

# Spectral Methods for Mesh Processing and Analysis

Hao Zhang    Oliver van Kaick    Ramsay Dyer

GrUVi Lab, School of Computing Science, Simon Fraser University, BC, Canada

---

## Abstract

*Spectral methods for mesh processing and analysis rely on the eigenvalues, eigenvectors, or eigenspace projections derived from appropriately defined mesh operators to carry out desired tasks. Early works in this area can be traced back to the seminal paper by Taubin in 1995, where spectral analysis of mesh geometry based on a combinatorial Laplacian aids our understanding of the low-pass filtering approach to mesh smoothing. Over the past ten years or so, the list of applications in the area of geometry processing which utilize the eigenstructures of a variety of mesh operators in different manners have been growing steadily. Many works presented so far draw parallels from developments in fields such as graph theory, computer vision, machine learning, graph drawing, numerical linear algebra, and high-performance computing. This state-of-the-art report aims to provide a comprehensive survey on the spectral approach, focusing on its power and versatility in solving geometry processing problems and attempting to bridge the gap between relevant research in computer graphics and other fields. Necessary theoretical background will be provided and existing works will be classified according to different criteria — the operators or eigenstructures employed, application domains, or the dimensionality of the spectral embeddings used — and described in adequate length. Finally, despite much empirical success, there still remain many open questions pertaining to the spectral approach, which we will discuss in the report as well.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

---

## 1. Introduction

A great number of spectral methods have been proposed in the computing science literature in recent years, appearing in the fields of graph theory, computer vision, machine learning, visualization, graph drawing, high performance computing, and computer graphics. Generally speaking, a spectral method solves a problem by examining or manipulating the eigenvalues, eigenvectors, eigenspace projections, or a combination of these quantities, derived from an appropriately defined linear operator. More specific to the area of geometry processing and analysis, spectral methods have been developed with the intention of solving a diversity of problems including mesh compression, correspondence, parameterization, segmentation, sequencing, smoothing, watermarking, surface reconstruction, and remeshing.

As a consequence of these developments, researchers are now faced with an extensive literature related to spectral methods and it might be a laborious task for those new to the field to collect the necessary references in order to obtain an overview of the different methods, as well as an under-

standing of their similarities and differences. Furthermore, this is a topic that still instigates much interest, and there are still many open problems to be addressed, which provide numerous potential possibilities for further investigation. Although introductory and short surveys which cover particular aspects of the spectral approach have been given before, e.g., by Gotsman [Got03] on spectral partitioning, layout, and geometry coding, and more recently by Lévy [Lev06] on a study of Laplace-Beltrami eigenfunctions, we believe a comprehensive survey is still called for. In presenting this report, our goal is to provide a sufficient theoretical background, informative insights, as well as a thorough and up-to-date reference on the topic so as to draw interested researchers into this area and facilitate future research. Our effort should also serve to bridge the gap between past and on-going developments in several related disciplines.

The rest of the document is organized as follows. We start with a brief historical account of the use of spectral methods. Section 3 offers an overview of the spectral approach, presenting the general solution paradigm and discussing possi-

ble classifications. Section 4 motivates the use of the spectral approach through a few examples and mentions at a high level several natural applications. In Section 5, we provide some theoretical background with several theorems from linear algebra and other results that are frequently encountered in the literature covering spectral methods. Section 6 surveys existing operators used for spectral mesh processing and analysis, while Section 7 outlines how the different eigenstructures can be utilized to solve respective problems. Computational issues are addressed in Section 8. Section 9 finally provides a detailed survey of specific spectral methods. Finally, we offer a few open questions for future consideration in Section 10.

## 2. A historical account

Long before spectral methods came about in the computer graphics and geometry processing community, a great deal of knowledge from the field of spectral graph theory had been accumulated, following the pioneering work of Fielder [Fie73] in the 1970's. A detailed account of results from this theory can be found in the book by Chung [Chu97], two survey papers by Mohar [MP93, Moh97], as well as other graph theory texts, e.g., [Bo198].

The focus in spectral graph theory has been to derive relationships between the eigenvalues of the Laplacian or adjacency matrices of a graph and various fundamental properties of the graph, e.g., its diameter and connectivity [Chu97]. It has long been known that the graph Laplacian can be seen as a combinatorial version of the Laplace-Beltrami operator from Riemannian geometry [Moh97]. Thus the interplay between spectral Riemannian geometry [Cha84] and spectral graph theory has also been a subject of study [Chu97].

One major development stemming from spectral graph theory that has found many practical applications involves the use of the *Fielder vector*, the eigenvector of a graph Laplacian corresponding to the smallest non-zero eigenvalue. These applications include graph or mesh layout [DPS02, Kor03], image segmentation [SM00], graph partitioning for parallel computing [AKY99], as well as sparse matrix reordering [BPS93] in numerical linear algebra. For the most part, these works had not received a great deal of attention from the graphics community until recently.

Treating the mesh vertex coordinates as a 3D signal defined over the underlying mesh graph, Taubin [Tau95] first introduced the use of mesh Laplacian operators for discrete geometry processing in his SIGGRAPH 1995 paper. What had motivated this development were not results from spectral graph theory but an analogy between spectral analysis with respect to the mesh Laplacian and the classical discrete Fourier analysis. Such an analysis was then applied to the irregular grids characterizing general meshes and mesh smoothing was done via low-pass filtering without expensive optimization. Subsequently, projections of a

mesh signal into the eigenspaces of mesh Laplacians have been studied for different problems, e.g., implicit mesh fairing [DMSB99, KR05, ZF03], geometry compression [KG00, SCOT03], and mesh watermarking [OTMM01, OMT02]. A summary of the filtering approach to mesh geometry processing was given by Taubin in his Eurographics state-of-the-art report [Tau00]. Mesh Laplacians operators also allow us to define *differential coordinates* to represent mesh geometry, which is useful in applications such as mesh editing and shape interpolation. These works have been surveyed in the state-of-the-art report given by Sorkine [Sor05].

While mesh filtering [Tau95, DMSB99, ZF03] can be efficiently carried out in the spatial domain via convolution, methods which require explicit computation of the eigenvectors of the mesh Laplacian, e.g., [KG00, OTMM01], had suffered from the problem of high computation cost. A remedy was proposed to first partition the mesh into smaller patches and then perform spectral processing on a per patch basis [KG00]. In fact, one may even choose to perform regular resampling and conduct the classical Fourier analysis in each patch [PG01]. However, artifacts emerging at the artificially introduced patch boundaries may occur and it would still be desirable to perform global spectral analysis over the whole mesh surface in a seamless manner.

Recently, efficient schemes for eigenvector computation, e.g., with the use of algebraic multi-grid methods [KCH02] or spectral shift [DBG\*06, VL07], as well as eigenvector approximation via the Nyström method [FBCM04], have feuded renewed interests in spectral methods for mesh processing and analysis. At the same time, developments in fields such as computer vision and machine learning on spectral techniques have started to exert more influence on the computer graphics community. These inspiring developments include spectral graph matching and point correspondence from computer vision, dating back to the works of Umeyama [Ume88] and Shapiro and Brady [SB92] in the late 1980's and early 1990's. Extremal properties of the eigenvectors known from linear algebra provided the theoretical background. These techniques have been extended to the correspondence between 3D meshes, e.g., [JZvK07].

The method of spectral clustering [vL06] from machine learning has also been utilized in geometry processing, e.g., [KSO04, LZ04]. An important result from graph theory that plays a crucial role in spectral clustering relates the number of connected components in a graph with the multiplicity of the zero eigenvalue of the graph Laplacian. Recently, studies in the machine learning community on spectral clustering have focused on explaining its success [NJW02] as well as discovering its limitations [vLBB05].

The use of the Fielder vector for linear graph layout has found applications in mesh sequencing [IL05] and mesh segmentation recently [ZL05, LZ07]. The study of Laplacian eigenfunctions over mesh surfaces has yielded a new application, that of quadrilateral remeshing [DBG\*06]. Last

but not the least, multidimensional scaling [CC94] and null-space graph embeddings [LS99] have been applied to planar [ZKK02] and spherical [Got03] mesh parameterizations, respectively.

### 3. Overview of the spectral approach

Most spectral methods have a basic framework in common, which can be roughly divided into three steps.

1. A matrix  $M$  which represents a discrete operator based on the structure of the input mesh is defined. This matrix can be seen as incorporating pairwise relations between mesh elements. That is, each entry  $M_{ij}$  possesses a value that represents the relation between the vertices (faces or other primitives)  $i$  and  $j$  of the mesh. The pairwise relations can take into account only vertex connectivity or combine topological and geometric information.
2. An eigendecomposition of the matrix  $M$  is performed, that is, its eigenvalues and eigenvectors are computed.
3. The eigendecomposition is further employed in a problem-dependent manner to obtain a desired solution.

In view of the above framework, the variations for the different spectral methods arise in how the matrix  $M$  is composed and how the eigendecomposition is employed to obtain the result, since eigenvalues, eigenvectors, or eigenspace projections can all be used. These lead to a few possible classifications of spectral methods, which we outline below.

- **Based on the operator used:**

Depending on whether the matrix  $M$  should be defined by the geometry of the input mesh, one can classify linear mesh operators used for spectral analysis as either *combinatorial* or *geometric*.

It is also possible to distinguish between matrices which encode graph *adjacency* information, along with their extensions, and matrices which approximate the *Laplacian* operator, e.g., the graph Laplacian used in spectral graph theory [Bol98, Chu97, Moh97]. In graph-theoretic terminology, the adjacency matrix is sometimes said to model the *Lagrangian* of a graph [Bol98]. One possible extension of the graph Laplacian operator is to the class of *discrete Schrödinger operators*, e.g., see [BHL\*04, DGLS01]. The precise definition of these and other operators mentioned in this section will be given in Section 6.

Both the graph adjacency and the Laplacian matrices can also be extended to incorporate higher-order neighborhood information. That is, relationships between all pairs of mesh elements are modeled instead of only considering element pairs that are adjacent in a mesh graph. A particularly important class of such operators are the so-called *Gram matrices*, e.g., see [STWCK05]. These matrices play a crucial role in several techniques from machine learning, including spectral clustering [vL06] and kernel-

based methods [SS02], e.g., kernel principal component analysis (Kernel PCA) [SSM98].

- **Based on the eigenstructures used:**

In graph theory, the focus has been placed on the eigenvalues of graph adjacency or Laplacian matrices. Many results are known which relate these eigenvalues to graph-theoretical properties [Chu97]. While from a theoretical point of view, it is of interest to obtain various bounds on the graph invariants from the eigenvalues, several practical applications simply rely on the eigenvalues of appropriately defined graphs to characterize geometric shapes, e.g., [JZ07, RWP06, SMD\*05, SSGD03].

Indeed, eigenvalues and eigenspace projections are primarily used to derive shape descriptors (or signatures) for shape matching and retrieval applications, where the latter, obtained by projecting a mesh representation along the appropriate eigenvectors, mimics the behavior of Fourier descriptors [ZR72] in the classical setting.

Eigenvectors, on the other hand, are most frequently used to derive a *spectral embedding* of the input data, e.g., a mesh shape. Often, the new (spectral) domain is more convenient to operate on, e.g., it is low dimensional, while the transform still retains as much information about the input data as possible. This issue, along with the use of eigenvalues and Fourier descriptors for shape characterization, will be discussed further in Section 7 and 9.

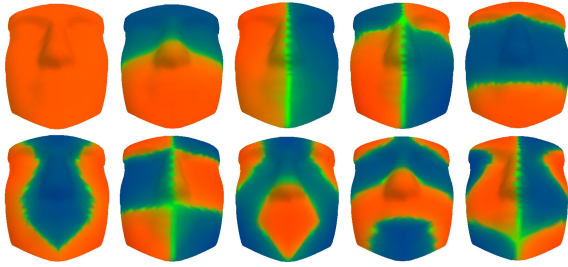
- **Based on the dimensionality of the eigenstructure:**

This classification is more relevant to the use of eigenvectors for the construction of spectral embeddings. One-dimensional embeddings typically serve as solutions to ordering or sequencing problems, where some specific optimization criterion is to be met. In many instances, the optimization problem is NP-hard and the use of an eigenvector provides a good heuristic [DPS02, MP93]. Of particular importance is the Fiedler vector [Fie73]. For example, it has been used by the well-known normalized cut algorithm for image segmentation [SM00].

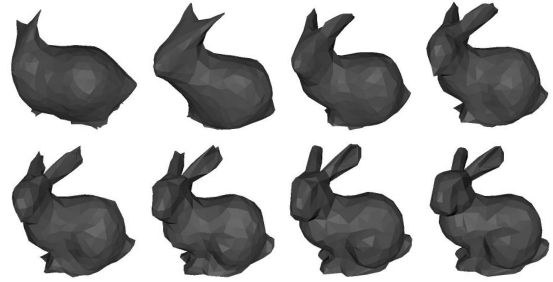
Two-dimensional spectral embeddings have been used for graph drawing [KCH02] and mesh flattening [ZSGS04, ZKK02], and three-dimensional embeddings have been applied to spherical mesh parameterization [Got03]. Generally speaking, low-dimensional embeddings can be utilized to facilitate solutions to several geometric processing problems, including mesh segmentation [LZ04, ZL05, LZ07] and correspondence [JZvK07]. These works are inspired by the use of the spectral approach for clustering [vL06] and graph matching [SB92, Ume88].

## 4. Motivation

In this section, we motivate the use of the spectral approach for mesh processing and analysis from several perspectives. These discussions naturally reveal which classes of prob-



**Figure 1:** Color plot of the first 10 eigenvectors of the graph Laplacian for a face mesh. A nonlinear color transfer function is applied to enhance the contrast. The apparent alignment between the color patch boundaries and face features is due to the feature-sensitive tessellation of the face mesh.



