feature references; (6) Repeat step (4) and (5) until all valid points are registered.

The rationale of our approach is that distinguishing elements that are distant from each other on the surface is much more accurate than nearby elements. Even with a small number of features, we can achieve very good registration on a heavily downsampled version of the original shapes. The registration result of a coarse resolution can serve as seed correspondences when performing registration in a finer level. The large number of available seeds significantly reduce the chances of correspondences being trapped in an incorrect location. Moreover, the multiresolution process enables us to pick additional features from already registered points. This greatly enhances the discriminative strength as the meshes become more refined.

Hierarchical Registration. Given a source shape S and a target shape T,

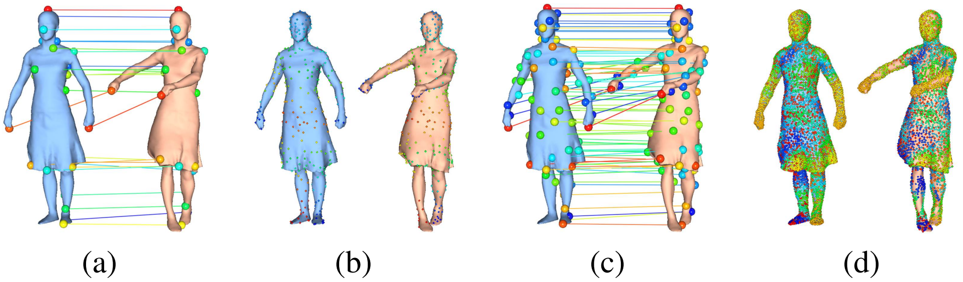


Figure 6.7: Major steps of our hierarchical registration algorithm. The blue shape is the source and the red one is the target. We use a three level hierarchy in this example. (a) Initial feature correspondences; (b) Coarse registration result (Third level); (c) Expanded feature correspondences (Third level); (d) Final registration result.

both represented as triangular meshes, and let $V^S = \{s_i\}$ and $V^T = \{t_i\}$ be their respective vertex sets, the objective of dense registration is to find an optimal mapping $\tau : V^S \rightarrow V^T$. In practice, we represent the registration results as a set of correspondences $R = \{v_i^S, v_i^T\}$. When the shapes in question are not complete, some vertices in V^S may not have correspondences in R.

Initial Feature Detection and Matching. The goal of this step is to obtain a small feature correspondence set C^*. In this work, we adopt the heat kernel signature (HKS) [146] to extract multi-scale features and spectral graph matching method [81] to match them.

Multiresolution Structure. Once we obtained features correspondence set C^*, we can use it as reference to propagate the correspondences by searching in the vicinity of already matched vertices, until every source vertex is mapped to a vertex in the target shape [61, 64]. However, when the size of C^* is small, simple propagation approaches often cannot produce satisfactory registration. On one hand, with insufficient features as anchors, it is difficult to distinguish nearby vertices no matter what kind of parameterization scheme we employ. On the other hand, since the sources for propagation are few, wrong correspondences are more likely to accumulate following a mismatch.

To address this issue, instead of computing registration in a single run, we perform it hierarchically in a coarse-to-fine manner. We construct a multi-resolution structure of the original shapes, and in each level we only register vertices that belong to the current resolution. Given a triangular mesh $M_0 = (V_0, F_0)$ and constants $d, m \in \mathbb{Z}$, we downsample M and obtain the mesh hierarchy $\{M_0, M_1, \ldots, M_m\}$. Assume $M_i = (V_i, F_i)$ and $n_i = |V_i|$, we enforce that $n_{i+1} = n_i/d$. We adopt the method in [51] for mesh downsampling. In our implementation, we select $d = 4$.

Correspondence Propagation and Feature Expansion. Let both the initial correspondence set R_{m+1} and initial feature set C_{m+1} be C^*. In level l, we input the previous level's registration result R_{l+1} and feature set C_{l+1}. The goal is to find the l-th level correspondence set R_l that registers meshes S_l and T_l, with an augmented feature set C_l.

For each vertex x in S_l and T_l, we compute its heat kernel coordinates

$$\text{HKC}(x) = (h_t(x, c_1), \ldots, h_t(x, c_z)), c_i \in C_{l+1}. \tag{6.2.1}$$

Inheriting R_{l+1} as the initial correspondence set, we propagate correspondence to match the rest vertices in S_l and T_l. We use a heap to determine the order by which the vertices in S_l are processed, prioritizing on the magnitude of HKC. For an already matched pair (s_j, t_j) and one of s_j's immediate neighbor s_i, we search for s_i's best correspondence $t_i \in V_k^T$ in the neighborhood of t_j, and add (s_i, t_i) into the correspondence set. t_i is selected using the following criterion

$$t_i = \operatorname*{arg\,min}_{t \in n(t_j, T_k)} \|\text{HKC}_S(s_i) - \text{HKC}_T(t)\|_2 \tag{6.2.2}$$

where $n(t_j, T_k)$ represent the set of t_j's neighboring vertices in T_k, and HKC_S and HKC_T denote the heat kernel coordinates of points on S and T.

The correspondence propagation continues until all vertices in S_k have been matched and we get the correspondence set $\{(s_i, t_i)\} \subset V_{m-1}^S \times V_{m-1}^T$. Note that for each correspondence (s_i, t_i), the endpoint t_i actually represent a set of vertices $K(t_i)$ in the original mesh T_0. To find the precise correspondence of s_i in the original target mesh, we search $K(t_i)$ and replace t_i with $t_j \in K(t_i)$ if t_j is closer

