Feature Encoding of Spectral Descriptors for 3D Shape Recognition

Majid Masoumi

A Thesis
in
The Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy (Information and Systems Engineering) at
Concordia University
Montreal, Quebec, Canada

April 2017

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Majid Masoumi**

Entitled: **Feature Encoding of Spectral Descriptors for 3D Shape Recognition**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Information and Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____Chair
Dr. Subhash Rakheja

_____External Examiner
Dr. Mohand Said Allili

_____External to Program
Dr. M. Zahangir Kabir

_____Examiner
Dr. Jamal Bentahar

_____Examiner
Dr. Nizar Bouguila

_____Thesis Supervisor
Dr. A. Ben Hamza

Approved by _____
Dr. Chun Wang, Graduate Program Director

June 28, 2017 _____
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Feature Encoding of Spectral Descriptors for 3D Shape Recognition

Majid Masoumi, Ph.D.
Concordia University, 2017

Feature descriptors have become a ubiquitous tool in shape analysis. Features can be extracted and subsequently used to design discriminative signatures for solving a variety of 3D shape analysis problems. In particular, shape classification and retrieval are intriguing and challenging problems that lie at the crossroads of computer vision, geometry processing, machine learning and medical imaging.

In this thesis, we propose spectral graph wavelet approaches for the classification and retrieval of deformable 3D shapes. First, we review the recent shape descriptors based on the spectral decomposition of the Laplace-Beltrami operator, which provides a rich set of eigenbases that are invariant to intrinsic isometries. We then provide a detailed overview of spectral graph wavelets. In an effort to capture both local and global characteristics of a 3D shape, we propose a three-step feature description framework. Local descriptors are first extracted via the spectral graph wavelet transform having the Mexican hat wavelet as a generating kernel. Then, mid-level features are obtained by embedding local descriptors into the visual vocabulary space using the soft-assignment coding step of the bag-of-features model. A global descriptor is subsequently constructed by aggregating mid-level features weighted by a geodesic exponential kernel, resulting in a matrix representation that describes the frequency of appearance of nearby codewords in the vocabulary. In order to analyze the performance of the proposed algorithms on 3D shape classification, support vector machines and deep belief networks are applied to mid-level features. To assess the performance of the proposed approach for nonrigid 3D shape retrieval, we compare the global descriptor of a query to the global descriptors of the rest of shapes in the dataset using a dissimilarity measure and find the closest shape. Experimental results on three standard 3D shape benchmarks demonstrate the effectiveness of the proposed classification and retrieval approaches in comparison with state-of-the-art methods.

Acknowledgements

I am forever thankful and indebted to my supervisor Prof. **A. Ben Hamza** for his guidance, endless support, and encouragement during my Ph.D. He has routinely gone beyond his duties to fire fight my worries, concerns, and anxieties. I am more grateful to him than he will ever know.

My deepest gratitude is to my wife **Mahsa** who has been tirelessly supporting me and making even the most tough times truly wonderful. She spent sleepless nights and was always my support in the moments when there was no one to answer my queries. I would like to dedicate my thesis to Mahsa who has inspired me and given me the confidence to work as well as the happiness to live.

I would like to express my special thanks to **my parents** for their unremitting encouragement, and unconditional love. Words cannot express how grateful I am to my mother and father.

Table of Contents

List of Tables

List of Figures

List of Acronyms

LBO Laplace-Beltrami Operator

HKS Heat Kernel Signature

CBIR Content-Based Image Retrieval

BoF Bag-of-Features

SGWC Spectral Graph Wavelet Codes

SGWC-BoF Spectral Graph Wavelet Codes Bag-of-Features

DBN Deep Belief Networks

DeepSGW Deep Spectral Graph Wavelets

SVMs Support Vector Machines

RBM Restricted Boltzmann Machines

CNN Convolutional Neural Networks

DCG Discounted Cumulative Gain

NN Nearest Neighbor

FT First-Tier

ST Second-Tier

SHREC Shape Retrieval Contest

CHAPTER 1

Introduction

1.1 Framework and Motivation

Recent advances in 3D imaging and processing, graphics hardware and networks have led to a whopping increase in geometry models available freely or commercially on the Web. As a result, the task of efficiently measuring the 3D object similarity to find and retrieve relevant objects for a given query and categorize an object into one of a set of classes has become of paramount importance in a wide range of applications, including computer-aided design, video gaming, special effects and film production, medicine, and archaeology. The main challenge in 3D shape retrieval and classification algorithms is to compute an invariant shape descriptor that captures well the geometric and topological properties of a shape [1–5].

In computer graphics and geometry processing, a 3D shape is usually represented as a triangle mesh. Other effective representations methods are based on medial [6] or multiple views [7]. Content-based shape retrieval based on the comparison of shape properties is complicated by the fact that many 3D objects manifest rich variability, and invariance to different classes of transformations and shape variations is often required. One of the most challenging settings addressed is the case of nonrigid or deformable shapes, in which shapes undergo changes that can be well-approximated by intrinsic isometries, i.e. deformations that preserve geodesic distances between all pairs of points. This class of deformations is much richer than rigid motions and can be approximated. Recently, various methods have been proposed to tackle nonrigid 3D shape recognition problem, particularly with the isometric invariant representation. These methods can be mainly categorized into two main classes: skeleton-based [6, 8] and surface-based [9–12]. The former approaches usually capture the global topological structure of the shape, and a dissimilarity is often

determined as the cost function to match two or more shapes. The latter methods, on the other hand, often represent a shape as a frequency histogram of deformation invariant local distances or vertex signatures.

Over the past decade, there has been a flurry of research activity on surface based shape recognition due largely to two key reasons: First, surface-based 3D models are more popular because of their highly-effective representation ability and less memory storage. Second, humans are taught to differentiate between shapes mainly by surface features, and in many shape applications only the surface is of interest. Therefore, in this thesis, we focus on surface-based shape recognition with local vertex descriptors.

Research efforts on spectral shape analysis have recently resulted in numerous spectral descriptors [9–14], which are predominately based on the LBO [15,16]. However, to date, no comprehensive comparison has been conducted in the literature, which often results in intractable situation when choosing appropriate descriptors for certain applications.

1.2 Literature Review

In recent years, considerable research efforts on shape analysis have been conducted in a bid to design an appropriate shape descriptor aimed at finding the most relevant shapes. In the literature, there are several survey works [1–5] that have keen interest in systematic shape classification and retrieval. Early research works on 3D shape description have been centered primarily on invariance under global Euclidean transformations (i.e. rigid transformations). These works include the shape context [17], shape distributions [18] and spherical harmonics [19]. Recently, significant efforts have been invested in exploring the invariance properties of shapes to nonrigid deformations. An intuitive approach is to replace the Euclidean distance with the geodesic one. The primary motivation is that unlike the Euclidean distance, which is basically a straight line between two points in 3D space, the geodesic distance captures the global nonlinear structure and the intrinsic geometry of the data. The main drawback of the geodesic distance is that it suffers from strong sensitivity to topological noise, which might heavily damage the shape invariants.

Other similar spectral distances include the commute time distance and the biharmonic distance [20]. Since the eigensystem of the LBO is isometric invariant, it is well-suited for the analysis and retrieval of nonrigid shapes, and it is more robust than the geodesic distance. By integrating the local distribution of features, the intrinsic shape context was proposed in [21] as a natural extension of the 2D Shape Context to 3D nonrigid surfaces, and it was shown to outperform individual vertex descriptors in 3D shape matching.

The overwhelming majority of 3D object rendering techniques proposed in the literature of computer graphics and computer vision are initially based on geometric and topological representations

which represent the features of an object [22, 23]. For example, Siddiqi *et al*. [24] introduced a shock detection approach based on singularity theory to generate a skeletal shape model. Also, Siddiqi *et al*. [25] proposed a directed acyclic graph representation for 3D retrieval using medial surfaces. This approach utilizes the geometric information associated with each graph node along with an eigenvalue labeling of the adjacency matrix of the subgraph rooted at that node. Cornea *et al*. [26] designed a 3D matching framework for 3D volumetric objects using a many-to-many matching algorithm. This algorithm is based on establishing correspondences among two skeletal representations via distribution-based matching in metric spaces. Hassouna *et al*. [27] proposed a level set based framework for robust centerline extraction of 2D shapes and 3D volumetric objects. This approach is based on the gradient vector flow and uses a wave propagation technique, which identifies the curve skeletons as the wave points of maximum positive curvatures. Tagliasacchi *et al*. [28] introduced a curve skeleton extraction algorithm from imperfect point clouds. A major drawback of curve skeletons is that they cannot capture general shape features such as surface ridges, and are essentially restricted to objects which resemble connected tubular forms.

Global approaches have been proposed as alternatives to feature-based representations, which represent a 3D object by a global measure or shape distribution defined on the surface of the object [18, 19, 29]. Ankerst *et al*. [29] used shape histograms to analyze the similarity of 3D molecular surfaces. These histograms are built from uniformly distributed surface points taken from the molecular surfaces, and are defined on concentric shells and sectors around the centroid of the surface. Osada *et al*. [18] proposed a global approach for computing shape signatures of arbitrary 3D models. The key idea is to represent an object by a global histogram based on the Euclidean distance defined on the surface of an object. More recently, Ion *et al*. [30] presented an articulation-insensitive shape matching approach by constructing histograms from the eccentricity transform using geodesic distances. Kazhdan *et al*. [19] proposed a rotation invariant spherical harmonic representation that transforms rotation dependent shape descriptors into rotation independent ones. Chen *et al*. [31] presented a lightfield descriptor for 3D object retrieval by comparing ten silhouettes of the 3D shape obtained from ten viewing angles distributed uniformly on the viewing enclosing sphere. The dissimilarity between two shapes is computed as the minimal distance obtained by rotating the viewing sphere of one lightfield descriptor relative to the other lightfield descriptor. The computation of this descriptor is, however, significantly time consuming compared to spherical harmonics [32].

The intriguing field of diffusion geometry provides a generic framework for many methods in the analysis of geometric shapes [33]. This framework formulates the heat diffusion processes on manifolds. Spectral shape analysis is a methodology that relies on the eigensystem (eigenvalues and/or eigenfunctions) of the Laplace-Beltrami operator to compare and analyze geometric shapes. Levy [34] showed that eigenfunctions can be well-adapted to the geometry and the topology of a

3D model. Coifman and Lafon [33] constructed diffusion distances as the L_2-norm difference of energy distribution between two points initialized with unit impulse functions after a given time. Finally, shape google algorithm [35] was proposed as a classic method for deformable shape retrieval. It uses the multi-scale diffusion heat kernels as "geometric words", and constructs compact and informative shape representation using vocabulary method.

The past decade has witnessed the surge in popularity of the vocabulary model in image processing domain. Vocabulary model was first introduced in text retrieval, and later was applied to image categorization in the seminal paper [36]. Subsequent research has focused on overcoming its two intrinsic limitations to improve discrimination, namely the information loss of the assignment of local features to visual words, and then the lack of information on the spatial layout of the local features.

Increase in size of vocabulary is often addressed as a way to enhance the performance of dictionary model. However, it leads to a higher computational complexity for making dictionary and feature assignment. On the other hand, when the vocabularies are more compact, the information lost in the quantization steps becomes more significant, specifically when hard assignment [37] is applied. Boiman *et al.* [38] showed that by directly using of image-to-class distances without descriptor quantization, the discrimination ability is considerably decreased due to the rough quantization of the feature space. But with the soft-assignment coding of signatures to multiple visual words, the loss can be compensated as reported in [39, 40]. Inspired by compressive sensing methodology, other approaches for assignment were guided by sparsity constraints [41] and locality constraints [42].

Bag-of-Features (BoF) usually encodes the zero-order statistics of the distribution of signatures. The Fisher vector extends the BoF by encoding high-order statistics (first and, optionally, second order). This description vector is the gradient of the sample's likelihood with respect to the parameters of this distribution, scaled by the inverse square root of the Fisher information matrix [43]. A simplified version of Fisher kernels, namely vector of locally aggregated descriptors (VLAD) was introduced in [44]. The three aforementioned various ways of aggregating local image descriptors into a vector were evaluated by Jegou *et al.* in [45]. Furthermore, Picard *et al.* [46] expanded the VLAD approach by adding an aggregation of the tensor product of descriptors.

Similar to the image domain, the vocabulary model representation for 3D surfaces is a frequency histogram of quantized local geometric appearance, where the spatial layout of the geometric appearance is completely ignored [35]. Clearly, the spatial information may convey useful cues to improve the discrimination between 3D shapes. Before modeling the spatial layout on surfaces, it is necessary to review the technique for images. In the literature, two different ways to encode spatial information have been explored, which are based on local relative positions of pairwise features, and on global absolute positions.

Modeling pairwise spatial features into the vocabulary model is an intuitive way to amalgamate spatial information. A spatially-sensitive affine-invariant image descriptor was constructed by Bronstein *et al.* [47] using canonical relation, in which both the features and their relation are affine-invariant. They also generalized the pairwise spatially-sensitive descriptors called "Expression" for 3D surface using the heat kernel as the relation [35]. In order to give the signature more descriptive ability, they also considered the relationship between the visual words. Saverese *et al.* [48] used correlograms of visual words to model the spatial correlations between quantized local descriptors. Ling and Soatto [49] characterized the relative locations of visual words. Their proximity distribution representation is a 3D structure which records the number of times a visual word appears within a particular number of nearest neighbors of another word. Finally, besides pairwise relation, more complex relation such as the graph manner layout of groups of quantized local invariant descriptors was proposed by Behmo *et al.* [50], which can preserve translational relations between features. Liu *et al.* [51] calculated spatial histograms where the co-occurrences of local features are computed in circular regions of varying distances.

One of the initial works to address the lack of spatial information in the BoF representation is spatial pyramid matching (SPM) which introduced by Lazebnik *et al.* [52]. Their spatial pyramid representation was motivated by an earlier work, termed pyramid matching by Grauman and Darrell [53], on finding approximate correspondences between sets of points in high-dimensional feature spaces. The key idea behind pyramid matching is to partition the feature space into a sequence of increasingly coarser grids and then compute a weighted sum over the number of matches that occur at each level of resolution. However, SPM and relative spatial relation modeling are still too weak. Recently, stronger spatially encoding methods include encoding geometric information of objects within the images. Local features of an image are projected onto different directions or points to generate a series of ordered BoF, based on which families of spatial partitions can guarantee the invariance of object to affine transformation [54]. Additionally, there are some approaches characterizing both the absolute and relative spatial layout of an image. Spatial pyramid co-occurrence [55] computes local co-occurrence with respect to spatial layout over a hierarchical spatial partitioning of an image. In addition to co-occurrences, geometry-preserving visual phrases [56] can encode more spatial information through capturing the local and long-range spatial layouts of the words. Unlike manually defined spatial regions for pooling, Jia *et al.* [57] proposed to learn more adaptive receptive fields to increase the performance even with a significantly smaller vocabulary size at the coding layer. In [58], the Gaussian mixture model was encoded with spatial layout to improve the performance of Fisher kernel for image classification.

1.3 Spectral Geometry

Spectral geometry is concerned with the eigenvalue spectrum of the Laplace-Beltrami operator (LBO) on a compact Riemannian manifold, and aims at describing the relationships between such a spectrum and the geometric structure of the manifold.

1.3.1 Triangle Mesh

A 3D shape is usually modeled as a triangle mesh $\mathbb{M}$ whose vertices are sampled from a Riemannian manifold. A triangle mesh $\mathbb{M}$ may be defined as a graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ or $\mathbb{G} = (\mathcal{V}, \mathcal{T})$, where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges, and $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_m\}$ is the set of triangles, as depicted in the enlarged view of Figure 1.1 (left). Each edge $e_{ij} = [\mathbf{v}_i, \mathbf{v}_j]$ connects a pair of vertices $\{\mathbf{v}_i, \mathbf{v}_j\}$. Two distinct vertices $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}$ are adjacent (denoted by $\mathbf{v}_i \sim \mathbf{v}_j$ or simply $i \sim j$) if they are connected by an edge, i.e. $e_{ij} \in \mathcal{E}$.

1.3.2 Laplace-Beltrami Operator

Given a compact Riemannian manifold $\mathbb{M}$, the space $L^2(\mathbb{M})$ of all smooth, square-integrable functions on $\mathbb{M}$ is a Hilbert space endowed with inner product $\langle f_1, f_2 \rangle = \int_{\mathbb{M}} f_1(\mathbf{x}) f_2(\mathbf{x})\, da(\mathbf{x})$, for all $f_1, f_2 \in L^2(\mathbb{M})$, where $da(x)$ (or simply dx) denotes the measure from the area element of a Riemannian metric on $\mathbb{M}$. Given a twice-differentiable, real-valued function $f : \mathbb{M} \to \mathbb{R}$, the LBO is defined as $\Delta_{\mathbb{M}} f = -\mathrm{div}(\nabla_{\mathbb{M}} f)$, where $\nabla_{\mathbb{M}} f$ is the intrinsic gradient vector field and div is the divergence operator [15, 59]. The LBO is a linear, positive semi-definite operator acting on the space of real-valued functions defined on $\mathbb{M}$, and it is a generalization of the Laplace operator to non-Euclidean spaces.

Discretization A real-valued function $f : \mathcal{V} \to \mathbb{R}$ defined on the mesh vertex set may be represented as an m-dimensional vector $\mathbf{f} = (f(i)) \in \mathbb{R}^m$, where the ith component $f(i)$ denotes the function value at the ith vertex in $\mathcal{V}$. Using a mixed finite element/finite volume method on triangle meshes [60], the value of $\Delta_{\mathbb{M}} f$ at a vertex $\mathbf{v}_i$ (or simply i) can be approximated using the cotangent weight scheme as follows:

$$\Delta_{\mathbb{M}} f(i) \approx \frac{1}{a_i} \sum_{j \sim i} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} \big(f(i) - f(j)\big), \tag{1.1}$$

where α_{ij} and β_{ij} are the angles $\angle(\mathbf{v}_i \mathbf{v}_{k_1} \mathbf{v}_j)$ and $\angle(\mathbf{v}_i \mathbf{v}_{k_2} \mathbf{v}_j)$ of two faces $\mathbf{t}^\alpha = \{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_{k_1}\}$ and $\mathbf{t}^\beta = \{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_{k_2}\}$ that are adjacent to the edge $[i, j]$, and a_i is the area of the Voronoi cell (shaded polygon) at vertex i, as shown in Figure 1.1 (right). It should be noted that the cotangent weight scheme is numerically consistent and preserves several important properties of the continuous LBO, including symmetry and positive semi-definiteness [61].

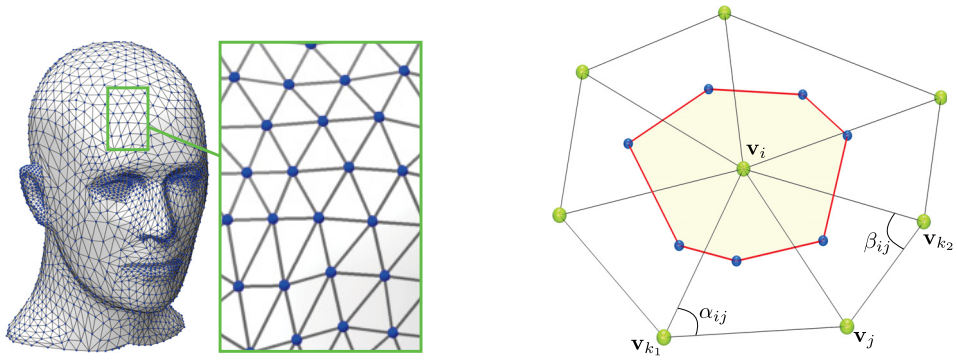


Figure 1.1: Triangular mesh representation (left); Cotangent scheme angles (right).

Spectral Analysis The $m \times m$ matrix associated to the discrete approximation of the LBO is given by $\mathbf{L} = \mathbf{A}^{-1}\mathbf{W}$, where $\mathbf{A} = \mathrm{diag}(a_i)$ is a positive definite diagonal matrix (mass matrix), and $\mathbf{W} = \mathrm{diag}(\sum_{k \neq i} c_{ik}) - (c_{ij})$ is a sparse symmetric matrix (stiffness matrix). Each diagonal element a_i is the area of the Voronoi cell at vertex i, and the weights c_{ij} are given by

$$c_{ij} = \begin{cases} \dfrac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} & \text{if } i \sim j \\ 0 & \text{o.w.} \end{cases} \tag{1.2}$$

where α_{ij} and β_{ij} are the opposite angles of two triangles that are adjacent to the edge $[i, j]$.

The eigenvalues and eigenvectors of $\mathbf{L}$ can be found by solving the generalized eigenvalue problem $\mathbf{W}\boldsymbol{\varphi}_\ell = \lambda_\ell \mathbf{A}\boldsymbol{\varphi}_\ell$ using for instance the Arnoldi method of ARPACK, where λ_ℓ are the eigenvalues and $\boldsymbol{\varphi}_\ell$ are the unknown associated eigenfunctions (i.e., eigenvectors which can be thought of as functions on the mesh vertices). We may sort the eigenvalues in ascending order as $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_m$ with associated orthonormal eigenfunctions $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_m$, where the orthogonality of the eigenfunctions is defined in terms of the $\mathbf{A}$-inner product, i.e.

$$\langle \boldsymbol{\varphi}_k, \boldsymbol{\varphi}_\ell \rangle_{\mathbf{A}} = \sum_{i=1}^{m} a_i \varphi_k(i) \varphi_\ell(i) = \delta_{k\ell}, \quad \text{for all } k, \ell = 1, \ldots, m. \tag{1.3}$$

We may rewrite the generalized eigenvalue problem in matrix form as $\mathbf{W}\boldsymbol{\Phi} = \mathbf{A}\boldsymbol{\Phi}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_i)$ is an $m \times m$ diagonal matrix with the λ_ℓ on the diagonal, and $\boldsymbol{\Phi}$ is an $m \times m$ orthogonal matrix whose ℓth column is the unit-norm eigenvector $\boldsymbol{\varphi}_\ell$. It should be noted that since the first eigenvalue λ_1 is zero, its associated eigenvector $\boldsymbol{\varphi}_1$ is an m-dimensional constant vector given by

$$\boldsymbol{\varphi}_1 = \left(\frac{1}{\sqrt{a}}, \frac{1}{\sqrt{a}}, \ldots, \frac{1}{\sqrt{a}} \right)^{\mathsf{T}}, \tag{1.4}$$

where $a = \mathrm{area}(\mathbb{M})$ is the total area of the mesh.

The successful use of the LBO eigenvalues and eigenfunctions in shape analysis is largely attributed to their isometry invariance and robustness to noise. Moreover, the eigenfunctions associated to the smallest eigenvalues capture well the large-scale properties of a shape. As shown in Figure 1.2, the (non-trivial) eigenfunctions of the LBO encode important information about the intrinsic global geometry of a shape. Notice that the eigenfunctions associated with larger eigenvalues oscillate more rapidly. Blue regions indicate negative values of the eigenfunctions and red colors regions indicate positive values, while green and yellow regions in between.

Figure 1.2: Visualization of the first four (non-trivial) eigenfunctions of the LBO. From left to right: a 3D frog model Gouraud shaded and color-coded by the values of the first, second, third and fourth eigenfunctions. **Best viewed in color**.

1.3.3 Spectral Shape Signatures

In recent years, several local descriptors based on the eigenvalues and eigenfunctions of the LBO have been proposed in the 3D shape analysis literature such as ShapeDNA [13], global point signature (GPS) [9], heat mean signature (HMS), heat kernel signature (HKS) [10] and wave kernel signature (WKS) [12].

ShapeDNA One of the first spectral shape descriptors is ShapeDNA [13] which is a normalized sequence of the first eigenvalues of the LBO. Its main advantages are the simple representation (a vector of numbers) and scale invariance. Despite its simplicity, the shapeDNA yields a better performance in the retrieval of nonrigid shapes. However, the eigenvalues are a global descriptor, therefore the shapeDNA cannot be used for local or partial shape analysis. The Eigenvalue Descriptor (EVD) [5], on the other hand, is a sequence of the eigenvalues of the geodesic distance matrix. Both ShapeDNA and EVD can be normalized by the second eigenvalue.

Global Point Signature The global point signature (GPS) [9] at a surface point is a vector of scaled eigenfunctions of the LBO. The GPS is a global feature in the sense that it cannot be used for partial shape matching. It is defined in terms of the eigenvalues and eigenfunctions of $\Delta_{\mathbb{M}}$ as follows:

$$\mathrm{GPS}(x) = \left(\frac{\varphi_2(x)}{\sqrt{\lambda_2}}, \frac{\varphi_3(x)}{\sqrt{\lambda_3}}, \ldots, \frac{\varphi_i(x)}{\sqrt{\lambda_i}}, \ldots \right) \tag{1.5}$$

GPS is invariant under isometric deformations of the shape, but it suffers for the problem of eigenfunctions switching whenever the associated eigenvalues are close to each other.

Heat Mean Signature The Heat Mean Signature (HMS) [62] quantitatively evaluate the temperature distribution resulting from the heat flow process

$$\mathrm{HMS}_t(x) = \frac{1}{m} \sum_{y \neq x} k_t(x, y), \tag{1.6}$$

where k_t is heat kernel and $\mathrm{HMS}_t(x)$ can be physically interpreted as the average temperature on the surface obtained by applying a unit amount of heat on the vertex x and after a certain amount of time of heat dissipation. A relatively smaller parameter t is often empirically chosen to preserve a higher resolution version of the original surface [63]. Fang *et al.* also proposed the temperature distribution descriptor [64], which is based on the distribution of the values of average temperature for all of the vertices on the mesh. We construct a multi-scale HMS to compare temperature distribution with multiple diffusion times as follows:

$$\mathrm{HMS}(x) = \left(\mathrm{HMS}_{t_1}, \mathrm{HMS}_{t_2}, \ldots, \mathrm{HMS}_{t_n}\right). \tag{1.7}$$

Heat Kernel Signature and Wave Kernel Signature Both HKS and WKS have an elegant physical interpretation: the HKS describes the amount of heat remaining at a mesh vertex $j \in \mathcal{V}$ after a certain time, whereas the WKS is the probability of measuring a quantum particle with the initial energy distribution at j. The HKS at a vertex j is defined as:

$$\mathfrak{s}_{t_k}(j) = \sum_{\ell=1}^{m} e^{-\lambda_\ell t_k} \varphi_\ell^2(j), \tag{1.8}$$

where λ_ℓ and φ_ℓ are the eigenvalues and eigenfunctions of the LBO. In other words, HKS ($k_t(x, x)$) is the diagonal of the heat kernel matrix ($k_t(x, y)$) at multiple scales. Figure 1.3 depicts a clear representation of heat kernel versus heat kernel signature on human shape.