**Figure 2:** A 557-vertex bunny model reconstructed using 10, 30, 60, 90, 120, 240, 480, and 557 eigenvectors of the graph Laplacian.

lems are suitable for the spectral approach. Several examples are presented to better illustrate the ideas.

#### 4.1. Harmonic behavior of Laplacian eigenvectors

One of the main reasons that combinatorial and geometric Laplacians are often considered for spectral mesh processing is that their eigenvectors possess similar properties as the Fourier basis functions. By representing mesh geometry using a discrete signal defined over the manifold mesh surface, it is commonly believed that a Fourier transform of such a signal can be obtained by an eigenspace projection of the signal along the eigenvectors of a mesh Laplacian. This stipulation was first applied by Taubin [Tau95] to develop a signal processing framework for mesh fairing, where an analogy to the classical 1D situation was taken advantage of.

Indeed, the classical Fourier transform of a periodic 1D signal can be seen as the decomposition of the signal into a linear combination of the eigenvectors of the Laplacian operator.<sup>†</sup> A combinatorial mesh Laplacian is then adopted to conduct Fourier analysis on a mesh signal. A distinction between the mesh case and the classical Fourier transform is that while the latter uses a fixed set of basis functions, the eigenvectors which serve as “Fourier-like” bases for mesh signal processing would change depending on mesh connectivity, geometry, and which type of Laplacian operator is adopted. Nevertheless, the eigenvectors of the mesh Laplacians all appear to exhibit harmonic behavior and are seen as the *vibration modes* or the *harmonics* of the mesh surface and their corresponding eigenvalues are the associated *frequencies* [Tau95]. Consequently, mesh fairing can be carried out via low-pass filtering. This approach and subsequent developments have been described in details in [Tau00].

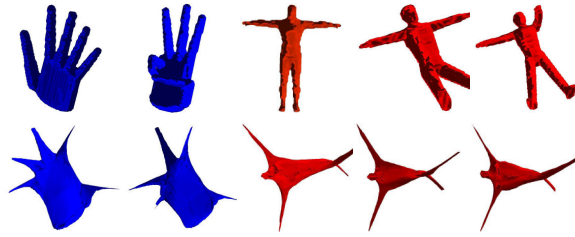
<sup>†</sup> It is worth noting here that this statement still holds if we replace the Laplacian operator by any *circulant matrix* [Jai89].

In Figure 1, we give color plots of the first 10 eigenvectors of the combinatorial graph Laplacian of a face mesh, where the entries of an eigenvector are color-mapped. As we can see, the harmonic behavior of the eigenvectors is evident. Although the filtering approach proposed by Taubin does not fall strictly into the category of spectral methods since neither the eigenvalues nor the eigenvectors of the mesh Laplacian are explicitly computed, the resemblance to classical Fourier analysis implies that any application which utilizes the Fourier transform can be applied in the mesh setting, e.g., geometry compression [KG00]. In Figure 2, we show a simplified Stanford bunny model (with 557 vertices and 999 faces) reconstructed using a few truncated spectrums derived from the graph Laplacian.

#### 4.2. Modeling of global characteristics

Although each entry in a linear mesh operator may encode only local information, it is widely held that the eigenvalues and eigenvectors of the operator can reveal meaningful global information about the mesh shape. This is hardly surprising from the perspective of spectral graph theory, where many results are known which relate extremal properties of a graph, e.g., its diameter and Cheeger constant, with the eigenvalues of the graph Laplacian.

As Chung stated in her book [Chu97], results from spectral theory suggest that the Laplacian eigenvalues are closely related to almost all major graph invariants. Thus if a matrix models the structures of a shape, either in terms of topology or geometry, then we would expect its set of eigenvalues to provide an adequate characterization of the shape. Indeed, this has motivated the use of graph spectra for shape matching and retrieval in computer vision [SMD\*05,SSGD03] and geometry processing [JZ07,RWP06]. The eigenvalues serve as compact global shape descriptors. They are sorted by their magnitudes so as to establish a correspondence for computing the similarity distance between two shapes. In this context, it is not necessary to consider what particular characteristic an individual eigenvalue reveals.



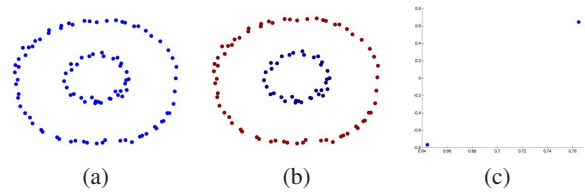
**Figure 3:** Spectral embeddings (bottom row) of some articulated 3D shapes (top row) from the McGill 3D shape benchmark database [McG]. Since the mesh operator is constructed from geodesic distances, the embeddings are normalized with respect to shape bending.

Compared with eigenvalues, eigenvectors provide more refined shape characterization which also tends to have a global nature. For instance, it can be shown [SFYD04] that pairwise distances between points given by the spectral embeddings derived from the graph Laplacian model the so-called commute-time distances [Lov93], a global measure related to the behavior of random walks on a graph. Eigenvectors also possess extremal properties, highlighted by the Courant-Fischer theorem (given in Section 5), which enable spectral techniques to provide high-quality results for several NP-hard global optimization problems, including normalized cuts [SM97] and the linear arrangement problem [DPS02], among others [MP93].

#### 4.3. Structure revelation

Depending on the requirement of the problem at hand, the operator we use to compute the spectral embeddings can be made to incorporate any intrinsic measure on a shape in order to obtain useful invariance properties, e.g., with respect to part articulation. In Figure 3, we show 3D spectral embeddings of a few mesh models obtained from an operator derived from geodesic distances over the mesh surface. As geodesic distance is bending-invariant, the resulting embeddings are normalized with respect to bending and can facilitate shape retrieval under part articulation [EK03, JZ07].

Generally speaking, with an appropriately designed linear operator, the resulting spectral embedding can better reveal, single out, or even exaggerate useful underlying structures in the input data. The above example shows that via a transformation into the spectral domain, certain intrinsic shape structures, e.g., part composition of the shape, are better revealed by removing the effects of other features, e.g., those resulting from shape bending. In other instances, the spectral approach can present the nonlinear structures in an input data in a high-dimensional feature space so that they become much easier to handle. In particular, the nonlinear structures may be “unfolded” into linear ones so that methods based on



**Figure 4:** Result of spectral clustering, shown in (b), on the 2-ring data set (a). (c) shows the 2D spectral embedding.

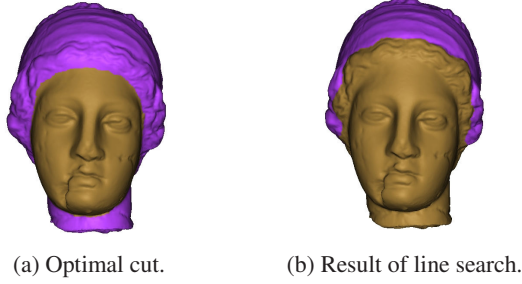
linear transformations and linear separators, e.g., PCA and  $k$ -means clustering, can be applied.

One classical example to illustrate this, is the problem of clustering the 2D data set shown in Figure 4(a). Although to a human observer, the data should clearly be clustered into an outer and an inner ring, conventional clustering methods such as  $k$ -means would fail. However, by constructing a Gram matrix based on Euclidean distances and a Gaussian kernel and then spectrally embedding the data into the 2D domain, we arrive at the set shown in Figure 4(c). This set is trivial to cluster via  $k$ -means to obtain the desired result in (b). This is the spectral clustering method [vL06].

#### 4.4. Dimensionality reduction

Typically, the dimensionality of the linear operator used for spectral mesh analysis is equal to the size of the input mesh, which can become quite large. By properly selecting a small number of leading eigenvectors of the operator to construct an embedding, the dimensionality of the problem is effectively reduced while the global characteristics of the original data set are still retained. In fact, the extremal properties of the eigenvectors ensure that the spectral embeddings are “information-preserving”; this is suggested by a theorem due to Eckart and Young [EY36], which we give in Section 5 as Theorem 5.5. Furthermore, there is evidence that the clustering structures in the input data may be enhanced in a low-dimensional embedding space, as hinted by the Polarization Theorem [BH03] (Theorem 5.6).

Some of the advantages of dimensionality reduction include computational efficiency and problem simplification. One such example is image segmentation using normalized cuts [SM00]. In a recursive setting, each iteration of the segmentation algorithm corresponds to a line search along a 1D embedding obtained by the Fiedler vector of a weighted graph Laplacian. The same idea has been applied to mesh segmentation [ZL05, LZ07] where the simplicity of the line search allows to incorporate any efficiently computable (but not necessarily easy to optimize) search criteria, e.g., part saliency [HS97]. In Figure 5(a), we show the best cut present in the mesh face sequence obtained using the 1D spectral embedding technique given by Liu and Zhang [LZ07]. This example reflects the ability of spectral embeddings to reveal meaningful global shape characteristics for a model that is



**Figure 5:** First cut on the Igea model (example taken from [LZ07]). (a) The best cut present in the mesh face sequence. (b) Result from line search based on part saliency.

difficult to segment and in only one dimension. However, line search based on part saliency does not return the best result, as shown in (b). This is due to the inability of the part saliency measure to capture the most meaningful cut.

## 5. Theoretical background

In this section, we list a few theorems from linear algebra related to eigenstructures of general matrices, as well as a few useful results concerning spectral embeddings. These include the Spectral Theorem, the Courant-Fischer Theorem, the Ky-Fan theorem [Bha97], and the Polarization Theorem [BH03]. These theorems are stated here without proofs. Proofs of some of these theorems can be found in the associated references. The Spectral and Courant-Fischer theorems are well known in linear algebra, whose proofs can be found in many standard linear algebra texts, e.g., [Mey01, TB97].

Given an  $n \times n$  matrix  $M$ , to obtain its eigendecomposition, we compute its eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  and associated eigenvectors  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ . By definition,

$$M\mathbf{v}_i = \lambda_i \mathbf{v}_i \text{ and } \mathbf{v}_i \neq 0, \text{ for } i \in \{1 \dots n\}.$$

The set of eigenvalues  $\lambda(M) = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$  is known as the *spectrum* of the matrix.

When the matrix  $M$  is generalized to a linear operator acting on a Hilbert space, the eigenvectors become *eigenfunctions* of the operator. For our purpose, we will focus on real symmetric matrices, whose counterpart in functional analysis are compact self-adjoint operators. The main advantage offered by symmetric matrices is that they possess real eigenvalues whose eigenvectors form an orthogonal basis, that is,  $\mathbf{v}_i^T \mathbf{v}_j = 0$  for  $i \neq j$ . The eigendecomposition of a real symmetric matrix is described by the Spectral Theorem:

**Theorem 5.1 (The Spectral Theorem)** Let  $S$  be a real symmetric matrix of dimension  $n$ . Then we have

$$S = V\Lambda V^T = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{v}_i^T,$$

the eigendecomposition of  $S$ , where  $V = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_n]$  is

the matrix of eigenvectors of  $S$  and  $\Lambda$  is the diagonal matrix of the eigenvalues of  $S$ . The eigenvalues of  $S$  are real and its eigenvectors are orthogonal, i.e.,  $V^T V = I$ , where  $M^T$  denotes the transpose of a matrix  $M$  and  $I$  is the identity matrix.

One of the most fundamental theorems which characterize eigenvalues and eigenvectors of a symmetric matrix is the Courant-Fischer theorem. It reveals certain extremal property of eigenvectors, which has frequently motivated the use of eigenvectors and the embeddings they define for solving a variety of optimization problems.

**Theorem 5.2 (Courant-Fischer)** Let  $S$  be a real symmetric matrix of dimension  $n$ . Then its eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  satisfy the following,

$$\lambda_i = \min_{\substack{\mathcal{V} \subset \mathbb{R}^n \\ \dim \mathcal{V} = i}} \max_{\substack{\mathbf{v} \in \mathcal{V} \\ \|\mathbf{v}\|_2 = 1}} \mathbf{v}^T S \mathbf{v}$$

where  $\mathcal{V}$  is a subspace of  $\mathbb{R}^n$  with the given dimension. When considering only the smallest eigenvalue of  $S$ , we have

$$\lambda_1 = \min_{\|\mathbf{v}\|_2 = 1} \mathbf{v}^T S \mathbf{v}.$$

Similarly, the largest eigenvalue

$$\lambda_n = \max_{\|\mathbf{v}\|_2 = 1} \mathbf{v}^T S \mathbf{v}.$$

When the unit length constraint is removed, the quadratic form  $\mathbf{v}^T S \mathbf{v}$  in Theorem 5.2 becomes the well-known *Rayleigh quotient*  $\mathbf{v}^T S \mathbf{v} / \mathbf{v}^T \mathbf{v}$ . Another way of characterizing the eigenstructures is the following result, which can be seen as a corollary of the Courant-Fischer Theorem.

**Theorem 5.3** Let  $S$  be a real symmetric matrix of dimension  $n$ . Then its eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  satisfy the following,

$$\lambda_i = \min_{\substack{\|\mathbf{v}\|_2 = 1 \\ \mathbf{v}^T \mathbf{v}_k = 0, \forall 1 \leq k \leq i-1}} \mathbf{v}^T S \mathbf{v},$$

where  $\mathbf{v}_1, \dots, \mathbf{v}_{i-1}$  are the eigenvectors of  $S$  corresponding to eigenvalues  $\lambda_1, \dots, \lambda_{i-1}$ , respectively.

Another useful theorem which relates the sum of the partial spectrum of a symmetric matrix to the respective eigenvectors is also known.

**Theorem 5.4 (Ky-Fan)** Let  $S$  be a real symmetric matrix with eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ . Then

$$\sum_{i=1}^k \lambda_i = \min_{\substack{U \in \mathbb{R}^{n \times k} \\ U^T U = I_k}} \text{tr}(U^T S U),$$

where  $\text{tr}(M)$  denotes the trace of a matrix  $M$  and  $I_k$  is the  $k \times k$  identity matrix.