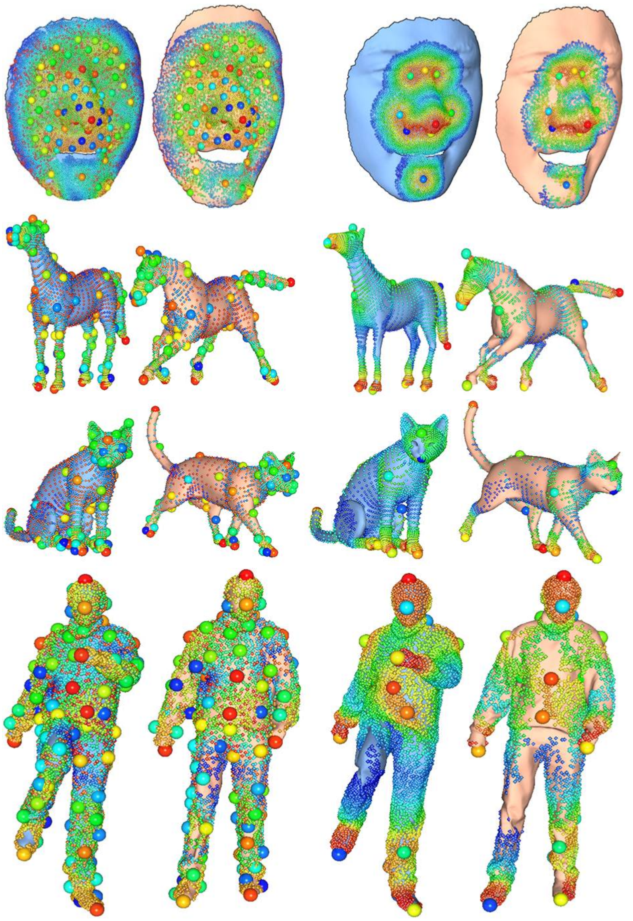


Figure 6.8: Comparison of registration results using our multi-resolution method (*Left*) and the single-resolution method [61] (*Right*). Large colored dots represent matched features.

to s_i in the embedding space. The result is the l-th level correspondence set R_k that relates points $s_i \in S_k$ to $t_i \in T_0$. For each correspondence (s_i, t_j), we assign a matching score

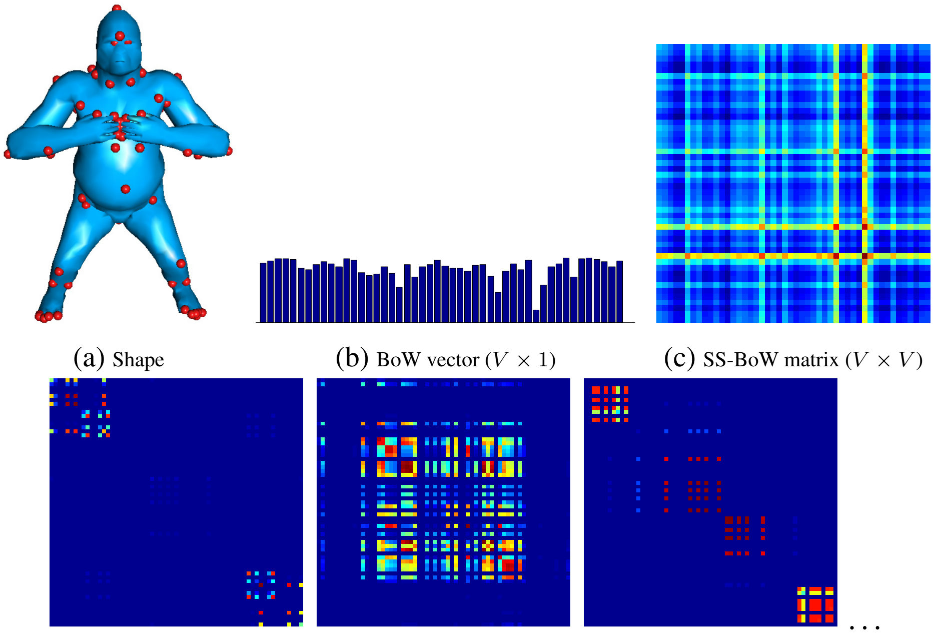
$$\text{score}(s_i, t_j) = \exp(-\|\text{HKC}_S(s_i) - \text{HKC}_T(t_j)\|_2). \tag{6.2.3}$$

We then select from R_l some vertex pairs as new features and insert them into the feature set. These new added feature pairs should be both reliable (having great matching score) and not in the δ-neighborhood of any existing feature points. The expanded feature set C_l enables a more discriminative HKC in the next level. We carry on this process from the coarsest level to the finest level until we obtain the final registration set R_0 between the original meshes S_0 and T_0. Fig. 6.7 shows the major steps of our algorithm.

6.3 Bag-of-Feature-Graphs Shape Retrieval

To effectively characterize a shape for retrieval, only using the distribution of word frequency as in traditional bag-of-words method is not enough. We need to incorporate the spatial relations between features into our representation, and the most effective way to encode pairwise relation is through graph-based feature aggregation method, such as spatially-sensitive bags-of-words (SS-BoW) [14] or relevance diffusion [48].

We present a new paradigm, called bag-of-feature-graphs (BoFG), for non-rigid shape retrieval. The basic idea is to represent a shape by constructing graphs among its features, which significantly reduces the number of points involved in computation. Given a vocabulary of geometric words, for each word the BoFG builds a graph that records spatial information of features, weighted by their similarities to this word. This eliminates unlikely points in a word category, during shape comparison. Feature graphs are governed by their affinity matrices of weighted heat kernels, whose eigenvalues form a concise shape descriptor. Given



(a) Shape

(b) BoW vector ($V \times 1$)

(c) SS-BoW matrix ($V \times V$)

(d) BoFG matrices ($|F| \times |F| \times V$)

Figure 6.9: Different representations of a given shape.

a vocabulary of geometric words, corresponding to each word we build a graph that records spatial information between features, weighted by their similarities to this word. Specific characteristics of the BoFG include:

- It is concise by significantly reducing the number of points involved in representation, and thus, is fast to compute.
- It explicitly records spatial information among features.
- It is representative, since features are salient points containing important information of the shape.
- Graphs have different dominating features associated with corresponding words. This greatly improves the accuracy in shape comparison by eliminating unlikely word-distributions.