The HKS contains information mainly from low frequencies, which correspond to macroscopic features of the shape; and thus exhibits a major discrimination ability in shape retrieval and classification tasks. With multiple scaling factors t_k, a collection of low-pass filters are established. The larger is t_k, the more high frequencies are suppressed. However, different frequencies are always mixed in the HKS, and high-precision localization task may fail due in part to the suppression of the high frequency information, which corresponds to microscopic features. To circumvent these disadvantages, Aubry *et al.* [12] introduced the WKS, which is defined at a vertex j as follows:

$$\mathfrak{s}_{t_k}(j) = \sum_{\ell=1}^{m} C_{t_k} \exp\left(-\frac{(\log t_k - \log \lambda_\ell)^2}{\sigma^2}\right) \varphi_\ell^2(j), \tag{1.9}$$

(a) $k_t(x, y)$ (b) $k_t(x, x)$

Figure 1.3: (a) Propagation of heat ($k_t(x, y)$) from a specified point on the elbow of human shape to the rest of the shape in a given time t. (b) Representation of heat kernel signature acquired by the diagonal of the heat kernel matrix. As shown, heat is raised when color changes from black to red. Also, positive and negative values of Gaussian curvatures relate to high and low amount of $k_t(x, x)$, respectively.

where C_{t_k} is a normalization constant. The WKS explicitly separates the influences of different frequencies, treating all frequencies equally. Thus, different spatial scales are naturally separated, making the high-precision feature localization possible.

Given a range of discrete scales t_k, a bank of filters is constructed for each signature, and thus a vertex j on the mesh surface can be described by a p-dimensional point signature vector given by

$$\mathbf{s}_j = \{\mathfrak{s}_{t_k}(j) \mid k = 1, \ldots, p\}, \quad \text{for } j = 1, \ldots, m. \tag{1.10}$$

For the sake of notational simplicity, we use $s(x)$ to represent the types of the above spectral signatures evaluated at a surface point x, i.e. HKS and WKS.

1.3.4 Spectral Graph Wavelets

Similar to the Fourier transform which decomposes a signal into its constituent frequencies, the wavelet transform is a powerful multiresolution analysis tool that enable decomposition of a signal

into a wavelet basis which allows simultaneous localization in space and frequency. Wavelet analysis provides a time-scale representation and extends frequency analysis to scale, while Fourier analysis only gives the frequency information [65]. The idea of wavelets is based on the use of two main operations on the signal, namely shifting and scaling. Using these two operations, a signal f can be represented as the sum of shifted and scaled versions of the so-called mother wavelet function, ψ, and shifted versions of the so-called scaling function, ϕ. The mother wavelet and scaling functions act as band-pass and low-pass functions, respectively.

Classical Continuous Wavelet Transform The continuous wavelet transform maps the original signal, which is a function of just one variable (time) into a function of two variables (time and frequency), providing highly redundant information. More specifically, for a given mother wavelet ψ, a family $\psi_{t,a}$ of daughter wavelets can be obtained by simply scaling and translating ψ as follows:

$$\psi_{t,a}(x) = \frac{1}{t}\psi\left(\frac{x-a}{t}\right), \tag{1.11}$$

where t is a positive scaling parameter that controls the width of the wavelet, and a is a translation parameter that controls the location of the wavelet. Scaling a wavelet simply means stretching it (if $t > 1$) or compressing it (if $t < 1$), while translating a wavelet simply means shifting its position in time. Note that the translation parameter does not have a counterpart in the Fourier basis functions, where the position information is totally missing. It should also be noted that the scaling parameter in the wavelet analysis is similar to the scale used in maps. As in the case of maps, high scales correspond to a non-detailed global view of the signal, while low scales correspond to a detailed view.

Given a square-integrable signal f, the continuous wavelet transform (CWT) with respect to the mother wavelet ψ is expressed by the following integral

$$W_f(t,a) = \langle \psi_{t,a}, f\rangle = \frac{1}{t}\int_{-\infty}^{\infty} f(x)\psi^*\left(\frac{x-a}{t}\right)dx, \tag{1.12}$$

which is also referred to as the wavelet coefficient at scale t and location a. Also, ψ^* denotes the complex conjugate of ψ. The position of the wavelet in the time domain is given by the translational value a, while its position in the frequency domain is given by the scale t. Thus, the CWT gives us information simultaneously on time and frequency. Unlike Fourier transform, the CWT possesses the ability to construct a time-frequency representation of a signal that offers very good time and frequency localization. The CWT may be invertible when the mother wavelet ψ satisfies the admissibility condition, $C_\psi = \int_0^\infty \frac{\left|\hat{\psi}(\omega)\right|^2}{\omega}\mathrm{d}\omega < \infty$, where $\hat{\psi}$ is the Fourier transform of ψ. The inverse CWT is given by

$$f(x) = \frac{1}{C_\psi}\int_0^\infty\int_{-\infty}^{\infty} W_f(t,a)\psi_{t,a}(x)\frac{da\,dt}{t}\cdot \tag{1.13}$$

For a fixed scale t, the CWT may be interpreted as an operator taking a function f and returning the function $(T^t f)(a) = W_f(t, a)$. In other words, the translation parameter can be considered as the independent variable of the function returned by the operator T^t. The CWT may also be expressed in the Fourier domain as [66]:

$$(T^t f)(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\psi}^*(t\omega) \hat{f}(\omega) e^{\imath\omega x} d\omega, \tag{1.14}$$

where $\hat{\psi}^*(t\omega)$ is the complex conjugate of the Fourier transform of the wavelet ψ at scale t, and $\hat{f}(\omega)$ is the Fourier transform of the signal f. The scaling parameter t appears only in the argument of $\hat{\psi}^*(t\omega)$, showing that the scaling operation can be completely transferred to the Fourier domain. It is clear that the wavelet transform at each scale can be viewed as a Fourier multiplier operator, determined by filters that are derived from scaling a single filter $\hat{\psi}^*(\omega)$. This idea was adopted by Hammond *et al.* [66] to provide the analogue of the wavelet transform on weighted graphs via spectral graph theory.

For any graph signal $f : \mathcal{V} \to \mathbb{M}$, the forward and inverse graph Fourier transforms (also called manifold harmonic and inverse manifold harmonic transforms) are defined as

$$\hat{f}(\ell) = \langle \mathbf{f}, \boldsymbol{\varphi}_\ell \rangle = \sum_{i=1}^{m} a_i f(i) \varphi_\ell(i), \quad \ell = 1, \ldots, m \tag{1.15}$$

and

$$f(i) = \sum_{\ell=1}^{m} \hat{f}(\ell) \varphi_\ell(i) = \sum_{\ell=1}^{m} \langle \mathbf{f}, \boldsymbol{\varphi}_\ell \rangle \varphi_\ell(i), \quad i \in \mathcal{V}, \tag{1.16}$$

respectively, where $\hat{f}(\ell)$ is the value of f at eigenvalue λ_ℓ (i.e. $\hat{f}(\ell) = \hat{f}(\lambda_\ell)$). In particular, the graph Fourier transform of a delta function δ_j centered at vertex j is given by

$$\hat{\delta}_j(\ell) = \sum_{i=1}^{m} a_i \delta_j(i) \varphi_\ell(i) = \sum_{i=1}^{m} a_i \delta_{ij} \varphi_\ell(i) = a_j \varphi_\ell(j). \tag{1.17}$$

The forward and inverse graph Fourier transforms may be expressed in matrix-vector multiplication as follows:

$$\hat{\mathbf{f}} = \boldsymbol{\Phi}^\intercal \mathbf{A} \mathbf{f} \quad \text{and} \quad \mathbf{f} = \boldsymbol{\Phi} \hat{\mathbf{f}}, \tag{1.18}$$

where $\mathbf{f} = (f(i))$ and $\hat{\mathbf{f}} = (\hat{f}(\ell))$ are m-dimensional vectors whose elements are given by (1.15) and (1.16), respectively.

Wavelet Function The spectral graph wavelet transform is determined by the choice of a spectral graph wavelet generating kernel $g : \mathbb{R}^+ \to \mathbb{R}^+$, which is analogous to the Fourier domain wavelet. To act as a band-pass filter, the kernel g should satisfy $g(0) = 0$ and $\lim_{x\to\infty} g(x) = 0$.

Let g be a given kernel function and denote by T_g^t the wavelet operator at scale t. Similar to the Fourier domain, the graph Fourier transform of T_g^t is given by

$$\widehat{T_g^t f}(\ell) = g(t\lambda_\ell)\hat{f}(\ell), \tag{1.19}$$

where g acts as a scaled band-pass filter. Thus, the inverse graph Fourier transform is given by

$$(T_g^t f)(i) = \sum_{\ell=1}^{m} \widehat{T_g^t f}(\ell)\varphi_\ell(i) = \sum_{\ell=1}^{m} g(t\lambda_\ell)\hat{f}(\ell)\varphi_\ell(i). \tag{1.20}$$

Applying the wavelet operator T_g^t to a delta function centered at vertex j (i.e. $f(i) = \delta_j(i) = \delta_{ij}$), the spectral graph wavelet $\psi_{t,j}$ localized at vertex j and scale t is then given by

$$\psi_{t,j}(i) = (T_g^t \delta_j)(i) = \sum_{\ell=1}^{m} g(t\lambda_\ell)\hat{\delta}_j(\ell)\varphi_\ell(i) = \sum_{\ell=1}^{m} a_j g(t\lambda_\ell)\varphi_\ell(j)\varphi_\ell(i). \tag{1.21}$$

This indicates that shifting the wavelet to vertex j corresponds to a multiplication by $\varphi_\ell(j)$. It should be noted that $g(t\lambda_\ell)$ is able to modulate the spectral wavelets $\psi_{t,j}$ only for λ_ℓ within the domain of the spectrum of the LBO. Thus, an upper bound on the largest eigenvalue $\lambda_{\max}$ is required to provide knowledge on the spectrum in practical applications.

Hence, the spectral graph wavelet coefficients of a given function f can be generated from its inner product with the spectral graph wavelets:

$$W_f(t,j) = \langle \mathbf{f}, \boldsymbol{\psi}_{t,j} \rangle = \sum_{\ell=1}^{m} a_j g(t\lambda_\ell)\hat{f}(\ell)\varphi_\ell(j). \tag{1.22}$$

Scaling Function Similar to the low-pass scaling functions in the classical wavelet analysis, a second class of waveforms $h : \mathbb{R}^+ \to \mathbb{R}$ are used as low-pass filters to better encode the low-frequency content of a function f defined on the mesh vertices. To act as a low-pass filter, the function h should satisfy $h(0) > 0$ and $h(x) \to 0$ as $x \to \infty$. Similar to the wavelet kernels, the scaling functions are given by

$$\phi_j(i) = (T_h \delta_j)(i) = \sum_{\ell=1}^{m} h(\lambda_\ell)\hat{\delta}_j(\ell)\varphi_\ell(i) = \sum_{\ell=1}^{m} a_j h(\lambda_\ell)\varphi_\ell(j)\varphi_\ell(i). \tag{1.23}$$

and their spectral coefficients are

$$S_f(j) = \langle \mathbf{f}, \boldsymbol{\phi}_j \rangle = \sum_{\ell=1}^{m} a_j h(\lambda_\ell)\hat{f}(\ell)\varphi_\ell(j). \tag{1.24}$$

A major advantage of using the scaling function is to ensure that the original signal f can be stably recovered when sampling scale parameter t with a discrete number of values t_k. As demonstrated

in [66], given a set of scales $\{t_k\}_{k=1}^K$, the set $F = \{\phi_j\}_{j=1}^m \cup \{\psi_{t_k,j}\}_{k=1}^K{}_{j=1}^m$ forms a spectral graph wavelet frame with bounds

$$A = \min_{\lambda \in [0,\lambda_{\max}]} G(\lambda) \quad \text{and} \quad B = \max_{\lambda \in [0,\lambda_{\max}]} G(\lambda), \tag{1.25}$$

where

$$G(\lambda) = h(\lambda)^2 + \sum_k g(t_k\lambda)^2. \tag{1.26}$$

The stable recovery of f is ensured when A and B are away from zero. Additionally, the crux of the scaling function is to smoothly represent the low-frequency content of the signal on the mesh. Thus, the design of the scaling function h is uncoupled from the choice of wavelet generating kernel g.

1.4 Shape Classification

The task in the shape classification problem is to assign a shape to a class chosen from a predefined set of classes. Broadly speaking, shape classification is the process of organizing a dataset of shapes into a known number of classes, and the task is to assign new shapes to one of these classes. It is common practice in classification to randomly split the available data into training and test sets. Classification aims to learn a classifier (also called predictor or classification model) from labeled training data. The training data consist of a set of training examples or instances that are labeled with predefined classes. The resulting, trained model is subsequently applied to the test data to classify future (unseen) data instances into these classes. The test data, which consists of data instances with unknown class labels, is used to evaluate the performance of the classification model and determine its accuracy in terms of the number of test instances correctly or incorrectly predicted by the model. A good classifier should result in high accuracy, or equivalently, in few misclassifications. In our work, we propose two approaches to perform 3D shape classification on spectral graph wavelet codes that are obtained from spectral graph wavelet signatures (i.e. local descriptors) via the soft-assignment coding step of the BoF model in conjunction with a geodesic exponential kernel for capturing the spatial relations between features.

1.5 Deep Learning

Deep learning is a machine learning paradigm that mimics the way the human brain works to varying degrees. The popularity of deep learning is largely attributed not only to its huge success in a wide range of tasks such as handwritten character recognition, image and video recognition, text analysis and speech recognition, but also to tech industry giants such as Google, Apple, IBM,

Microsoft, Facebook, Twitter, PayPal and Baidu that have acquired most of the dominant players in this field to improve their product offerings and services. Inspired by the actual structure of the brain, deep learning refers to a powerful class of machine learning techniques that learn multi-level representations of data in deep hierarchical architectures composed of multiple layers, where each higher layer corresponds to a more abstract (i.e. higher level) representation of information [67]. The process of deep learning is hierarchical in the sense that it takes low-level features at the bottom layer and then constructs higher and higher level features through the composition of layers.

Deep learning is a rapidly growing discipline that models high-level features in data as complex multilayered networks, where each layer can learn features at a different level of abstraction. As a branch of the broader discipline of machine learning, deep learning emulates a biological system by creating a simulated software network of mathematical neurons, and the resulting neural network builds a model that is capable of adapting itself to new data.

As a type of artificial intelligence, deep learning has been successfully applied in areas ranging from voice, image and text recognition to game playing, cybersecurity and emotion identification. Success in all of these fields is rooted in the ability of deep learning networks to extract useful information from unstructured real-world data such as collections of pictures or webcam videos of human faces. Deep learning is proving to be a powerful tool for extracting useful information from unstructured data in order to provide solutions across a broad range of field. For instance, a network may learn to identify brand logos in pictures posted on social media, health-threatening abnormalities in x-rays and MRIs, or human emotions from facial expressions captured by web-cams. The learning process involves an extended period of training in which the network is given examples to learn, extracts information from the examples, tests itself to determine whether the information it extracted improves its ability to recognize the examples, and then adjusts itself so that the next time it tries it does a better job. This learning process repeats until the network has achieved a predetermined level of accuracy.

In contrast to the shallow architectures which only contain a fixed feature layer (or base function) and a weight-combination layer (usually linear), deep architectures refers to the multilayer network where each two adjacent layers are linked to each other in some way. Deep architectures assist deep learning to model more complex data for better performance and even for less computation time in some cases [68].

1.6 Overview and Contributions

The organization of this thesis is as follows

- Chapter 1 contains a brief review of essential concepts and definitions which we refer to throughout the thesis, provides a literature review, and presents a short summary of material relevant to 3D shape retrieval and classification in the spectral geometric framework.

- In Chapter 2, we introduce a spectral graph wavelet framework for 3D shape classification that employs the BoF paradigm in an effort to design a global shape descriptor defined in terms of mid-level features and a geodesic exponential kernel [69]. The proposed approach not only takes into consideration the spatial relations between features, but also substantially excels state-of-the-art methods both in classification accuracy and in scalability. The effectiveness of the method was demonstrated on two standard 3D shape benchmarks.

- In Chapter 3, we propose a deep learning-based approach for classification of 3D objects [70]. Our proposed DeepSGW framework incorporates the vertex area into the definition of spectral graph wavelet in a bid to capture more geometric information and, hence, further improve its discriminative ability. Moreover, we use spectral graph wavelet codes in conjunction with a geodesic exponential kernel for capturing the spatial relations between features. Then, in order to perform 3D object classification, deep belief networks (DBN) is employed as a classifier to be learned from the training data. Finally, the learned model is evaluated using a set of test data to predict the class labels for the DBN classifier and hence assess the classification accuracy. Experimental results on two datasets depict the superiority of the proposed DeepSGW framework in comparison with the other state-of-the-art methods.

- In Chapter 4, we present a spectral graph wavelet framework for the analysis and design of efficient shape signatures for nonrigid 3D shape retrieval [71]. Although this work focuses primarily on shape retrieval, our approach is, however, fairly general and can be used to address other 3D shape analysis problems. In a bid to capture the global and local geometry of 3D shapes, we employ a multi-resolution signature via a Mexican hat wavelet generating kernel. Then, mid-level features are obtained by embedding local descriptors into the visual vocabulary space using the soft-assignment coding step of the BoF model. Finally, by aggregating mid-level features weighted by a geodesic exponential kernel, a global descriptor is constructed. The parameters of the proposed signature can be easily determined as a tradeoff between effectiveness and compactness. Experimental results on two standard 3D shape benchmarks demonstrate that the proposed shape retrieval approach outperforms other state-of-the-art methods.

- In Chapter 5, we summarize the contributions of this thesis, and propose several future research directions that are directly or indirectly related to the ideas developed therein.

CHAPTER 2

Spectral Graph Wavelets for Shape Classification

Spectral shape descriptors have been used extensively in a broad spectrum of geometry processing applications ranging from shape retrieval and segmentation to classification. In this chapter, we propose a spectral graph wavelet approach for 3D shape classification using the BoF paradigm. In an effort to capture both the local and global geometry of a 3D shape, we present a three-step feature description framework. First, local descriptors are extracted via the spectral graph wavelet transform having the Mexican hat wavelet as a generating kernel. Second, mid-level features are obtained by embedding local descriptors into the visual vocabulary space using the soft-assignment coding step of the BoF model. Third, a global descriptor is constructed by aggregating mid-level features weighted by a geodesic exponential kernel, resulting in a matrix representation that describes the frequency of appearance of nearby codewords in the vocabulary. Experimental results on two standard 3D shape benchmarks demonstrate the effectiveness of the proposed classification approach in comparison with state-of-the-art methods.

2.1 Introduction

The recent surge of interest in the spectral analysis of the LBO has resulted in a glut of spectral shape signatures that have been successfully applied to a broad range of areas, including manifold learning [72], object recognition and deformable shape analysis [9, 13, 34, 35, 70, 73], medical imaging [74], multimedia protection [75], and shape classification [76]. The diversified nature of these applications is a powerful testimony of the practical usage of spectral shapes signatures, which are usually defined as feature vectors representing local and/or global characteristics of a shape and may be broadly classified into two main categories: local and global descriptors. Local

descriptors (also called point signatures) are defined on each point of the shape and often represent the local structure of the shape around that point, while global descriptors are usually defined on the entire shape.

Most point signatures may easily be aggregated to form global descriptors by integrating over the entire shape. Rustamov [9] proposed a local feature descriptor referred to as the global point signature (GPS), which is a vector whose components are scaled eigenfunctions of the LBO evaluated at each surface point. The GPS signature is invariant under isometric deformations of the shape, but it suffers from the problem of eigenfunctions' switching whenever the associated eigenvalues are close to each other. This problem was lately well handled by the heat kernel signature (HKS) [10], which is a temporal descriptor defined as an exponentially-weighted combination of the LBO eigenfunctions. HKS is a local shape descriptor that has a number of desirable properties, including robustness to small perturbations of the shape, efficiency and invariance to isometric transformations. The idea of HKS was also independently proposed by Gębal *et al.* [77] for 3D shape skeletonization and segmentation under the name of auto diffusion function. From the graph Fourier perspective, it can be seen that HKS is highly dominated by information from low frequencies, which correspond to macroscopic properties of a shape. To give rise to substantially more accurate matching than HKS, the wave kernel signature (WKS) [12] was proposed as an alternative in an effort to allow access to high-frequency information. Using the Fourier transform's magnitude, Kokkinos *et al.* [11] introduced the scale invariant heat kernel signature (SIHKS), which is constructed based on a logarithmically sampled scale-space.

One of the simplest spectral shape signatures is Shape-DNA [13], which is an isometry-invariant global descriptor defined as a truncated sequence of the LBO eigenvalues arranged in increasing order of magnitude. Gao *et al.* [76] developed a variant of Shape-DNA, referred to as compact Shape-DNA (cShape-DNA), which is an isometry-invariant signature resulting from applying the discrete Fourier transform to the area-normalized eigenvalues of the LBO. Chaudhari *et al.* [74] presented a slightly modified version of the GPS signature by setting the LBO eigenfunctions to unity. This signature, called GPS embedding, is defined as a truncated sequence of inverse square roots of the area-normalized eigenvalues of the LBO. A comprehensive list of spectral descriptors can be found in [78, 79].

From the graph Fourier perspective, it can be seen that HKS is highly dominated by information from low frequencies, which correspond to macroscopic properties of a shape. Wavelet analysis has some major advantages over Fourier transform, which makes it an interesting alternative for many applications. In particular, unlike the Fourier transform, wavelet analysis is able to perform local analysis and also makes it possible to perform a multiresolution analysis. Classical wavelets are constructed by translating and scaling a mother wavelet, which is used to generate a set of functions through the scaling and translation operations. The wavelet transform coefficients

are then obtained by taking the inner product of the input function with the translated and scaled waveforms. The application of wavelets to graphs (or triangle meshes) is, however, problematic and not straightforward due in part to the fact that it is unclear how to apply the scaling operation on a signal (or function) defined on the mesh vertices. To tackle this problem, Coifman *et al.* [33] introduced the diffusion wavelets, which generalize the classical wavelets by allowing for multiscale analysis on graphs. The construction of diffusion wavelets interacts with the underlying graph through repeated applications of a diffusion operator, which induces a scaling process. Hammond *et al.* [66] showed that the wavelet transform can be performed in the graph Fourier domain, and proposed a spectral graph wavelet transform that is defined in terms of the eigensystem of the graph Laplacian matrix. More recently, a spectral graph wavelet signature (SGWS) was introduced in [14, 80]. SGWS is a multiresolution local descriptor that is not only isometric invariant, but also compact, easy to compute and combines the advantages of both band-pass and low-pass filters.

A popular approach for transforming local descriptors into global representations that can be used for 3D shape recognition and classification is the bag-of-features (BoF) model [35]. The task in the shape classification problem is to assign a shape to a class chosen from a predefined set of classes. The BoF model represents each shape in the training dataset as a collection of unordered feature descriptors extracted from local areas of the shape, just as words are local features of a document. A baseline BoF approach quantizes each local descriptor to its nearest cluster center using K-means clustering and then encodes each shape as a histogram over cluster centers by counting the number of assignments per cluster. These cluster centers form a visual vocabulary or codebook whose elements are often referred to as visual words or codewords.

Although the BoF paradigm has been shown to provide significant levels of performance, it does not, however, take into consideration the spatial relations between features, which may have an adverse effect not only on its descriptive ability but also on its discriminative power. To account for the spatial relations between features, Bronstein *et al.* introduced a generalization of a BoF, called spatially sensitive bags-of-features (SS-BoF) [35]. The SS-BoF is a global descriptor defined in terms of mid-level features and the heat kernel, and can be represented by a square matrix whose elements represent the frequency of appearance of nearby codewords in the vocabulary. In the same spirit, Bu *et al.* [81] recently proposed the geodesic-aware bags-of-features (GA-BoF) for 3D shape classification by replacing the heat kernel in SS-BoF with a geodesic exponential kernel.

In this thesis, we propose a 3D shape classification approach, called SGWC-BoF, which employs spectral graph wavelet codes (SGWC) obtained from spectral graph wavelet signatures (i.e. local descriptors) via the soft-assignment coding step of the BoF model in conjunction with a geodesic exponential kernel for capturing the spatial relations between features. Broadly speaking, shape classification is the process of organizing a dataset of shapes into a known number of classes, and the task is to assign new shapes to one of these classes. It is common practice in classification to

randomly split the available data into training and test sets. Classification aims to learn a classifier (also called predictor or classification model) from labeled training data. The training data consist of a set of training examples or instances that are labeled with predefined classes. The resulting, trained model is subsequently applied to the test data to classify future (unseen) data instances into these classes. The test data, which consists of data instances with unknown class labels, is used to evaluate the performance of the classification model and determine its accuracy in terms of the number of test instances correctly or incorrectly predicted by the model. A good classifier should result in high accuracy, or equivalently, in few misclassifications.

In addition to taking into consideration the spatial relations between features via a geodesic exponential kernel, the proposed approach performs classification on SGWC, thereby seamlessly capturing the similarity between these mid-level features. We not only show that our formulation allows us to take into account the spatial layout of features, but we also demonstrate that the proposed framework yields better classification accuracy results compared to state-of-the-art methods, while remaining computationally attractive.

The rest of this chapter is organized as follows. In Section 2.2, we present our proposed framework for 3D shape classification and its main algorithmic steps. Also, the notion of support vector machine (SVMs) and the bag-of-features (BoF) paradigm are explained in this section. Experimental results are discussed in Section 2.3.

2.2 Method

In this section, we provide a detailed description of our 3D shape classification method that utilizes spectral graph wavelets in conjunction with the BoF paradigm. Each 3D shape in the dataset is first represented by local descriptors, which are arranged into a spectral graph wavelet signature matrix. Then, we perform soft-assignment coding by embedding local descriptors into the visual vocabulary space, resulting in mid-level features which we refer to as spectral graph wavelet codes (SGWC). It is important to point out that the vocabulary is computed offline by concatenating all the spectral graph wavelet signature matrices into a data matrix, followed by applying the K-means algorithm to find the data cluster centers.

In a bid to capture the spatial relations between features, we compute a global descriptor of each shape in terms of a geodesic exponential kernel and mid-level features, resulting in a SGWC-BoF matrix which is then transformed into a SGWC-BoF vector by stacking its columns one underneath the other. The last stage of the proposed approach is to perform classification on the SGWC-BoF vectors using a classification algorithm. The flowchart of the proposed framework is depicted in Figure 2.1.