One may also interpret Theorem 5.4 as saying that a set of  $k$  orthogonal vectors which minimizes the matrix trace in the theorem is given by the  $k$  eigenvectors corresponding

to the  $k$  smallest eigenvalues. Clearly, if  $U$  consists of the  $k$  eigenvectors of  $S$  corresponding to eigenvalues  $\lambda_1, \dots, \lambda_k$ , then we have  $\text{tr}(U^T S U) = \sum_{i=1}^k \lambda_i$ .

Taking an alternative view, we will see that the set of leading eigenvectors of a symmetric matrix plays a role in low-rank approximation of matrices, as given by a theorem due to Eckart and Young [EY36]. This result is useful in studying the properties of spectral embeddings, e.g., [BH03, dST04].

**Theorem 5.5 (Eckart-Young)** Let  $S$  be a real, symmetric and positive semi-definite matrix of dimension  $n$  and let  $S = V\Lambda V^T$  be the eigendecomposition of  $S$ . Suppose that the eigenvalues, given along the diagonal of  $\Lambda$ , are in descending order. Let  $X = V\Lambda^{1/2}$  be the matrix of eigenvectors that are scaled by the square root of their respective eigenvalues. Denote by  $X_{(k)} \in \mathbb{R}^{n \times k}$  a truncated version of  $X$ , i.e., its columns consist of the  $k$  leading columns of  $X$ . Then

$$X_{(k)} = \underset{U \in \mathbb{R}^{n \times k}}{\text{argmin}} \quad \frac{\|S - UU^T\|_F}{\text{rank}(U) = k},$$

where  $\text{rank}(M)$  denotes the rank of a matrix  $M$  and  $\|\cdot\|_F$  is the Frobenius norm.

Theorem 5.5 states that the outer product of the  $k$  largest eigenvectors of  $S$  (eigenvectors corresponding to the largest eigenvalues), when scaled using the square root of their respective eigenvalues, provides the *best rank- $k$  approximation* of  $S$ . As a related result, we mention an interesting theorem which suggests that the clustering structures in a data set are somewhat *exaggerated* as the dimensionality of the spectral embedding decreases. This is the Polarization theorem due to Brand and Huang [BH03].

**Theorem 5.6 (Polarization Theorem)** Denote by  $S_{(k)} = X_{(k)}X_{(k)}^T$  the best rank- $k$  approximation of  $S$  with respect to the Frobenius norm, where  $X_{(k)}$  is as defined in Theorem 5.5. As  $S$  is projected to successively lower ranks  $S_{(n-1)}, S_{(n-2)}, \dots, S_{(2)}, S_{(1)}$ , the sum of squared angle-cosines,

$$s_k = \sum_{i \neq j} (\cos \theta_{ij}^{(k)})^2 = \sum_{i \neq j} \frac{\mathbf{x}_i^{(k)\top} \mathbf{x}_j^{(k)}}{\|\mathbf{x}_i^{(k)}\|_2 \cdot \|\mathbf{x}_j^{(k)}\|_2}$$

is strictly increasing, where  $\mathbf{x}_i^{(k)}$  is the  $i$ -th row of  $X_{(k)}$ .

This theorem states that as the dimensionality of the representation is reduced, the distribution of the cosines migrates away from 0 towards two poles  $+1$  or  $-1$ , such that the angles migrate from  $\theta_{ij} = \pi/2$  to  $\theta_{ij} \in \{0, \pi\}$ .

## 6. Linear operators for spectral mesh processing

We devote this section to the definition of and discussion on linear operators for spectral mesh processing and analysis. After introducing notations to be used, coverage will be provided on combinatorial mesh Laplacians, geometric mesh Laplacians, discrete Schrödinger operators, and higher-order operators (in particular, Gram matrices).

### 6.1. Notation

A triangle mesh with  $n$  vertices is represented as  $\mathcal{M} = (\mathcal{G}, P)$ , where  $\mathcal{G} = (V, E)$  models the mesh graph, with  $V$  denoting the set of mesh vertices and  $E \subseteq V \times V$  the set of edges, and  $P \in \mathbb{R}^{n \times 3}$  represents the geometry of the mesh, given by an array of 3D vertex coordinates. Each vertex  $i \in V$  has an associated position vector, denoted by  $\mathbf{p}_i = [x_i \ y_i \ z_i]$ ; it corresponds to the  $i$ -th row of  $P$ . In subsequent discussions, matrices are denoted in upper-case letters (e.g.,  $M$ ), vectors in lower-case bold (e.g.,  $\mathbf{v}$ ), and scalars or functions in lower-case roman (e.g.,  $s$ ). The  $i$ -th element of a vector  $\mathbf{v}$  is denoted  $v_i$ , and the  $(i, j)$ -th element of a matrix  $M$  is denoted as  $M_{ij}$ .

### 6.2. Mesh Laplacians: an overview

Let us first provide a general view of mesh Laplacians and then define specific combinatorial and geometric Laplacians in subsequent sections. Mesh Laplacian operators are linear operators that act on functions defined on a mesh. These functions are specified by their values at the vertices. Thus if a mesh  $\mathcal{M}$  has  $n$  vertices, then functions on  $\mathcal{M}$  will be represented by vectors with  $n$  components and a mesh Laplacian will be described by an  $n \times n$  matrix.

Loosely speaking, a mesh Laplacian operator locally takes a weighted average of the differences between the value of a function at a vertex and its value at the first-order or immediate neighbour vertices. Specifically, for our purposes, a Laplacian will have a local form given by

$$(Lf)_i = b_i^{-1} \sum_{j \in N(i)} w_{ij} (f_i - f_j), \quad (1)$$

where  $N(i)$  is the set of neighbours of vertex  $i$ , and the  $w_{ij}$ 's are the edge weights and they are symmetric:  $w_{ij} = w_{ji}$ . The factor  $b_i^{-1}$  is a positive number. Its expression as an inverse will appear natural in subsequent developments.

An operator that is locally expressed by (1) can be factored into the product of a diagonal and a symmetric matrix

$$L = B^{-1}S, \quad (2)$$

where  $B^{-1}$  is a diagonal matrix whose diagonal entries are the  $b_i^{-1}$ 's and  $S$  is a symmetric matrix whose diagonal entries are given by  $S_{ii} = \sum_{j \in N(i)} w_{ij}$  and whose off diagonal entries are  $-w_{ij}$ . Although  $L$  itself is not symmetric in general, it is similar to the symmetric matrix  $O = B^{-1/2}SB^{-1/2}$  since

$$L = B^{-1}S = B^{-1/2}B^{-1/2}SB^{-1/2}B^{1/2} = B^{-1/2}OB^{1/2}.$$

Thus  $L$  and  $O$  have the same real eigenvalues. And if  $\mathbf{v}_i$  is an eigenvector of  $O$  with eigenvalue  $\lambda_i$ , then  $\mathbf{u}_i = B^{-1/2}\mathbf{v}_i$  is an eigenvector of  $L$  with the same eigenvalue.

The eigenvectors of  $O$  are mutually orthogonal, since  $O$  is symmetric. This is not generally true for  $L$ . However, if we define a scalar product by

$$\langle f, g \rangle_B = f^T B g,$$

then the eigenvectors of  $L$  are orthogonal with respect to that product:

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle_B = \mathbf{u}_i^T B \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}.$$

### 6.3. Combinatorial mesh Laplacians

A combinatorial mesh Laplacian is completely defined by the graph associated with the mesh. The adjacency matrix  $W$  of a mesh  $\mathcal{M} = (V, E, P)$ , where  $|V| = n$  and  $|E| = m$ , is given by

$$W_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

#### 6.3.1. Graph, normalized graph, and Tutte Laplacians

Given  $W$ , the degree matrix  $D$  is defined as

$$D_{ij} = \begin{cases} d_i = |N(i)| & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

where  $d_i$  is said to be the degree of vertex  $i$ . We define the *graph Laplacian* matrix  $K$  as

$$K = D - W.$$

Referring to equation (1), we see that  $K$  can be defined from (1) by setting  $b_i = w_{ij} = 1$  for all  $i, j$ . The operator  $K$  is also known as the *Kirchoff operator* [OTMM01], as it has been encountered in the study of electrical networks by Kirchoff. In that context, the weighted adjacency matrix  $W$  is referred as the *conductance matrix* [GM00].

Note that in the current literature, there is no complete agreement as to what should be called a graph Laplacian. In Chung's book [Chu97], for example, the following normalized version of  $K$ ,

$$Q = D^{-1/2} K D^{-1/2},$$

with

$$Q_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -1/\sqrt{d_i d_j} & \text{if } (i, j) \in E, \\ 0 & \text{otherwise,} \end{cases}$$

is called a graph Laplacian. In this paper, we call  $Q$  the *normalized graph Laplacian*.

There is yet another operator that has been applied as a combinatorial mesh Laplacian. It was used by Taubin [Tau95] in his signal processing approach to mesh fairing, as well as in the context of planar graph drawing [Kor03], first studied by Tutte [Tut63]. Following the terminology used by Gotsman et al. [Got03], we call this operator the *Tutte Laplacian*. It is defined as

$$T = D^{-1} K,$$

where

$$T_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -1/d_i & \text{if } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Referring to equation (1), we see that  $T$  can be defined from (1) by setting  $b_i = d_i$  for all  $i$  and  $w_{ij} = 1$  for all  $i, j$ . It is also easy to see that  $T$  and  $Q$  are similar to each other, as  $T = D^{-1/2} Q D^{1/2}$ .

Clearly, for a manifold triangle mesh, the matrices  $L, Q$ , and  $T$  are all expected to be sparse with the average number of non-zero entries per row about seven.

Finally, it is trivial to extend the above definitions to weighted graphs, where the graph adjacency matrix  $W$  would be defined by  $W_{ij} = w(e_{ij}) = w_{ij}$ , for some edge weight  $w : E \rightarrow \mathbb{R}^+$ , whenever  $(i, j) \in E$ . Then, it is necessary to define the diagonal entries of the degree matrix  $D$  as  $D_{ii} = \sum_{j \in N(i)} w_{ij}$ .

#### 6.3.2. Graph Laplacian and Laplace-Beltrami operator

Following Mohar [Moh97], we show below that the graph Laplacian  $K$  can be seen as a combinatorial analogue of the Laplace-Beltrami operator defined on a manifold.

Let us first define an *oriented incidence matrix*  $R$  of a mesh graph  $G$  as follows. Orient each of the  $m$  edges of  $G$  in an arbitrary manner. Then  $R \in \mathbb{R}^{n \times m}$  is a vertex-edge incidence matrix where

$$R_{ie} = \begin{cases} -1 & \text{if } i \text{ is the initial vertex of edge } e, \\ +1 & \text{if } i \text{ is the terminal vertex of edge } e. \end{cases}$$

It is not hard to show that  $K = R R^T$  regardless of the assignment of edge orientations [Moh97].

The Laplace-Beltrami operator  $\Delta$  on a Riemannian manifold is a second-order differential operator defined as the divergence of the gradient [Ros97]. Given an infinitely continuous real scalar function  $\phi$  defined over the manifold,

$$\Delta(\phi) = \text{div}(\text{grad}(\phi)).$$

Now let  $G = (V, E)$  be the graph of a triangulation of the Riemannian manifold. Consider the scalar function  $f : V \rightarrow \mathbb{R}$  which is a restriction of  $\phi$  to  $V$ . Let  $R$  be an oriented incidence matrix corresponding to  $G$ . Then the operator  $R^T f : \hat{E} \rightarrow \mathbb{R}$  acts on the set  $\hat{E}$ , with  $|\hat{E}| = 2|E|$ , of oriented edges,

$$(R^T f)(e) = f(e^+) - f(e^-),$$

where  $e^+$  and  $e^-$  are the terminal and initial vertices of edge  $e$ , respectively. One can view the above as a natural analogue of the gradient of  $\phi$  at the edge  $e$ . It follows that  $K = R R^T$  provides an analogue of the divergence of the gradient, giving a combinatorial version of the Laplace-Beltrami operator. As the triangulation  $G$  becomes sufficiently dense, the restriction of  $\Delta(\phi)$  to  $V$  is expected to be close to  $K f$ .

#### 6.3.3. Rayleigh quotient of graph Laplacian

One of the most important properties of the graph Laplacian is that its associated Rayleigh quotient has a particularly use-



ful form. Let  $f$  be any scalar function defined over the vertices of a mesh with graph Laplacian  $K$ . Then

$$\frac{f^\top K f}{f^\top f} = \frac{\frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2}{\|f\|_2^2}, \quad (3)$$

where  $w_{ij}$  is the weight at edge  $(i, j)$ . It follows that  $K$  is positive semi-definite. Similar results can be obtained for the normalized graph Laplacian and Tutte Laplacian.