We first revisit the ShapeGoogle method originally introduced in [113], which requires a point descriptor $K(x)$ and a distance measure $\gamma(x, y)$ defined on surface $\mathcal{M}$. Let $W = \{W_1, \ldots, W_V\}$ be a vocabulary of geometric words with size $V = |W|$. The words $\{W_i\}$ are representative vectors in the descriptor space clustered by the k-means algorithm. For each point x, the ShapeGoogle method computes its word distribution $\Theta(x) = [\theta_1(x), \ldots, \theta_V(x)]^T$. The similarity between x and word W_i is given by

$$\theta_i(x) = c(x)e^{-\frac{\|K(x)-W_i\|^2}{2\sigma^2}}, \tag{6.3.1}$$

where σ is a parameter, and $c(x)$ is the normalization factor selected such that $\|\theta(x)\|_1 = 1$. The BoW descriptor of $\mathcal{M}$ is computed by integrating word similarities over the entire shape

$$f(\mathcal{M}) = \int_{\mathcal{M}} \Theta(x)d\mu(x), \tag{6.3.2}$$

where $\mu(x)$ denotes the surface area of x. As shown in Fig. 6.9, the BoW descriptor is a $|W| \times 1$ vector that measures the frequencies of words appearing on the shape.

The ShapeGoogle algorithm also introduced the Spatially-Sensitive Bag-of-Words (SS-BoW) descriptor, given by

$$F(\mathcal{M}) = \int_{\mathcal{M}\times\mathcal{M}} \Theta(x)\Theta^T(y)\gamma(x, y)d\mu(x)d\mu(y). \tag{6.3.3}$$

As shown in Fig. 6.9, $F(\mathcal{M})$ is a $|W| \times |W|$ matrix which measures frequencies of word pairs modulated by their spatial relations. The BoW needs to evaluate the word distributions for all points in the query shape, while the SS-BoW needs to comptue all word-to-word distributions for all pair of points.

In practice, heat kernel signature (HKS) or scale-invariant heat kernel signature (SI-HKS) is often employed as point descriptors thanks to the properties of robustness and isometry-invariance. To make comparison more efficient, the ShapeGoogle algorithm highly suppresses the geometric information by computing the frequencies of words or word-pairs on the shape. It ends up with concise descriptors for comparison, yet completely loses spatial information. Besides, the shape-google algorithms are time-consuming, since they are working on all the points in the data.

To reduce the complexity of ShapeGoogle, it is desirable to reduce the number of points involved in representing the shape. A straightforward solution is to select feature points, which keep most information of the shape geometry. Because of the multi-scale property, HKS features contain geometry information ranging from points in small scales to the entire shape in large scales. However, one concern is that a reduced number of points may not be sufficient to faithfully represent the shape. Therefore, instead of counting word frequencies, we construct graphs on detected features, giving rise to a bag-of-feature-graphs (BoFG) paradigm. The graphs encode spatial relations between features, which contain much more geometry information in representing the shape.

We adopt weighted heat kernel matrices to capture global structures of graphs. Specifically, for a shape M with feature set F, only points $x \in F$ are involved in computing word distributions $\Theta(x)$, which significantly reduces computation. Features are vector-quantized by a fuzzy classification, which assigns $\theta_i(x)$ portion of similarity to word W_i in the distribution of feature x. The distribution $\Theta(x)$ is computed by Eq. (6.3.1) with σ set as $\frac{1}{4}$ of the average distance of words in the vocabulary. This fuzzy classification reduces ambiguities in graph comparison, and also avoids misclassification in a hard quantization. For a geometric word W_i, we construct a matrix G_i, whose entry $G_i(x, y)$ with $(x, y) \in F \times F$ is computed by

$$G_i(x, y) = \theta_i(x)\theta_i(y)h_t(x, y). \tag{6.3.4}$$

G_i the heat kernel between x and y weighted by their similarities to the geometric word W_i.

The matrix set $\mathbf{G}(M) = \{G_1, \ldots, G_V\}$ comprises a BoFG representation of the shape M. As shown in the bottom row of Fig. 6.9, matrices characterize spa-

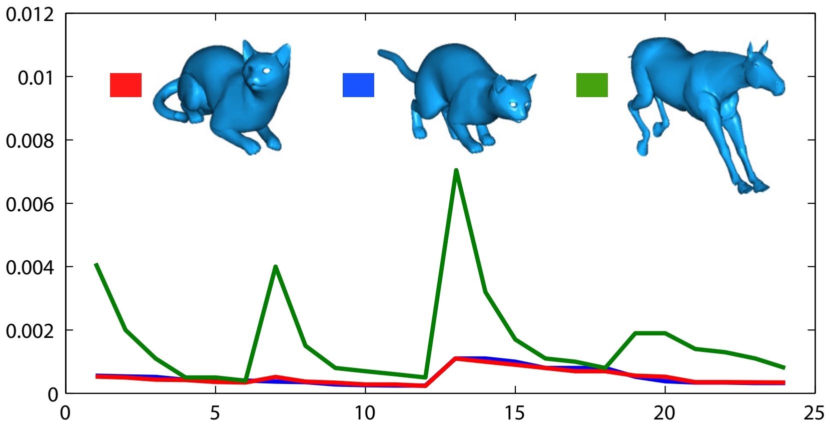


Figure 6.10: Some non-rigid shapes and their BoFG descriptors.

tial information of features assigned to different word categories. The near-zero entries in a matrix indicate they are hardly classified to this category, and therefore, not considered in this graph. It contains all the geometric information of features in a multi-scale way, which faithfully characterizes the shape. The computation complexity for this matrix representation is O($|\mathrm{F}|^2$D), as the computed heat kernels can be shared by all matrices. Considering that the size of feature set is always much fewer than the total number of points on the shape, the BoFG is much faster than ShapeGoogle.