Multiclass support vector machines (SVMs) are arguably the most popular and effective super-

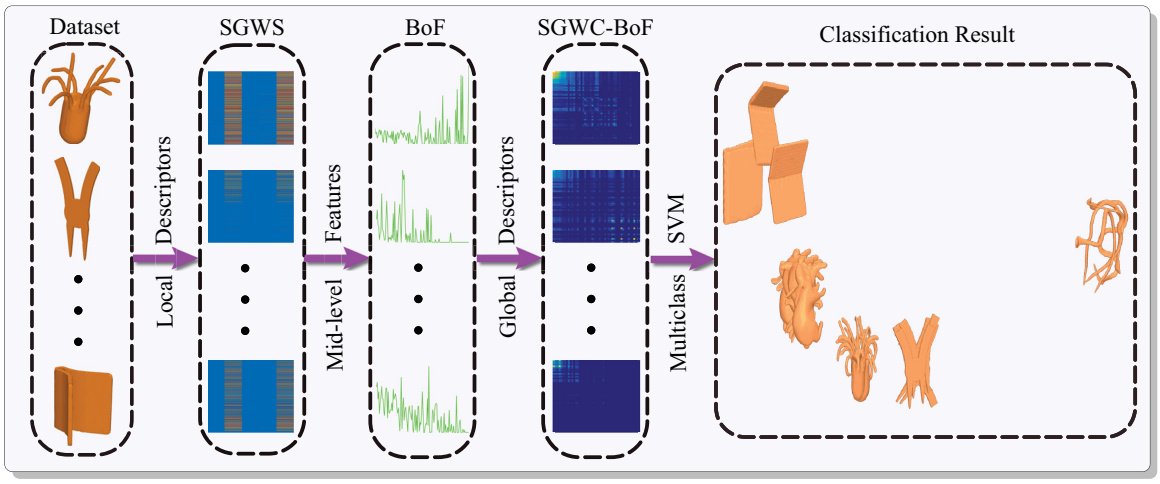


Figure 2.1: Flowchart of the proposed approach.

vised learning methods used for classification. Broadly speaking, supervised learning algorithms consist of two main steps: training step and test step. In the training step, a classification model (classifier) is learned from the training data by a learning algorithm (e.g., SVMs). In the test step, the learned model is evaluated using a set of test data to predict the class labels for the classifier and hence assess the classification accuracy.

2.2.1 Local Descriptors

Wavelets are useful in describing functions at different levels of resolution. To characterize the localized context around a mesh vertex $j \in \mathcal{V}$, we assume that the signal on the mesh is a unit impulse function, that is $f(i) = \delta_j(i)$ at each mesh vertex $i \in \mathcal{V}$. Thus, it follows from (1.20) that the spectral graph wavelet coefficients are

$$W_{\delta_j}(t, j) = \langle \boldsymbol{\delta}_j, \boldsymbol{\psi}_{t,j} \rangle = \sum_{\ell=1}^{m} a_j^2 g(t\lambda_\ell) \varphi_\ell^2(j), \tag{2.1}$$

and that the coefficients of the scaling function are

$$S_{\delta_j}(j) = \sum_{\ell=1}^{m} a_j^2 h(\lambda_\ell) \varphi_\ell^2(j). \tag{2.2}$$

Following the multiresolution analysis, the spectral graph wavelet and scaling function coefficients are collected to form the the spectral graph wavelet signature at vertex j as follows:

$$\mathbf{s}_j = \{\mathbf{s}_L(j) \mid L = 1, \ldots, R\}, \tag{2.3}$$

where R is the resolution parameter, and $\mathbf{s}_L(j)$ is the shape signature at resolution level L given by

$$\mathbf{s}_L(j) = \{W_{\delta_j}(t_k, j) \mid k = 1, \ldots, L\} \cup \{S_{\delta_j}(j)\}. \tag{2.4}$$

The wavelet scales t_k ($t_k > t_{k+1}$) are selected to be logarithmically equispaced between maximum and minimum scales t_1 and t_L, respectively. Thus, the resolution level L determines the resolution of scales to modulate the spectrum. At resolution $R = 1$, the spectral graph wavelet signature $\mathbf{s}_j$ is a 2-dimensional vector consisting of two elements: one element, $W_{\delta_j}(t_1, j)$, of spectral graph wavelet function coefficients and another element, $S_{\delta_j}(j)$, of scaling function coefficients. And at resolution $R = 2$, the spectral graph wavelet signature $\mathbf{s}_j$ is a 5-dimensional vector consisting of five elements (four elements of spectral graph wavelet function coefficients and one element of scaling function coefficients). In general, the dimension of a spectral graph wavelet signature $\mathbf{s}_j$ at vertex j can be expressed in terms of the resolution R as follows:

$$p = \frac{(R+1)(R+2)}{2} - 1. \tag{2.5}$$

Hence, for a p-dimensional signature $\mathbf{s}_j$, we define a $p \times m$ spectral graph wavelet signature matrix as $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_m)$, where $\mathbf{s}_j$ is the signature at vertex j and m is the number of mesh vertices. In our implementation, we used the Mexican hat wavelet as a kernel generating function g. In addition, we used the scaling function h given by

$$h(x) = \gamma \exp\left(-\left(\frac{x}{0.6\lambda_{\min}}\right)^4\right), \tag{2.6}$$

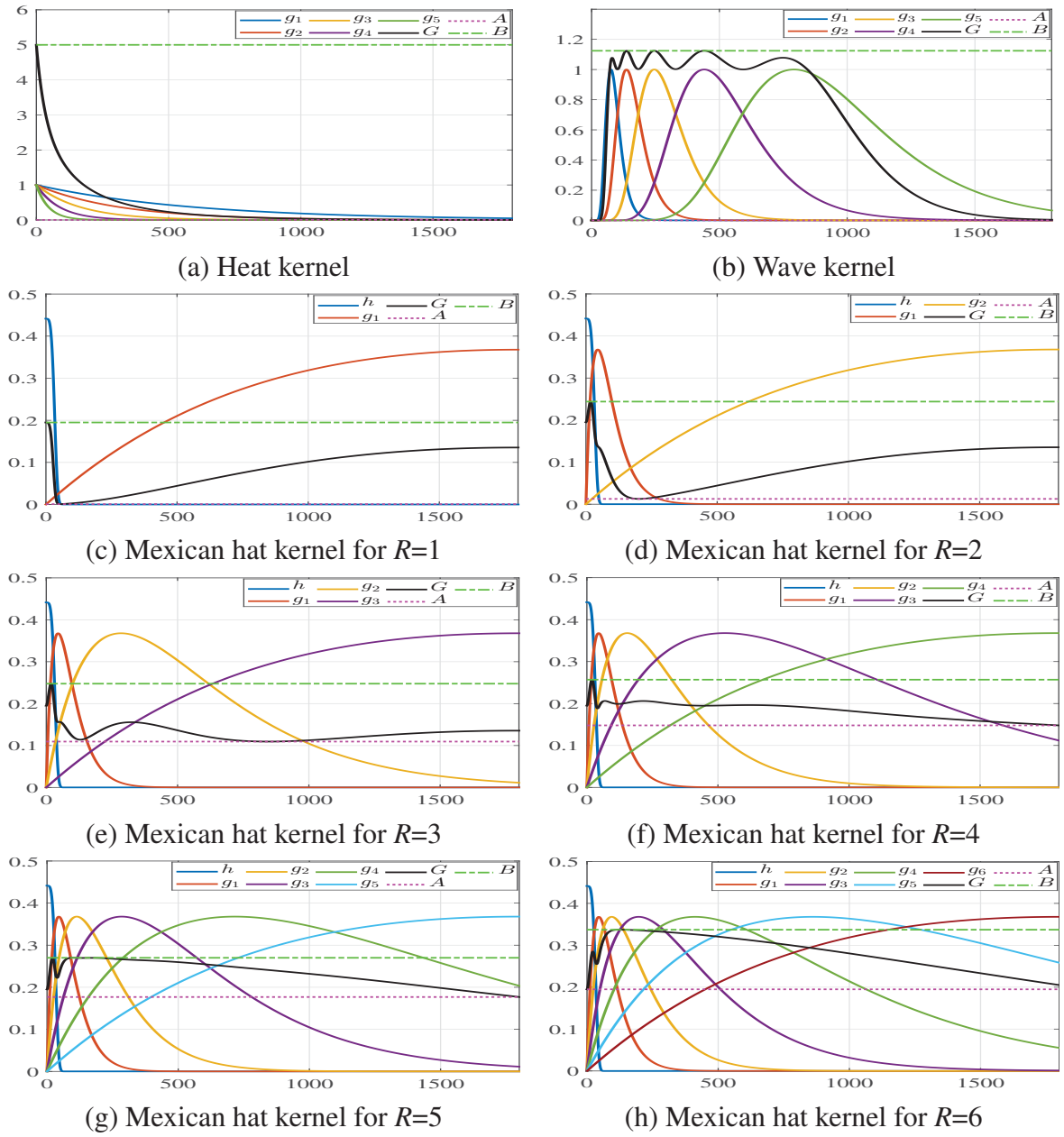
where $\lambda_{\min} = \lambda_{\max}/20$ and γ is set such that $h(0)$ has the same value as the maximum value of g. The maximum and minimum scales are set to $t_1 = 2/\lambda_{\min}$ and $t_L = 2/\lambda_{\max}$.

The geometry captured at each resolution R of the spectral graph wavelet signature can be viewed as the area under the curve G shown in Figure 2.2. For a given resolution R, we can understand the information from a specific range of the spectrum as its associated areas under G. As the resolution R increases, the partition of spectrum becomes tighter, and thus a larger portion of the spectrum is highly weighted.

2.2.2 Mid-Level Features

Broadly speaking, the BoF model aggregates local descriptors of a shape in an effort to provide a simple representation that may be used to facilitate comparison between shapes. We model each 3D shape as a triangle mesh $\mathbb{M}$ with m vertices. The BoF model consists of four main steps: feature extraction and description, codebook design, feature coding and feature pooling. The idea of the BoF paradigm on 3D shapes is illustrated in Figure 2.3.

Feature Extraction and Description In the BoF paradigm, a 3D shape $\mathbb{M}$ is represented as a collection of m local descriptors of the same dimension p, where the order of different feature vectors is of no importance. Local descriptors may be classified into two main categories: dense



(a) Heat kernel (b) Wave kernel

(c) Mexican hat kernel for R=1 (d) Mexican hat kernel for R=2

(e) Mexican hat kernel for R=3 (f) Mexican hat kernel for R=4

(g) Mexican hat kernel for R=5 (h) Mexican hat kernel for R=6

Figure 2.2: Spectrum modulation using different kernel functions; (a) heat kernel, (b) wave kernel, (c)-(h) Mexican hat kernel at various resolutions. The dark line is the squared sum function G, while the dash-dotted and the dotted lines are upper and lower bounds (B and A) of G, respectively.

and sparse. Dense descriptors are computed at each point (vertex) of the shape, while sparse descriptors are computed by identifying a set of salient points using a feature detection algorithm. In our proposed framework, we represent $\mathbb{M}$ by a $p \times m$ matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_m)$ of spectral graph wavelet signatures, where each p-dimensional feature vector $\mathbf{s}_i$ is a dense, local descriptor that encodes the local structure around the ith vertex of $\mathbb{M}$.

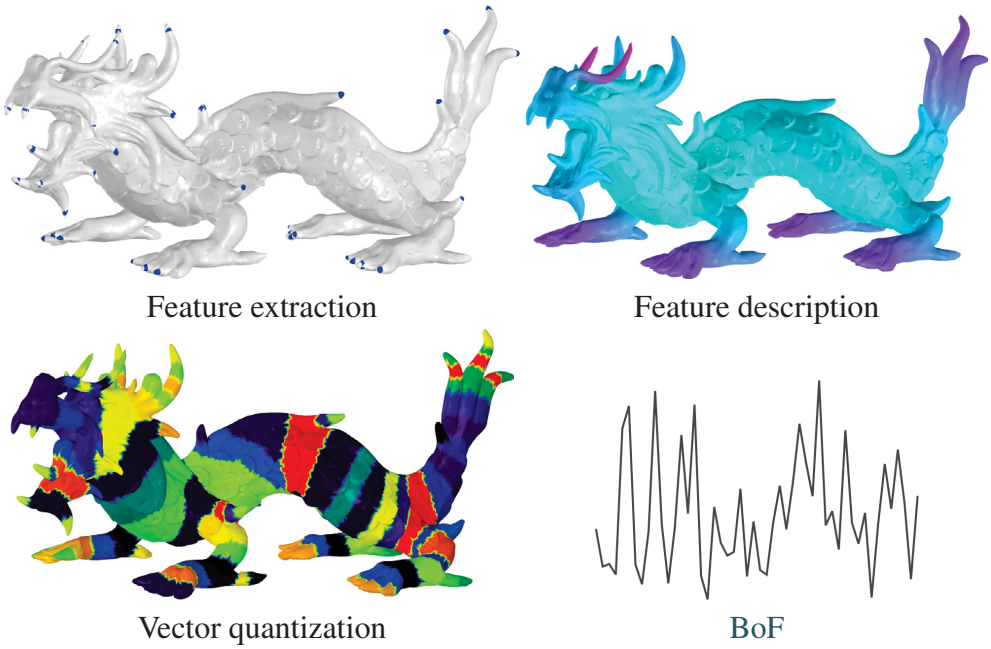


Figure 2.3: Flow of the BoF model.

Codebook Design A codebook (or visual vocabulary) is constructed via clustering by quantizing the m local descriptors (i.e. spectral graph wavelet signatures) into a certain number of codewords. These codewords are usually defined as the centers $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of k clusters obtained by performing an unsupervised learning algorithm (e.g., vector quantization via K-means clustering) on the signature matrix $\mathbf{S}$. The codebook is the set $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ of size k, which may be represented by a $p \times k$ vocabulary matrix $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_k)$.

Feature Coding The goal of feature coding is to embed local descriptors in the vocabulary space. Each spectral graph wavelet signature $\mathbf{s}_i$ is mapped to a codeword $\mathbf{v}_r$ in the codebook via the $k \times m$ cluster soft-assignment matrix $\mathbf{U} = (u_{ri}) = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$ whose elements are given by

$$u_{ri} = \frac{\exp(-\alpha \|\mathbf{s}_i - \mathbf{v}_r\|_2^2)}{\sum_{\ell=1}^{k} \exp(-\alpha \|\mathbf{s}_i - \mathbf{v}_\ell\|_2^2)}, \tag{2.7}$$

where $\|\cdot\|_2$ denotes the L_2-norm, and α is a smoothing parameter that controls the softness of the assignment. Unlike hard-assignment coding in which a local descriptor is assigned to the nearest cluster, soft-assignment coding assigns descriptors to every cluster center with different probabilities in an effort to improve quantization properties of the coding step. We refer to the coefficient vector $\mathbf{u}_i$ as the *spectral graph wavelet code* (SGWC) of the descriptor $\mathbf{s}_i$, with u_{ri} being the coefficient with respect to the codeword $\mathbf{v}_r$.

Histogram Representation (Feature Pooling) Each spectral graph wavelet signature is mapped to a certain codeword through the clustering process and the shape is then represented by the

histogram $\mathbf{h}$ of the codewords, which is a k-dimensional vector given by

$$\mathbf{h} = \mathbf{U}\mathbf{1}_m = (h_r)_{r=1,\ldots,k}, \tag{2.8}$$

where $h_r = \sum_{i=1}^{m} u_{ri}$. That is, the histogram consists of the column-sums of the cluster assignment matrix $\mathbf{U}$. Other feature pooling methods include average- and max-pooling. In general, any predefined pooling function that aggregates the information of different codewords into a single feature vector can be used.

2.2.3 Global Descriptors

A major drawback of the BoF model is that it only considers the distribution of the codewords and disregards all information about the spatial relations between features, and hence the descriptive ability and discriminative power of the BoF paradigm may be negatively impacted. To circumvent this limitation, various solutions have been recently proposed including the spatially sensitive bags-of-features (SS-BoF) [35] and geodesic-aware bags-of-features (GA-BoF) [81]. The SS-BoF, which is defined in terms of mid-level features and the heat kernel, can be represented by a square matrix whose elements represent the frequency of appearance of nearby codewords in the vocabulary. Similarly, the GA-BoF matrix is obtained by replacing the heat kernel in the SS-BoF with a geodesic exponential kernel. Unlike the heat kernel which is time-dependent, the geodesic exponential kernel avoids the possible effect of time scale and shape size [81]. In the same vein, we define a global descriptor of a shape as a $k \times k$ SGWC-BoF matrix $\mathbf{F}$ defined in terms of SGWC and a geodesic exponential kernel as follows:

$$\mathbf{F} = \mathbf{U}\mathbf{K}\mathbf{U}^{\mathsf{T}}, \tag{2.9}$$

where $\mathbf{U}$ is a $k \times m$ matrix of SGWC (i.e. mid-level featues), and $\mathbf{K} = (\kappa_{ij})$ is an $m \times m$ geodesic exponential kernel matrix whose elements are given by

$$\kappa_{ij} = \exp\left(-\frac{d_{ij}}{\epsilon}\right), \tag{2.10}$$

with d_{ij} denoting the geodesic distance between any pair of mesh vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, and ϵ is a positive, carefully chosen parameter that determines the width of the kernel. It should be noted that the geodesic distance is computed using Fast Marching algorithm [82]. Intuitively, the parameter ϵ controls the linearity of the kernel function, i.e. the larger the width, the more linear the function. It is worth pointing out that the proposed SGWC-BoF is similar in spirit to SS-BoF and GA-BoF. The main distinction of our work is that we use multiresolution local descriptors that may be regarded as generalized signatures for those in [35, 81]. In addition, our spectral graph wavelet signature combines the advantages of both band-pass and low-pass filters.

2.2.4 Multiclass Support Vector Machines

SVMs are supervised learning models that have proven effective in solving classification problems. SVMs are based upon the idea of maximizing the margin, i.e. maximizing the minimum distance from the separating hyperplane to the nearest example. Although SVMs were originally designed for binary classification, several extensions have been proposed in the literature to handle the multiclass classification. The idea of multiclass SVM is to decompose the multiclass problem into multiple binary classification tasks that can be solved efficiently using binary SVM classifiers. One of the simplest and most widely used coding designs for multiclass classification is the one-vs-all approach, which constructs K binary SVM classifiers such that for each binary classifier, one class is positive and the rest are negative. In other words, the one-vs-all approach requires K binary SVM classifiers, where the ith classifier is trained with positive examples belonging to class i and negative examples belonging to the remaining $K-1$ classes. When testing an unknown example, the classifier producing the maximum output (i.e. largest value of the decision function) is considered the winner, and this class label is assigned to that example.

2.2.5 Proposed Algorithm

Shape classification is a supervised learning method that assigns shapes in a dataset to target classes. The objective of 3D shape classification is to accurately predict the target class for each 3D shape in the dataset. Our proposed 3D shape classification algorithm consists of four main steps. The first step is to represent each 3D shape in the dataset by a spectral graph wavelet signature matrix, which is a feature matrix consisting of local descriptors. More specifically, let $\mathcal{D}$ be a dataset of n shapes modeled by triangle meshes $\mathbb{M}_1, \ldots, \mathbb{M}_n$. We represent each 3D shape in the dataset $\mathcal{D}$ by a $p \times m$ spectral graph wavelet signature matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_m) \in \mathbb{R}^{p \times m}$, where $\mathbf{s}_i$ is the p-dimensional local descriptor at vertex i and m is the number of mesh vertices.

In the second step, the spectral graph wavelet signatures $\mathbf{s}_i$ are mapped to high-dimensional mid-level feature vectors using the soft-assignment coding step of the BoF model, resulting in a $k \times m$ matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$ whose columns are the k-dimensional mid-level feature codes (i.e. SGWC). In the third step, the $k \times k$ SGWC-BoF matrix $\mathbf{F}$ is computed using the mid-level feature codes matrix and a geodesic exponential kernel, followed by reshaping $\mathbf{F}$ into a k^2-dimensional global descriptor $\mathbf{x}_i$. In the fourth step, the SGWC-BoF vectors $\mathbf{x}_i$ of all n shapes in the dataset are arranged into a $k^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. Finally, a one-vs-all multiclass SVM classifier is performed on the data matrix $\mathbf{X}$ to find the best hyperplane that separates all data points of one class from those of the other classes.

The task in multiclass classification is to assign a class label to each input example. More precisely, given a training data of the form $\mathcal{X}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i \in \mathbb{R}^{k^2}$ is the ith example

(i.e. SGWC-BoF vector) and $y_i \in \{1, \ldots, K\}$ is its ith class label, we aim at finding a learning model that contains the optimized parameters from the SVM algorithm. Then, the trained SVM model is applied to a test data $\mathcal{X}_{\text{test}}$, resulting in predicted labels $\hat{y}_i$ of new data. These predicted labels are subsequently compared to the labels of the test data to evaluate the classification accuracy of the model.

To assess the performance of the proposed framework, we employed two commonly-used evaluation criteria, the confusion matrix and accuracy, which will be discussed in more detail in the next section. The main algorithmic steps of our approach are summarized in Algorithm 1.

Algorithm 1 Proposed Algorithmic Steps

Input: Dataset $\mathcal{D} = \{\mathbb{M}_1, \ldots, \mathbb{M}_n\}$ of n 3D shapes
Output: n-dimensional vector $\hat{\mathbf{y}}$ containing predicted class labels for each 3D shape

1: **for** $i = 1$ to n **do**
2: {**Step 1**} Compute the $p \times m$ spectral graph wavelet signature matrix $\mathbf{S}_i$ of each shape $\mathbb{M}_i$
3: {**Step 2**} Apply soft-assignment coding to find the $k \times m$ mid-level feature matrix $\mathbf{U}_i$, where $k > p$
4: {**Step 3**} Compute the $k \times k$ SGWC-BoF matrix $\mathbf{F}_i$, and reshape it into a k^2-dimensional vector $\mathbf{x}_i$
5: **end for**
6: {**Step 4**} Arrange all the n SGWC-BoF vectors into a $k^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$
7: {**Step 5**} Perform multiclass SVM on $\mathbf{X}$ to find the n-dimensional vector $\hat{\mathbf{y}}$ of predicted class labels.

Remark: It is important to point out that in our implementation the vocabulary is computed offline by applying the K-means algorithm to the $p \times mn$ matrix obtained by concatenating all SGWS matrices of all n meshes in the dataset. As a result, the vocabulary is a matrix $\mathbf{V}$ of size $p \times k$, where $k > p$.

2.3 Experimental Results

In this section, we conduct extensive experiments to evaluate the performance of the proposed SGWC-BoF framework for 3D shape classification. The effectiveness of our approach is validated by performing a comprehensive comparison with several state-of-the-art methods.

Datasets The performance of the proposed framework is evaluated on two standard and publicly available 3D shape benchmarks: SHREC 2010 and SHREC 2011. Sample shapes from these two benchmarks are shown in Figure 2.4.

Performance Evaluation Measures In practice, the available data (which has classes) $\mathcal{X}$ for classification is usually split into two disjoint subsets: the training set $\mathcal{X}_{\text{train}}$ for learning, and the

Figure 2.4: Sample shapes from SHREC-2010 (top) and SHREC-2011 (bottom).

test set $\mathcal{X}_{\text{test}}$ for testing. The training and test sets are usually selected by randomly sampling a set of training instances from $\mathcal{X}$ for learning and using the rest of instances for testing. The performance of a classifier is then assessed by applying it to test data with known target values and comparing the predicted values with the known values. One important way of evaluating the performance of a classifier is to compute its confusion matrix (also called contingency table),

which is a $K \times K$ matrix that displays the number of correct and incorrect predictions made by the classifier compared with the actual classifications in the test set, where K is the number of classes.

Another intuitively appealing measure is the classification accuracy, which is a summary statistic that can be easily computed from the confusion matrix as the total number of correctly classified instances (i.e. diagonal elements of confusion matrix) divided by the total number of test instances. Alternatively, the accuracy of a classification model on a test set may be defined as follows

$$\text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}} = \frac{|\mathbf{x} \,:\, \mathbf{x} \in \mathcal{X}_{\text{test}} \wedge \hat{y}(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} \,:\, \mathbf{x} \in \mathcal{X}_{\text{test}}|}, \tag{2.11}$$

where $y(\mathbf{x})$ is the actual (true) label of $\mathbf{x}$, and $\hat{y}(\mathbf{x})$ is the label predicted by the classification algorithm. A correct classification means that the learned model predicts the same class as the original class of the test case. The error rate is equal to one minus accuracy.

Baseline Methods For each of the 3D shape benchmarks used for experimentation, we will report the comparison results of our method against various state-of-the-art methods, including Shape-DNA [13], compact Shape-DNA [76], GPS embedding [74], GA-BoF [81], and F1-, F2-, and F3-features [83]. The latter features, which are defined in terms of the Laplacian matrix eigenvalues, were shown to have good inter-class discrimination capabilities in 2D shape recognition [76], but they can easily be extended to 3D shape analysis using the eigenvalues of the LBO.

Implementation Details The experiments were conducted on a desktop computer with an Intel Core i5 processor running at 3.10 GHz and 8 GB RAM; and all the algorithms were implemented in MATLAB. The appropriate dimension (i.e. length or number of features) of a shape signature is problem-dependent and usually determined experimentally. For fair comparison, we used the same parameters that have been employed in the baseline methods, and in particular the dimensions of shape signatures. In our setup, a total of 201 eigenvalues and associated eigenfunctions of the LBO were computed. For the proposed approach, we set the resolution parameter to $R = 2$ (i.e. the spectral graph wavelet signature matrix is of size $5 \times m$, where m is the number of mesh vertices) and the kernel width of the geodesic exponential kernel to $\epsilon = 0.1$. Moreover, the parameter of the soft-assignment coding is computed as $\alpha = 1/(8\mu^2)$, where μ is the median size of the clusters in the vocabulary [35]. We also considered a linear kernel as SVM kernel function. For shape-DNA, GPS embedding, and F1-, F2-, and F3-features, the selected number of retained eigenvalues equals 10. As suggested in [76], the dimension of the compact Shape-DNA signature was set to 33.

2.3.1 SHREC-2010 Dataset

SHREC 2010 is a dataset of 3D shapes consisting of 200 watertight mesh models from 10 classes [84]. These models are selected from the McGill Articulated Shape Benchmark dataset.

Each class contains 20 objects with distinct postures. Moreover, each shape in the dataset has approximately $m = 1002$ vertices.