Since the sum of each row of  $K$  is zero and  $K$  is positive semi-definite, the smallest eigenvalue of  $K$  is 0 and one of its associated eigenvectors is the constant vector. It is known [MP93, Moh97] that the multiplicity of the zero eigenvalue of  $K$  equals the number of connected components in the graph. By Theorem 5.3 and equation (3), we can characterize the Fiedler vector  $\mathbf{v}_2(K)$  of a connected graph, the eigenvector associated with the smallest non-zero eigenvalue of  $K$ , as follows,

$$\mathbf{v}_2(K) = \operatorname{argmin}_{\mathbf{u}^\top \mathbf{1} = 0, \|\mathbf{u}\|_2 = 1} \sum_{i,j=1}^n w_{ij} (\mathbf{u}_i - \mathbf{u}_j)^2.$$

This shows the usefulness of the Fiedler vector in dealing with the *minimum linear arrangement* problem, which seeks a permutation  $\pi : V \rightarrow \{1, 2, \dots, n\}$  of the vertices of the graph  $G$  so as to minimize

$$\sum_{i,j=1}^n w_{ij} |\pi(i) - \pi(j)|.$$

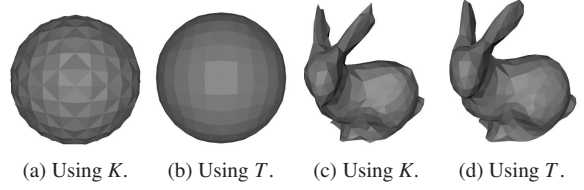
### 6.3.4. Nodal domain theorems

An interesting property of the Laplacians is the relation between their eigenfunctions and the number of nodal domains that they possess. A *nodal set* associated with an eigenfunction is defined as the set composed of points at which the eigenfunction takes on the value zero, and it partitions the surface into a set of *nodal domains*, each taking on positive or negative values in the eigenfunction. The nodal sets and domains are bounded by the following theorem [JNT01]:

**Theorem 6.1 (Courant’s Nodal Domain Theorem)** Let the eigenvectors of the Laplace operator be labeled in increasing order. Then, the  $i$ -th eigenfunction can have at most  $i$  nodal domains, i.e., the zero set of the  $i$ -th eigenfunction can separate the domain into at most  $i$  connected components.

This theorem only gives an upper bound for the number of nodal domains. The direct relation between a specific eigenfunction and its nodal domains is not clear. One possible application of nodal domains is pointed out by Dong et al. for spectral mesh quadrangulation [DBG\*06], explained in Section 9.2.1. By carefully selecting a suitable eigenfunction, they take advantage of the partitioning given by the nodal domains and remesh an input surface.

Discrete analogues of Courant’s Nodal Domain Theorem are known [DGLS01]. In fact, these results are applicable to a larger class of discrete operators, called the discrete Schrödinger operators, which we define in Section 6.5.



**Figure 6:** Comparing results of spectral compression using Tutte Laplacian  $T$  and graph Laplacian  $K$ , showing artifacts resulting from the use of the latter. Note that the two meshes in (a) and (b) both have a 4-8 connectivity.

### 6.3.5. Additional properties and variations

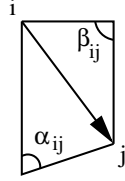
Zhang [Zha04] has examined various matrix-theoretic properties of the three (unweighted) combinatorial Laplacians defined thus far. While the eigenvectors of the three operators all exhibit harmonic behavior, it is shown that depending on the application, there are some subtle differences between them. For example, unlike  $K$  and  $T$ , the eigenvector of  $Q$  corresponding to the smallest (zero) eigenvalue is not a constant vector, where we assume that the mesh in question is connected. It follows that the normalized graph Laplacian  $Q$  cannot be used for low-pass filtering as done in [Tau95].

In the context of spectral mesh compression, Ben-Chen and Gotsman [BCG05] demonstrate that if a specific distribution of geometries is assumed, then the graph Laplacian  $K$  is optimal in terms of capturing the most spectral power for a given number of leading eigenvectors. This result is based on the idea that the spectral decomposition of a mesh signal of a certain class is equivalent to its PCA, when this class is equipped with the specific probability distribution.

However, it has been shown [ZB04, Zha04] that although optimal for a specific singular multivariate Gaussian distribution, the graph Laplacian  $L$  tends to exhibit more sensitivity towards vertex degrees, resulting in artifacts in meshes reconstructed from a truncated spectrum, as shown in Figure 6. In comparison, the Tutte Laplacian appears to possess more desirable properties in this regard and as well as when they are applied to spectral graph drawing [Kor03], but the Tutte Laplacian is not symmetric in general.

One possible way of obtaining a symmetric version from the initial non-symmetric Laplacian  $T$  is by applying the simple transformation  $T' = \frac{1}{2}(T + T^\top)$ , as suggested by Lévy [Lev06]. Zhang [Zha04] proposes a new symmetric operator which approximates the Tutte Laplacian. The new operator does exhibit nice properties. But a common drawback of both suggestions is that it is unclear whether the discrete operators have meaningful continuous counterparts.

Finally, the Tutte Laplacian  $T$  can also be “symmetrized” into a second-order operator  $T'' = T^\top T$ , where the non-zero entries of the matrix extend to the second-order neighbors of a vertex [Zha04]. The eigendecomposition of  $T''$  is re-



**Figure 7:** Angles involved in the calculation of cotangent weights for geometric mesh Laplacians.

lated to the singular value decomposition of  $T$ . When used in relevant applications,  $T''$  exhibits nice properties.

#### 6.4. Geometric mesh Laplacians

In defining a geometric mesh Laplacian, we are interested in discrete approximations to the Laplace-Beltrami operator  $\Delta_{\mathcal{S}}$  on a smooth surface  $\mathcal{S}$ . Given a triangle mesh  $\mathcal{M}$  that approximates  $\mathcal{S}$ , we want an operator  $L_{\mathcal{M}}$  on  $\mathcal{M}$  that will play the role that  $\Delta_{\mathcal{S}}$  plays on  $\mathcal{S}$ . As before, functions on  $\mathcal{M}$  are defined by values at the vertices. Barycentric interpolation is used to obtain function values on the faces of the mesh so that the resulting piecewise linear function on  $\mathcal{M}$  is continuous.

Pinkall and Polthier [PP93] derive a geometric mesh Laplacian  $P$  based on the cotangent weights, where for any scalar function  $f: V \rightarrow \mathbb{R}$ ,

$$(Pf)_i = \sum_{j \in N(i)} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij}) (f_i - f_j). \quad (4)$$

The angles  $\alpha_{ij}$  and  $\beta_{ij}$  are subtended by the edge  $(i, j)$ , as shown in Figure 7. Referring to equation (1), we see that  $P$  can be defined from (1) by setting  $b_i = 1$  for all  $i$  and  $w_{ij} = \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})$  for all  $i, j$ .

The operator  $P$  can be derived by computing the Dirichlet energy for piecewise linear functions and noting that the Laplace equation arises as the Euler-Lagrange equation which must be satisfied by its minimiser [PP93]. It can be shown that the quadratic form  $f^T P f$  evaluates to a total discrete Dirichlet energy over the mesh, implying that  $P$  is positive semi-definite. Also,  $P$  is a symmetric matrix so that the self-disjoint property of the Laplacian operator is satisfied.

The expression (4) for  $P$  can be arrived at in several different ways, e.g., see also [MDSB02]. One weakness of  $P$  as a representative of the Laplacian is that  $(Pf)_i$  evaluates to a nodal value that represents the integral of the Laplacian over a neighbourhood, rather than a point sample. A solution proposed by Meyer et al. [MDSB02] is to divide the expression in (4) by the area of a local neighbourhood  $\Omega_i$ , whose boundary  $\partial\Omega_i$  is formed by straight lines connecting the barycentres of the triangles adjacent to vertex  $i$  with the midpoints of edges incident to the triangles. We refer to cells constructed in this way as *barycells*.

The resulting operator  $Y$  is specified by

$$(Yf)_i = \frac{1}{|\Omega_i|} \sum_{j \in N(i)} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij}) (f_i - f_j), \quad (5)$$

where  $|\Omega_i|$  is the area of the barycell  $\Omega_i$ . Effectively, the  $(Yf)_i$ 's would yield values that are local spatial averages of the Laplacian. If  $D$  is the diagonal matrix whose diagonal entries are  $|\Omega_i|$ , then the Laplacian proposed is  $Y = D^{-1}P$ .

Although  $Y$  is not a symmetric matrix, which means that the self-adjoint character of the Laplacian is sacrificed, the situation can be salvaged by modifying the definition of the scalar product as done in Section 6.2.

Finally, using a finite element formulation, another candidate for the Laplacian, which we call the *FEM Laplacian*, can be derived [DZM05],

$$F = B^{-1}P.$$

The matrix  $B$  is the *mass matrix* encountered in finite element analysis. It is a sparse matrix defined by

$$B_{ij} = \int_{\mathcal{M}} \varphi_i \varphi_j da,$$

where  $\varphi_i$  is the piecewise linear ‘‘hat’’ function defined at vertex  $i$ . Specifically,  $\varphi_i(i) = 1$ ,  $\varphi_i(j) = 0$  if  $j \neq i$ , and function values on the mesh faces are obtained via barycentric interpolation. This integral which defines  $B$  can be computed analytically [DZM05]. If vertices  $i$  and  $j$  are neighbours, then  $B_{ij}$  is  $1/12$  of the total area of the two triangles adjacent to edge  $(i, j)$ . Each diagonal entry  $B_{ii}$  is  $1/6$  of the total area of the triangles adjacent to vertex  $i$ . All other entries are zero.

The matrix  $F$  is self-adjoint with respect to the inner product  $\langle f, g \rangle_B = f^T B g$ , which also renders its eigenvectors orthogonal in this context. Recently, Vallet and Lévy [VL07] provide an alternative derivation of the FEM Laplacian using exterior calculus.

The geometric mesh Laplacians have appeared frequently in the computer graphics literature, e.g., for implicit mesh fairing [DMSB99] and more general signal processing on manifolds [KR05, VL07], shape characterization [RWP06], as well as remeshing [DBG\*06]. Compared to combinatorial mesh Laplacians, which depend on mesh connectivity and not geometry, the geometric mesh Laplacians more closely approximate the Laplace-Beltrami operator for Riemannian manifolds.

#### 6.5. Discrete Schrödinger operator

The discrete Schrödinger operator is defined by supplementing the discrete Laplacian with a *potential* function, which is again a term arising from the study of electrical networks. The potential function is a real function, taking on both negative and positive values, defined on the vertices of a graph. Specifically, for a given graph  $G = (V, E)$ ,  $H$  is a discrete

Schrödinger operator for  $G$  if

$$H_{ij} = \begin{cases} \text{a negative real number} & \text{if } (i, j) \in E, \\ \text{any real number} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Such operators have been considered by Yves Colin de Verdière [dV90] in the study of a particular spectral graph invariant, as well as by Davies et al. [DGLS01] who have proved discrete analogues of Courant's nodal domain theorem [JNT01].

A special sub-class of discrete Schrödinger operators, those having exactly one negative eigenvalue, have drawn particular attention. Lovász and Schrijver [LS99] have proved that if a graph  $G$  is planar and 3-connected, then any matrix  $M$  in that special sub-class for  $G$  with co-rank 3 (dimensionality of the null-space of  $M$ ) admits a valid null-space embedding on the unit sphere. The null-space embedding is obtained by the three eigenvectors corresponding to the zero eigenvalue of  $M$ . This result has subsequently been utilized by Gotsman et al. [Got03] to construct valid spherical mesh parameterizations.

## 6.6. Higher-order operators

In works related to image segmentation and clustering, spectral methods usually employ a different operator, the so-called *affinity matrix* [SM00, Wei99]. Each entry  $W_{ij}$  of the affinity matrix  $W$  represents a numerical relation between two data points  $i$  and  $j$ , e.g., pixels in an image. If this value is large, the two data points are highly correlated, otherwise, we say that they possess a small affinity. This notion can also be applied to meshes, where the entry  $W_{ij}$  represents the affinity between vertices  $i$  and  $j$ .

It should be noticed that the affinity matrix differs from the Laplacian in that affinities between all pairs of elements being considered have to be defined. This implies that this matrix is not sparse, in opposition to a mesh Laplacian, which only relates neighboring elements and is thus sparse. In practice, this non-sparse structure implies more memory requirements and more expensive computations.

### 6.6.1. Gram matrices

A particularly important class of affinity matrices are the so-called *Gram matrices*, which are frequently encountered in machine learning. An  $n \times n$  Gram matrix is typically derived by applying a *kernel* function to pairwise distances between a set of data points. For example, when a Gaussian kernel is used, we obtain a Gram matrix  $W$  with

$$W_{ij} = e^{-\text{dist}(i,j)^2/2\sigma^2}$$

where the Gaussian width  $\sigma$  is a free parameter. Furthermore, a normalized affinity matrix  $N$  can be obtained,

$$N = D^{-1/2}WD^{-1/2},$$

where the degree matrix  $D$  is defined as before.

While some authors advocate the use of the normalized affinity matrix for spectral analysis, e.g., in spectral clustering [NJW02], others employ directly only the matrix  $W$ . Although normalization may have an influence on the obtained results, what visibly controls the behavior of the operator is how the affinity  $W_{ij}$  between two vertices is defined; this turns out to be one of the key points of this approach. This is also an important observation in similar techniques, such as Multi-dimensional Scaling (MDS) [CC94]. MDS employs spectral embedding to obtain low-dimensional, typically 2D, data representations to facilitate visualization. Given as input the pairwise distances between data points in the original domain, the spectral embeddings are such that these distances are closely approximated by Euclidean distances in the embedding domain. The theoretical set up here is given in part by Theorem 5.5 from Section 5.

Different ways to define the affinities exist for mesh processing. One possibility is to define the affinity between two mesh vertices as their distance in the graph implied by the mesh connectivity [LZvK06]. More elaborate approaches consider the distance on the surface of the model, known as the geodesic distance, for the purpose of obtaining bending invariant shape signatures [EK03, JZvK07] or achieving low distortion in texture mapping [ZKK02]. In the latter, MDS is applied to flatten a mesh surface for texture mapping, where Euclidean distances in the plane approximate geodesic distances over the input mesh surface. However, computing geodesic distances on meshes can be an expensive task. Therefore, other approaches have been proposed, such as refining the graph distances by considering the more global *traversal distances*. These consist in defining the affinity between two vertices as the number of paths in the graph between these two elements [SFYD04].