The mechanism of shape retrieval is to build concise BoFG descriptors of shape models in a database in an off-line process, and retrieve related shapes for a query by the approximate nearest neighbor (ANN) search. The BoFG descriptor is constructed by concatenating the most significant eigenvalues of each BoFG matrix. Each G_i is a real symmetric matrix, whose eigenvalues are all real and eigenvectors are orthogonal to each other. We choose its six largest eigenvalues, denoted as $S_i(M)$, which contributes to a $6V \times 1$ vector $[S_1(M), \ldots, S_V(M)]^T$ as a concise descriptor. This reduces the dimension of the matrix by multi-dimensional scaling (MDS) [15]. Fig. 6.10 shows some non-rigid shapes and their BoFG descriptors. The deformed cat-models have very similar BoFG descriptors, while the horse-model has a quite different one. It projects the matrix to its main direc-

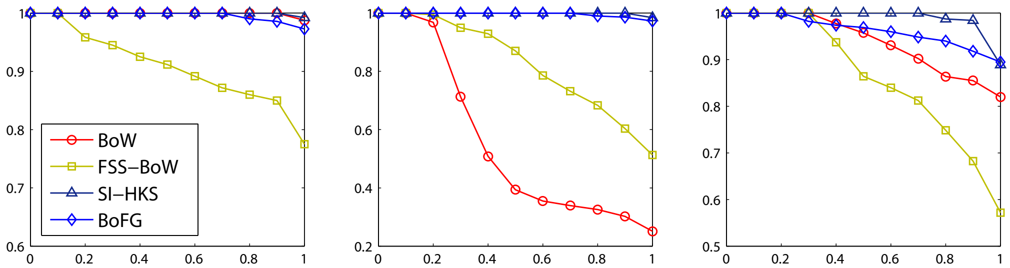


Figure 6.11: Precision-recall curves of evaluated methods, with categories of (from *Left* to *Right*) null, scale change, and hole.

tions in the subspace spanned by the eigenvectors corresponding to $S_i(M)$, which are stable to a small amount of outliers. Then, we define the similarity distance between two models M_1 and M_2 as

$$d(M_1, M_2) = \sum_{i=1}^{V} \|S_i(M_1) - S_i(M_2)\|_2. \tag{6.3.5}$$

The above distance is based on one-scale heat kernels, which can be easily extended to multi-scale by averaging distances of heat kernels at different values of t.

To test the methods under some challenging cases, we apply transformations to the query shapes. This leads to categorized experiments, including null (no transformation), scale change (scaling vertex coordinates), and hole (topological change and missing information). For comparison purpose, we also evaluate some state-of-the-art methods that are similar to ours, including the BoW shape google, the SS-BoW ShapeGoogle, and the SI-HKS. Since the SS-BoW runs extremely slow, we only consider feature points in its implementation, denoted as FSS-BoW.

The methods are quantitatively evaluated by the *precision-recall* (PR) curve that is often adopted for evaluating retrieval performance [47]. It plots the trade-off between precision (ratio of the number of relevant shapes retrieved and the total number of shapes retrieved) and recall (ratio of the number of relevant shapes retrieved and the total number of existing relevant shapes that could be ideally retrieved). Fig. 6.11 shows the PR curves of evaluated methods, with categories of null, scale change, and hole. The BoFG has competitive results comparing with some state-of-the-art methods.

6.4 Chapter Summary

In this chapter, we present several algorithms for feature-driven shape correspondence and retrieval. We adopt heat kernel tensor for high-order graph matching of shape features and propose a hierarchical coarse-to-fine registration method to achieve better dense correspondence. We also propose a novel paradigm for shape retrieval called the bag-of-feature-graph (BoFG). In BoFG, both the similarity between features and words in vocabulary and the spatial relations among features are explicitly encoded. By only considering feature points, the required computation is significantly reduced comparing with traditional spatially-sensitive shape retrieval methods.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This dissertation presents our research work on shape modeling and analysis based on spectral representations and sparsity-driven algorithms. Our proposed methods provide solutions to a wide range of shape modeling problems with competitive performance, demonstrating the great potential of spectral methods and sparse representations in the mesh domain.

We present an innovative approach to 3D mesh compression by incorporating redundant spectral graph wavelets in dictionary design and using greedy pursuits to find compressed coefficient approximation.

By formulating surface inpainting as a sparse signal recovery problem, we propose a novel variational approach for surface inpainting, integrating data fidelity constraints on the shape domain with coefficient sparsity constraints on the transformed domain.

We propose a novel informative shape descriptor that combines SGW coefficients and contour-based statistics for better characterization of feature shape regions. Based on this descriptor, we develop a generalized feature detection framework, which can integrate different types of regional descriptors to facilitate widespread applications including partial matching and model recognition.

We develop an effective feature-driven shape correspondence and retrieval algorithms. For coarse correspondence, we adopt tensor-based high-order graph matching to maximizes the geometric compatibility between features tuples; and for dense matching, we present a hierarchical shape registration algorithm, generating correspondence in multiples levels in a coarse-to-fine manner. We also propose the Bag-of-Feature-Graph (BoFG) descriptor for shape retrieval, which significantly reduce the number of points required for computing distributions, comparing with traditional Bag-of-Features descriptors.

7.2 Future Work

There are many possible extensions and improvements related to our current research work in spectral representations and sparse modeling. Here we introduce some potential topics.

Shape Editing

Classical Laplacian mesh editing [12] can be written as a regularized least square problem. Taking the x-coordinate for example, soft constraint Laplacian mesh editing constructs the new coordinate function by solving the following optimization problem

$$\min_{\mathbf{x}'} \sum_{j \in C} (x'_j - c_j)^2 + \gamma \|L\mathbf{x}' - L\mathbf{x}\|_2^2, \tag{7.2.1}$$

where L denotes Laplacian operator, C represents the set of handle vertices, c_j denotes the coordinate of the handle vertex indexed at j, and $\mathbf{x}'$ and $\mathbf{x}$ represent the original and deformed coordinates, respectively. This formulation ensures that the deformed shape will preserve the second-order differential geometric information of the original shape while conforming to the constraints of new handle positions.

Let function $\mathbf{f} = \mathbf{x}' - \mathbf{x}$ and denote the values of $\mathbf{f}$ at handle vertices as $d_j = c_j - x_j, j \in C$. The optimization problem can be rewritten as

$$\min_{\mathbf{f}} \sum_{j \in C} (f_j - d_j)^2 + \gamma \mathbf{f}^T L^2 \mathbf{f}. \tag{7.2.2}$$

Expanding $\mathbf{f}$ w.r.t. the Laplacian eigenbasis $\mathbf{\Phi} = \{\phi_{\mathbf{k}}\}$ as $\mathbf{f} = \sum_k a_k \phi_{\mathbf{k}}$, Eq. 7.2.2 becomes

$$\min_{\mathbf{f}} \sum_{j \in C} (f_j - d_j)^2 + \gamma \sum_k \lambda_k^2 a_k^2, \tag{7.2.3}$$

where $\{\lambda_k\}$ denote the Laplacian eigenvalues.