Performance Evaluation We randomly selected 50% shapes in the SHREC-2010 dataset to hold out for the test set, and the remaining shapes for training. That is, the test data consists of 100 shapes. A one-vs-all multiclass SVM is first trained on the training data to learn the model (i.e. classifier), which is subsequently used on the test data with known target values in order to predict the class labels. Figure 2.5 displays the confusion matrix for SHREC 2010 on the test data. This 10×10 confusion matrix shows how the predictions are made by the model. Its rows correspond to the actual (true) class of the data (i.e. the labels in the data), while its columns correspond to the predicted class (i.e. predictions made by the model). The value of each element in the confusion matrix is the number of predictions made with the class corresponding to the column for instances with the correct value as represented by the row. Thus, the diagonal elements show the number of correct classifications made for each class, and the off-diagonal elements show the errors made. As can be seen in Figure 2.5, the proposed approach was able to accurately classify all shapes in the test data, except the hand, octopus and spider models which were misclassified only once as teddy, crab and ant, respectively. Also, the human shape was misclassified three times as a spider. Such a good performance strongly suggests that our method captures well the discriminative features of the shapes.

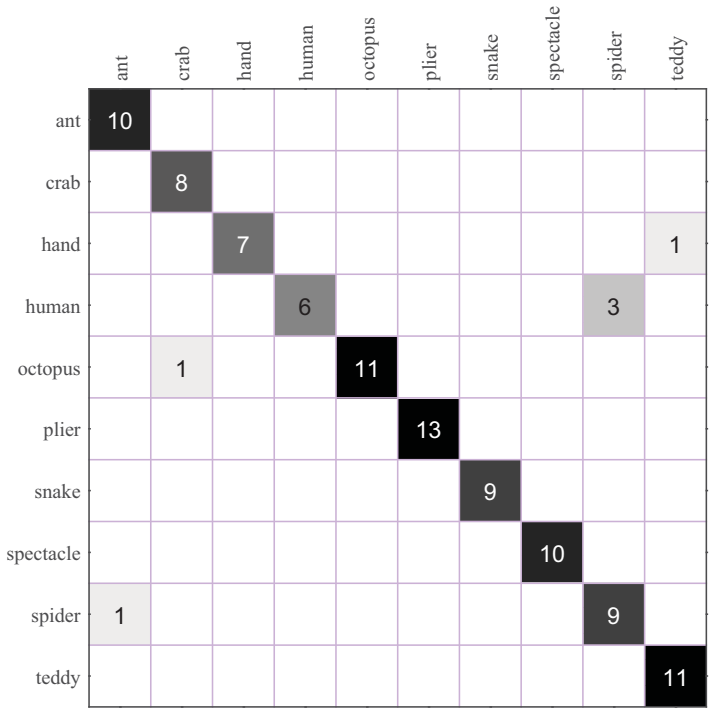


Figure 2.5: Confusion matrix for SHREC-2010 using linear multiclass SVM.

Results In our approach, each 3D shape in the SHREC-2010 dataset is represented by a 5×1002 matrix of spectral graph wavelet signatures. Setting the number of codewords to $k = 128$, we computed offline a 5×128 vocabulary matrix $\mathbf{V}$ via K-means clustering. The pre-computation of the vocabulary of size 128 took approximately 15 minutes. The soft-assignment coding of the BoF model yields a 128×1002 matrix $\mathbf{U}$ of spectral graph wavelet codes, resulting in a SGWC-BoF data matrix $\mathbf{X}$ of size $128^2 \times 200$.

We compared the proposed method to Shape-DNA, compact shape-DNA, GPS embedding, and F1-, F2-, and F3-features. In order to compute the accuracy, we repeated the experimental process 10 times with different randomly selected training and test data in an effort to obtain reliable results, and the accuracy for each run was recorded, then we selected the best result of each method. The classification accuracy results are summarized in Table 2.1, which shows the results of the baseline methods and the proposed framework. As can be seen, our SGWC-BoF method achieves better performance than Shape-DNA, compact Shape-DNA, GPS embedding, GA-BoF, and F1-, F2-, and F3-features. The proposed approach yields the highest classification accuracy of 95.66%, with performance improvements of 2.76% and 4.70% over the best baseline methods cShape-DNA and Shape-DNA, respectively. To speed-up experiments, all shape signatures were computed offline, albeit their computation is quite inexpensive due in large part to the fact that only a relatively small number of eigenvalues of the LBO need to be calculated.

Table 2.1: Classification accuracy results on the SHREC-2010 dataset. Boldface number indicates the best classification performance.

Method	Average accuracy %
Shape-DNA	90.96
cShape-DNA	92.90
GPS-embedding	88.87
F1-features	86.49
F2-features	84.11
F3-features	87.72
GA-BoF	86.02
SGWC-BoF	**95.66**

2.3.2 SHREC-2011 Dataset

SHREC 2011 is a dataset of 3D shapes consisting of 600 watertight mesh models, which are obtained from transforming 30 original models [85]. Each shape in the dataset has approximately $m = 1502$ vertices.

Performance Evaluation We randomly selected 50% shapes in the SHREC-2011 dataset to hold out for the test set, and the remaining shapes for training. That is, the test data consists of 300 shapes. First, we trained a one-vs-all multiclass SVM on the training data to learn the classification model. Then, we used the resulting, trained model on the test data to predict the class labels. With the exception of the horse, man and paper models, which were misclassified once as dog1, hand and bird1, respectively. Moreover, the ant shape was misclassified nine times as a spider. Therefore, the proposed approach was able to accurately classify all shapes in the test data, as shown in Figure 2.6.

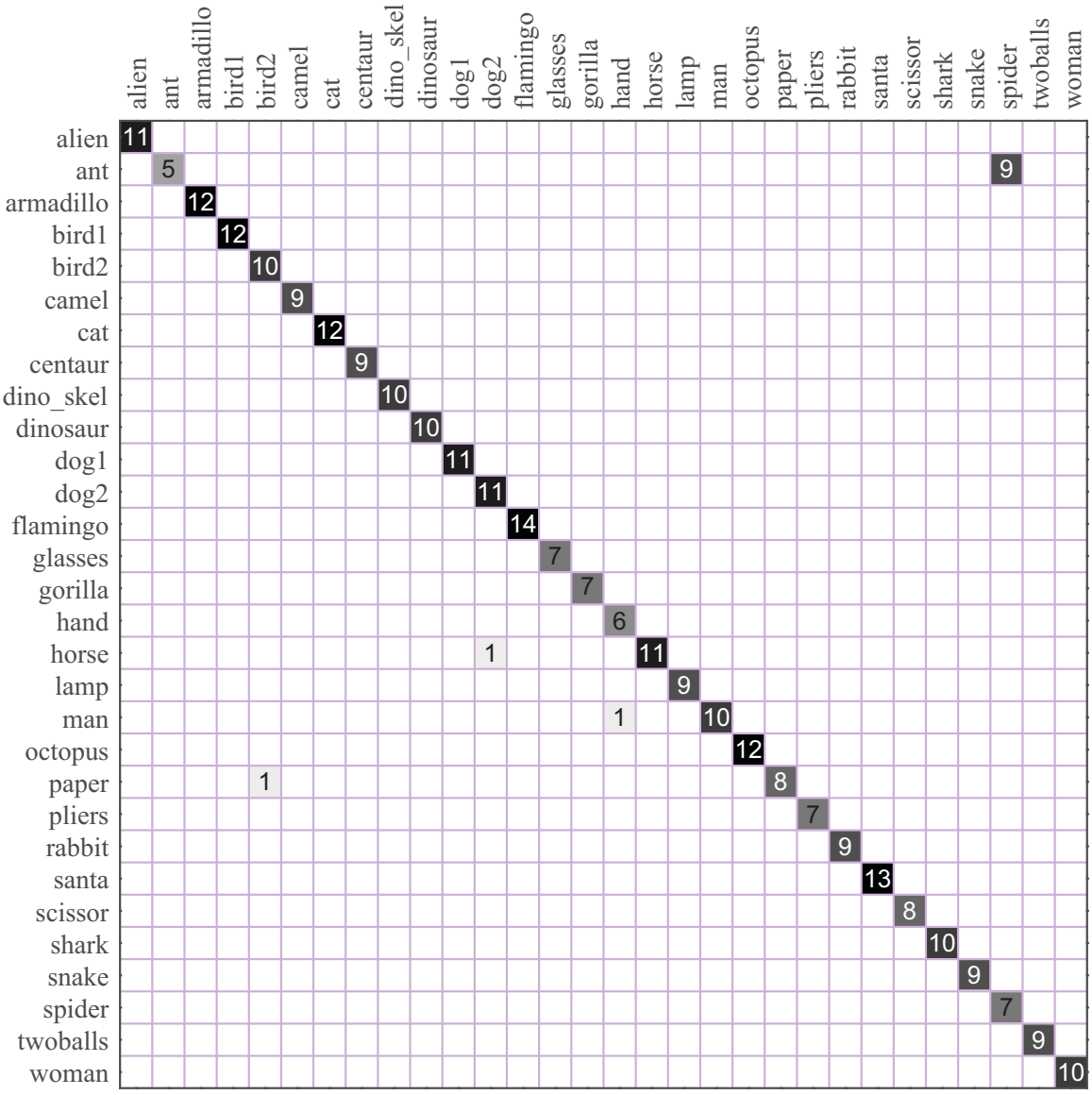


Figure 2.6: Confusion matrix for SHREC-2011 using linear multiclass SVM.

Results Following the setting of the previous experiment, each 3D shape in the SHREC-2011 dataset is represented by a 5×1502 spectral graph wavelet signature matrix. We pre-computed offline a vocabulary of size $k = 128$, and it took about 70 minutes. The soft-assignment coding yields a 128×1502 matrix $\mathbf{U}$ of mid-level features. Hence, the SGWC-BoF data matrix $\mathbf{X}$ for SHREC 2011 is of size $128^2 \times 600$. Figure 2.7 shows the spectral graph wavelet code matrices of two shapes from two different classes of SHREC 2011. As can be seen, the global descriptors are quite different and hence they may be used efficiently to discriminate between shapes in classification tasks.

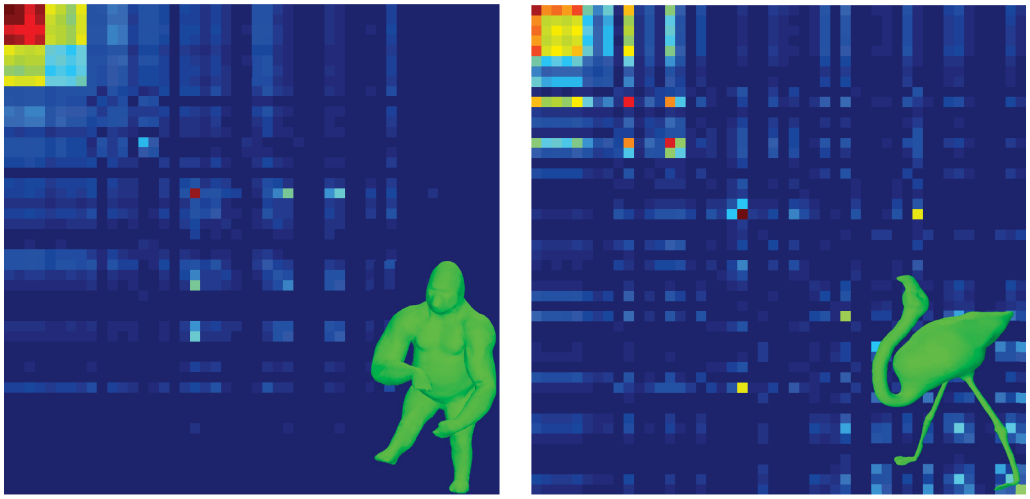


Figure 2.7: SGWC of two shapes (gorilla and flamingo) from two different classes of the SHREC-2011 dataset.

We repeated the experimental process 10 times with different randomly selected training and test data in an effort to obtain reliable results, and the accuracy for each run was recorded. The average accuracy results are reported in Table 2.2. As can be seen, the proposed method performs the best compared to all the seven baseline methods. The highest classification accuracy of 97.66% corresponds to our method, with performance improvements of 4.77% and 3.25% over the best performing baseline methods Shape-DNA and cShape-DNA, respectively.

2.3.3 Parameter Sensitivity

The proposed approach depends on two key parameters that affect its overall performance. The first parameter is the kernel width ϵ of the geodesic exponential kernel. The second parameter k is the size of the vocabulary, which plays an important role in the SGWC-BoF matrix $\mathbf{F}$. As shown in Figure 2.8, the best classification accuracy on SHREC 2011 is achieved using $\epsilon = 0.1$ and $k = 128$. In addition, the classification performance of proposed method is satisfactory for a

Table 2.2: Classification accuracy results on the SHREC-2011 dataset. Boldface number indicates the best classification performance.

Method	Average accuracy %
Shape-DNA	92.89
cShape-DNA	94.41
GPS-embedding	88.40
F1-features	91.90
F2-features	89.47
F3-features	92.48
GA-BoF	93.20
SGWC-BoF	**97.66**

wide range of parameter values, indicating the robustness of the proposed framework to the choice of these parameters.

2.4 Conclusions

In this chapter, we introduced a spectral graph wavelet framework for 3D shape classification that employs the BoF paradigm in an effort to design a global shape descriptor defined in terms of mid-level features and a geodesic exponential kernel. An important facet of our approach is the ability to combine the advantages of wave and heat kernel signatures into a compact yet discriminative descriptor, while allowing a multiresolution representation of shapes. The proposed spectral shape descriptor also combines the advantages of both band-pass and low-pass filters. In addition to taking into consideration the spatial relations between features via a geodesic exponential kernel, the proposed approach performs classification on SGWC, thereby seamlessly capturing the similarity between these midlevel features. We not only showed that our formulation allows us to take into account the spatial layout of features, but we also demonstrated that the proposed framework yields better classification accuracy results compared to state-of-the-art methods, while remaining computationally attractive. This better performance is largely attributed to the discriminative global descriptor constructed by aggregating mid-level features weighted by a geodesic exponential kernel. Extensive experiments were carried out on two standard 3D shape benchmarks to demonstrate the effectiveness of the proposed method and its robustness to the choice of parameters. We evaluated the results using several metrics, including the confusion matrix and average accuracy. For future work, we plan to apply the proposed approach to other 3D shape analysis problems.

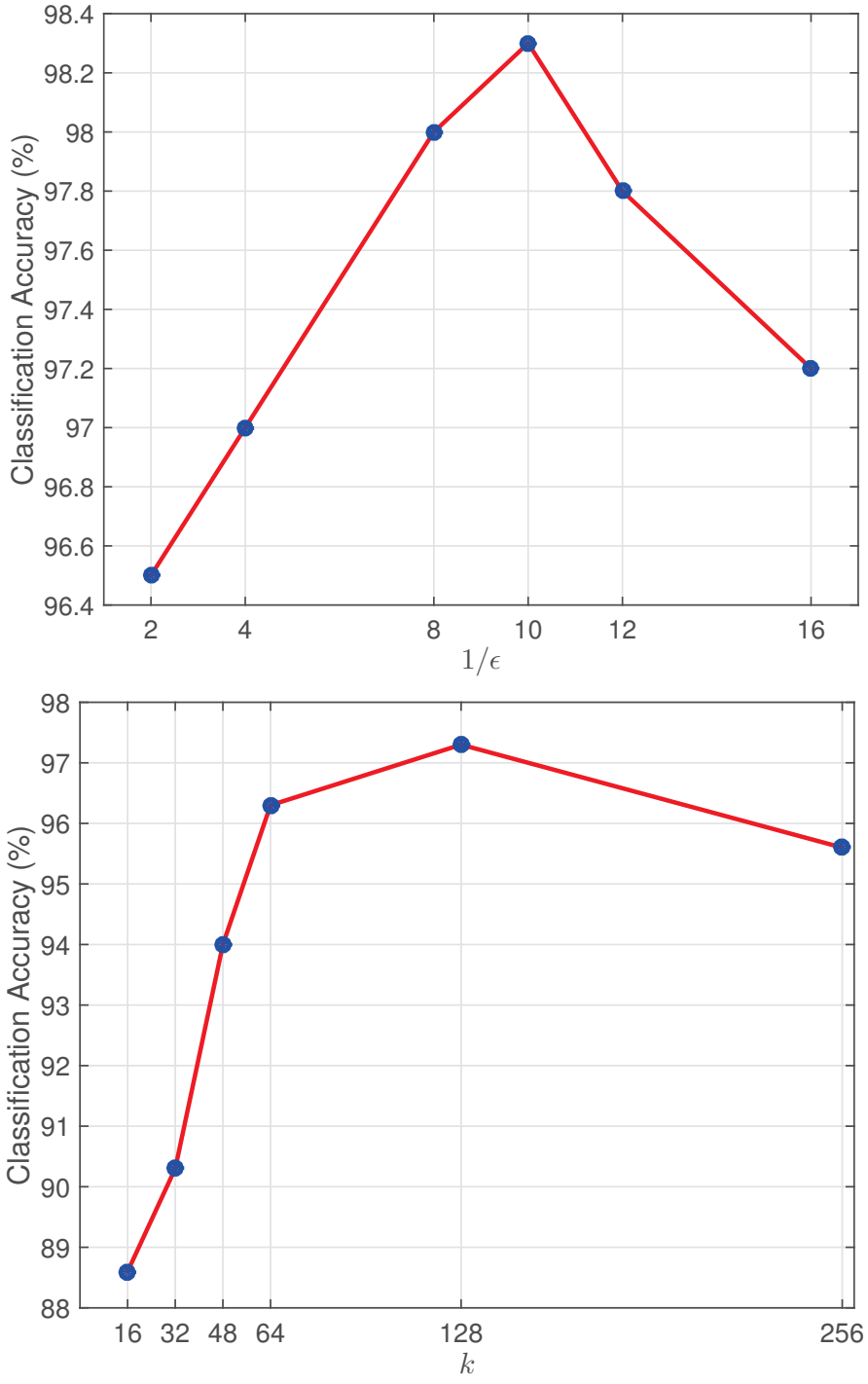


Figure 2.8: Effects of the parameters on the classification accuracy for SHREC 2011.

CHAPTER 3

Spectral Shape Classification using Deep Learning

The soaring popularity of deep learning in a wide variety of fields ranging from computer vision and speech recognition to self-driving vehicles has sparked a flurry of research interest from both academia and industry. In this chapter, we present a deep learning approach to 3D shape classification using spectral graph wavelets that are obtained from spectral graph wavelet signatures (i.e. local descriptors) via the soft-assignment coding step of the BoF model in conjunction with a geodesic exponential kernel which helps to capture the spatial relations between features. Experimental results on two different datasets will show our approach substantially outperforms state-of-the-art methods both in classification accuracy and in scalability.

3.1 Introduction

3D model classification is an intriguing and challenging problem that lies at the crossroads of computer vision, geometry processing and machine learning. While the overwhelming majority of prior work on 3D shape analysis has concentrated primarily on rigid shape classification, many real objects such as articulated motions of humans are nonrigid and hence can exhibit a variety of poses and deformations.

Over the past decade, deep neural networks have been successfully applied not only in classification tasks [86, 87], but also in regression [88], dimensionality reduction [89], modeling textures [90], modeling motion [91], object segmentation [92], information retrieval [93], 3D shape recognition [94, 95], robotics [96], natural language processing [97], and collaborative filtering [98]. Unlike conventional machine learning approaches which usually utilize shallow architectures, deep learning emulates the way human brain works and processes information through

multiple stages of transformation and representation. By applying deep architectures to learn features at multiple level of abstracts from data automatically, deep learning approaches allow a system to learn complex functions that directly map raw input data to the output, without relying on human-crafted features [99].

With improved computational power and an overwhelming availability of 3D shape data, the burgeoning area of deep learning has dramatically transformed how shape classification methods are studied due in large part to the recent theoretical developments in deep learning representations that are essential not only to improving the classification accuracy, but also to producing state-of-the-art results.

In this chapter, we introduce a deep learning framework, namely DeepSGW for 3D shape classification. As a machine learning paradigm, deep learning mimics the way the human brain works to varying degrees. The process of deep learning is hierarchical in the sense that it takes low-level features at the bottom layer and then constructs higher and higher level features through the composition of layers.

The proposed DeepSGW approach performs 3D shape classification on SGWC that are obtained from spectral graph wavelet signatures (i.e. local descriptors) via the soft-assignment coding step of the BoF model in conjunction with a geodesic exponential kernel for capturing the spatial relations between features. In addition to taking into consideration the spatial relations between features via a geodesic exponential kernel, the proposed approach performs classification on SGWC, thereby seamlessly capturing the similarity between these mid-level features. We not only show that our formulation allows us to take into account the spatial layout of features, but we also demonstrate that the proposed framework yields better classification accuracy results compared to state-of-the-art methods, while remaining computationally attractive.

The remainder of this chapter is organized as follows. In Section 3.2, we briefly overview deep learning concept. In Section 3.3, we introduce a deep learning approach for 3D shape classification, and we discuss its main algorithmic steps in detail. Experimental results are extensively performed in Section 3.4.

3.2 Deep Learning

Deep learning has its roots on neural networks, and focuses on learning deep feature hierarchies with each layer learning new features from the output of its preceding layer [67, 100–103]. One of the most widely used deep architectures is the so-called deep belief network (DBN), which is a generative graphical model composed of a layer of visible units and multiple layers of hidden units, with unsupervised Restricted Boltzmann Machines (RBMs) as their building blocks [101]. Each layer of a DBN encodes correlations in the units in the previous layer, and the network parameters

obtained from the unsupervised learning phase are subsequently fine-tuned using backpropagation or other discriminative algorithms. The visible units correspond to the attributes of the input data vector (training example), and the hidden layers act as feature detectors.

3.2.1 Restricted Boltzmann Machines (RBMs)

An RBM is a two-layer, undirected graphical model that consists of a visible (input) layer of stochastic binary visible units $\mathbf{v} = (v_i)$ of dimension I and a hidden layer of stochastic binary hidden units $\mathbf{h} = (h_j)$ of dimension J, where v_i is the state of visible unit i and h_j is the state of hidden unit j. Each visible unit is connected to each hidden unit, but there are no intra-visible or intra-hidden connections, as shown in Figure 3.1. The symmetric connections between the two layers of an RBM are represented by an $I \times J$ weight matrix $\mathbf{W} = (w_{ij})$, where w_{ij} is a real-valued weight characterizing the relative strength of the undirected edge between visible unit i and hidden unit j.

In a standard RBM, the visible and hidden units are assumed to be binary, meaning that they can only be in one of two states $\{0, 1\}$, where 1 indicates the unit is "on" and 0 indicates the unit is "off" (i.e. activated or deactivated, respectively).

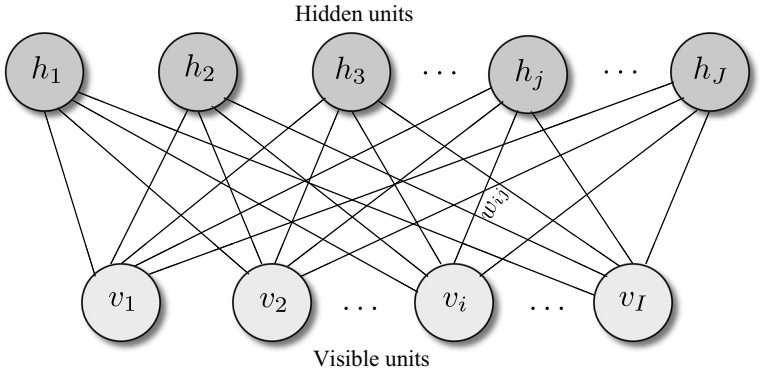


Figure 3.1: An RBM with visible units $\mathbf{v} = (v_i)$ and hidden units $\mathbf{h} = (h_j)$.

The energy of the joint configuration of the visible and hidden units $(\mathbf{v}, \mathbf{h})$ is given by

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{I}\sum_{j=1}^{J} v_i w_{ij} h_j - \sum_{j=1}^{J} b_j h_j - \sum_{i=1}^{I} c_i v_i = -\mathbf{v}^\intercal \mathbf{W}\mathbf{h} - \mathbf{b}^\intercal \mathbf{h} - \mathbf{c}^\intercal \mathbf{v}, \tag{3.1}$$

where and b_j is the real-valued bias of hidden unit j and c_i is the real-valued bias of visible unit i. This energy defines a joint probability distribution for configuration $(\mathbf{v}, \mathbf{h})$ as follows

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})), \tag{3.2}$$

where $Z = \sum_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}))$ is a normalization constant, obtained by summing up the energies of all possible $(\mathbf{v},\mathbf{h})$ configurations [101, 103]. Therefore, configurations with high energy are assigned low probability, while configurations with low energy are assigned high probability.

Because there are no intra-visible or intra-hidden connections in an RBM, the visible units are conditionally independent of one another given the hidden layer, and vice versa. For a simple RBM with Bernoulli distribution for both the visible and hidden layers (i.e. Bernoulli-Bernoulli RBM), the probability that h_j is activated, given visible unit vector $\mathbf{v}$ is

$$p(h_j = 1|\mathbf{v}) = \sigma\left(b_j + \sum_{i=1}^{I} w_{ij} v_i\right), \tag{3.3}$$

and the probability that v_i is activated, given hidden unit vector $\mathbf{h}$ is

$$p(v_i = 1|\mathbf{h}) = \sigma\left(c_i + \sum_{j=1}^{J} w_{ij} h_j\right), \tag{3.4}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the logistic sigmoidal activation function, whose output values lie in the interval $(0,1)$. In other words, the probability that a hidden unit is activated is independent of the states of the other hidden units, given the states of the visible units. Similarly, the probability that a visible unit is activated is independent of the states of the other visible units, given the states of the hidden units. This nice property of RBMs makes Gibbs sampling from (3.3) and (3.4) highly efficient, as one can sample all the hidden units simultaneously and then all the visible units simultaneously.

Training RBMs Given a training dataset $\mathbf{V}$ of visible vectors, RBMs are trained to maximize the average log probability (or equivalently minimize the energy) of $\mathbf{V}$ over the RBM's parameters $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$, i.e.

$$\arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{v}\in\mathbf{V}} \log p(\mathbf{v}), \tag{3.5}$$

where $p(\mathbf{v})$ is the marginal probability (over the visible vector $\mathbf{v}$) given by

$$\begin{aligned} p(\mathbf{v}) &= \sum_{\mathbf{h}} p(\mathbf{v},\mathbf{h}) \\ &= \frac{1}{Z}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h})) \\ &= \frac{1}{Z}\exp(\mathbf{c}^{\intercal}\mathbf{v}) \prod_{j=1}^{J}\left(1 + \exp\left(b_j + \sum_{i=1}^{I} w_{ij} v_i\right)\right). \end{aligned} \tag{3.6}$$

Taking the derivative of the log probability with respect to w_{ij} yields the following learning rule that performs stochastic gradient ascent in the log probability of the training data

$$\Delta w_{ij} = \varepsilon(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{model}}), \tag{3.7}$$

where ε is a learning rate, and $\langle\cdot\rangle_{\text{data}}$ and $\langle\cdot\rangle_{\text{model}}$ are the expectations under the distributions defined by the data and the model, respectively. Since $\langle\cdot\rangle_{\text{model}}$ is prohibitively expensive to compute, the single-step version (CD_1) of the contrastive divergence (CD) algorithm [101] is often used to optimize the model parameters (i.e. weights and biases) and it works well in practice. The new update rule becomes

$$\Delta w_{ij} = \varepsilon(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{recon}}), \tag{3.8}$$

where $\langle\cdot\rangle_{\text{recon}}$ is the expectation with respect to the distribution of samples from running the Gibbs sampler initialized at the data for one full step. The intuition behind the weight update rule is that the reconstructed data should be as close as possible to the input data. Similar updates rules are applied to the biases (i.e. bias vectors **b** and **c**).