### 6.6.2. Non-sparse Laplacian

The affinity matrix and the Laplacian can also be combined to provide a *non-sparse Laplacian*, e.g., given by  $D - W$ , where  $D$  and  $W$  are as defined in the previous section. Basically, it can be seen as giving the overall structure of a Laplacian to the affinity matrix. Shi and Malik [SM00] employ this operator, for example, to perform image segmentation based on normalized cuts.

As described before, after the operator that is being used by the method is constructed, its eigendecomposition is performed. The next sections present the different ways in which the extracted eigendecomposition can be employed.

## 7. Use of different eigenstructures

The eigendecomposition of a linear mesh operator provides a set of eigenvalues and eigenvectors, which can be directly used by an application to accomplish different tasks. Moreover, the eigenvectors can also be used as a basis onto which

a signal defined on a triangle mesh is projected. The resulting coefficients can be further analyzed or manipulated. In this section, we expand our discussion on these issues.

### 7.1. Use of eigenvalues

Drawing analogies from discrete Fourier analysis, one would treat the eigenvalues of a mesh Laplacian as measuring the frequencies of their corresponding eigenfunctions [Tau95]. However, it is not easily seen what the term frequency means exactly in the context of eigenfunctions that oscillate irregularly over a manifold. Furthermore, since different meshes would generally possess different operators and thus different eigenbases, using the magnitude of the eigenvalues to pair up corresponding eigenvectors between the two meshes for shape analysis, e.g., correspondence, is unreliable [JZ06]. Despite of these issues, much empirical success has been obtained using eigenvalues as global shape descriptors for graph [SMD\*05] and shape matching [JZ07]. These applications are described in more detail in Section 9.1.

Besides directly employing the eigenvalues as graph or shape descriptors, spectral clustering methods use the eigenvalues to scale the corresponding eigenvectors so as to obtain some form of normalization. Caelli and Kosinov [CK04] scale the eigenvectors by the squares of the corresponding eigenvalues, while Jain and Zhang [JZ07] provide justification for using the square root of the eigenvalues as a scaling factor. The latter choice is consistent with the scaling used in spectral clustering [NJW02], normalized cuts [SM00], and multidimensional scaling [CC94].

### 7.2. Use of eigenvectors

Eigenvectors are typically used to obtain an embedding of the input shape in the spectral domain, in which the task at hand is more easily carried out or the embedding itself possesses certain desirable properties. Thus, after obtaining the eigendecomposition of a specific operator, the coordinates of vertex  $i$  in a  $k$ -dimensional embedding are given by the  $i$ -th row of matrix  $V_k = [\mathbf{v}_1, \dots, \mathbf{v}_k]$ , where  $\mathbf{v}_1, \dots, \mathbf{v}_k$  are the first  $k$  eigenvectors from the spectrum (possibly after scaling). Whether the eigenvectors should be in ascending or descending order of eigenvalues depends on the operator that is being used. In the case of Gram matrices, eigenvectors corresponding to the largest eigenvalues are used to compute spectral embeddings. While for the various Laplacian operators, the opposite end of the spectrum is considered.

For example, spectral clustering makes use of such embeddings. Ng et al. [NJW02] present a method where the entries of the first  $k$  eigenvectors corresponding to the largest eigenvalues of a normalized affinity matrix (see Section 6.6.1) are used to obtain the transformed coordinates of the input points. Additionally, the embedded points are projected onto the unit  $k$ -sphere. Points that possess high affinities tend to be grouped together in the spectral domain,

where a simple clustering algorithm, such as  $k$ -means, can reveal the final clusters. Furthermore, the ability of the spectral methods to unfold nonlinearity in the input data has been demonstrated via numerous examples, including data sets similar to the one shown in Figure 4.

### 7.3. Use of eigenprojections

The full set of eigenvectors can be used as a basis to project a signal defined on a mesh into a different domain. If a set of orthogonal eigenvectors are obtained by the eigendecomposition of an operator and given by the columns of a matrix  $V$ , and a function defined on each vertex of the mesh is given by a vector  $\mathbf{x}$ , we can project the signal into the spectral domain simply by

$$\tilde{\mathbf{x}} = V^T \mathbf{x},$$

where  $\tilde{\mathbf{x}}$  contains the obtained spectral coefficients. For an orthogonal set of eigenvectors, this can be seen as a change of basis for the signal  $\mathbf{x}$ . As well, the projection is energy-preserving in the sense that the Euclidean 2-norm is preserved:  $\|\tilde{\mathbf{x}}\|_2 = \|\mathbf{x}\|_2$ . The inverse transform is obtained by

$$\mathbf{x} = V \tilde{\mathbf{x}}.$$

As in the case of Fourier analysis, the intuition is that when the signal is transformed into the spectral domain, it might be easier to carry out certain tasks because of the relation of the coefficients to low and high frequency information. For example, when the signal considered is the actual geometry of the mesh, the projections of the signal with respect to the eigenvectors of the graph Laplacian can be used for mesh compression [KG00]. That is, a set of the transformed coefficients from the high-frequency end of the spectrum can be removed without affecting too much the approximation quality of the mesh, when it is reconstructed by the inverse transform. For spectral watermarking of meshes [OTMM01] however, it is the low-frequency end of the spectrum that is to be modulated. This way, the watermark is less perceptible and the watermarked mesh can become resilient against such attacks as smoothing. We elaborate more on these in Section 9.3.

Note that the observation that the human eye is less sensitive to low-frequency errors in geometric shape was first made by Sorkine et al. [SCOT03] in their work on high-pass quantization for mesh compression. This work, along with related developments, has been described in details in a previous state-of-the-art report [Sor05].

## 8. Efficient computations

Since spectral methods all require the construction of a mesh operator, which may be non-sparse [LZ04], as well as its eigendecomposition, efficiency is a concern for large meshes. To this end, we survey several speed-up techniques that accelerate the eigendecomposition and reduce the  $O(n^3)$

complexity of the problem, where  $n$  denotes the mesh size (equivalently, dimensionality of the operator).

Methods that either compute a good approximation of the eigenvectors or that explore the structure of a specific type of matrices were proposed. When using a mesh Laplacian operator, it is natural to explore its sparse structure. One method proposed to accomplish this task is described in Section 8.1, which makes use of a multiscale approach. An alternative to this method, which allows to compute a specific set of eigenvectors, is to introduce a spectral shift into the operator. This can be combined with iterative methods to speed up the computation, as described in more detail in Section 8.2. On the other hand, to compute the eigendecomposition of dense affinity matrices, the Nyström method [FBCM04] can be employed. It is based on computing an approximation of the eigenvectors given by a number of sampled elements, as described in Section 8.3. Thus, it also avoids the construction of the full matrix describing the operator.

### 8.1. Exploring sparsity

Koren et al. [KCH02] propose ACE, from *Algebraic multi-grid Computation of Eigenvectors*, a multi-scale method to accelerate the computation of eigenvectors of Laplacian operators. Multi-scale methods consist in two steps: *coarsening* and *refinement*. Basically, an initial high-dimensional problem is progressively reduced to lower and lower dimensions by applying a coarsening step, which creates less complex instances of the problem. The exact solution of one of these low-dimensional versions of the problem is then computed. Furthermore, the refinement step progressively translates the solution of the problem in lower dimensions to higher ones, usually performing some adjustments to the solution, until a solution to the problem in the original dimension is obtained.

However, the design of the coarsening and refinement steps is usually application dependent, since both steps rely on exploring a special feature of the problem being solved and need to preserve the essence of the initial problem. In the case of Laplacian operators, the sparsity of the related matrices is what allows to speed up the computation of eigenvectors by means of a multi-scale method.

To carry out the coarsening and refinement steps, the key concept introduced by Koren et al. [KCH02] is that of an interpolation matrix  $A$ , which is an  $n \times m$  matrix that interpolates  $m$ -dimensional vectors  $y$  into  $n$ -dimensional ones  $x$ , given by  $x = Ay$ . The interpolation matrix is employed to obtain a coarsened Laplacian matrix given by  $L^c = A^T L A$ , where  $L$  is the original Laplacian. The same interpolation matrix is then used for the refinement step, computing the eigenvectors of the problem at higher and higher resolutions.

The interpolation matrix is created either by contracting edges on the underlying graph, or by performing a weighted

interpolation of a node from several nodes in the coarser version of the problem. The contractions or weighted interpolations are what define the entries of the interpolation matrix, which can be very sparse, when computed by the contractions, or less sparse but conveying a more accurate interpolation, when weighted interpolations are used. Determining which method should be preferentially used depends mainly on whether the underlying graphs are homogeneous or not.

### 8.2. Spectral shift and iterative methods

Iterative algorithms compute the eigenvectors of large sparse matrices in a more efficient manner [TB97]. However, these methods only allow to obtain the leading eigenvectors of a matrix. In order to compute eigenvectors associated to a specific set of eigenvalues, it is necessary to modify the eigenproblem being solved. Dong et al. [DBG\*06] and Vallet and Lévy [VL07] accomplish that by utilizing a *spectral shift*.

The original problem  $L\mathbf{v} = \lambda\mathbf{v}$  is modified to the form

$$(L - \sigma I)\mathbf{v} = (\lambda - \sigma)\mathbf{v}$$

so that when this eigenproblem is solved, the eigenvectors that are obtained correspond to eigenvalues close to  $\sigma$ . This is valid due to the fact that, if  $\mathbf{v}$  is an eigenvector of  $L$  with associated eigenvalue  $\lambda$ , then it is also an eigenvector of  $L - \sigma I$  with associated eigenvalue  $\lambda - \sigma$ .

Moreover, to compute the eigenvectors associated with the smallest eigenvalues instead of the leading ones, Vallet and Lévy [VL07] also resort to the idea of swapping the spectrum of a matrix by inverting it. This comes from the observation that the leading eigenvalues of the eigenproblem  $L^{-1}\mathbf{v} = (1/\lambda)\mathbf{v}$  are the smallest eigenvalues of  $L\mathbf{v} = \lambda\mathbf{v}$ , with the same set of eigenvectors  $\mathbf{v}$ . This is combined with the spectral shift to obtain the *shift-invert* spectral transform, which can be used to split the computation of the eigenvectors of large matrices into multiple bands. Each band can be computed in linear time, and this process can be further accelerated by the use of out-of-core factorization methods.

### 8.3. Nyström approximation

In order to compute the eigendecomposition of large affinity matrices, one technique that can be used to approximate the leading eigenvectors of sampled matrices is the Nyström method [FBCM04]. Given a set of mesh vertices  $\mathcal{Z}$  of size  $n$ , whose affinities are given in the matrix  $W \in \mathbb{R}^{n \times n}$ , the first step in Nyström's method is to divide the set  $\mathcal{Z}$  into a subset of samples  $\mathcal{X}$  of size  $l$ , with  $l \ll n$ , and a subset  $\mathcal{Y}$  of size  $m$ , which contains the remaining points.

Next, the affinities between the points in the subsets  $\mathcal{X}$  and  $\mathcal{Y}$  are stored in the matrices  $A \in \mathbb{R}^{l \times l}$  and  $C \in \mathbb{R}^{m \times m}$ , respectively. The *cross-affinities* between points of  $\mathcal{X}$  and  $\mathcal{Y}$  are stored in matrix  $B \in \mathbb{R}^{l \times m}$ . Thus, the matrix  $W$  can be

written in the following block form

$$W = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}.$$

After obtaining the eigendecomposition of the matrix  $A$ , given by  $A = U\Lambda U^T$ , the columns of  $\bar{U}$ , expressed below, are the approximation for the  $l$  leading eigenvectors of  $W$ , that is, the  $l$  eigenvectors related to the largest eigenvalues.  $\bar{U}$  is given by Nyström's method as

$$\bar{U} = \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix}.$$

Therefore, only the affinities  $A$  between the sampled points and the cross-affinities  $B$  need to be computed in order to obtain the approximation. Moreover, the original matrix  $W$  can be reconstructed by using the approximated eigenvectors. The approximation of  $W$  is given by

$$\bar{W} = \bar{U}\Lambda\bar{U}^T = \begin{bmatrix} A & B \\ B^T & B^T A^{-1} B \end{bmatrix}.$$

The quality of this approximation is given by the quantity  $\|W - \bar{W}\|$ , which is equivalent to computing only the norm of the *Schur complement*  $\|C - B^T A^{-1} B\|$ . However, it is expensive to directly compute this quantity since it requires the matrix  $C$ , which ideally contains the largest number of affinity values and would imply  $O(n^2)$  running time. Therefore, methods that compute an indirect measure of the quality of the approximation should usually be employed.

The overall complexity of this method is  $O(l^3)$  for computing the eigendecomposition of matrix  $A$ , and  $O(ml^2)$  for obtaining the approximated eigenvectors via extrapolation. Therefore, the problem of computing the leading eigenvectors is reduced from  $O(n^3)$  to only  $O(ml^2 + l^3)$ , recalling that  $l \ll n$ . In practical applications such as spectral image segmentation [FBCM04], spectral mesh segmentation [LJZ06, ZL05], and spectral shape correspondence [JZvK07] and retrieval [JZ07],  $l$  can be as small as less than 1% of  $n$  while still ensuring satisfactory results.

Nevertheless, there are a few issues which emerge when using the Nyström's method. First of all, the approximated eigenvectors are not orthogonal. Fowlkes et al. [FBCM04] present two techniques to orthogonalize the obtained eigenvectors, depending on whether the affinity matrix  $A$  is positive definite or indefinite. However, these techniques may introduce additional numerical errors on the approximated eigenvectors. Moreover, the key factor for the accuracy of the eigenvectors obtained by Nyström's method is the sampling technique that determines the subset  $\mathcal{X}$ . Different techniques were proposed for accomplishing this step of the method, such as random sampling [FBCM04], max-min farthest point sampling [dST04], and greedy sampling based on maximizing the trace of the matrix  $B^T A^{-1} B$  [LJZ06]. But these schemes all judge the quality of a sampling by the approximation quality of the eigenvectors obtained, measured

by standard matrix norms, and they do not take into consideration the application at hand.