Analyzing Eq. 7.2.3, we can see that the formulation of Laplace editing penalizing high frequency components (corresponding to larger λ_k) [133].

Replacing the bi-Laplacian in Eq. 7.2.2 with a spectral graph wavelet kernel W affords flexible design choices regarding the dominating frequencies. Suppose $W = \mathbf{\Phi} g(t\Lambda) \mathbf{\Phi}^{\mathbf{T}}$, where $g(\cdot)$ is the spectral domain generator function. Instead of preserving the differential or Laplacian coordinates, we may choose to preserve wavelet coefficients of different scales after deformation. The spectral expansion of the optimization problem then becomes

$$\min_{\mathbf{f}} \sum_{j \in C} (f_j - d_j)^2 + \gamma \sum_k g(t\lambda_k) a_k^2. \tag{7.2.4}$$

Clearly, changing the generator function $g(\cdot)$ or the scale parameter t will influence the contribution of components of different frequencies in the final reconstruction.

Sparse Representation Learning

Our current research focuses on *reconstructive* sparse models with predefined dictionaries composed of analytical functions such as wavelets. Nevertheless, sparse representations also have great potential as *discriminative* models for pattern recognition, as most meaningful high-dimensional signals probably reside in some low-dimensional subspace which can more better revealed by proper sparse models.

To construct a highly discriminative sparse models, predefined dictionaries are usually not sufficiently expressive. To improve the sparsity and discriminative power, a dictionary learned from the data set is much preferred. Given training signals $\{y_i\}_{i=1}^{N}$, a dictionary that minimize the overall coefficient sparsity of the training signals can be formulated as follows:

$$\hat{D} = \arg\min_{D} \sum_{i=1}^{N} \min_{x_i} \{\|Dx_i - y_i\|_2^2 + \lambda \|x_i\|_1\}. \tag{7.2.5}$$

The dictionary learning can be handled by algorithms such as K-SVD [1].

After the dictionary D is learned, for an input signal y, a sparse coefficient representation $\hat{x}$ can be easily computed with respect D using common l_1 formulation:

$$\hat{x} = \arg\min_{x} \{\|Dx - y\|_2^2 + \lambda \|x\|_1\}. \tag{7.2.6}$$

For shape signals, we cannot directly apply the dictionary learning formulation in Eq. 7.2.5, since different models generally have vastly different graph structures and cannot be easily mapped. Nevertheless, we can adopt or devise appropriate feature descriptors as a proxy instead of encoding shape signals directly. A highly discriminative sparse model in the descriptor space should have great potential for tasks such as classification and retrieval.

General Graph Signal Analysis

In this dissertation, our research work concentrates on graph spectral analysis in the context of shape modeling. Nonetheless, discrete mesh is only a special type of graph structure. For modeling various types of tangible or intangible networks, such as transportation networks and social networks, graph representation is the natural choice. It is also a common practice to infer graph structures from data sets without pre-defined networks to help organize the data in a specific way. Even for data with existing graph structure, such as 2D images, it is sometimes helpful to cluster the original data points into some super nodes and create an alternative graph topology. Graph is therefore a very generic data structure as it allows flexible representations of relations among data points.

In recent years, with the explosive increase of data sets generated from Internet and devices with sensors, there bas been a growing interest towards adapting and applying signal processing techniques for analyzing and modeling graph signals [135,139]. Currently, there is very few existing works in literature employing sparse and redundant modeling methods for graph signal, and it is a promising direction to extend our research in spectral representations and sparsity-driven algorithms for the analysis of general graph signals.

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, Nov 2006.

[2] J.-P. Antoine, D. Roşca, and P. Vandergheynst. Wavelet transform on manifolds: Old and new approaches. *Applied and Computational Harmonic Analysis*, 28(2):189–202, 2010. Special Issue on Continuous Wavelet Transform in Memory of Jean Morlet, Part I.

[3] M. Attene, M. Campen, and L. Kobbelt. Polygon mesh repairing: An application perspective. *ACM Computing Surveys (CSUR)*, 45(2):15, 2013.

[4] A. Bac, N.-V. Tran, and M. Daniel. A multistep approach to restoration of locally undersampled meshes. In *Proceedings of the 5th International Conference on Advances in Geometric Modeling and Processing*, GMP'08, pages 272–289, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] M. Belkin, J. Sun, and Y. Wang. Discrete laplace operator on meshed surfaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, SCG '08, pages 278–287, New York, NY, USA, 2008. ACM.

[6] M. Ben-Chen and C. Gotsman. On the optimality of spectral compression of mesh data. *ACM Transactions on Graphics (TOG)*, 24(1):60–80, 2005.

[7] M. Ben-Chen and C. Gotsman. Characterizing shape using conformal factors. In *Proceedings of the 1st Eurographics Conference on 3D Object Retrieval*, pages 1–8, 2008.

[8] M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In *Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000)*, VISUALIZATION '00, Washington, DC, USA, 2000. IEEE Computer Society.

[9] S. Biasotti and S. Marini. Sub-part correspondence using structure and geometry. In *Eurographics Italian Chapter Conference*, pages 23–28, 2006.

[10] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[11] G.-P. Bonneau. Optimal triangular haar bases for spherical data. In *Proceedings of the conference on Visualization '99: celebrating ten years*, VIS '99, pages 279–284, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[12] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on*, 14(1):213–230, 2008.

[13] J. Branch, F. Prieto, and P. Boulanger. Automatic hole-filling of triangular meshes using local radial basis function. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 727–734. IEEE, 2006.

[14] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)*, 30(1):1, 2011.

[15] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences of the United States of America*, 103(5):1168–1172, 2006.

[16] M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1704 –1711, june 2010.

[17] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

[18] J.-F. Cai, R. H. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, 24(2):131–149, 2008.

[19] E. Candes and J. Romberg. l1-magic: Recovery of sparse signals via convex programming. *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, 4:14, 2005.

[20] E. J. Candes, D. L. Donoho, et al. *Curvelets: A surprisingly effective nonadaptive representation for objects with edges*. DTIC Document, 1999.