The CD algorithm starts by setting the states of the visible units to a training vector. Given a randomly selected training example **v**, a binary vector of hidden units is obtained from sampling the conditional probability distribution (3.3) and then backpropagated using (3.4), resulting in a reconstruction of the original input data. After RBM training, hidden units can be considered to act as feature detectors, as they form a high-level representation of the input data.

Gaussian-Bernoulli RBMs If the visible units are real-valued, then exponential family distributions such as the Gaussian distribution are more suitable for modeling real-valued and count data (e.g., grayscale images and speech signals). Hence, for a Gaussian-Bernoulli RBM with Gaussian distribution for the visible layer and Bernoulli distribution for the hidden layer (i.e. $\mathbf{v} \in \mathbb{R}^I$ and $\mathbf{h} \in \{0,1\}^J$), the energy of the joint configuration $(\mathbf{v}, \mathbf{h})$ is defined as

$$E(\mathbf{v}, \mathbf{h}) = \sum_{i=1}^{I} \frac{(v_i - c_i)^2}{2\sigma_i^2} - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{j=1}^{J} b_j h_j, \tag{3.9}$$

where σ_i is the standard deviation associated with the Gaussian visible unit v_i, and the conditional probabilities are given by

$$p(h_j = 1|\mathbf{v}) = \sigma\left(b_j + \sum_{i=1}^{I} w_{ij}\frac{v_i}{\sigma_i}\right), \tag{3.10}$$

and

$$p(v_i = x|\mathbf{h}) = \mathcal{N}\left(c_i + \sigma_i \sum_{j=1}^{J} w_{ij} h_j, \sigma_i^2\right), \tag{3.11}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2. In other words, each visible unit is modeled with a Gaussian distribution given the hidden layer. In practice, it is a good idea, prior to fitting a DBN to input data, to standardize each input variable to have zero mean and unit standard deviation. Therefore, the energy of the joint configuration $(\mathbf{v}, \mathbf{h})$ becomes

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2}\|\mathbf{v} - \mathbf{c}\|_2^2 - \mathbf{v}^\intercal \mathbf{W}\mathbf{h} - \mathbf{b}^\intercal \mathbf{h}. \tag{3.12}$$

3.3 Method

In this section, we provide a detailed description of our DeepSGW classification method that utilizes spectral graph wavelets in conjunction with the BoF paradigm. Each 3D shape in the dataset is first represented by local descriptors, which are arranged into a spectral graph wavelet signature matrix. Then, we perform soft-assignment coding by embedding local descriptors into the visual vocabulary space, resulting in mid-level features which we refer to as SGWC. It is important to point out that the vocabulary is computed offline by concatenating all the spectral graph wavelet signature matrices into a data matrix, followed by applying the K-means algorithm to find the data cluster centers.

In an effort to capture the spatial relations between features, we compute a global descriptor of each shape in terms of a geodesic exponential kernel and mid-level features, resulting in a SGWC-BoF matrix which is then transformed into a SGWC-BoF vector by stacking its columns one underneath the other. The last stage of the proposed approach is to perform classification on the SGWC-BoF vectors using a deep belief networks (DBNs). The flowchart of the proposed framework is depicted in Figure 3.2.

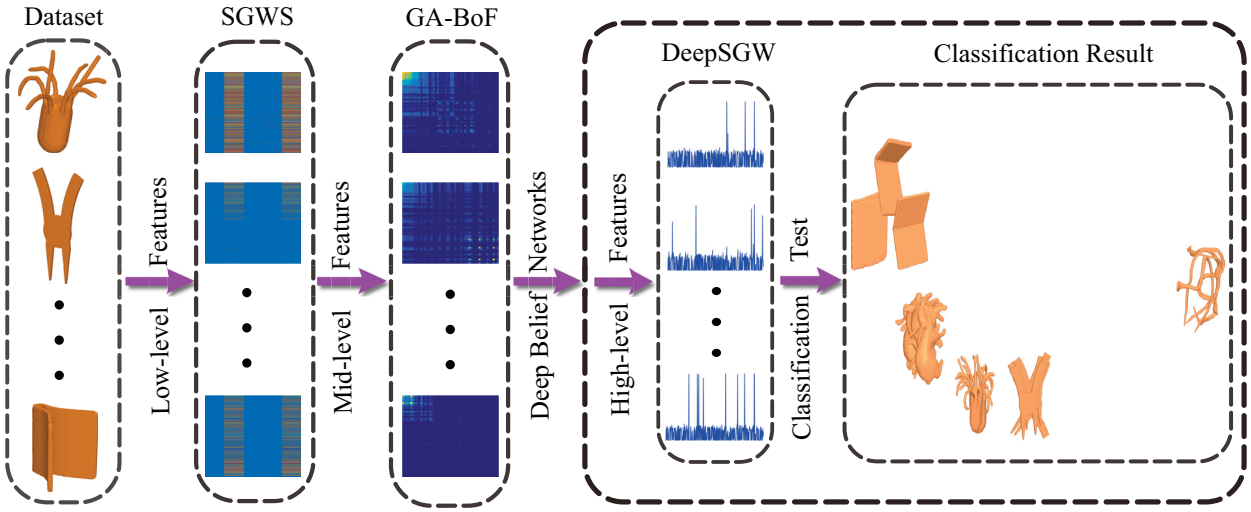


Figure 3.2: Flowchart of the proposed deep learning approach.

DBNs are highly effective supervised learning methods for classification. Broadly speaking, supervised learning algorithms consist of two main steps: training step and test step. In the training step, a classification model (classifier) is learned from the training data by a DBN learning algorithm. In the test step, the learned model is evaluated using a set of test data to predict the class labels for the DBN classifier and hence assess the classification accuracy.

3.3.1 Deep Belief Networks

A DBN is a probabilistic, generative model consisting of multiple layers of RBMs stacked on top of each other, starting with the visible (input) layer and first hidden layer that form the first RBM. It is made up of a visible layer $\mathbf{v}$ and S hidden layers $\mathbf{h}_s$, $s = 1, \ldots, S$, with the number of RBMs also equals S, which can be determined empirically to obtain the best model performance. Each RBM is trained in a greedy layer-wise manner, with the hidden layer of the sth RBM acting as a visible layer for the $(s+1)$th RBM, as shown in Figure 3.3.

A DBN consists of two main learning phases: pre-training and fine-tuning. Pre-training is an unsupervised phase that learns the weights (and biases) between layers from the bottom-up, i.e. from one layer at a time using an RBM on each layer. Pre-training treats the current layer as the hidden units of an RBM and the previous layer as the visible units of the same RBM. The pre-training starts by training the first RBM to obtain features in the first hidden layer from the training (input) data. In subsequent layers, the hidden activations of the previous layer are used as input data, i.e. the learned feature activations of one RBM are used as the input data for training the next RBM in the stack. Features at different layers contain different information about data with higher-level features constructed from lower-level features. This greedy, layer-wise training is iteratively performed until reaching the top hidden layer. To speed up the pretraining, it is common practice to subdivide the input data into mini-batches and the weights are updated after each mini-batch. The fine-tuning, on the other hand, is a supervised, discriminative phase that fine-tunes the model parameters (weights and biases) at the top layer by backpropagation error derivatives.

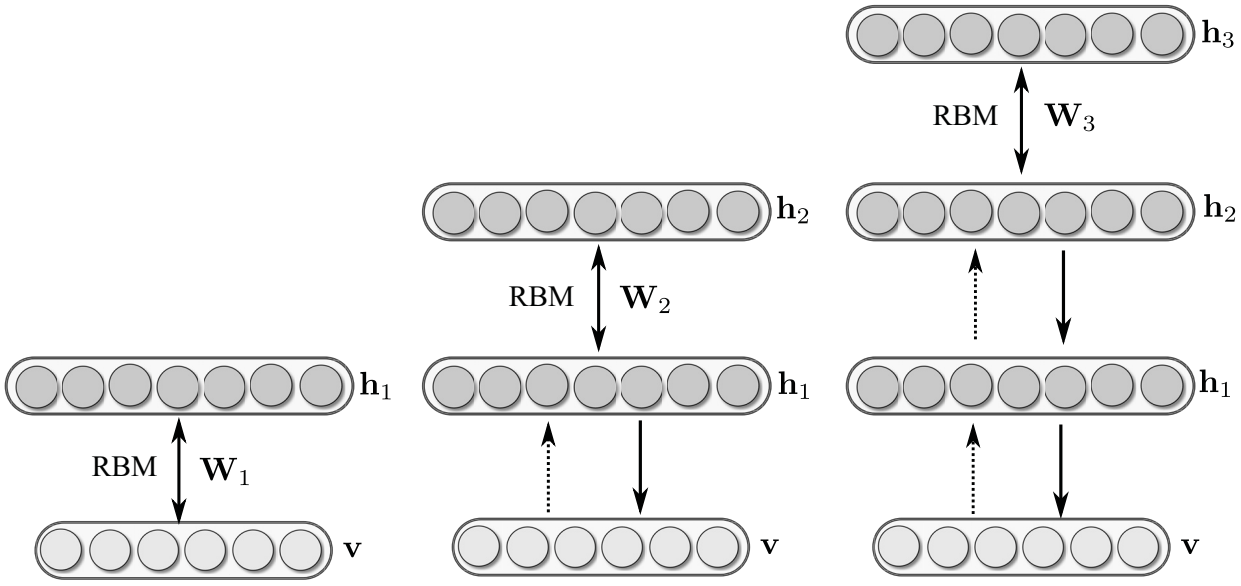


Figure 3.3: DBN architecture with three RBMs stacked on top of each other.

For classification tasks, an output layer $\mathbf{y} = (y_k)$ of K classes (units) is added on top of the stacked RBMs learned in the first phase to construct a discriminative model, where each output node of the softmax layer corresponds to a single unique class. The output (softmax) layer acts as a classifier, and is trained using labeled data. Each output node is represented by the output probability of each class label, and the probabilities will all sum up to 1. The node with the largest probability is usually used to predict the class of an instance (example) in the test set, and hence to compute the classification error/accuracy. More precisely, each output node y_k is represented by a probability p_k given by the softmax activation function

$$p_k = \frac{e^{a_k}}{\sum_{k=1}^{K} e^{a_k}}, \quad \text{with} \quad a_k = \sum_j w_{jk} h_j, \tag{3.13}$$

where $\mathbf{h} = (h_j)$ is the top hidden layer and $\mathbf{W} = (w_{jk})$ is a weight matrix of symmetric connections between the top hidden layer and the softmax layer. The predicted class $\hat{k}$ is then given by

$$\hat{k} = \arg\max_k p_k = \arg\max_k a_k. \tag{3.14}$$

It should be noted that the softmax activation function is a generalization of the logistic function (it reduces to the logistic function when there are only two classes). The purpose of the softmax function is to provide an estimate of the posterior probability of each class, i.e. the probability that an instance belongs in a particular class, given the data.

3.3.2 Proposed Algorithm

Shape classification is a supervised learning method that assigns shapes in a dataset to target classes. The objective of 3D shape classification is to accurately predict the target class for each 3D shape in the dataset. Our proposed 3D shape classification algorithm consists of four main steps. The first step is to represent each 3D shape in the dataset by a spectral graph wavelet signature matrix, which is a feature matrix consisting of local descriptors. More specifically, let $\mathcal{D}$ be a dataset of n shapes modeled by triangle meshes $\mathbb{M}_1, \ldots, \mathbb{M}_n$. We represent each 3D shape in the dataset $\mathcal{D}$ by a $p \times m$ spectral graph wavelet signature matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_m) \in \mathbb{R}^{p \times m}$, where $\mathbf{s}_i$ is the p-dimensional local descriptor at vertex i and m is the number of mesh vertices.

In the second step, the spectral graph wavelet signatures $\mathbf{s}_i$ are mapped to high-dimensional mid-level feature vectors using the soft-assignment coding step of the BoF model, resulting in a $k \times m$ matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$ whose columns are the k-dimensional mid-level feature codes (i.e. SGWC). In the third step, the $k \times k$ SGWC-BoF matrix $\mathbf{F}$ is computed using the mid-level feature codes matrix and a geodesic exponential kernel, followed by reshaping $\mathbf{F}$ into a k^2-dimensional global descriptor $\mathbf{x}_i$. In the fourth step, the SGWC-BoF vectors $\mathbf{x}_i$ of all n shapes in the dataset are

arranged into a $k^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. Finally, a DBN classifier is performed on the data matrix $\mathbf{X}$ to find the best hyperplane that separates all data points of one class from those of the other classes.

The task in multiclass classification is to assign a class label to each input example. More precisely, given a training data of the form $\mathcal{X}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i \in \mathbb{R}^{k^2}$ is the ith example (i.e. SGWC-BoF vector) and $y_i \in \{1, \ldots, K\}$ is its ith class label, we aim at finding a learning model that contains the optimized parameters from the DBN classification algorithm. Then, the trained DBN model is applied to a test data $\mathcal{X}_{\text{test}}$, resulting in predicted labels $\hat{y}_i$ of new data. These predicted labels are subsequently compared to the labels of the test data to evaluate the classification accuracy of the model.

To assess the performance of the proposed DeepSGW framework, we employed two commonly-used evaluation criteria, the confusion matrix and accuracy, which will be discussed in more detail in the next section. The main algorithmic steps of our approach are summarized in Algorithm 2.

Algorithm 2 Proposed Algorithmic Steps

Input: Dataset $\mathcal{D} = \{\mathbb{M}_1, \ldots, \mathbb{M}_n\}$ of n 3D shapes
Output: n-dimensional vector $\hat{\mathbf{y}}$ containing predicted class labels for each 3D shape

1: **for** $i = 1$ to n **do**
2: {**Step 1**} Compute the $p \times m$ spectral graph wavelet signature matrix $\mathbf{S}_i$ of each shape $\mathbb{M}_i$
3: {**Step 2**} Apply soft-assignment coding to find the $k \times m$ mid-level feature matrix $\mathbf{U}_i$, where $k > p$
4: {**Step 3**} Compute the $k \times k$ SGWC-BoF matrix $\mathbf{F}_i$, and reshape it into a k^2-dimensional vector $\mathbf{x}_i$
5: **end for**
6: {**Step 4**} Arrange all the n SGWC-BoF vectors into a $k^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$
7: {**Step 5**} Perform DBN classification on $\mathbf{X}$ to find the n-dimensional vector $\hat{\mathbf{y}}$ of predicted class labels.

Remark: It is important to point out that in our implementation the vocabulary is computed offline by applying the K-means algorithm to the $p \times mn$ matrix obtained by concatenating all SGWS matrices of all n meshes in the dataset. As a result, the vocabulary is a matrix $\mathbf{V}$ of size $p \times k$, where $k > p$.

3.4 Experimental Results

In this section, we organize extensive experiments for 3D shape classification problem to evaluate the proposed DeepSGW approach. The effectiveness of our method is validated by performing a comprehensive comparison with several state-of-the-art methods.

Datasets The performance of the proposed DeepSGW framework is evaluated on two standard and publicly available 3D shape benchmarks: SHREC 2010 and SHREC 2011. Sample shapes from these two benchmarks are shown in Figure 3.4.

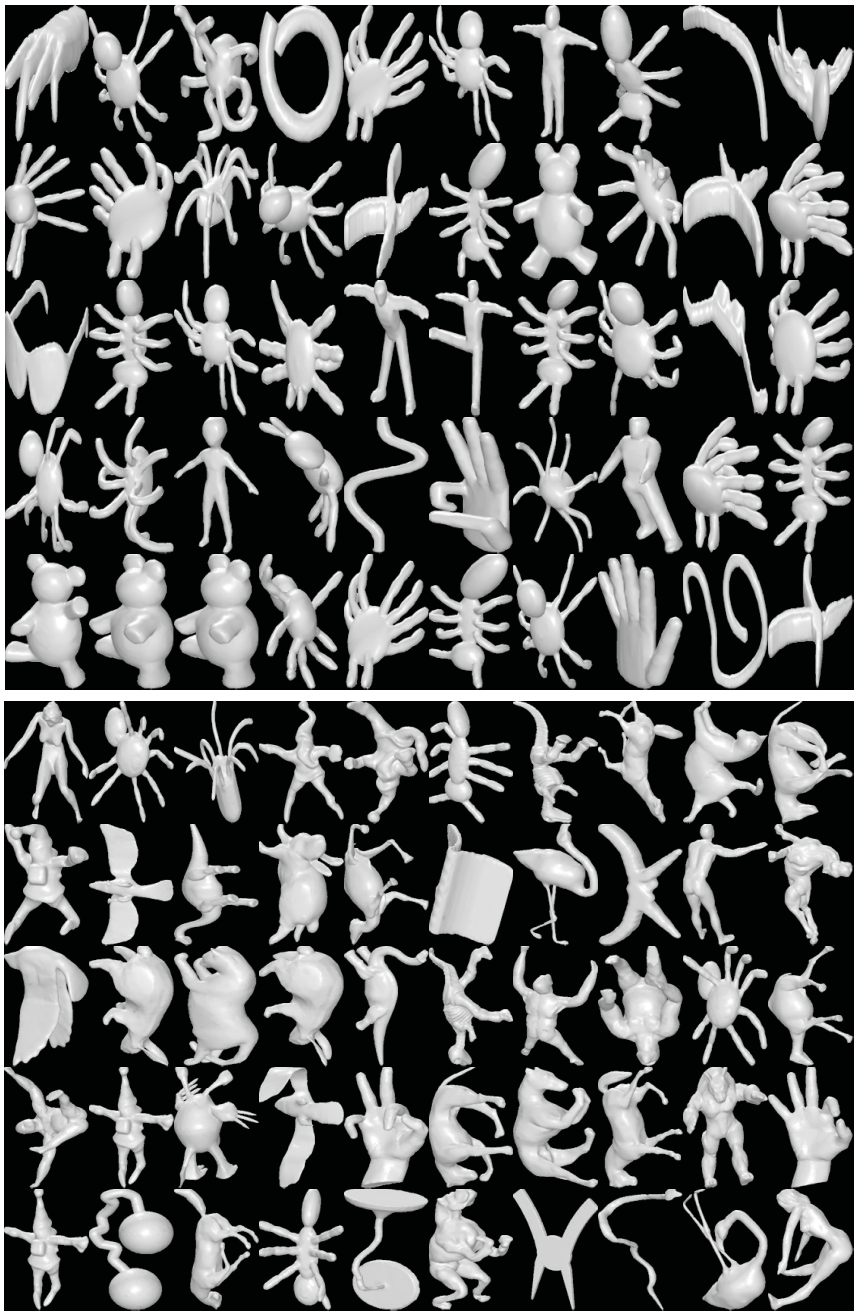


Figure 3.4: Sample shapes from SHREC-2010 (top) and SHREC-2011 (bottom).

Performance Evaluation Measures In practice, the available data (which has classes) $\mathcal{X}$ for classification is usually split into two disjoint subsets: the training set $\mathcal{X}_{\text{train}}$ for learning, and the test set $\mathcal{X}_{\text{test}}$ for testing. The training and test sets are usually selected by randomly sampling

a set of training instances from $\mathcal{X}$ for learning and using the rest of instances for testing. The performance of a classifier is then assessed by applying it to test data with known target values and comparing the predicted values with the known values. One important way of evaluating the performance of a classifier is to compute its confusion matrix (also called contingency table), which is a $K \times K$ matrix that displays the number of correct and incorrect predictions made by the classifier compared with the actual classifications in the test set, where K is the number of classes.

Another intuitively appealing measure is the classification accuracy, which is a summary statistic that can be easily computed from the confusion matrix as the total number of correctly classified instances (i.e. diagonal elements of confusion matrix) divided by the total number of test instances. Alternatively, the accuracy of a classification model on a test set may be defined as follows

$$\text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}} = \frac{|\mathbf{x} \,:\, \mathbf{x} \in \mathcal{X}_{\text{test}} \wedge \hat{y}(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} \,:\, \mathbf{x} \in \mathcal{X}_{\text{test}}|}, \tag{3.15}$$

where $y(\mathbf{x})$ is the actual (true) label of $\mathbf{x}$, and $\hat{y}(\mathbf{x})$ is the label predicted by the classification algorithm. A correct classification means that the learned model predicts the same class as the original class of the test case. The error rate is equal to one minus accuracy.

Baseline Methods For each of the 3D shape benchmarks used for experimentation, we will report the comparison results of our method against various state-of-the-art methods, including Shape-DNA [13], compact Shape-DNA [76], GPS embedding [74], GA-BoF [81], and F1-, F2-, and F3-features [83]. The latter features, which are defined in terms of the Laplacian matrix eigenvalues, were shown to have good inter-class discrimination capabilities in 2D shape recognition [76], but they can easily be extended to 3D shape analysis using the eigenvalues of the LBO.

Implementation Details The experiments were conducted on a desktop computer with an Intel Core i5 processor running at 3.10 GHz and 8 GB RAM; and all the algorithms were implemented in MATLAB. The appropriate dimension (i.e. length or number of features) of a shape signature is problem-dependent and usually determined experimentally. For fair comparison, we used the same parameters that have been employed in the baseline methods, and in particular the dimensions of shape signatures. In our setup, a total of 201 eigenvalues and associated eigenfunctions of the LBO were computed. For the proposed approach, we set the resolution parameter to $R = 2$ (i.e. the spectral graph wavelet signature matrix is of size $5 \times m$, where m is the number of mesh vertices) and the kernel width of the geodesic exponential kernel to $\epsilon = 0.1$. We used a DBN architecture consisting of two hidden layers. The first hidden layer has 400 units, while the second hidden layer contains 800 units. Each layer of hidden units learns to represent features that capture higher order correlations in the original input data. Moreover, the parameter of the soft-assignment coding is computed as $\alpha = 1/(8\mu^2)$, where μ is the median size of the clusters in the vocabulary [35]. For shape-DNA, GPS embedding, and F1-, F2-, and F3-features, the selected number of retained

eigenvalues equals 10. As suggested in [76], the dimension of the compact Shape-DNA signature was set to 33.

3.4.1 SHREC-2010 Dataset

SHREC 2010 is a dataset of 3D shapes consisting of 200 watertight mesh models from 10 classes [84]. These models are selected from the McGill Articulated Shape Benchmark dataset. Each class contains 20 objects with distinct postures. Moreover, each shape in the dataset has approximately $m = 1002$ vertices.

Performance Evaluation We randomly selected 50% shapes in the SHREC-2010 dataset to hold out for the test set, and the remaining shapes for training. That is, the test data consists of 100 shapes. A DBN with two hidden layers is first trained on the training data to learn the model (i.e. classifier), which is subsequently used on the test data with known target values in order to predict the class labels. Figure 3.5 displays the confusion matrix for SHREC 2010 on the test data. This 10×10 confusion matrix shows how the predictions are made by the model. Its rows correspond to the actual (true) class of the data (i.e. the labels in the data), while its columns correspond to the predicted class (i.e. predictions made by the model). The value of each element in the confusion matrix is the number of predictions made with the class corresponding to the column for instances with the correct value as represented by the row. Thus, the diagonal elements show the number of correct classifications made for each class, and the off-diagonal elements show the errors made. As shown in Figure 3.5, the proposed DeepSGW framework was able to classify all shapes of the test data with high accuracy, except the hand and human models which were misclassified only once as crab and hand, respectively. In addition, the octopus model was misclassified once as a human and also once as a crab. Such a good performance strongly suggests that our method captures well the discriminative features of the shapes.

Results In our DeepSGW approach, each 3D shape in the SHREC-2010 dataset is represented by a 5×1002 matrix of spectral graph wavelet signatures. Setting the number of codewords to $k = 48$, we computed offline a 5×48 vocabulary matrix $\mathbf{V}$ via K-means clustering. The soft-assignment coding of the BoF model yields a 48×1002 matrix $\mathbf{U}$ of SGWC, resulting in a SGWC-BoF data matrix $\mathbf{X}$ of size $48^2 \times 200$.

We compared the DeepSGW method to Shape-DNA, compact Shape-DNA, GPS embedding, GA-BoF, and F1-, F2-, and F3-features. In order to compute the accuracy, we repeated the experimental process 10 times with different randomly selected training and test data in an effort to obtain reliable results, and the accuracy for each run was recorded, then we selected the best result of each method. The classification accuracy results are summarized in Table 3.1, which shows the results of the baseline methods and the proposed framework. As can be seen, our method

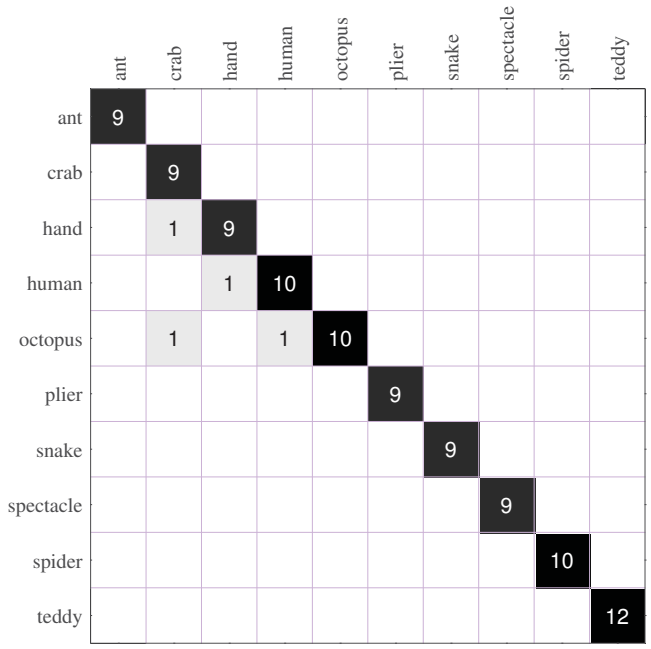


Figure 3.5: Confusion matrix for SHREC 2010 using the proposed DeepSGW approach.

achieves better performance than Shape-DNA, compact Shape-DNA, GPS embedding, GA-BoF, and F1-, F2-, and F3-features. The DeepSGW approach yields the highest classification accuracy of 96.00%, with performance improvements of 3.10% and 5.04% over the best baseline methods cShape-DNA and Shape-DNA, respectively. To speed-up experiments, all shape signatures were computed offline, albeit their computation is quite inexpensive due in large part to the fact that only a relatively small number of eigenvalues of the LBO need to be calculated.