## 9. Applications

In this section, we survey specific spectral methods. Although our focus will be on spectral methods for mesh processing and analysis, highly relevant and representative problems and techniques from other fields will also be covered for completeness.

Let us first list in Table 1 the relevant references grouped by applications. Most of these references will be discussed in detail in subsequent sections. Others have been discussed in other parts of the report, when appropriate.

Application	References
Clustering	[NJW02], [VM03], [BN03], [KVV00], [vLBB05], [vL06], [BH03], [MP04]
Graph drawing	[Hal70], [PST00], [KCH02], [DPS02], [Kor03]
Graph indexing	[SMD*05]
Graph matching	[Ume88], [SB92], [CK04]
Graph partitioning	[ST96], [SM97], [PF98], [Wei99], [AKY99], [BPS93]
Matrix reordering	[BPS93]
Mesh compression	[KG00], [ZB04], [Zha04], [BCG05]
Mesh parameterization	[GGS03], [ZSGS04]
Mesh reordering	[IL05], [LZvK06]
Mesh segmentation	[LZ04], [KLT05], [ZL05]
Mesh smoothing	[VL07]
Remeshing	[DBG*06]
Shape correspondence	[JZ06], [JZvK07]
Shape indexing	[EK03], [RWP06], [JZ07]
Surface reconstruction	[KSO04]
Texture mapping	[ZKK02]
Watermarking	[OTMM01], [OMT02]

Table 1: Applications addressed by spectral methods.

### 9.1. Use of eigenvalues

Although most applications in the field of geometry processing employ eigenvectors to accomplish different tasks, eigenvalues have been successfully used to address certain problems, such as graph and shape indexing.

- **Graph indexing:**

The use of graph spectra for indexing is well known in computer vision and machine learning, e.g., see a recent comparative study [ZW05]. Recently, Shokoufandeh et al. [SMD\*05] make use of eigenvalues for indexing graphs that represent hierarchical structures, such as

*shock graphs* that define image silhouettes. The eigenvalues provide a measure indicating which graphs are similar and should be compared with a more expensive matching algorithm. Basically, the index is defined as the sum of eigenvalues of the adjacency matrix of the graph, which allows to obtain a low-dimensional index. However, in order to also reflect local properties of the graph, one term corresponding to the sum of eigenvalues is stored for each subtree of the graph.

- **Shape indexing:**

For 3D shape indexing, Jain and Zhang [JZ07] propose to use the leading eigenvalues of an affinity matrix constructed using approximated geodesic distances over the shape surfaces. The similarity between two models is given by the  $\chi^2$ -distance between the selected  $k$  eigenvalues of the two models, where  $k$  is usually very small, for example, 6. This comparison between the first eigenvalues intuitively corresponds to comparing the variation of the models in each of the first  $k$  nonlinear principal components. As argued before, the bending invariance of geodesic distances should facilitate the retrieval of articulated shapes. This has indeed been confirmed by their experiments on the McGill articulated shape database [McG]. The simple eigenvalue-based descriptor did outperform two of the best shape descriptors, the spherical harmonics descriptor [MKR03] and the light field descriptor [CTSO03], even when they are applied to the bending-normalized spectral embeddings.

## 9.2. Use of eigenvectors

In this section, we survey eigenvector-based methods and classify them according to the dimensionality of the spectral embeddings used.

### 9.2.1. 1D embedding

Embedding in one dimension consists in producing a linear sequence of mesh elements based on the order given by the entries of one eigenvector. The Fiedler vector has been extensively used in a number of different applications to obtain a linear ordering of mesh vertices or faces. However, other applications also select different eigenvectors to obtain a linear sequencing of mesh elements.

- **Sparse matrix reordering:**

Barnard et al. [BPS93] use the Fiedler vector to reduce the *envelope* of sparse matrices. By reordering a matrix and optimizing its envelope, the locality of its elements is increased and the resulting matrix will become more “banded”. Several numerical algorithms would improve their performance when applied to the reordered matrix. The Fiedler vector is selected due to its property of minimizing the 2-sum in a continuous relaxation of the problem [MP93].

- **Mesh sequencing:**

Isenburg and Lindstrom [IL05] introduce the concept of *streaming meshes*. The idea is to process very large meshes that do not fit in main memory by streaming its components, i.e., transferring blocks of vertices and faces in an incremental manner from the hard disk to main memory and back. In this sense, it is highly desirable that the order in which the vertices and faces are traversed preserves neighboring relations the most, so that only a small number of mesh elements have to be maintained simultaneously in main memory. Therefore, one of the possibilities to obtain such a streaming sequence is also to employ the Fiedler vector to order the vertices and faces of the mesh, which provides a linear sequence that heuristically minimizes *vertex separation*.

In this context of obtaining a good ordering of mesh elements, Liu et al. [LZvK06] investigate how the embeddings given by an eigenvector of an affinity matrix differ from the ones given by the Laplacian matrix or other possible heuristics. These embeddings are evaluated in terms of various measures related to the interdependence of mesh elements, the best known ones being *span* and *width* of vertices and faces. Experiments show that the embeddings given by the affinity matrix provide a better trade-off between these two measures, when compared to other approaches.

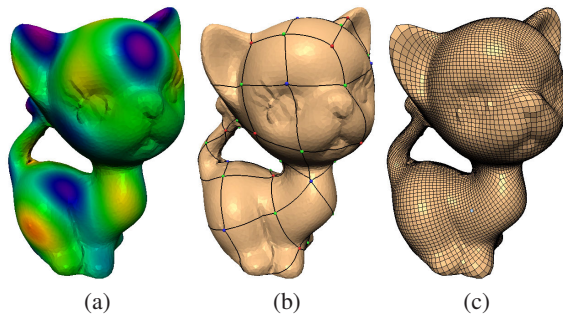
- **Image segmentation:**

In the case of image segmentation, instead of the Fiedler vector, the eigenvector related to the largest eigenvalue of an affinity matrix has also been used in a great number of works, since it essentially provides a heuristic method for obtaining satisfactory graph partitions [ST96]. Weiss [Wei99] presents a unified view of four well-known methods that follow this approach, such as the work of Shi and Malik [SM00] and Perona and Freeman [PF98].

These methods firstly define an affinity matrix  $W$  based on the distances between pixels in the image, which are seen as nodes in a graph. Next, the eigenvector related to the eigenvalue that is largest in magnitude or, equivalently, the second smallest eigenvector, in the case of the method of Shi and Malik [SM00], is computed. In the latter case, the matrix considered is the non-sparse Laplacian  $D - W$ . Finally, the entries of this specific eigenvector are used to convey a partition of the pixels into two groups. Either the signs of the entries of the eigenvector or a thresholding of these entries is used to obtain a binary partition of the pixels. The essence of this approach is that this specific eigenvector is a continuous solution to the discrete problem of minimizing the normalized cut between two sets of nodes in a graph.

- **Spectral clustering for surface reconstruction:**

Kolluri et al. [KSO04] follow basically the same approach for the reconstruction of surfaces from point clouds. After constructing a Delaunay tetrahedralization based on



**Figure 8:** Generating a quadrangulation of a model (Images generated by Dong et al. [DBG\*06]): (a) An adequate eigenfunction is selected. (b) The Morse-Smale complex is extracted. (c) The quadrangulation is constructed.

the input points, the tetrahedra are divided into two sets by a spectral graph partitioning method, which provides the indication of which tetrahedra are inside of the original object and which are outside. Finally, this labeling of tetrahedra defines a watertight surface. The partitioning of the tetrahedra is also given by the signs of the entries of the smallest eigenvector of a *pole matrix*, which is similar to a Laplacian.

- **Mesh segmentation:**

In a different setting, Zhang and Liu [ZL05] propose a mesh segmentation approach based on a recursive 2-way spectral clustering method. An affinity matrix encodes distances between mesh faces, which are a combination of geodesic and angular distances, that provide information for a visually meaningful segmentation. Next, only the first two largest eigenvectors are computed. This provides a one-dimensional embedding of faces given by the quotient of the entries of the two eigenvectors. Finally, the most salient cut in this embedding is located, given by a part salience measure. The cut provides a segmentation of the faces into two parts. This process is recursively repeated in order to obtain a hierarchical binary partitioning of the mesh.

- **Spectral mesh quadrangulation:**

Dong et al. [DBG\*06] propose to use one specific eigenvector of the geometric Laplacian to guide a remeshing process. Firstly, a suitable eigenfunction of the mesh has to be selected, which possesses a desired number of critical points. The critical points are points of minima, maxima, or saddle points. Next, the *Morse-Smale complex* is extracted from the mesh based on this eigenvector. The complex is obtained by connecting critical points with lines of steepest ascend/descend, and partitions the mesh into rectangular patches, which are then refined and parameterized, conveying a quadrangular remeshing of the original model. The whole process can be seen in Figure 8.

One of the key points of this method is in selecting an eigenfunction that provides an adequate partition of the mesh. This is achieved by computing all the eigenvectors of a simplified version of the mesh, choosing from these eigenvectors the most suitable for the remeshing task, and then computing the corresponding eigenvector in the full-resolution mesh by applying a *spectral shift* to the underlying matrix.

### 9.2.2. 2D and 3D embeddings

Instead of using only one eigenvector given by the eigendecomposition of a specific operator, the next possible step is to use two or three eigenvectors, to obtain a planar or three-dimensional embedding of the mesh.

- **Graph drawing:**

Methods that provide such embeddings have been successfully applied in the field of graph drawing, where the main goal is to obtain a disposition of nodes and edges on a plane or volume which looks organized and is aesthetically pleasing. Early graph drawing algorithms already proposed to use the Laplacian matrix, as in the work of Tutte [Tut63], whose method actually dates back to the work of Fáry [Far48]. However, in Tutte's method, the eigenvectors of the Laplacian matrix are not computed. Instead, the positions of the nodes of a graph are obtained by solving a linear system based on this matrix.

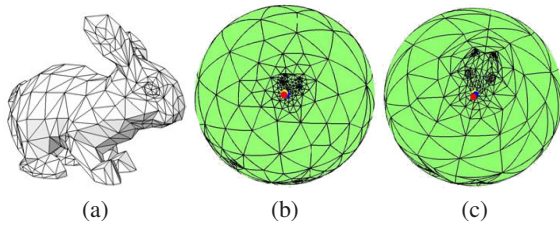
However, Hall [Hal70] later proposed to use the eigenvectors of the Laplacian matrix to embed the nodes of a graph in a space of arbitrary dimension. The entries of the  $k$  eigenvectors related to the first smallest non-zero eigenvalues are used as the coordinates of a node (a  $k$ -dimensional embedding). This method has been recently applied in different domains to provide planar embeddings of graphs. For example, Pisanski and Shawe-Taylor [PST00] use this method to obtain pleasing drawings of symmetrical graphs, such as fullerene molecules in chemistry. Koren et al. [KCH02, Kor03] employ the ACE algorithm (Section 8.1) to accelerate Hall's method in order to obtain drawings of large graphs.

- **Planar mesh parameterization via MDS:**

Zigelman et al. [ZKK02] use a variant of MDS to obtain a planar embedding of a mesh and then map a given texture on this planar surface. In this work, a geodesic distance matrix is computed by fast marching. Next, the matrix is centered and its eigendecomposition is computed. The two eigenvectors which are related to the two largest eigenvalues are used to provide a planar embedding of the mesh. The flattening that is obtained heuristically minimizes distortions, which is one of the desired properties in the case of texture mapping. However, no assurance is provided to prevent triangle fold-overs.

Following a similar approach, Zhou et al. [ZSGS04] employ MDS based on geodesic distances to obtain a pa-





**Figure 9:** Spherical parameterizations (Images generated by Gotsman et al. [GG03]) of the model shown in (a): using the Tutte Laplacian in (b), and conformal weights in (c).

parameterization and then a chartification of a given mesh. By growing patches around representative points, which are determined according to the spectral embedding, the mesh is divided into charts. The representatives that are selected are points that are well-spaced over the mesh and that are also points of minima or maxima, according to their coordinates in the spectral embedding.

- **MDS for mesh segmentation:**

MDS is also used by Katz et al. [KLT05] to perform segmentation of meshes, due to its potential of obtaining a *pose-invariant* embedding of a mesh. Three eigenvectors are selected to obtain a 3D embedding. Next, feature points are located in this normalized space, which guide the segmentation algorithm that partitions the mesh into meaningful parts.

- **Spherical mesh parameterization:**

Gotsman et al. [GG03] describe how a spherical parameterization of a mesh can be obtained from the eigenvectors of *Colin de Verdiere* matrices. Each graph has a class of these matrices associated with it. By using the entries of three eigenvectors of such a matrix as the coordinates of the mesh vertices, a valid spherical embedding is obtained. By solving a non-linear system based on a Laplacian or a similar operator, a Colin de Verdiere matrix is generated and the three eigenvectors that give the valid embedding (which are associated to repeated eigenvalues) are simultaneously computed. One spherical parameterization obtained by this method can be seen in Figure 9.

- **Mesh segmentation:**

Recently, Liu and Zhang [LZ07] propose a mesh segmentation algorithm via recursive bisection where at each step, a sub-mesh embedded in 3D is spectrally projected to 2D and then a contour is extracted from the planar embedding. Two operators are used in sequence to compute the projection: the well-known graph Laplacian and a geometric operator designed to emphasize concavity. The two embeddings reveal distinctive shape semantics of the 3D model and complement each other in capturing the structural or geometrical aspects of a segmentation. Transforming the shape analysis problem to the 2D domain also fa-

cilitates segmentability analysis and sampling, where the latter is needed to identify two samples residing on different parts of the sub-mesh. These two samples are used by the Nyström method in the construction of a 1D face sequence for finding an optimal cut, as in [ZL05].