[21] E. J. Candes, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[22] E. J. Candès and D. L. Donoho. Ridgelets: a key to higher-dimensional intermittency? *Phil. Trans. R. Soc. Lond. A*, pages 2495–2509, 1999.

[23] V. Caselles, G. Haro, G. Sapiro, and J. Verdera. On geometric variational models for inpainting surface holes. *Computer Vision and Image Understanding*, 111(3):351–373, 2008.

[24] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, 2008.

[25] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on*, pages 3869–3872. IEEE, 2008.

[26] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[27] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design*, 21(5):427–445, 2004.

[28] J. Claude Iehl and B. Péoche. An adaptive spectral rendering with a perceptual control. *Computer Graphics Forum*, 19(3):291–300, 2000.

[29] R. R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53 – 94, 2006.

[30] G. Craciun, M. Jiang, D. Thompson, and R. Machiraju. Spatial domain wavelet design for feature preservation in computational data sets. *IEEE Transactions on Visualization and Computer Graphics*, 11:149–159, 2005.

[31] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[32] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[33] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of*

the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[34] M. N. Do and M. Vetterli. The contourlet transform: an efficient directional multireso-lution image representation. *IEEE Transactions on Image Processing*, 14(12):2091–2106, 2005.

[35] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. *ACM Transactions on Graphics (TOG)*, 25(3):1057–1066, 2006.

[36] D. L. Donoho. For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829, 2006.

[37] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 58(2):1094–1121, 2012.

[38] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2383 –2395, dec. 2011.

[39] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[40] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005.

[41] J. Fadili and J.-L. Starck. Curvelets and ridgelets. In *Computational Complexity*, pages 754–773. Springer, 2012.

[42] M.-J. Fadili and J.-L. Starck. Em algorithm for sparse representation-based image inpainting. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–61. IEEE, 2005.

[43] M.-J. Fadili, J.-L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. *The Computer Journal*, 52(1):64–79, 2009.

[44] M. Figueiredo. Bayesian image segmentation using wavelet-based priors. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 437 – 443 vol. 1, june 2005.

[45] M. Fischler, R. Bolles, A. Varshney, and R. Farias. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[46] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. *ACM Transactions on Graphics*, 23(3):652–663, 2004.

[47] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. *ACM Transactions on Graphics (TOG)*, 22(1):83–105, 2003.

[48] T. Furuya and R. Ohbuchi. Diffusion-on-manifold aggregation of local features for shape-based 3d model retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, pages 171–178, New York, NY, USA, 2015. ACM.

[49] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):130–150, 2006.

[50] R. Gal, A. Shamir, T. Hassner, M. Pauly, and D. Cohen-Or. Surface reconstruction using local shape priors. In *Symposium on Geometry Processing*, number EPFL-CONF-149318, pages 253–262, 2007.

[51] M. Garland. *Quadric-Based Polygonal Surface Simplification*. PhD thesis, Carnegie Mellon University, 1999.

[52] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka. Curvature maps for local shape comparison. In *Shape Modeling and Applications*, pages 244–253, 2005.

[53] M. Gavish, B. Nadler, and R. R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *Proceedings of the 27th International Conference on Machine Learning*, pages 367–374, 2010.

[54] G. H. Golub, P. C. Hansen, and D. P. O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

[55] J. Gomes and L. Velho. From fourier analysis to wavelets. Course Notes - SIGGRAPH, 1999.

[56] O. G. Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part i: theory. *Image Processing, IEEE Transactions on*, 15(3):539–554, 2006.

[57] O. G. Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms. *Image Processing, IEEE Transactions on*, 15(3):555–571, 2006.

[58] S. Gumhold and W. Straßer. Real time compression of triangle mesh connectivity. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 133–140, New York, NY, USA, 1998. ACM.

[59] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[60] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-Theory and Methods*, 6(9):813–827, 1977.

[61] T. Hou and H. Qin. Robust dense registration of partial nonrigid shapes. *TVCG*, 99, 2011.

[62] T. Hou and H. Qin. Admissible diffusion wavelets and their applications in space-frequency processing. *Visualization and Computer Graphics, IEEE Transactions on*, 19(1):3–15, 2013.

[63] J. Hu and J. Hua. Salient spectral geometric features for shape matching and retrieval. *The Visual Computer*, 25(5-7):667–675, 2009.

[64] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-rigid registration under isometric deformations. *Computer Graphics Forum*, 27(5):1449–1457, 2008.

[65] A. Itskovich and A. Tal. Surface partial matching and application to archaeology. *Computers & Graphics*, 35(2):334–341, 2011.

[66] L. Jiang, X. Zhang, and G. Zhang. Partial shape matching of 3D models based on the laplace-beltrami operator eigenfunction. *Journal of Multimedia*, 8(6):655–661, 2013.

[67] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286. ACM Press/Addison-Wesley Publishing Co., 2000.

[68] Z. Karni and C. Gotsman. 3d mesh compression using fixed spectral bases. In *Graphics Interface*, volume 1, pages 1–8, 2001.

[69] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[70] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *ACM Symposium on Geometry Processing*, pages 156–164, 2003.

[71] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l 1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):606–617, 2007.

[72] W. H. Kim, D. Pachauri, C. Hatt, M. K. Chung, S. Johnson, and V. Singh. Wavelet based multi-scale shape features on arbitrary surfaces for cortical thickness discrimination. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1241–1249. Curran Associates, Inc., 2012.

[73] W. H. Kim, V. Singh, M. K. Chung, C. Hinrichsa, and D. Pachauri. Multi-resolutional shape features via non-euclidean wavelets: Applications to statistical analysis of cortical thickness. *NeuroImage*, 93(1):107 – 123, 2014.

[74] O. Kin-Chung Au, Y. Zheng, M. Chen, P. Xu, and C.-L. Tai. Mesh segmentation with concavity-aware fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1125–1134, 2012.

[75] I. Kokkinos, M. Bronstein, R. Litman, and A. Bronstein. Intrinsic shape context descriptors for deformable shapes. In *IEEE Computer Vision and Pattern Recognition*, pages 159–166, 2012.

[76] V. Kraevoy and A. Sheffer. Template-based mesh completion. In *Symposium on Geometry Processing*, pages 13–22, 2005.