Table 3.1: Classification accuracy results on the SHREC-2010 dataset. Boldface number indicates the best classification performance.

Method	**Average accuracy %**
Shape-DNA	90.96
cShape-DNA	92.90
GPS-embedding	88.87
F1-features	86.49
F2-features	84.11
F3-features	87.72
GA-BoF	86.02
DeepSGW	**96.00**

3.4.2 SHREC-2011 Dataset

SHREC 2011 is a dataset of 3D shapes consisting of 600 watertight mesh models, which are obtained from transforming 30 original models [85]. Each shape in the dataset has approximately $m = 1502$ vertices.

Performance Evaluation We randomly selected 50% shapes in the SHREC-2011 dataset to hold out for the test set, and the remaining shapes for training. That is, the test data consists of 300 shapes. First, we train a DBN with two hidden layers on the training data to learn the classification model. Then, we use the resulting, trained model on the test data to predict the class labels. With the exception of the cat model, which was misclassified once as a hand and also the bird2 model which was misclassified twice as bird1, the proposed DeepSGW approach was able to accurately classify all shapes in the test data, as shown in Figure 3.6.

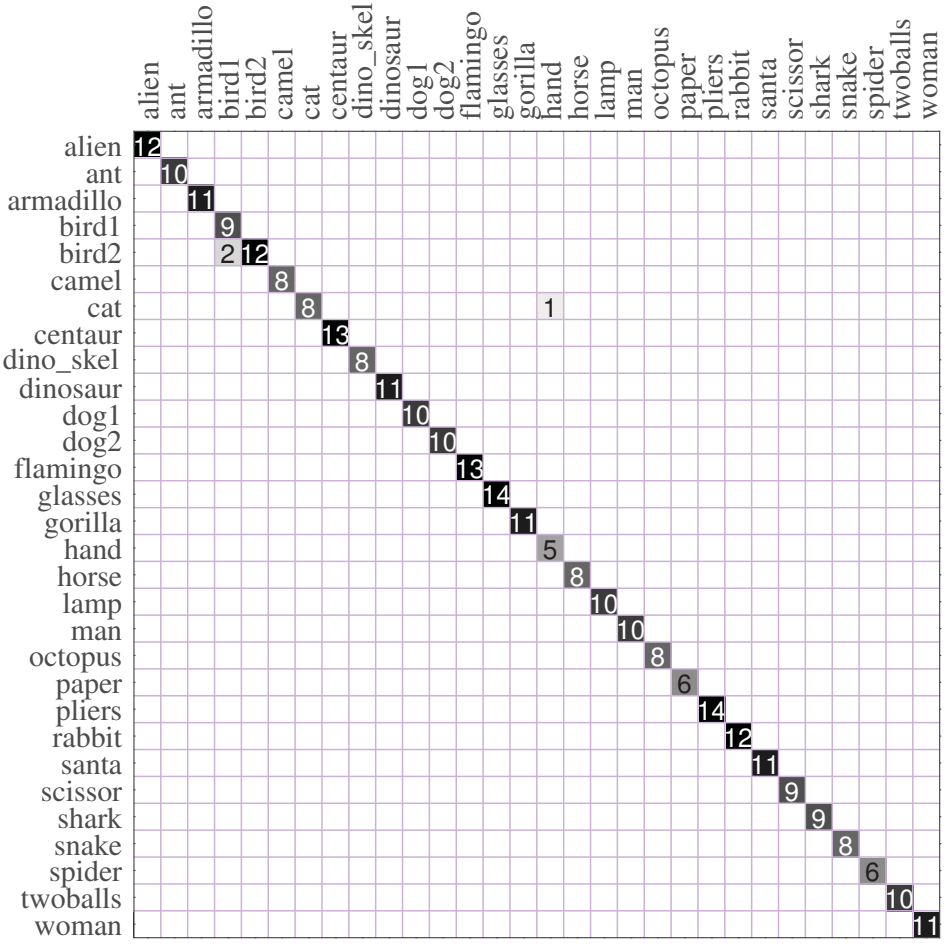


Figure 3.6: Confusion matrix for SHREC-2011 using the proposed DeepSGW approach.

Results Similar to the previous experiment, each 3D shape in the SHREC-2011 dataset is represented by a 5×1502 spectral graph wavelet signature matrix. We pre-computed offline a vocabulary of size $k = 48$. The soft-assignment coding yields a 48×1502 matrix $\mathbf{U}$ of mid-level features. Hence, the SGWC-BoF data matrix $\mathbf{X}$ for SHREC 2011 is of size $48^2 \times 600$. Figure 3.7 shows the SGWC matrices of two shapes (cat and centaur) from two different classes of SHREC 2011. As can be seen, the global descriptors are quite different and hence they may be used efficiently to discriminate between shapes in classification tasks.

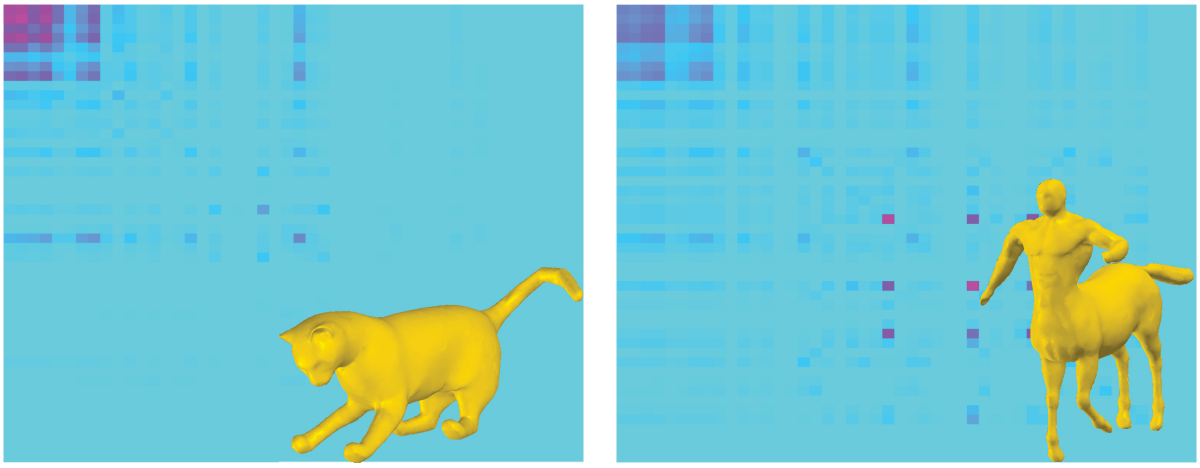


Figure 3.7: SGWC of two shapes (cat and centaur) from two different classes of the SHREC-2011 dataset.

We repeated the experimental process 10 times with different randomly selected training and test data in a bid to obtain reliable results, and the accuracy for each run was recorded. The average accuracy results are reported in Table 3.2. As can be seen, the DeepSGW approach outperforms all the seven baseline methods used for comparison. The highest classification accuracy of 99.79% corresponds to our method, with performance improvements of 6.50% and 5.29% compared to the best performing baseline methods GA-BoF and cShape-DNA, respectively.

Figure 3.8 shows the learned weights on DBN first and second layers for the the SHREC-2011 dataset. The DBN first layer consists of 400 units, while the second layer contains 800 units. Each square displays the incoming weights from all the visible units into one hidden unit. White encodes a positive weight and black encodes a negative weight. Figure 3.9 shows the first 64 training examples computed by DBN on the SHREC-2011 dataset.

Parameter Sensitivity The proposed approach depends on two key parameters that affect its overall performance. The first parameter is the kernel width ϵ of the geodesic exponential kernel. The second parameter k is the size of the vocabulary, which plays an important role in the SGWC-BoF matrix $\mathbf{F}$. As shown in Figure 3.10, the best classification accuracy on SHREC 2011 is achieved using $\epsilon = 0.1$ and $k = 48$. In addition, the classification performance of proposed

Table 3.2: Classification accuracy results on the SHREC-2011 dataset. Boldface number indicates the best classification performance.

Method	Average accuracy %
Shape-DNA	92.89
cShape-DNA	94.41
GPS-embedding	88.40
F1-features	91.90
F2-features	89.47
F3-features	92.48
GA-BoF	93.20
DeepSGW	**99.70**

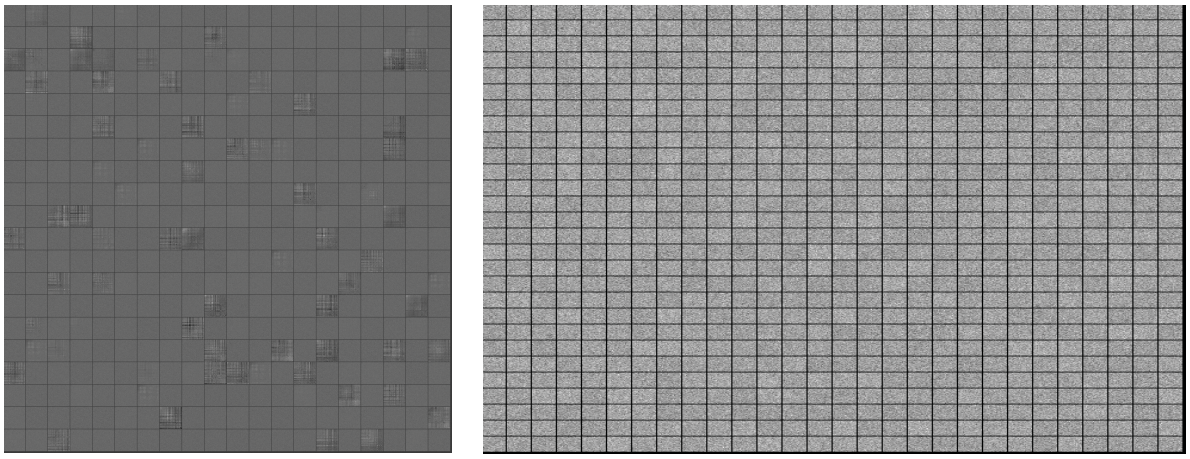


Figure 3.8: Training on the SHREC-2011 dataset. Learned weights on DBN first layer (left). Learned weights on DBN second layer (right).

method is satisfactory for a wide range of parameter values, indicating the robustness of the proposed framework to the choice of these parameters.We also tested the effect of the resolution parameter on the classification accuracy of the proposed approach. As can seen in Figure 3.10, the best classification accuracy on SHREC 2011 is achieved when $R = 2$.

3.5 Conclusions

In this chapter, we presented a deep learning approach to 3D shape classification using spectral graph wavelets and the BoF paradigm. This approach not only captures the similarity between feature descriptors via a geodesic exponential kernel, but also substantially outperforms state-of-the-art methods both in classification accuracy and in scalability. For future work, we plan to apply the proposed DeepSGW approach to other 3D shape analysis problems, and in particular segmentation and clustering.

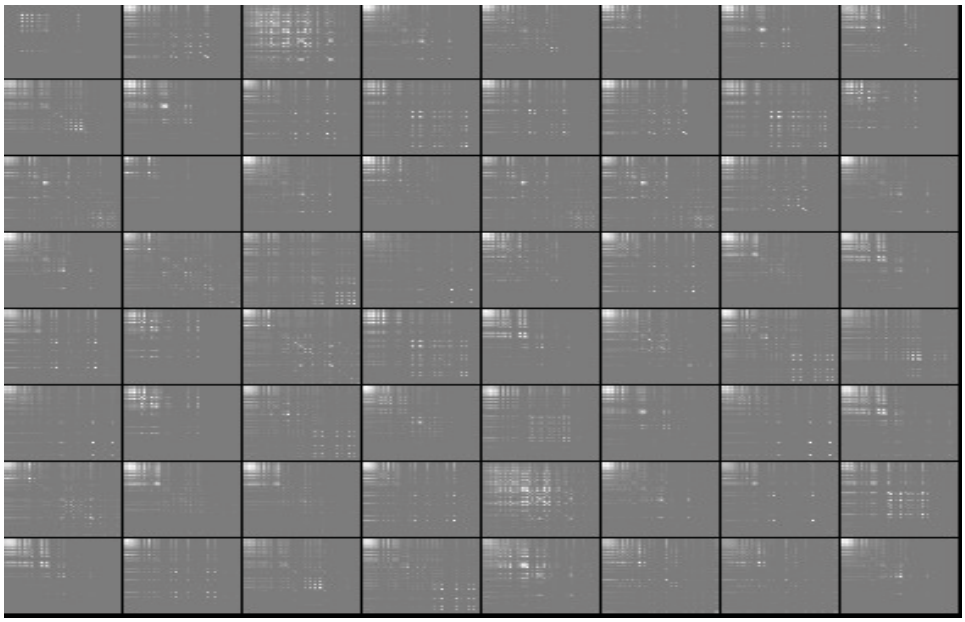


Figure 3.9: First 64 training examples computed by DBN on the SHREC-2011 dataset.

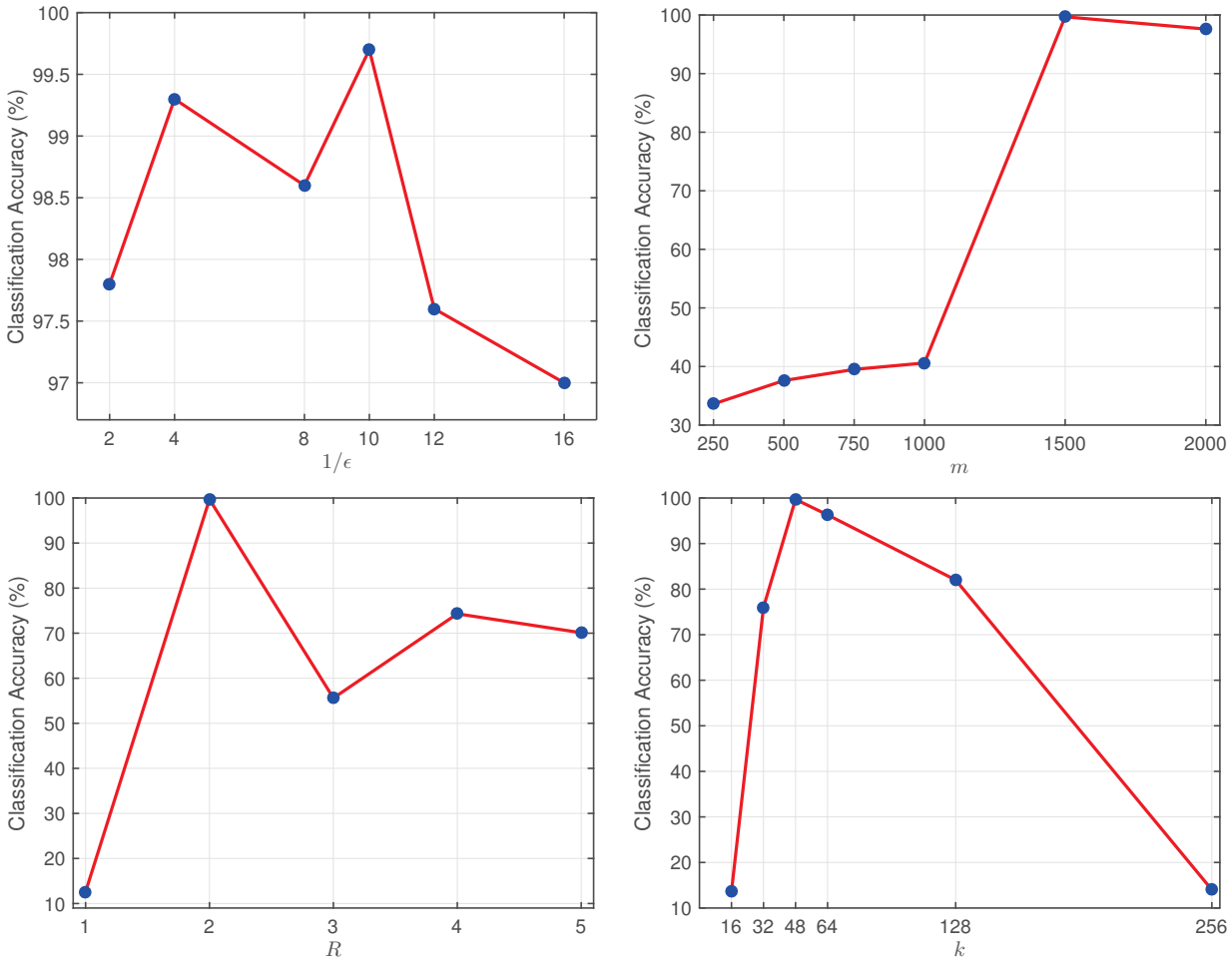


Figure 3.10: Effects of the parameters on the classification accuracy for SHREC 2011.

CHAPTER 4

Nonrigid Shape Retrieval using Spectral Graph Wavelets

In this chapter, we propose a nonrigid 3D shape retrieval framework, called SGWC-BoF, which employs spectral graph wavelet codes (SGWC) obtained from spectral graph wavelet signatures (i.e. local descriptors) via the soft-assignment coding step of the BoF model in conjunction with a geodesic exponential kernel for capturing the spatial relations between features. Broadly speaking, shape retrieval refers to the process of retrieving the most similar shapes to the queries from a dataset of 3D shapes. A good retrieval algorithm should result in high accuracy, or equivalently, in few dissimilar shapes. In addition to taking into consideration the spatial relations between features via a geodesic exponential kernel, the proposed approach performs retrieval on SGWC, thereby seamlessly capturing the similarity between these mid-level features. We not only show that our formulation allows us to take into account the spatial layout of features, but we also demonstrate that the proposed framework yields better retrieval accuracy results compared to state-of-the-art methods, while remaining computationally attractive.

4.1 Introduction

Three dimensional shape analysis has a wide range of applications such as mechanical design for CAD models, archaeology, cultural heritage, games, medical research studies and computer graphics. Recently, among the broad usage of 3D models in computer graphics like registration, matching, recognition, segmentation and classification, 3D shape retrieval has achieved more attentions because of its nice applications in search engines e.g., Google, Altavista, Bing and etc. Another

witness of growing trends to 3D model retrieval is the annual release of SHREC dataset [73], formed with the aim of evaluating the strength of different 3D object retrieval approaches.

A variety of models can be created as a result of deformations of a nonrigid 3D shape. Modeling of the produced shapes as well as analyzing their properties are the key issues in this domain [104]. On the other hand, a 3D object can be geometrically rendered in different forms like point clouds, triangular meshes, and parametric surfaces. In a bid to measure the dissimilarity among the nonrigid shapes, we require to find the properties that discriminate between the shapes.

While spectral signatures have received much attention in nonrigid 3D shape analysis [9, 13, 71, 105], view-based methods, on the other hand, have also been successfully applied to 3D shape retrieval [7, 106, 107]. Gao *et al.* presented a view-based 3D shape recognition and retrieval approach by exploring higher-order relationships among shapes via hypergraphs [106], where a vertex represents a shape and an edge delineates a cluster of views. More recently, Bai *et al.* proposed an interesting view-based method for 3D shape matching and retrieval using a two layer coding (TLC) framework that encodes view pairs rather than single views. Unlike many view-based methods, the TLC framework can be easily applied to encode features of 3D shapes in the same spirit as spectral signatures.

In this chapter, we built upon our previous work [14] to design an improved spectral graph wavelet signature by incorporating the vertex area into the definition of this signature in a bid to capture more geometric information and, hence, further improve its discriminative ability. We also used the Mexican hat wavelet as a generating kernel, which considers all frequencies equally-important overall as opposed to the cubic spline kernel [14]. More specifically, we propose a nonrigid 3D shape retrieval framework, called SGWC-BoF, which employs SGWC obtained from improved spectral graph wavelet signatures (i.e. local descriptors) via the soft-assignment coding step of the BoF model in conjunction with a geodesic exponential kernel for capturing the spatial relations between features. Broadly speaking, shape retrieval refers to the process of retrieving the most similar shapes to the queries from a dataset of 3D shapes. A good retrieval algorithm should result in high accuracy, or equivalently, in few dissimilar shapes.

An important facet of our approach [71] is the ability to combine the advantages of WKS and HKS into a single signature, while allowing a multiresolution representation of shapes. In addition to taking into consideration the spatial relations between features via a geodesic exponential kernel, the proposed approach performs retrieval on SGWC, thereby seamlessly capturing the similarity between these mid-level features. We not only show that our formulation allows us to take into account the spatial layout of features, but we also demonstrate that the proposed framework yields better retrieval accuracy results compared to state-of-the-art methods, while remaining computationally attractive.

The rest of this chapter is organized as follows. In Section 4.2, we explain the main steps of our

proposed framework for nonrigid 3D shape retrieval, and we discuss in detail its main algorithmic steps. Also, Section 4.3 focuses on experimental results. Ultimately, we briefly conclude in Section 4.4.

4.2 Method

In this section, we provide a detailed description of our nonrigid 3D shape retrieval method that utilizes spectral graph wavelets in conjunction with the BoF paradigm. Each 3D shape in the dataset is first represented by local descriptors, which are arranged into a spectral graph wavelet signature (SGWS) matrix. Then, we perform soft-assignment coding by embedding local descriptors into the visual vocabulary space, resulting in mid-level features which we refer to as SGWC. It is important to point out that the vocabulary is computed offline by concatenating all the spectral graph wavelet signature matrices into a data matrix, followed by applying the K-means algorithm to find the data cluster centers.

In a bid to capture the spatial relations between features, we compute a global descriptor of each shape in terms of a geodesic exponential kernel and mid-level features, resulting in a SGWC-BoF matrix which is then transformed into a SGWC-BoF vector by stacking its columns one underneath the other. The last stage of the proposed approach is to perform retrieval on the SGWC-BoF vectors by computing a dissimilarity metric between the SGWC-BoF vector of a given query and all SGWC-BoF vectors in the dataset in an effort to find the closest shape to the query. The flowchart of the proposed framework is depicted in Figure 4.1.

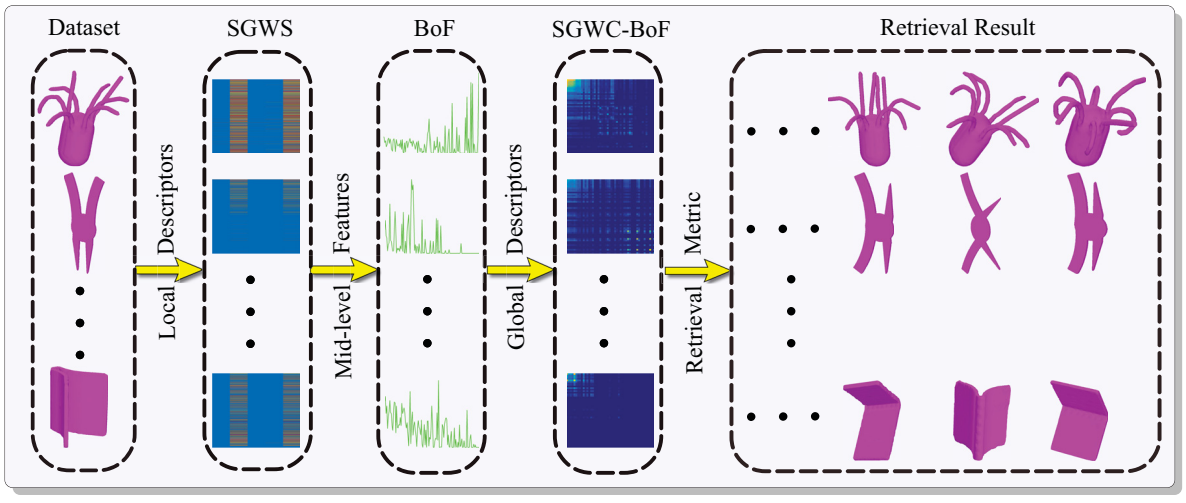


Figure 4.1: Flowchart of the proposed SGWC-BoF approach.

4.2.1 Proposed Algorithm

The goal of 3D shape retrieval is to search and extract the most relevant shapes to the queries from a dataset of 3D shapes. By relevant, we mean the objects that belong to the same class. The retrieval accuracy is usually evaluated by computing a dissimilarity measure between pairs of 3D shapes in the dataset. A commonly used dissimilarity measure for content-based retrieval is the ℓ_1-distance, also known as Manhattan or city-block metric, which quantifies the difference between each pair of 3D shapes.

Our proposed nonrigid 3D shape retrieval algorithm consists of four main steps. The first step is to represent each 3D shape in the dataset by a spectral graph wavelet signature matrix, which is a feature matrix consisting of local descriptors. More specifically, let $\mathcal{D}$ be a dataset of n shapes modeled by triangle meshes $\mathbb{M}_1, \ldots, \mathbb{M}_n$. We represent each 3D shape in the dataset $\mathcal{D}$ by a $p \times m$ spectral graph wavelet signature matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_m) \in \mathbb{R}^{p \times m}$, where $\mathbf{s}_i$ is the p-dimensional local descriptor at vertex i and m is the number of mesh vertices.

In the second step, the spectral graph wavelet signatures $\mathbf{s}_i$ are mapped to high-dimensional mid-level feature vectors using the soft-assignment coding step of the BoF model, resulting in a $k \times m$ matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$ whose columns are the k-dimensional mid-level feature codes (i.e. SGWC). In the third step, the $k \times k$ SGWC-BoF matrix $\mathbf{F}$ is computed using the mid-level feature codes matrix and a geodesic exponential kernel, followed by reshaping $\mathbf{F}$ into a k^2-dimensional global descriptor $\mathbf{x}_i$. In the fourth step, the SGWC-BoF vectors $\mathbf{x}_i$ of all n shapes in the dataset are arranged into a $k^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. Finally, we compare a query $\mathbf{x}$ to all data points in $\mathbf{X}$ using ℓ_1-distance to find the most relevant shapes to the query. The lower the value of this distance is, the more similar the shapes are. The main algorithmic steps of our approach are summarized in Algorithm 3.

Algorithm 3 SGWC-BoF

Input: Dataset $\mathcal{D} = \{\mathbb{M}_1, \ldots, \mathbb{M}_n\}$ of 3D shapes and a query.

1: **for** $i = 1$ to n **do**
2: {**Step 1**} Compute the $p \times m$ spectral graph wavelet signature matrix $\mathbf{S}_i$ of each shape $\mathbb{M}_i$
3: {**Step 2**} Apply soft-assignment coding to find the $k \times m$ mid-level feature matrix $\mathbf{U}_i$, where $k > p$
4: {**Step 3**} Compute the $k \times k$ SGWC-BoF matrix $\mathbf{F}_i$, and reshape it into a k^2-dimensional vector $\mathbf{x}_i$
5: **end for**
6: {**Step 4**} Arrange all the n SGWC-BoF vectors into a $k^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
7: {**Step 5**} Compute the ℓ_1-distance between the SGWC-BoF vector $\mathbf{x}$ of the query and all SGWC-BoF vectors in the dataset, and find the closest shape(s).