### 9.2.3. Higher-dimensional embedding

Since a set of  $n$  eigenvectors can be obtained from the eigendecomposition of an  $n \times n$  matrix, it is natural to intend to use more eigenvectors simultaneously to extract more information from the eigendecomposition.

- **Clustering and mesh segmentation:**

One of the most well-known techniques in this regard is spectral clustering [BN03, KVV00, NJW02]. Interested readers should refer to the recent survey by Luxburg [vL06] and the comparative study by Verma and Meilă [VM03]. The approach by Ng et al. [NJW02] has already been outlined in Section 7.2. Other approaches differ only slightly from the core solution paradigm, e.g., in terms of the operator used and the dimensionality of the embedding. Some works, e.g., [AKY99, NJW02, VM03], seem to suggest that clustering based on multiple eigenvectors tends to produce better results compared with recursive approaches using single eigenvectors.

Although the reasons for the empirical success of spectral clustering are still not fully understood, Ng et al. [NJW02] provide an analysis in terms of matrix perturbation theory to show that the algorithm is expected to work well even in situations where the cluster structure in the input data is far from an ideal case. There are other possible interpretations of spectral clustering, e.g., in terms of graph cuts or random walks [vL06].

The ubiquity of the clustering problem makes spectral clustering an extremely useful technique. Besides the work of Kolluri et al. [KSO04] mentioned in Section 9.2.1, Liu and Zhang [LZ04] perform mesh segmentation via spectral clustering. Basically, an affinity matrix is constructed as in [ZL05]. Next, the eigenvectors given by the eigendecomposition of this matrix guide a clustering method, which provides patches of faces that define the different segments of the mesh returned by the segmentation algorithm. It is shown by example that it can be advantageous to perform segmentation in the spectral domain, e.g., in terms of higher-quality cut boundaries as evidenced by the Polarization Theorem (Theorem 5.6 in Section 5). The downside however is the computational cost. In a follow-up work [LJZ06], Nyström approximation is applied to speed-up spectral mesh segmentation.

- **Shape correspondence and retrieval:**

Elad and Kimmel [EK03] use MDS to compute bending-invariant signatures of meshes. Each entry of the distance matrix is defined as the geodesic distance between two nodes on the surface, computed by fast marching. The idea behind this approach is that the eigenvectors of this

affinity matrix allow to embed the mesh in a space where geodesic distances are reproduced by Euclidean distances. As well, bending transformations are removed in the spectral domain. The similarity between two shapes is then given by the Euclidean distance between the moments of the first few eigenvectors, usually less than 15, and these similarity distances can be used for shape classification.

Motivated by works on spectral point correspondence [SB92] from computer vision, Jain and Zhang [JZvK07] rely on higher-dimensional embeddings based on the eigenvectors of an affinity matrix to obtain point correspondence between two mesh shapes. The first  $k$  eigenvectors of the affinity matrix encoding the geodesic distances between pairs of vertices are used to embed the model in a  $k$ -dimensional space. After this process is performed for two models, the two embeddings are non-rigidly aligned via thin-plate splines and the correspondence between the two shapes is given by the proximity of the vertices after such alignment. Any measure for the cost of a correspondence, e.g., sum of distances between corresponding vertices, can be used as a similarity distance for shape retrieval.

One of the key observations made in [JZvK07] is the presence of “eigenvector switching”, meaning that the eigenmodes of similar shapes do not line up with respect to the magnitude of their corresponding eigenvalues. This is illustrated in Figure 10. As a result, it is unreliable to sort the eigenvectors according to the magnitude of their respective eigenvalues, as has been done in all works on spectral correspondence so far.

- **Graph matching:**

Generally speaking, graphs are commonly used to model shape structures, e.g., skeletal graphs [SSGD03], shock graphs, or Reeb graphs [HSKK01]. The subsequent graph matching problem is well studied in the computer vision community, where a number of spectral approaches have been proposed, e.g., [Ume88, CK04], adding a geometric flavor to the problem. As a basic framework, a graph adjacency matrix, which may only encode topological information, is eigendecomposed, whereby the graph nodes are mapped into a low-dimensional vector space. The matching problem is solved in the embedding space.

### 9.3. Use of eigenprojections

Instead of directly using the entries of the eigenvectors to provide an embedding for a given model, the eigenvectors can also be used as a basis to transform signals defined on the vertices of the mesh. One example of such a signal is the actual geometry of the mesh (the 3D coordinates of its vertices). The set of eigenvectors given by the eigendecomposition can be used to project these signals into the spectral space, where a specific problem might be easier to solve.

- **Geometry compression:**

Karni and Gotsman [KG00] propose this approach to compress the geometry of triangle meshes. Firstly, the set of eigenvectors of the Tutte Laplacian is computed. Next, the geometry of the mesh is projected into the spectral space according to the orthogonal basis given by the computed eigenvectors. Moreover, part of the coefficients obtained by this transformation is eliminated in order to reduce the storage space required for the geometry of the mesh. The coefficients related to the eigenvectors associated to larger eigenvalues are firstly removed, which would correspond to high frequency detail, when following an analogy with Fourier analysis. The exact number of coefficients that is eliminated is given by the impact of the compression on the quality of the mesh, after the inverse transformation is performed.

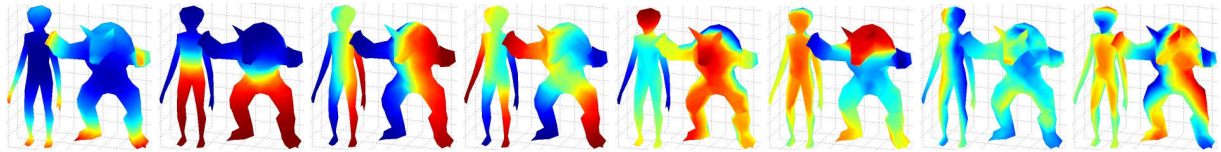
The main drawback of this method is that all eigenvectors need to be computed, so that the projection can be performed. Since in practice it is only feasible to compute the full set of eigenvectors for small matrices, Karni and Gotsman propose to partition the mesh into smaller sets of vertices. Although that alleviates the problem of computing the eigenvectors for large matrices, it still requires a good partitioning of the mesh for the efficiency of the compression, and artifacts along the partition boundaries are evident when higher compression rates are employed.

- **Watermarking:**

Ohbuchi et al. [OTMM01, OMT02] also employ the eigenprojection approach, to insert watermarks into triangle meshes. However, in this method, the eigenvectors of the *Kirchhoff* operator are used as the basis for the projection. After transforming the geometry of the mesh and obtaining the spectral coefficients, a watermark is inserted into the model by modifying coefficients at the low-frequency end of the spectrum. In this way, modifications on the geometry of the mesh are well-spread over the model and less noticeable than if they were directly added to the vertex coordinates. This method also requires the computation of eigenvectors of the Laplacian operator, which is prohibitive in the case of large meshes. Mesh partitioning is again used to address this problem.

- **Fourier descriptors:**

2D Fourier descriptors have been quite successful as a means to characterize 2D shapes. Using eigendecomposition with respect to the the mesh Laplacians, one can compute analogous Fourier-like descriptors to describe mesh geometry. However, we have not seen such mesh Fourier descriptors being proposed for shape analysis so far. There have been methods, e.g., [VS01], which rely on 3D Fourier descriptors for 3D shape retrieval. In this case, the mesh shapes are first voxelized and 3D Fourier descriptors are extracted from the resulting volumetric data. We suspect that the main difficulties with the use of mesh Fourier descriptors for shape matching include computational costs and the fact that when the eigenmodes vary



**Figure 10:** Eigenvector plots for two similar shapes, both with 252 vertices. The entries in an eigenvector are color-mapped. As we can see, there is an “eigenvector switching” occurring between the fifth and sixth eigenvectors. Such “switching” is difficult to detect from the magnitude of the eigenvalues. The first 8 eigenvalues for the two shapes are [205.6, 11.4, 4.7, 3.8, 1.8, 0.4, 0.26, 0.1] and [201.9, 10.9, 6.3, 3.4, 1.8, 1.2, 0.31, 0.25], respectively.

between the two mesh shapes to be matched, it becomes doubtful whether their associated eigenspace projections can serve as reliable shape descriptors.

## 10. Conclusion and discussions

In this report, we describe, motivate, and classify spectral methods for mesh processing and analysis. Related and representative developments from other fields, e.g., computer vision and machine learning, are also covered. Necessary theoretical background and illustrative examples are both provided to facilitate understanding of the various concepts. Finally, we give a detailed survey of specific spectral methods developed to solve a diversity of problems.

From a theoretical standpoint, we are still missing an adequate sampling theory for signals defined over 2-manifolds. We envision this theory to be one whose results and analysis tools resemble those from the theory of sampling and reconstruction in the regular setting using Fourier transforms. Fundamental questions concerning the proper definition of concepts such as frequencies, band-limited signals, shift-invariance, etc., should be addressed.

Take the concept of frequency for example. Our general belief is that eigenvalues of the mesh Laplacian represent frequencies. However, eigenvalues are only able to indicate global properties of the manifold or global properties of the associate eigenfunctions. The variability of eigenfunctions having the same or similar eigenvalues implies that eigenvalues alone cannot provide sufficient characterization of their related eigenfunctions. This has been the case when we seek a proper ordering of the eigenvectors in order to facilitate robust spectral shape correspondence [JZvK07]. The situation described here differs from the classical case, where the eigenfunctions all follow regular vibration patterns and frequencies alone would provide adequate characterization.

Ideally, we would like to find additional characteristic measures for the eigenfunctions. This, for example, might help us more easily locate the right eigenvector for deriving a high-quality surface quadrangulation automatically [DBG\*06]. As Lévy [Lev06] has eloquently put, Laplace-Beltrami eigenfunctions (or eigenfunctions of other geometric mesh operators) appear to “understand” geometry. However, it is not necessarily easy to interpret what the

eigenfunctions are presenting to us. A better understanding of the eigenvectors and how they relate to the shape of the underlying manifolds would certainly spark new research and allow for improvements in the spectral methods.

Other theoretical studies concerning mesh operators and their eigenstructures include convergence analyses for geometric mesh Laplacians (as in [Xu04]), analyses on the sensitivity of the eigenstructures against shape or connectivity variations (as in [DZM05]), as well as studies on sampling for Nyström approximation. In this setting, as for the development of a sampling theory, we are concerned with the interplay between the continuous and the discrete settings.

Moreover, a generalization of the Laplacian operators to Schrödinger operators allows to consider a new class of possible operators. However, it is not clear how to easily construct specific instances of these operators in an efficient manner, e.g., the Colin de Verdière matrices for spherical mesh parameterization [GGS03], or to explicitly design such an operator having the property that it is optimal for a specific application, e.g., compression or segmentation.

Another wide avenue for further research is the study of the theoretical aspects of spectral clustering algorithms. First of all, the reason for the good results obtained by these algorithms is still not completely understood. Fortunately there exist a number of studies and analyses which elucidate certain properties responsible for the exceptional behavior of these algorithms, e.g., [vL06]. These studies might serve as a starting point to explain the functioning of spectral clustering and lead to ideas for more complete explanations. Additionally, other aspects, such as how to select the dimensionality of the spectral embeddings or how to construct affinity matrices more suitable for specific applications, e.g., for proper handling of part stretching in shape characterization, still require further attention.

**Acknowledgments:** This research is supported in part by an NSERC Discovery Grant (No. 611370) and an MITACS research grant (No. 699127). Mesh models used in Figure 1 and Figure 3 are from the the GavabDB face database and the McGill 3D Shape Benchmark Database, respectively. The bunny model used was produced by the QSlim mesh simplification program written by Michael Garland and the origi-

nal model comes from the Stanford 3D Scanning Repository, with credit to the Stanford Computer Graphics Laboratory. The Igea model is from Cyberware Inc. Finally, the authors would like to thank the reviewers for their comments.