[77] V. Kreavoy, D. Julius, and A. Sheffer. Model composition from interchangeable components. In *Pacific Conference on Computer Graphics and Applications*, pages 129–138, 2007.

[78] Y. Lamdan and H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of ICCV*, pages 238–249, 1988.

[79] G. Lavoué. Bag of words and local spectral descriptor for 3D partial shape retrieval. In *Eurographics Conference on 3D Object Retrieval*, pages 41–48, 2011.

[80] G. Lavoué. Combination of bag-of-words descriptors for robust partial shape retrieval. *The Visual Computer*, 28(9):931–942, 2012.

[81] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV'05*, 2005.

[82] C. Lessig and E. Fiume. Soho: Orthogonal and symmetric haar wavelets on the sphere. *ACM Trans. Graph.*, 27(1):4:1–4:11, Mar. 2008.

[83] B. Lévy. Laplace-beltrami eigenfunctions towards an algorithm that ”understands” geometry. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, page 13, june 2006.

[84] C. Li and A. Ben Hamza. A multiresolution descriptor for deformable 3D shape retrieval. *The Visual Computer*, 29(6-8):513–524, 2013.

[85] X. Li, Z. Yin, L. Wei, S. Wan, W. Yu, and M. Li. Symmetry and template guided completion of damaged skulls. *Computers & Graphics*, 35(4):885–893, 2011.

[86] Z. Li, D. S. Meek, and D. J. Walton. Polynomial blending in a mesh hole-filling application. *Computer-Aided Design*, 42(4):340–349, 2010.

[87] P. Liepa. Filling holes in meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 200–205. Eurographics Association, 2003.

[88] Y. Lipman, R. M. Rustamov, and T. A. Funkhouser. Biharmonic distance. *ACM Transactions on Graphics*, 29(3):1–11, 2010.

[89] L. Lippert, T. Hochschule, P. F. D. Zurigo, M. H. Gross, and M. H. Gross. Fast wavelet based volume rendering by accumulation of transparent texture maps, 1995.

[90] R. Liu and H. Zhang. Mesh segmentation via spectral embedding and contour analysis. volume 26, pages 385–394. Blackwell Publishing Ltd, 2007.

[91] Y. Liu, H. Zha, and H. Qin. Shape topics: A compact representation and new algorithms for 3D partial shape retrieval. In *IEEE Computer Vision and Pattern Recognition*, pages 2025–2032, 2006.

[92] Z. Liu, J. Mitani, Y. Fukui, and S. Nishihara. A 3d shape retrieval method based on continuous spherical wavelet transform. In *International Conference on Computer Graphics and Imaging*, pages 21–26, 2007.

[93] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, Jan. 1997.

[94] A. M. Bronstein and M. M. Bronstein. Regularized partial matching of rigid shapes. In *European Conference on Computer Vision*, pages 143–154, 2008.

[95] M. Maggioni, J. C. Bremer, R. R. Coifman, and A. D. Szlam. Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs. In M. Papadakis, A. F. Laine, and M. A. Unser, editors, *Proceedings of the SPIE*, volume 5914, 2005.

[96] S. Mahadevan. Adaptive mesh compression in 3d computer graphics using multiscale manifold learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 585–592, New York, NY, USA, 2007. ACM.

[97] S. Mahadevan and M. Maggioni. Value function approximation with diffusion wavelets and laplacian eigenfunctions. In *NIPS*, 2005.

[98] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2008.

[99] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[100] A. Maximo, R. Patro, A. Varshney, and R. Farias. A robust and rotationally invariant local surface descriptor with applications to non-local mesh processing. *Graph. Models*, 73(5):231–242, 2011.

[101] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003.

[102] C. Moenning and N. A. Dodgson. Fast marching farthest point sampling for point clouds and implicit surfaces. Technical report, Cambridge University, 2003.

[103] M. Mortara, G. Patané, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for the multi-scale analysis and decomposition of triangle-meshes. *Algorithmica*, 38(1):227–248, 2003.

[104] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Com-*

puter graphics and interactive techniques in Australasia and Southeast Asia, pages 381–389. ACM, 2006.

[105] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[106] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):310–316, 2010.

[107] H. T.-M. Ngo and W.-S. Lee. Feature-first hole filling strategy for 3d meshes. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, pages 53–68. Springer, 2013.

[108] G. M. Nielson, I.-H. Jung, and J. Sung. Haar wavelets over triangular domains with applications to multiresolution models for flow over a sphere. In *Proceedings of the 8th conference on Visualization '97*, VIS '97, pages 143–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.

[109] T. Ogawa and M. Haseyama. Missing image data reconstruction based on adaptive inverse projection via sparse representation. *Multimedia, IEEE Transactions on*, 13(5):974–992, 2011.

[110] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.

[111] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389–403, 2000.

[112] R. S. Overbeck, C. Donner, and R. Ramamoorthi. Adaptive wavelet rendering. *ACM Trans. Graph.*, 28(5):140:1–140:12, Dec. 2009.

[113] M. Ovsjanikov, A. Bronstein, M. Bronstein, and L. Guibas. Shape google: a computer vision approach to isometry invariant shape retrieval. In *IEEE International Conference on Computer Vision Workshops (ICCV)*, pages 320–327, Sept 2009.

[114] M. Ovsjanikov, Q. Mrigot, F. Mmoli, and L. Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564, 2010.

[115] S. Park, X. Guo, H. Shin, and H. Qin. Shape and appearance repair for incomplete point surfaces. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1260–1267. IEEE, 2005.

[116] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[117] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, number EPFL-CONF-149337, pages 23–32, 2005.

[118] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas. Discovering structural regularity in 3d geometry. *ACM Trans. Graph.*, 27(3):43:1–43:11, Aug. 2008.

[119] F. Payan and M. Antonini. Temporal wavelet-based compression for 3d animated models. *Computers & Graphics*, 31(1):77–88, Jan. 2007.

[120] J.-P. Pernot, G. Moraru, and P. Véron. Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. *Computers & Graphics*, 30(6):892–902, 2006.

[121] R. Pfeifle and H.-P. Seidel. Triangular b-splines for blending and filling of polygonal holes. In *Proceedings of the conference on Graphics interface'96*, pages 186–193. Canadian Information Processing Society, 1996.