Output: Retrieved set of most relevant shapes to the query.

Remark: It is important to point out that in our implementation the vocabulary is computed offline by applying the K-means algorithm to the $p \times mn$ matrix obtained by concatenating all SGWS matrices of all n meshes in the dataset. As a result, the vocabulary is a matrix $\mathbf{V}$ of size $p \times k$, where $k > p$.

4.3 Experimental Results

In this section, we conduct extensive experiments to evaluate the performance of the proposed SGWC-BoF framework for nonrigid 3D shape retrieval. The effectiveness of our approach is validated by carrying out a comprehensive comparison with several state-of-the-art methods.

Datasets The performance of the proposed framework is evaluated on two standard and publicly available 3D shape benchmarks: SHREC 2011 and SHREC 2015. Sample shapes from these two benchmarks are shown in Figure 4.2.

Figure 4.2: Sample shapes from SHREC 2011 (top) and SHREC 2015 (bottom).

Performance Evaluation Measures The retrieval performance of the proposed SGWC-BoF approach is comprehensively evaluated using six commonly-used evaluation metrics: Precision-Recall (P-R) curve, Nearest Neighbor (NN), First-Tier (FT), Second-Tier (ST), E-Measure (E) and Discounted Cumulative Gain (DCG) [32].

The P-R curve is an informative graph that illustrates the tradeoff between precision as a function of recall, and it shows the retrieval performance at each point in the ranking. If, for instance, the $(r+1)$th shape retrieved is relevant, then both precision and recall increase. However, if it is irrelevant then recall is the same as for the top r shapes, but precision decreases. Hence, a P-R curve that is shifted upwards and to the right indicates superior performance.

The NN metric is the percentage of the closest matches that belong to the query's class, i.e. for each shape in the dataset, the second result (assuming that the first result is the shape itself) is checked whether it is a member of the same class the shape belongs to. The FT metric is the percentage of the $C-1$ matches retrieved that belong to the query's class, while the ST metric is the percentage of the $2(C-1)$ matches retrieved that belong to the query's class, where C is the size of the query's class. On the other hand, the DCG is a statistic that weights correct results near the front of the list more than correct results later in the ranked list, under the assumption that a user is less likely to consider elements near the end of the list. All these metric have scores ranging from 0 to 1 (or equivalently from 0% to 100% in terms of percentages), with a higher score indicating a better performance.

Baseline Methods For each of the 3D shape benchmarks used for experimentation, we will report the retrieval results of the proposed SGWC-BoF method against various baseline methods in the literature. For the SHREC-2011 dataset, we compared our approach to GA-BoF [81] and a variety of baseline methods (see [85, 108] and references therein), including features on geodesics (FOG), bag of words with local spectral descriptors (BOW-LSD), visual similarity based non-rigid 3D shape retrieval using multidimensional scaling (MDS-CM-BOF), bag of geodesic histograms (BOGH), localized statistical features (LSF), ShapeDNA, Harris 3D geodesic map (Hariss3DGeoMap), heat kernel signature (HKS) and scale invariant feature transform for meshes (MeshSIFT). We also compared our method to Shape Google [35], SGWS [14], and the two layer coding (TLC) framework [7].

For the SHREC-2015 dataset, we compared SGWC-BoF to several baseline approaches (see [109] and references therein), including geodesic distance distribution (SNU), heat kernel signature (HKS), surface area (SA), wave kernel signature (WKS), multi-feature, spectral geometry (SG), Fisher vector encoding framework-heat kernel signature (FVF-HKS), Fisher vector encoding framework-scaled invariant heat kernel signature (FVF-SIHKS), Fisher vector encoding framework-heat kernel signature-wave kernel signature (FVF-WKS), time series analysis for

shape retrieval (TSAR), sphere intersection descriptor (SID) and Euclidean distance based canonical forms (EDBCF-AV).

Implementation Details The experiments were conducted on a desktop computer with an Intel Core i5 processor running at 3.10 GHz and 8 GB RAM; and all the algorithms were implemented in MATLAB. The appropriate dimension (i.e. length or number of features) of a shape signature is problem-dependent and usually determined experimentally. For fair comparison, we used the same parameters that have been employed in the baseline methods, and in particular the dimensions of shape signatures. In our setup, a total of 201 eigenvalues and associated eigenfunctions of the LBO were computed. For the proposed approach, we set the resolution parameter to $R = 2$ (i.e. the spectral graph wavelet signature matrix is of size $5 \times m$, where m is the number of mesh vertices) and the kernel width of the geodesic exponential kernel to $\epsilon = 0.1$. Moreover, the parameter of the soft-assignment coding is computed as $\alpha = 1/(8\mu^2)$, where μ is the median size of the clusters in the vocabulary [35].

4.3.1 SHREC-2011 Dataset

SHREC 2011 is a dataset of 3D shapes consisting of 600 watertight mesh models, which are obtained from transforming 30 original models [85]. Each shape in the dataset has approximately $m = 1502$ vertices.

Performance Evaluation The retrieval performance of the proposed approach is evaluated by performing a pairwise comparison between the SGWC-BoF vector of a given query and all the SGWC-BoF vectors of the shapes in the SHREC-2011 dataset using the ℓ_1-distance, and then finding the closest shape to the query. A smaller value of the ℓ_1-distance indicates that two shapes are similar.

Figure 4.3 displays the P-R plots of the proposed approach and other state-of-the-art methods. As can be seen, SGWC-BoF achieves better performance compared to the baseline methods, indicating that our approach is able to retrieve correct shapes with a high degree of accuracy. Such a good performance strongly suggests that the proposed SGWT-BoF framework captures well the discriminative features of the shapes.

Results In our approach, each 3D shape in the SHREC-2011 dataset is represented by a 5×1502 matrix of spectral graph wavelet signatures. Setting the number of codewords to $k = 128$, we computed offline a 5×128 vocabulary matrix $\mathbf{V}$ via K-means clustering. The pre-computation of the vocabulary of size 128 took approximately 70 minutes. The soft-assignment coding of the BoF model yields a 128×1502 matrix $\mathbf{U}$ of SGWC, resulting in a SGWC-BoF data matrix $\mathbf{X}$ of size $128^2 \times 600$.

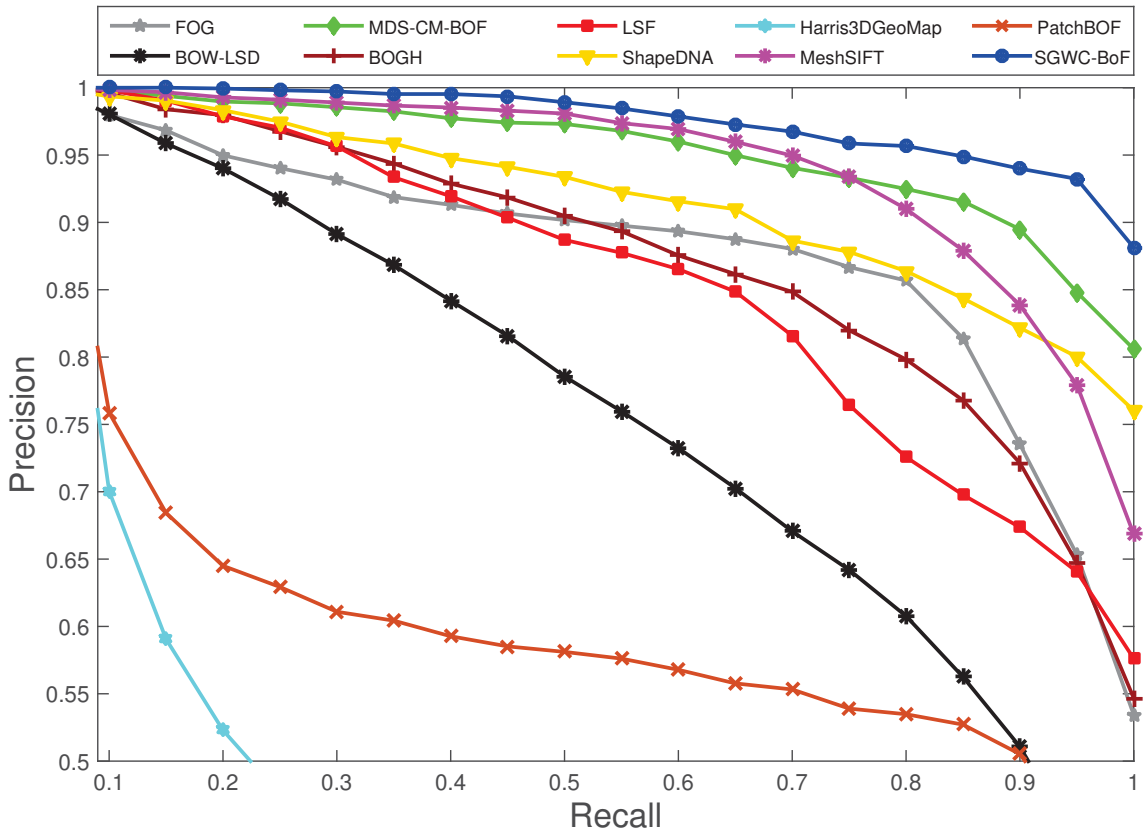


Figure 4.3: P-R plots comparing the performance of the proposed method and other state-of-the-art approaches on SHREC 2011.

We compared the proposed method to FOG, BOW-LSD, MDS-CM-BOF, BOGH, LSF, ShapeDNA, Hariss3DGeoMap, HKS, MeshSIFT, SD-GDM-meshSIFT, Shape Google [35], SGWS [14], TLC+J-PairTLC+I-Pair [7], TLC+I-Pair [7] and GA-BoF [81]. In order to evaluate the retrieval performance, we first computed the dissimilarity matrix between all SGWC-BoFs of the shapes in the SHREC-2011 dataset using ℓ_1-distance. The retrieval results are summarized in Table 4.1, which shows the scores of the evaluation metrics for the baseline methods and the proposed framework. With the exception of SD-GDM-meshSIFT, our SGWC-BoF approach outperforms all the baselines. This better performance is in fact consistent with all the retrieval evaluation metrics. For example, the NN value for SGWC-BoF is a perfect 100%, similar to SD-GDM-meshSIFT. From the table, we can also see that SGWC-BoF yields improvements of 0.5% in NN, 4.9% in FT, 1.3% in ST, 1.2% in E and 1.1% in DCG compared to MDS-CM-BOF, which is the best baseline performer.

Table 4.1: Retrieval results on the SHREC-2011 dataset. Boldface numbers indicate the best retrieval performance.

	Retrieval Evaluation Measures (%)				
Method	NN	FT	ST	E	DCG
FOG	96.8	81.7	90.3	66.0	94.4
BOW-LSD	95.5	67.2	80.3	57.9	89.7
MDS-CM-BOF	99.5	91.3	96.9	71.7	98.2
BOGH	99.3	81.1	88.4	64.7	94.9
LSF	99.5	79.9	86.3	63.3	94.3
ShapeDNA	99.2	91.5	95.7	70.5	97.8
Hariss3DGeoMap	56.2	32.5	46.6	32.2	65.4
HKS	83.7	40.6	49.7	35.3	73.0
MeshSIFT	99.5	88.4	96.2	70.8	98.0
SD-GDM-MeshSIFT	**100**	**97.2**	**99**	**73.6**	**99.4**
Shape Google	98.2	63.7	73.2	–	88.1
SGWS	91.1	80.8	86.5	61.7	89.48
TLC+J-Pair (SIFT)	98.2	86.4	94.1	–	96.4
TLC+I-Pair (SIFT)	99	86.5	93.3	–	96.3
GA-BoF	98.6	91.0	97.4	68.3	97.2
SGWC-BoF	**100**	96.2	98.2	72.9	99.3

4.3.2 SHREC-2015 Dataset

SHREC 2015 is a dataset of 3D shapes consisting of 1200 watertight mesh models from 50 classes [109], where each class contains 24 objects with distinct postures. Each shape in the dataset has approximately $m = 1502$ vertices.

Performance Evaluation The SGWC-BoF matrices of two shapes from two different classes of SHREC 2011 are shown in Figure 4.4. As can be seen, these global descriptors are quite different and hence they may be used efficiently to discriminate between shapes in retrieval tasks. To assess the retrieval performance of the proposed approach on the SHREC-2015 dataset, we plotted the P-R curves of SGWC-BoF and the baseline methods in Figure 4.5. As can be seen, the SGWC-BoF approach significantly outperforms the baselines, albeit 16.7% of shapes in each class of SHREC 2015 contain different topological structures compared to the SHREC-2011 dataset. This indicates that pose-resistant features of nonrigid 3D shapes are well-represented by our approach.

Results Following the setting of the previous experiment, each 3D shape in the SHREC-2015 dataset is represented by a 5×1502 spectral graph wavelet signature matrix. We pre-computed offline a vocabulary of size $k = 256$, and it took about 100 minutes. The soft-assignment coding yields a 256×1502 matrix $\mathbf{U}$ of mid-level features. Hence, the SGWC-BoF data matrix $\mathbf{X}$ for

Figure 4.4: SGWC of two shapes (buffalo and kangaroo) from two different classes of the SHREC-2015 dataset.

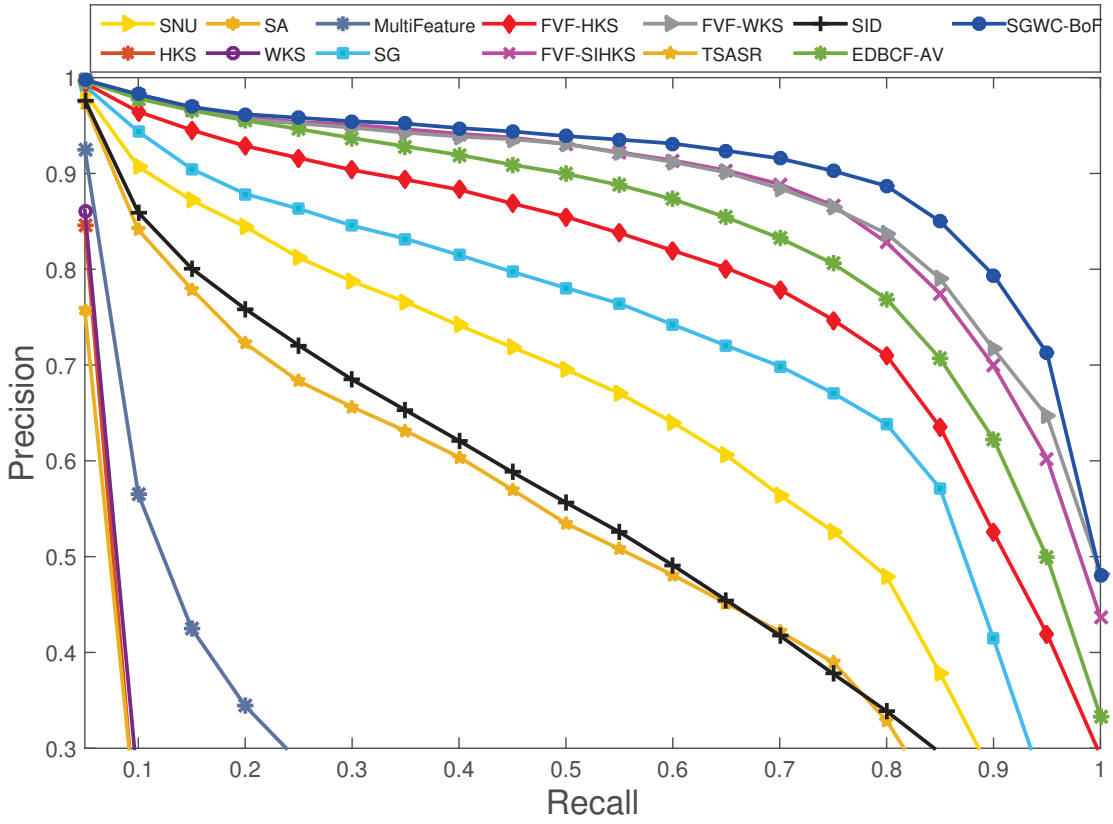


Figure 4.5: P-R plots comparing the performance of the proposed method and other state-of-the-art approaches on SHREC 2015

SHREC 2015 is of size $256^2 \times 1200$. The retrieval results are summarized in Table 4.2. As can be seen, the proposed approach outperforms the baseline methods. For instance, in terms of the

NN metric, the SGWC-BoF approach achieves a 98.3% score, with performance improvements of 0.3% and 0.7% over the best performing baselines FVF-SIHKS and FVF-WKS, respectively. In addition, SGWC-BoF outperforms the SG approach [14] by 4.7% in NN, 18.9% in FT, 18.3% in ST, 16% in E and 8.4% in DCG. This better performance is again consistent with all the retrieval evaluation metrics.

Table 4.2: Retrieval results on the SHREC-2015 dataset. Boldface numbers indicate the best retrieval performance.

	Retrieval Evaluation Measures (%)				
Method	NN	FT	ST	E	DCG
SNU	89.8	56.3	66.9	51.6	83.2
HKS	6.5	6.3	12.4	7.4	39.1
SA	6.5	6.7	12.8	7.8	39.3
WKS	13.4	7.4	13.7	8.3	40.8
Multi-feature	45.0	18.6	26.2	18.4	52.5
SG	93.6	66.8	73.6	58.7	87.5
FVF-HKS	96.0	72.5	80.9	64.4	91.3
FVF-SIHKS	98.0	82.4	88.2	71.7	95.0
FVF-WKS	97.6	82.2	89.4	72.4	95.1
TSAR	81.3	46.3	54.4	42.0	74.9
SID	79.5	48.4	61.4	45.9	77.8
EDBCF-AV	97.5	76.9	86.8	68.9	93.5
SGWC-BoF	**98.3**	**85.7**	**91.9**	**74.7**	**95.9**

For fair comparison, we compared our approach to baseline methods of the same category (i.e. BoF-based methods). In addition, approaches based on sparse coding suffer from the long running time of optimizing the sparse modeling problem to find the dictionary matrix. Although HAPT, SPH-SparseCoding and SV-LSF [109] perform slightly better than SGWC-BoF, the proposed framework consistently outperforms the baseline methods in most cases, as evidenced by our experimental results.

4.3.3 Sensitivity to Choice of Parameters

The proposed approach depends on two key parameters that affect its overall performance. The first parameter is the kernel width ϵ of the geodesic exponential kernel. The second parameter k is the size of the vocabulary, which plays an important role in the SGWC-BoF matrix $\mathbf{F}$. As shown in Figure 4.6, the highest DCG value on SHREC 2011 is achieved using $\epsilon = 0.1$ and $k = 128$.

Other two parameters that affect the SGWC-BoF approach to a lesser extent are the resolution parameter R and the mesh resolution. Figure 4.7 (left) indicates that increasing the number of

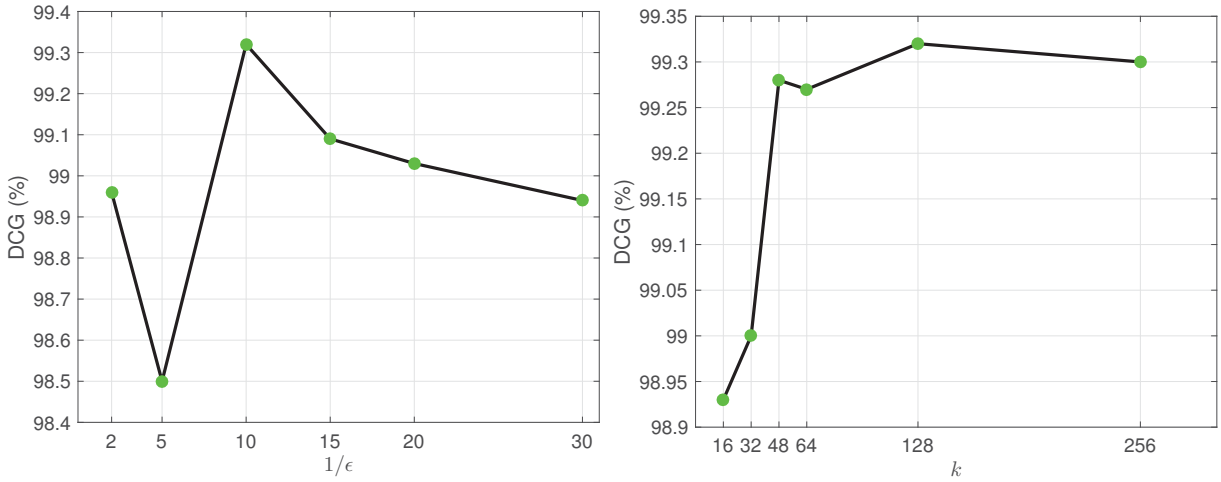


Figure 4.6: Effects of geodesic kernel width and size of vocabulary on the retrieval performance of SGWC-BoF for SHREC 2011.

mesh vertices slightly changes the DCG values, whereas Figure 4.7 (right) shows that best DCG value is obtained when $R = 2$. The effect of the signature resolution parameter R is further is illustrated in Figure 4.8, which depicts the normalized χ^2-distance between a reference point and other mesh vertices using SGWS for different values of R. As can be seen, changing the values R has practically an unnoticeable effect on the χ^2-distance.

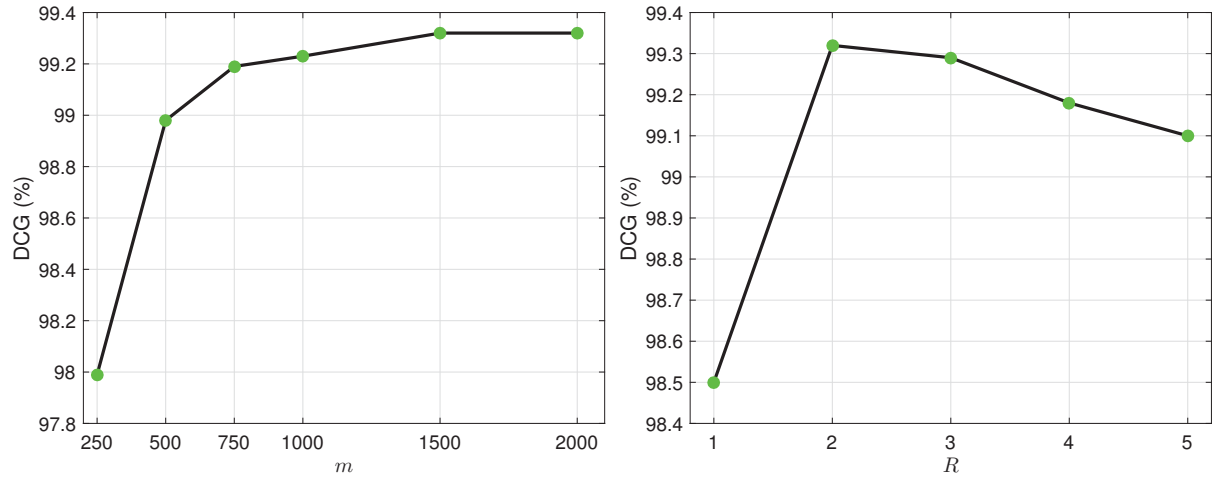


Figure 4.7: Effects of mesh resolution and signature resolution parameter on the retrieval performance of SGWC-BoF for SHREC 2011.

Overall, the retrieval performance of proposed method is satisfactory for a wide range of parameter values, indicating a slight sensitivity of our approach to the choice of parameters.

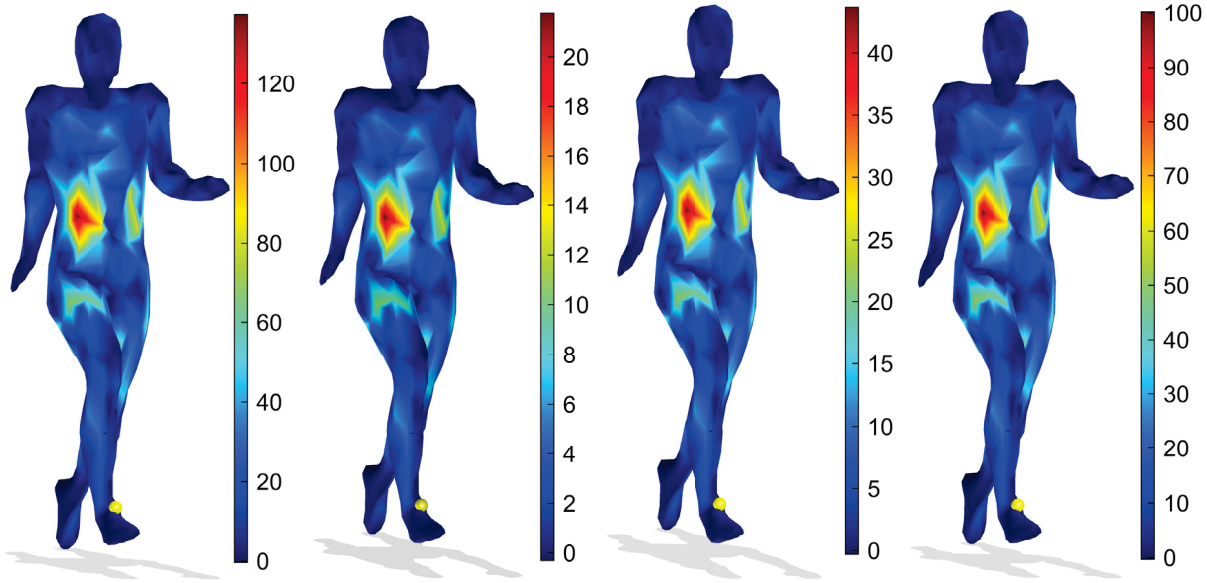


Figure 4.8: Normalized χ^2-distance between a reference point (yellow colored on the man's right foot) and other surface points using SGWS for different values of the resolution parameter $R = 1, 2, 3$ and 5 (left to right).

4.3.4 Robustness to Topological Noise

To assess the performance of the proposed approach in the presence of topological noise, we randomly selected a few shapes from SHREC 2011 and welded some selected vertices of each shape. Topological noise may arise not only from the triangulation process of point clouds, but also from various nonrigid deformations of shapes. Figure 4.9 shows sample shapes contaminated with topological noise.

We performed retrieval on the noisy SHREC-2011 dataset by computing the evaluation metrics for SGWC-BoF, and the results are NN $= 99.6$, FT $= 95.4$, ST $= 97.4$, E $= 72.3$ and DCG $= 98.9$. These values indicate that the performance of SGWC-BoF deteriorates slightly in the presence of topological noise, albeit the geodesic distance is notably sensitive to topological transformations.