[122] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics*, 2(1):15–36, 1993.

[123] M. D. Plumbley. Geometry and homotopy for l1 sparse representations. *Proceedings of SPARS*, 5:206–213, 2005.

[124] J. Pokrass, A. M. Bronstein, and M. M. Bronstein. Partial shape matching without point-wise correspondence. *Numerical Mathematics*, 6(1):223–244, 2013.

[125] I. Ram, M. Elad, and I. Cohen. Redundant wavelets on graphs and high dimensional data clouds. *Signal Processing Letters, IEEE*, 19(5):291–294, 2012.

[126] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado. Subset selection in noise based on diversity measure minimization. *Signal Processing, IEEE Transactions on*, 51(3):760–770, 2003.

[127] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers and Graphics*, 33(3):381 – 390, 2009.

[128] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-beltrami spectra as 'shape-dna' of surfaces and solids. *Computer-Aided Design*, 27(4):342 – 366, 2008. Symposium on Solid and Physical Modeling 2005.

[129] G. Rong, Y. Cao, and X. Guo. Spectral mesh deformation. *The Visual Computer*, 24(7-9):787–796, 2008.

[130] S. Rosenberg. *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifold*. Cambridge University Press, 1997.

[131] J. Rossignac. Edgebreaker: connectivity compression for triangle meshes. *Visualization and Computer Graphics, IEEE Transactions on*, 5(1):47–61, Jan 1999.

[132] R. M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Fifth Eurographics Symposium on Geometry Processing*, pages 225–234, July 2007.

[133] R. M. Rustamov. On mesh editing, manifold learning, and diffusion wavelets. In *Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII*, pages 307–321, Berlin, Heidelberg, 2009. Springer-Verlag.

[134] R. M. Rustamov. Multiscale biharmonic kernels. *Computer Graphics Forum*, 30(5):1521–1531, 2011.

[135] A. Sandryhaila and J. M. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *Signal Processing Magazine, IEEE*, 31(5):80–90, 2014.

[136] P. Schröder and W. Sweldens. Spherical wavelets: efficiently representing functions on the sphere. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 161–172, New York, NY, USA, 1995. ACM.

[137] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang. Contextual part analogies in 3D objects. *International Journal of Computer Vision*, 89(2-3):309–326, 2010.

[138] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 878–887. ACM, 2004.

[139] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *Signal Processing Magazine, IEEE*, 30(3):83–98, 2013.

[140] I. Sipiran and B. Bustos. Key-component detection on 3d meshes using local features. In *Proceedings of the 5th Eurographics Conference on 3D Object Retrieval*, pages 25–32, 2012.

[141] A. Skodras, C. Christopoulos, and T. Ebrahimi. The jpeg 2000 still image compression standard. *Signal Processing Magazine, IEEE*, 18(5):36–58, 2001.

[142] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin. Mesh saliency via spectral processing. *ACM Trans. Graph.*, 33(1):1–17, 2014.

[143] O. Sorkine. Laplacian mesh processing. In *Eurographics State-of-the Art Report*, 2005.

[144] O. Sorkine and D. Cohen-Or. Least-squares meshes. In *Shape Modeling Applications, 2004. Proceedings*, pages 191–199. IEEE, 2004.

[145] J.-L. Starck, F. Murtagh, and J. M. Fadili. *Sparse image and signal processing: wavelets, curvelets, morphological diversity*. Cambridge University Press, 2010.

[146] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5), 2009.

[147] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 351–358, New York, NY, USA, 1995. ACM.

[148] R. Toldo, U. Castellani, and A. Fusiello. Visual vocabulary signature for 3D object retrieval and partial matching. In *Proceedings of the 2nd Eurographics Conference on 3D Object Retrieval*, pages 21–28, 2009.

[149] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, 2004.

[150] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[151] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.

[152] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[153] S. Valette and P. Prost. Wavelet-based multiresolution analysis of irregular surface meshes. *Visualization and Computer Graphics, IEEE Transactions on*, 10(2):113 –122, march-april 2004.

[154] B. Vallet and B. Lévy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum*, 27(2):251–260, 2008.

[155] G. K. Wallace. The jpeg still picture compression standard. *Consumer Electronics, IEEE Transactions on*, 38(1):18–34, 1992.

[156] J. Wang and M. M. Oliveira. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image and Vision Computing*, 25(1):103–113, 2007.

[157] X. Wang, X. Liu, L. Lu, B. Li, J. Cao, B. Yin, and X. Shi. Automatic hole-filling of cad models with feature-preserving. *Computers & Graphics*, 36(2):101–110, 2012.

[158] L. Wei, W. Yu, M. Li, and X. Li. Skull assembly and completion using template-based surface matching. In *3DIMPVT: The First Joint 3DIM/3DPVT Conference, 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 413–420, 2011.

[159] N. Werghi and Y. Xiao. Wavelet moments for recognizing human body posture from 3D scans. In *International Conference on Pattern Recognition*, pages 319–322, 2002.

[160] G. Xu. Convergent discrete laplace-beltrami operators over triangular surfaces. In *Geometric Modeling and Processing, 2004. Proceedings*, pages 195 – 204, 2004.

[161] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *IEEE Computer Vision and Pattern Recognition*, pages 373–380, 2009.

[162] S. Zelinka and M. Garland. Similarity-based surface modelling using geodesic fans. In *Eurographics Symposium on Geometry Processing*, pages 209–218, 2004.

[163] H. Zhang, O. Van Kaick, and R. Dyer. Spectral mesh processing. *Computer Graphics Forum*, 29(6):1865–1894, 2010.

[164] K. Zhang, W. Yu, M. Manhein, W. Waggenspack, and X. Li. 3d fragment reassembly using integrated template guidance and fracture-region matching. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2138–2146, 2015.

[165] W. Zhao, S. Gao, and H. Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007.

[166] K. Zhou, J. Synder, B. Guo, and H.-Y. Shum. Iso-charts: stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the*

2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pages 45–54. ACM, 2004.

[167] X. Zhu and M. Rabbat. Graph spectral compressed sensing for sensor networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2865–2868. IEEE, 2012.