We also compared our approach with the Shape Google method on the noisy SHREC-2011 dataset. The resulting values for Shape Google are NN $= 96.7$, FT $= 59$, ST $= 67.5$, E $= 51.2$ and DCG $= 73.6$, indicating a lower performance than our proposed approach.

4.4 Conclusions

In this chapter, we introduced a spectral graph wavelet framework for 3D shape retrieval that employs the BoF paradigm in an effort to design a global shape descriptor defined in terms of mid-level features and a geodesic exponential kernel. The proposed approach not only takes into consideration the spatial relations between features, but also achieves better performance compared with state-of-the-art retrieval methods. The effectiveness of our method was demonstrated on two standard 3D shape benchmarks. For future work, we plan to apply the proposed approach to other 3D shape analysis problems.

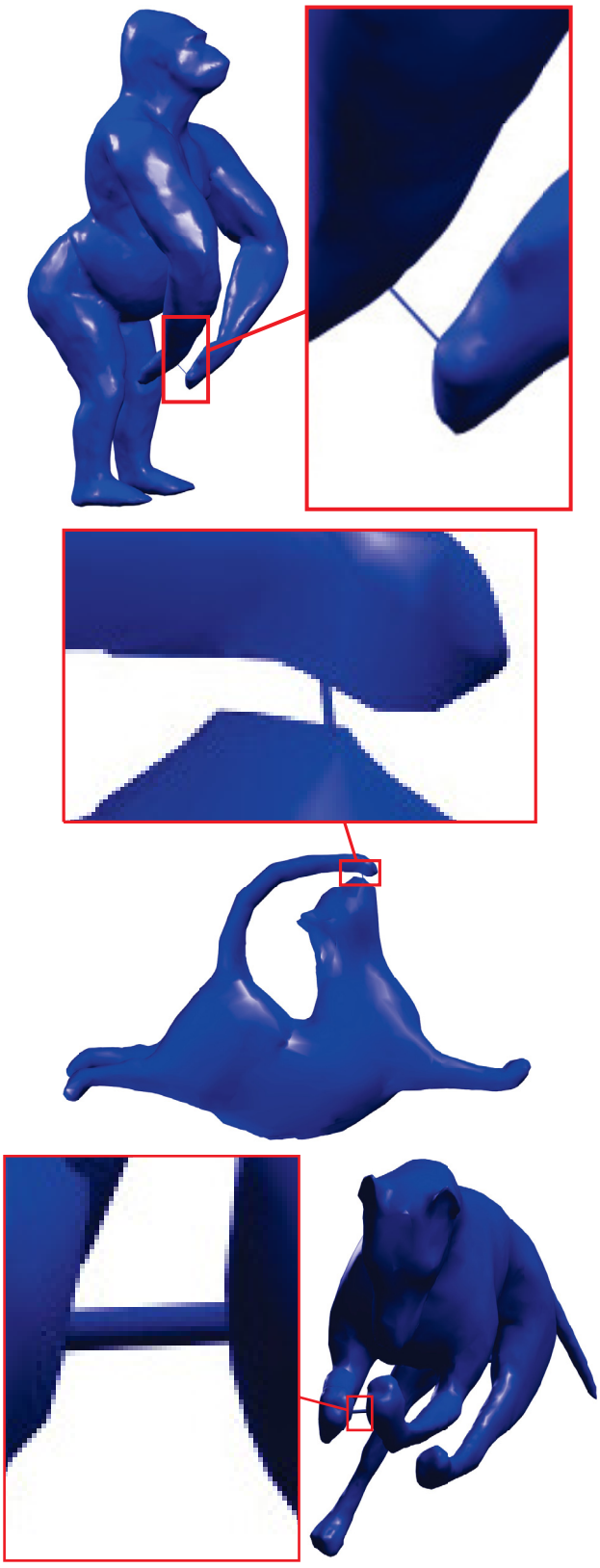


Figure 4.9: Sample noisy 3D shapes, where the enlarged views show the simulated topological noise.

CHAPTER 5

Conclusions and Future Work

This thesis has presented two techniques for classification of 3D objects, namely SGWC-BoF and DeepSGW. Furthermore, a spectral geometric approach for retrieval of nonrigid 3D shape using the LBO and the graph wavelet transform has been proposed. We have demonstrated through extensive experiments the much better performance of the proposed methods in comparison with other state-of-the-arts methods in the literature.

In Section 5.1, the contributions made in each of the previous chapters and the concluding results drawn from the associated research work are presented. Suggestions for future research directions related to this thesis are also provided in Section 5.2.

5.1 Contributions of the Thesis

5.1.1 Shape Classification using Spectral Graph Wavelets

In Chapter 2, we first reviewed and compared recent spectral descriptors for shape analysis. Then, we introduced a spectral graph wavelet framework which utilizes BoF paradigm in conjunction with geodesic exponential kernel for classification of 3D models. The main advantage of our proposed approach is that ours accounts the spatial relations between the features. The experimental results showed that our proposed technique is more accurate and outperforms existing approaches.

5.1.2 Spectral Shape Classification via Deep Learning

In Chapter 3, we presented a DeepSGW approach which provides a general and flexible framework for 3D object classification [70]. The proposed approach not only takes into consideration

the spatial relations between features, but also significantly improves the discriminative ability of signature. Experimental results on two datasets demonstrate that our proposed approach outperforms state-of-the-art methods both in classification accuracy and in scalability.

5.1.3 Nonrigid 3D Shape Retrieval using Spectral Graph Wavelets

In Chapter 4, we proposed a spectral graph wavelet framework for analysis and design of efficient shape descriptor for nonrigid 3D shape retrieval [71]. Although this work focuses primarily on shape retrieval, our approach is, however, fairly general and can be used to address other 3D shape analysis problems. By concentrating on finding informative spectrum for 3D shape retrieval, we devised a surface representation that is multiresolution, compact, highly discriminative, and parameter-insensitive. We also demonstrated through extensive experiments the effectiveness of the SGWC-BoF by achieving state-of-the-art results on two standard repositories of 3D shapes.

5.2 Future Research Directions

Several interesting research directions, motivated by this thesis, are discussed below:

5.2.1 Improvement of 3D Shape Retrieval using Deep Learning

The availability and widespread usage of large databases coupled with the need to explore 3D models in depth as well as in breadth has sparked the need to organize and search these vast data collections, retrieve the most relevant selections, and permit them to be effectively reused. 3D objects consist of geometric and topological information, and their compact representation is an important step towards a variety of computer vision applications, particularly matching and retrieval in a database of 3D models. The first step in 3D object matching usually involves finding a reliable shape descriptor which efficiently encodes the 3D shape information. We are interested in training the descriptors using deep learning to achieve high-level features which describe the 3D objects more precisely. As a result, by employing the high-level features the overall process of 3D object retrieval will be improved.

Inspired by recent successes of deep learning techniques in content-based image retrieval (CBIR) [99], we intend to investigate the state-of-the-art deep learning approaches including DBN [101], deep Boltzmann machine (DBM) [110], and deep neural networks (DNN) [111] for learning high-level features. Recent results [99] from the extensive empirical studies on CBIR show that deep CNN model pre-trained on large-scale dataset can be directly utilized for capturing high semantic information in new CBIR tasks. Moreover, features extracted by pre-trained CNN model in conjunction with proper feature refining frameworks, consistently outperform conventional hand-

crafted features on all datasets [99]. In future work, we will investigate more advanced deep learning techniques and assess more other diverse datasets to give more insights for bridging the semantic gap of 3D model retrieval. In particular, we will explore convolutional neural networks CNN [112] for retrieving nonrigid 3D shape.

5.2.2 Medical Shape Analysis

Detecting unique phenotypes across populations can be achieved by quantitative analysis of bone shape, provided that the databases of normal and abnormal pathologies are available. For future work directions, we plan to perform statistical analysis on carpal bones of the human wrist by representing the cortical surface of the carpal bone using spectral graph wavelet descriptor to supply a means for comparing shapes of the carpal bones across populations. Figure 5.1 shows an example of carpal bone for a healthy male. Furthermore, we will utilize this representation in two applications: (1) analysis of the differences in carpal bone shapes between women and men, and (2) analysis of carpal bone shape differences between the right and left hand across the population. More precisely, unlike our current SGWC-BoF method in which first aggregates local descriptors of a shape and then subsequently represent each object by a global signature, we will propose a novel framework of directly extracting global descriptor so-called global spectral graph wavelet (GSGW). Thus, we will circumvent all the procedure of BoF paradigm which leads to a lower computation time as well as higher analysis accuracy. Furthermore, we will evaluate the accuracy of our proposed framework in terms of multi-variant analysis of variance (MANOVA) and permutation test for different sexes and carpal bones.

5.2.3 3D Shape Watermarking

Recent advances in designing and processing digital contents has led to the representation of the valuable data in digital forms, which can be distributed through internet. Since digital contents may easily be duplicable, we need to protect such contents for the purposes of ownership claiming and authentication. Watermarking techniques have been used as effective solutions for solving the copyright and ownership verification issues by embedding the watermark information directly in a 3D object by modifying either the 3D mesh geometry or the topology of the triangles. However, these techniques are often susceptible to various kinds of attacks. We intend to develop 3D watermarking techniques using multi-resolution mesh analysis (spectral decomposition and graph wavelet transform) in an effort to show good resistance against attacks.

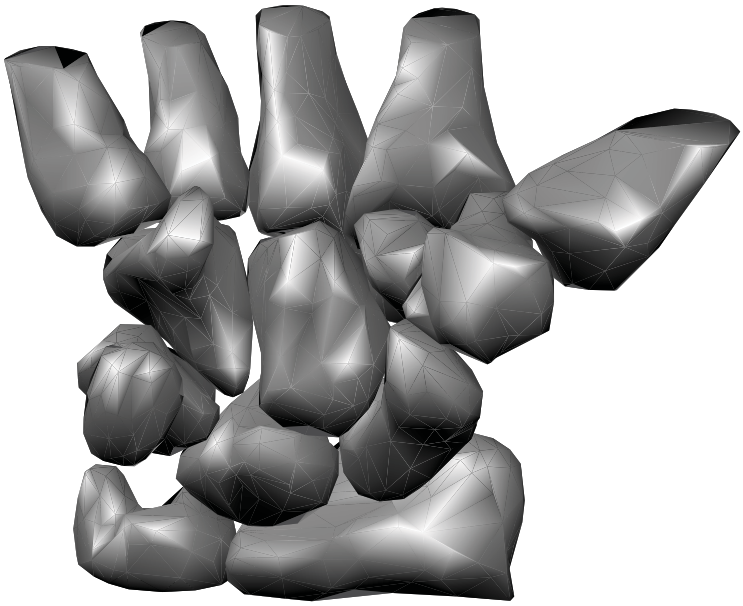


Figure 5.1: 3D representation of left carpal bone for a healthy male.

5.2.4 Design of Wavelet Generating Kernels

In its current form, the proposed SGWC-BoF is generated using a Mexican-hat kernel, and it has been shown to yield superior performance only with isometric or near-isometric transformations. In the future, we will look more carefully into the optimal choice of other wavelet generating kernel functions, thus extending the scope of the SGWC-BoF to more general classes of deformations. Additionally, designing appropriate signatures for other shape analysis applications such as surface denoising is a promising future work direction that we plan to explore.

5.2.5 From Image Processing to Geometry Processing

Generally speaking, this thesis provides a bridge to borrow ideas from image processing for geometry processing, namely the wavelet framework for shape descriptors' design. Abstractly, it generalizes methods in the Euclidean space to the weighted graph space, resulting in a fruitful way to understand 3D shapes by extending sophisticated methods in image domain via these tools. Our future plan is to explore other tools to link these two fields, such as finding a proper generalization of sparse coding and low rank matrix recovery based methods in the image domain for 3D surfaces.

References

[1] Y. Yang, H. Lin, and Y. Zhang, "Content-based 3-D model retrieval: A survey," *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 37, no. 6, pp. 1081–1098, 2007.

[2] A. DelBimbo and P. Pala, "Content-based retrieval of 3D models," *ACM Trans. Multimedia Computing, Communications, and Applications*, vol. 2, no. 1, pp. 20–43, 2006.

[3] J. Tangelder and R. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441–471, 2008.

[4] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranic, "Feature-based similarity search in 3D object databases," *ACM Computing Surveys*, vol. 37, no. 4, pp. 345–387, 2005.

[5] V. Jain and H. Zhang, "A spectral approach to shape-based retrieval of articulated 3D models," *Computer-Aided Design*, vol. 39, no. 5, pp. 398–407, 2007.

[6] K. Siddiqi and S. Pizer (Eds.), *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 2008.

[7] X. Bai, S. Bai, Z. Zhu, and L. Latecki, "3D shape matching via two layer coding," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2361–2373, 2015.

[8] M. Hassouna and A. Farag, "Variational curve skeletons using gradient vector flow," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2257–2274, 2009.

[9] R. Rustamov, "Laplace-Beltrami eigenfunctions for deformation invariant shape representation," in *Proc. Symp. Geometry Processing*, pp. 225–233, 2007.

[10] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[11] M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *CVPR*, pp. 1704–1711, 2010.

[12] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *Proc. Computational Methods for the Innovative Design of Electrical Devices*, pp. 1626–1633, 2011.

[13] M. Reuter, F. Wolter, and N. Peinecke, "Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids," *Computer-Aided Design*, vol. 38, no. 4, pp. 342–366, 2006.

[14] C. Li and A. Ben Hamza, "A multiresolution descriptor for deformable 3D shape retrieval," *The Visual Computer*, vol. 29, pp. 513–524, 2013.

[15] S. Rosenberg, *The Laplacian on a Riemannian Manifold*. Cambridge University Press, 1997.

[16] A. Bronstein, M. Bronstein, and R. Kimmel, *Numerical Geometry of Non-rigid Shapes*. Springer, 2008.

[17] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[18] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

[19] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing*, pp. 156–164, 2003.

[20] Y. Lipman, R. Rustamov, and T. Funkhouser, "Biharmonic distance," *ACM Trans. Graphics*, vol. 29, no. 3, pp. 1–11, 2010.

[21] I. Kokkinos, M. Bronstein, R. Litman, and A. Bronstein, "Intrinsic shape context descriptors for deformable shapes," in *CVPR*, pp. 159–166, 2012.

[22] Y. Shinagawa, T. Kunii, and Y. Kergosien, "Surface coding based on morse theory," *IEEE Computer Graphics and Applications*, vol. 11, no. 5, pp. 66–78, 1991.

[23] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proc. SIGGRAPH*, pp. 203–212, 2001.

[24] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker, "Shock graphs and shape matching," *Int. Journal of Computer Vision*, vol. 35, no. 1, pp. 13–32, 1999.

[25] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," *Machine vision and applications*, vol. 19, no. 4, pp. 261–275, 2008.

[26] N. Cornea, M. Demirci, D. Silver, A. Shokoufandeh, S. Dickinson, and P. Kantor, "3D object retrieval using many-to-many matching of curve skeletons," in *Proc. Int. Conf. Shape Modeling and Applications*, pp. 368–373, 2005.

[27] M. Hassouna and A. Farag, "Variational curve skeletons using gradient vector flow," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2257–2274, 2009.

[28] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Trans. Graphics*, vol. 28, no. 3, 2009.

[29] M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl, "3D shape histograms for similarity search and classification in spatial databases," in *Int. Symposium on Spatial Databases*, pp. 207–226, 1999.

[30] A. Ion, N. Artner, G. Peyré, W. Kropatsch, and L. Cohen, "Matching 2D and 3D articulated shapes using the eccentricity transform," *Computer Vision and Image Understanding*, vol. 115, pp. 817–834, 2011.

[31] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoun, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[32] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton shape benchmark," in *Proc. Shape Modeling International (SMI'04)*, pp. 167–178, 2004.

[33] R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[34] B. Lévy, "Laplace-Beltrami eigenfunctions: Towards an algorithm that "understands" geometry," in *Proc. IEEE Int. Conf. Shape Modeling and Applications*, p. 13, 2006.

[35] A. Bronstein, M. Bronstein, L. Guibas, and M. Ovsjanikov, "Shape Google: Geometric words and expressions for invariant shape retrieval," *ACM Trans. Graphics*, vol. 30, no. 1, 2011.

[36] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, pp. 1470–1477, 2003.

[37] J. Gemert, C. Snoek, C. Veenman, A. Smeulders, and J. Geusebroek, "Comparing compact codebooks for visual categorization," *Computer Vision and Image Understanding*, pp. 450–462, 2010.

[38] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *CVPR*, 2008.

[39] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.

[40] J. Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1271–1283, 2010.

[41] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, pp. 1794–1801, 2009.

[42] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, pp. 3360–3367, 2010.

[43] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *CVPR*, 2007.

[44] H. Jégou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *CVPR*, pp. 3304–3311, 2010.

[45] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local images descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1704–1716, 2012.

[46] D. Picard and P. Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *ICIP*, pp. 669–672, 2011.

[47] A. Bronstein and M. Bronstein, "Spatially-sensitive affine-invariant image descriptors," in *ECCV*, pp. 197–208, 2010.

[48] S. Savarese, J. Winn, and A. Criminisi, "Discriminative object class models of appearance and shape by correlatons," in *CVPR*, pp. 2033–2040, 2006.

[49] H. Ling and S. Soatto, "Proximity distribution kernels for geometric context in category recognition," in *ICCV*, pp. 1–8, 2007.

[50] R. Behmo, N. Paragios, and V. Prinet, "Graph commute times for image representation," in *CVPR*, 2008.

[51] D. Liu, G. Hua, P. Viola, and T. Chen, "Integrated feature selection and higher-order spatial feature extraction for object categorization," in *CVPR*, 2008.

[52] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, pp. 2169–2178, 2006.

[53] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *ICCV*, pp. 1458–1465, 2005.

[54] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Spatial-bag-of-features," in *CVPR*, pp. 3352–3359, 2010.

[55] Y. Yang and S. Newsam, "Spatial pyramid co-occurrence for image classification," in *ICCV*, pp. 1465–1472, 2011.

[56] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *CVPR*, pp. 809–816, 2011.

[57] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *CVPR*, 2012.

[58] J. Krapac, J. Verbeek, and F. Jurie, "Modeling spatial layout with fisher vectors for image categorization," in *ICCV*, pp. 1487–1494, 2011.

[59] H. Krim and A. Ben Hamza, *Geometric methods in signal and image analysis*. Cambridge University Press, 2015.

[60] M. Meyer, M. Desbrun, P. Schröder, and A. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," *Visualization and mathematics III*, vol. 3, no. 7, pp. 35–57, 2003.

[61] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete Laplace operators: no free lunch," in *Proc. Eurographics Symp. Geometry Processing*, pp. 33–37, 2007.

[62] Y. Fang, M. Sun, M. Kim, and K. Ramani, "Heat-mapping: A robust approach toward perceptually consistent mesh segmentation," in *CVPR*, pp. 2145–2152, 2011.

[63] A. Vaxman, M. Ben-Chen, and C. Gotsman, "A multi-resolution approach to heat kernels on discrete surfaces," *ACM Trans. Graph.*, 2010.

[64] Y. Fang, M. Sun, and K. Ramani, "Temperature distribution descriptor for robust 3D shape retrieval," in *CVPR*, 2011.

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

[66] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[67] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[68] D. Mo, "A survey on deep learning: one small step toward AI," *Dept. Computer Science, Univ. of New Mexico, USA*, 2012.

[69] M. Masoumi and A. Ben Hamza, "Shape classification using spectral graph wavelets," *Applied Intelligence*, accepted, 2017.

[70] M. Masoumi and A. Ben Hamza, "Spectral shape classification: A deep learning approach," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 198–211, 2017.

[71] M. Masoumi, C. Li, and A. Ben Hamza, "A spectral graph wavelet approach for nonrigid 3D shape retrieval," *Pattern Recognition Letters*, vol. 83, pp. 339–348, 2016.

[72] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[73] E. Rodola, L. Cosmo, O.Litany, M. M. Bronstein, A. M. Bronstein, N. Audebert, A. B. Hamza, A. Boulch, U. Castellani, M. N. Do, A.-D. Duong, T. Furuya, A. Gasparetto, Y. Hong, J. Kim, B. L. Saux, R. Litman, M. Masoumi, G. Minello, H.-D. Nguyen, V.-T. Nguyen, R. Ohbuchi, V.-K. Pham, T. V. Phan, M. Rezaei, A. Torsello, M.-T. Tran, Q.-T. Tran, B. Truong, L. Wan, and C. Zou11, "SHREC'17 track: Deformable shape retrieval with missing parts," in *Proc. Eurographics Workshop on 3D Object Retrieval 2017*, pp. 1–9, 2017.

[74] A. Chaudhari, R. Leahy, B. Wise, N. Lane, R. Badawi, and A. Joshi, "Global point signature for shape analysis of carpal bones," *Physics in Medicine and Biology*, vol. 59, pp. 961–973, 2014.

[75] K. Tarmissi and A. Ben Hamza, "Information-theoretic hashing of 3D objects using spectral graph theory," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9409–9414, 2009.

[76] Z. Gao, Z. Yu, and X. Pang, "A compact shape descriptor for triangular surface meshes," *Computer-Aided Design*, vol. 53, pp. 62–69, 2014.

[77] K. Gębal, J. A. Bærentzen, H. Aanæs, and R. Larsen, "Shape analysis using the auto diffusion function," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1405–1513, 2009.

[78] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen, "A comparison of methods for non-rigid 3D shape retrieval," *Pattern Recognition*, vol. 46, no. 1, pp. 449–461, 2013.

[79] C. Li and A. Ben Hamza, "Spatially aggregating spectral descriptors for nonrigid 3D shape retrieval: A comparative survey," *Multimedia Systems*, vol. 20, no. 3, pp. 253–281, 2014.

[80] C. Li and A. Ben Hamza, "Intrinsic spatial pyramid matching for deformable 3D shape retrieval," *International Journal of Multimedia Information Retrieval*, vol. 2, no. 4, pp. 261–271, 2013.

[81] S. Bu, Z. Liu, J. Han, J. Wu, and R. Ji, "Learning high-level feature by deep belief networks for 3-D model retrieval and recognition," *IEEE Trans. Multimedia*, vol. 24, no. 16, pp. 2154–2167, 2014.

[82] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *Proc. of the national academy of Sciences*, vol. 95, no. 15, pp. 8431–8435, 1998.

[83] M. Khabou, L. Hermi, and M. Rhouma, "Shape recognition using eigenvalues of the dirichlet laplacian," *Pattern Recognition*, vol. 40, pp. 141–153, 2007.

[84] Z. Lian, A. Godil, T. Fabry, T. Furuya, J. Hermans, R. Ohbuchi, C. Shu, D. Smeets, P. Suetens, D. Vandermeulen, and S. Wuhrer, "SHREC'10 track: Non-rigid 3D shape retrieval," in *Proc. Eurographics/ACM SIGGRAPH Sympo. 3D Object Retrieval*, pp. 101–108, 2010.

[85] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoue, H. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, , F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen, "SHREC'11 track: Shape retrieval on non-rigid 3D watertight meshes," in *Proc. Eurographics/ACM SIGGRAPH Symp. 3D Object Retrieval*, pp. 79–88, 2011.

[86] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *ICML*, pp. 473–480, 2007.

[87] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, pp. 609–616, 2009.

[88] G. Hinton and R. Salakhutdinov, "Using deep belief nets to learn covariance kernels for gaussian processes," in *NIPS*, pp. 1249–1256, 2008.

[89] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *AISTATS*, vol. 11, 2007.

[90] S. Osindero and G. Hinton, "Modeling image patches with a directed hierarchy of markov random fields," in *NIPS*, pp. 1121–1128, 2008.

[91] G. Taylor, G. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," *NIPS*, vol. 19, 2007.

[92] Z. Tu and S. Zhu, "Image segmentation by data-driven markov chain monte carlo," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 657–673, 2002.

[93] M. Welling, M. Rosen-Zvi, and G. Hinton, "Modeling image patches with a directed hierarchy of markov random fields," in *NIPS*, vol. 4, pp. 1481–1488, 2004.

[94] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *CVPR*, pp. 1912–1920, 2015.

[95] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *ICCV*, pp. 945–953, 2015.

[96] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller, and Y. LeCun, "Deep belief net learning in a long-range vision system for autonomous off-road driving," in *Int. Conf. Intelligent Robots and Systems*, pp. 628–633, 2008.

[97] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*, pp. 639–655, 2012.

[98] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *ICML*, pp. 791–798, 2007.

[99] J. Wan, D. Wang, S. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. 22nd ACM Int. Conf. on Multimedia*, pp. 157–166, 2014.

[100] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[101] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[102] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[103] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, pp. 609–616, 2009.

[104] M. Abdelrahman, M. El-Melegy, and A. Farag, "3D object classification using scale invariant heat kernels with collaborative classification," in *ECCV*, pp. 22–31, 2012.

[105] S. Biasotti, E. M. Thompson, M. Aono, A. B. Hamza, B. B. stos, S. Dong, B. Du, A. Fehri, H. Li, F. A. Limberger, M. Masoumi, M. Rezaei, I. Sipiran, L. Sun, A. Tatsuma, S. V. Forero, R. C. Wilson, Y. Wu, J. Zhang, T. Zhao, F. Fornasa, and A. Giachetti, "SHREC'17 track: Retrieval of surfaces with similar relief patterns," in *Proc. Eurographics Workshop on 3D Object Retrieval 2017*, pp. 1–10, 2017.

[106] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-D object retrieval and recognition with hypergraph analysis," *IEEE Trans. Image Processing*, vol. 21, no. 9, pp. 4290–4303, 2012.

[107] S. Zhao, H. Yao, Y. Zhang, Y. Wang, and S. Liu, "View-based 3D object retrieval via multimodal graph learning," *Signal Processing*, vol. 112, pp. 110–118, 2015.

[108] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoue, H. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen, "A comparison of methods for non-rigid 3D shape retrieval," *Pattern Recognition*, vol. 46, no. 1, pp. 449–461, 2013.

[109] Z. Lian, J. Zhang, S. Choi, H. ElNaghy, J. El-Sana, T. Furuya, A. Giachetti, R. Guler, L. Isaia, L. Lai, C. Li, H. Li, F. Limberger, R. Martin, R. Nakanishi, A. Neto, L. Nonato, R. Ohbuchi, K. Pevzner, D. Pickup, P. Rosin, A. Sharf, L. Sun, X. Sun, S. Tari, G. Unal, and R. Wilson, "SHREC'15 track: Non-rigid 3D shape retrieval," in *Proc. Eurographics/ACM SIGGRAPH Symp. 3D Object Retrieval*, 2015.

[110] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Artificial Intelligence and Statistics*, pp. 448–455, 2009.

[111] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[112] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.