## References

- [AKY99] ALPERT C. J., KAHNG A. B., YAO S. Z.: Spectral partitioning with multiple eigenvectors. *Discrete Applied Mathematics* 90 (1999), 3–26.
- [BCG05] BEN-CHEN M., GOTSMAN C.: On the optimality of spectral compression of mesh data. *ACM Trans. Graph.* 24, 1 (2005), 60–80.
- [BH03] BRAND M., HUANG K.: A unifying theorem for spectral embedding and clustering. In *Proc. of the 9th Int. Conf. on Artificial Intelligence and Statistics* (Key West, Florida, 2003).
- [Bha97] BHATIA R.: *Matrix Analysis*. Springer-Verlag, 1997.
- [BHL\*04] BIYIKOĞLU T., HORDIJK W., LEYDOLD J., PISANSKI T., STADLER P. F.: Graph Laplacians, nodal domains, and hyperplane arrangements. *Linear Algebra and Its Applications* 390 (2004), 155–174.
- [BN03] BELKIN M., NIYOGI P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. *Neural Computations* 15, 6 (2003), 1373–1396.
- [Bol98] BOLLOBÁS B.: *Modern Graph Theory*. Springer, 1998.
- [BPS93] BARNARD S. T., POTHEN A., SIMON H. D.: A spectral algorithm for envelope reduction of sparse matrices. In *Proc. of ACM Conference on Supercomputing* (1993), pp. 493–502.
- [CC94] COX T. F., COX M. A. A.: *Multidimensional Scaling*. Chapman & Hall, 1994.
- [Cha84] CHAVEL I.: *Eigenvalues in Riemannian Geometry*. Academic Press, 1984.
- [Chu97] CHUNG F. R. K.: *Spectral Graph Theory*. AMS, 1997.
- [CK04] CAELLI T., KOSINOV S.: An eigenspace projection clustering method for inexact graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 4 (2004), 515–519.
- [CTSO03] CHEN D.-Y., TIAN X.-P., SHEN Y.-T., OUHYOUNG M.: On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3 (2003), 223–232.
- [DBG\*06] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. In *Proc. of ACM SIGGRAPH* (2006), pp. 1057–1066.
- [DGLS01] DAVIES E. B., GLADWELL G. M. L., LEYDOLD J., STADLER P. F.: Discrete nodal domain theorems. *Lin. Alg. Appl.* 336 (2001), 51–60.
- [DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. of ACM SIGGRAPH* (1999), pp. 317–324.
- [DPS02] DÍAZ J., PETIT J., SERNA M.: A survey of graph layout problems. *ACM Computing Survey* 34, 3 (2002), 313–356.
- [dST04] DE SILVA V., TENENBAUM B.: *Sparse Multidimensional Scaling using Landmark Points*. Tech. rep., Stanford University, June 2004.
- [dV90] DE VERDIÈRE Y. C.: Sur un nouvel invariant des graphes et un critère de planarité. *Journal of Combinatorial Theory, Series B* 50 (1990), 11–21.
- [DZM05] DYER R., ZHANG H., MÖLLER T.: *An Investigation of the Spectral Robustness of Mesh Laplacians*. Tech. rep., Simon Fraser University, March 2005.
- [EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 10 (2003), 1285–1295.
- [EY36] ECKART C., YOUNG G.: The approximation of one matrix by another of lower rank. *Psychometrika* 1 (1936), 211–218.
- [Far48] FARY I.: On straight line representations of planar graphs. *Acta Sci. Math.* 11 (1948), 229–233.
- [FBCM04] FOWLKES C., BELONGIE S., CHUNG F., MALIK J.: Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 2 (2004), 214–225.
- [Fie73] FIEDLER M.: Algebraic connectivity of graphs. *Czech. Math. J.* 23 (1973), 298–305.
- [GGS03] GOTSMAN C., GU X., SHEFFER A.: Fundamentals of spherical parameterization for 3d meshes. *ACM Trans. Graph.* 22, 3 (2003), 358–363.
- [GM00] GUATTERY S., MILLER G. L.: Graph embeddings and laplacian eigenvalues. *SIAM Journal on Matrix Analysis and Applications* 21, 3 (2000), 703–723.
- [Got03] GOTSMAN C.: On graph partitioning, spectral analysis, and digital mesh processing. In *Proc. of Shape Modeling International (SMI)* (2003), pp. 165–171.
- [Hal70] HALL K. M.: An r-dimensional quadratic placement algorithm. *Manage. Sci.* 17, 8 (1970), 219–229.
- [HS97] HOFFMAN D. D., SINGH M.: Saliency of visual parts. *Cognition* 63 (1997), 29–78.
- [HKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. of ACM SIGGRAPH* (2001), pp. 203–212.
- [IL05] ISENBURG M., LINDSTROM P.: Streaming meshes. In *Proc. of IEEE Visualization* (2005), pp. 231–238.
- [Jai89] JAIN A. K.: *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [JNT01] JAKOBSON D., NADIRASHVILI N., TOTI J.: Geometric properties of eigenfunctions. *Russian Mathematical Surveys* 56, 6 (2001), 1085–1105.
- [JZ06] JAIN V., ZHANG H.: Robust 3D shape correspondence in the spectral domain. In *Proc. of Shape Modeling International (SMI)* (2006), pp. 118–129.
- [JZ07] JAIN V., ZHANG H.: A spectral approach to shape-based retrieval of articulated 3D models. *Computer Aided Design* 39 (2007), 398–407.
- [JZvK07] JAIN V., ZHANG H., VAN KAICK O.: Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling* (2007). to appear.
- [KCH02] KOREN Y., CARMEL L., HAREL D.: ACE: A fast multiscale eigenvector computation for drawing huge graphs. In *Proc. of the IEEE Symposium on Information Visualization (InfoVis)* (2002), pp. 137–144.

- [KG00] KARNI Z., GOTSMAN C.: Spectral compression of mesh geometry. In *Proc. of ACM SIGGRAPH* (2000), pp. 279–286.
- [KLT05] KATZ S., LEIFMAN G., TAL A.: Mesh segmentation using feature point and core extraction. *The Visual Computer (Pacific Graphics)* 21, 8-10 (2005), 649–658.
- [Kor03] KOREN Y.: On spectral graph drawing. In *Proc. of the International Computing and Combinatorics Conference* (2003).
- [KR05] KIM B. M., ROSSIGNAC J.: Geofilter: Geometric selection of mesh filter parameters. *Computer Graphics Forum (Special Issue of Eurographics 2005)* 24, 3 (2005), 295–302.
- [KSO04] KOLLURI R., SHEWCHUK J. R., O'BRIEN J. F.: Spectral surface reconstruction from noisy point clouds. In *Proc. of Eurographics Symposium on Geometry Processing* (2004), pp. 11–21.
- [KVV00] KANNAN R., VEMPALA S., VETTA A.: On clustering - good, bad, and spectral. In *FOCS* (2000), pp. 367–377.
- [Lev06] LEVY B.: Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *Proc. of Shape Modeling International (SMI)* (2006).
- [LJZ06] LIU R., JAIN V., ZHANG H.: Subsampling for efficient spectral mesh processing. In *Proc. of Computer Graphics International* (2006).
- [Lov93] LOVÁSZ: Random walks on graphs: a survey. In *Combinatorics, Paul Erdős is eighty*, Keszthely, (Ed.), vol. 2. Budapest: János Bolyai Math. Soc., 1993, pp. 353–397.
- [LS99] LOVÁSZ L., SCHRIJVER A.: On the null space of a colin de verdière matrix. *Annales de l'institut Fourier* 49, 3 (1999), 1017–1026.
- [LZ04] LIU R., ZHANG H.: Segmentation of 3D meshes through spectral clustering. In *Pacific Graphics* (2004), pp. 298–305.
- [LZ07] LIU R., ZHANG H.: Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum (Special Issue of Eurographics 2007)* 26 (2007), to appear.
- [LZvK06] LIU R., ZHANG H., VAN KAICK O.: *An Investigation into Spectral Sequencing using Graph Distance*. Tech. Rep. TR 2006-08, School of Computing Science, SFU, May 2006.
- [McG] McGill 3D Shape Benchmark: <http://www.cim.mcgill.ca/~shape/benchMark/>.
- [MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Proc. of VisMath* (2002).
- [Mey01] MEYER C. D.: *Matrix analysis and applied linear algebra*. SIAM, 2001, p. 550.
- [MKR03] M. KAZHDAN T. F., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. of Symposium on Geometry Processing (SGP)* (2003), pp. 156–165.
- [Moh97] MOHAR B.: Some applications of Laplacian eigenvalues of graphs. In *Graph Symmetry: Algebraic Methods and Applications*, Hahn G., Sabidussi G., (Eds.). Kluwer, 1997, pp. 225–275.
- [MP93] MOHAR B., POLJAK S.: *Eigenvalues in Combinatorial Optimization*, vol. 50 of *IMA Volumes in Mathematics and Its Applications*. Springer-Verlag, 1993, pp. 107–151.
- [MP04] MANOR L., PERONA P.: Self-tuning spectral clustering. In *Neural Information Processing Systems* (2004).
- [NJW02] NG A. Y., JORDAN M. I., WEISS Y.: On spectral clustering: analysis and an algorithm. In *Neural Information Processing Systems* (2002), vol. 14.
- [OMT02] OHBUCHI R., MUKAIYAMA A., TAKAHASHI S.: A frequency-domain approach to watermarking 3D shapes. *Computer Graphics Forum* 21, 3 (2002), 373–382.
- [OTMM01] OHBUCHI R., TAKAHASHI S., MIYAZAWA T., MUKAIYAMA A.: Watermarking 3D polygonal meshes in the mesh spectral domain. In *Proc. of Graphics Interface* (2001), pp. 9–18.
- [PF98] PERONA P., FREEMAN W.: A factorization approach to grouping. In *Proc. of European Conference on Computer Vision* (1998), pp. 655–670.
- [PG01] PAULY M., GROSS M.: Spectral processing of point-sampled geometry. In *Proc. of ACM SIGGRAPH* (2001), pp. 379–386.
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 2, 1 (1993), 15–36.
- [PST00] PISANSKI T., SHAW-TAYLOR J.: Characterizing graph drawing with eigenvectors. *Journal of Chemical Information and Computer Sciences* 40, 3 (2000), 567–571.
- [Ros97] ROSENBERG S.: *The Laplacian on a Riemannian Manifold*. Cambridge University Press, 1997.
- [RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplacian-Beltrami spectra as 'shape-DNA' for shapes and solids. *Computer Aided Geometric Design* 38 (2006), 342–366.
- [SB92] SHAPIRO L. S., BRADY J. M.: Feature-based correspondence: an eigenvector approach. *Image and Vision Computing* 10, 5 (1992), 283–288.
- [SCOT03] SORKINE O., COHEN-OR D., TOLEDO S.: High-pass quantization for mesh encoding. In *Proc. of Symposium on Geometry Processing* (2003), pp. 42–51.
- [SFYD04] SAERENS M., FOUSS F., YEN L., DUPONT P.: *The Principal Components Analysis of a Graph, and its Relationship to Spectral Clustering*, vol. 3201 of *Lecture Notes in Artificial Intelligence*. Springer, 2004, pp. 371–383.
- [SM97] SHI J., MALIK J.: Normalized cuts and image segmentation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (1997), pp. 731–737.
- [SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.
- [SMD\*05] SHOKOUFHANDEH A., MACRINI D., DICKINSON S., SIDDIQI K., ZUCKER S.: Indexing hierarchical structures using graph spectra. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 7 (2005), 1125–1140.
- [Sor05] SORKINE O.: Laplacian mesh processing. In *Eurographics State-of-the-Art Report* (2005).
- [SS02] SCHÖLKOPF B., SMOLA A. J.: *Learning with Kernels*. MIT Press, 2002.
- [SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON

- S.: Skeleton based shape matching and retrieval. In *Proc. of Shape Modelling International (SMI)* (2003), pp. 130–142.
- [SSM98] SCHÖLKOPF B., SMOLA A. J., MÜLLER K.-R.: Non-linear component analysis as a kernel eigenvalue problem. *Neural Computations* 10 (1998), 1299–1319.
- [ST96] SPIELMAN D. A., TENG S.-H.: Spectral partitioning works: Planar graphs and finite element meshes. In *IEEE Symposium on Foundation of Computer Science* (1996), pp. 96–105.
- [STWCK05] SHAWE-TAYLOR J., WILLIAMS C. K. I., CRISTIANINI N., KANDOLA J.: On the eigenspectrum of the gram matrix and the generalization error of kernel-PCA. *IEEE Trans. on Information Theory* 51, 7 (2005), 2510–2522.
- [Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Proc. of ACM SIGGRAPH* (1995), pp. 351–358.
- [Tau00] TAUBIN G.: Geometric signal processing on polygonal meshes. In *Eurographics State-of-the-Art Report* (2000).
- [TB97] TREFETHEN L. N., BAU D.: *Numerical Linear Algebra*. SIAM, 1997.
- [Tut63] TUTTE W. T.: How to draw a graph. *Proc. London Math. Society* 13 (1963), 743–768.
- [Ume88] UMEYAMA S.: An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.* 10, 5 (1988), 695–703.
- [vL06] VON LUXBURG U.: *A Tutorial on Spectral Clustering*. Tech. Rep. TR-149, Max Plank Institute for Biological Cybernetics, August 2006.
- [VL07] VALLET B., LÉVY B.: *Spectral Geometry Processing with Manifold Harmonics*. Tech. rep., ALICE-2007-001, March 2007.
- [vLBB05] VON LUXBURG U., BOUSQUET O., BELKIN M.: Limits of spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)* (2005), vol. 17, pp. 857–864.
- [VM03] VERMA D., MEILA M.: *A comparison of spectral clustering algorithms*. Tech. Rep. UW-CSE-03-05-01, University of Washington, 2003.
- [VS01] VRANIĆ D. V., SAUPE D.: 3D shape descriptor based on 3D Fourier transform. In *Proc. EURASIP Conf. on Digital Signal Processing for Multimedia Communications and Services* (2001).
- [Wei99] WEISS Y.: Segmentation using eigenvectors: A unifying view. In *Proc. of the International Conference on Computer Vision* (1999), pp. 975–983.
- [Xu04] XU G.: Discrete Laplace-Beltrami operators and their convergence. *Comput. Aided Geom. Des.* 21, 8 (2004), 767–784.
- [ZB04] ZHANG H., BLOK H. C.: Optimal mesh signal transforms. In *Proc. IEEE Geometric Modeling and Processing* (2004), pp. 373–379.
- [ZF03] ZHANG H., FIUME E.: Butterworth filtering and implicit fairing of irregular meshes. In *Proc. of Pacific Graphics* (2003), pp. 502–506.
- [Zha04] ZHANG H.: Discrete combinatorial Laplacian operators for digital geometry processing. In *Proc. SIAM Conf. on Geom. Design and Comp.* (2004), pp. 575–592.
- [ZKK02] ZIGELMAN G., KIMMEL R., KIRYATI N.: Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. on Vis. and Comp. Graph.* 8, 2 (2002), 198–207.
- [ZL05] ZHANG H., LIU R.: Mesh segmentation via recursive and visually salient spectral cuts. In *Proc. of Vision, Modeling, and Visualization* (2005).
- [ZR72] ZAHN C. T., ROSKIES R. Z.: Fourier descriptors for plane closed curves. *IEEE Trans. on Computers* 21, 3 (1972), 269–281.
- [ZSGS04] ZHOU K., SNYDER J., GUO B., SHUM H.-Y.: Isocharts: Stretch-driven mesh parameterization using spectral analysis. In *Proc. of Symposium on Geometry Processing* (2004).
- [ZW05] ZHU P., WILSON R. C.: A study of graph spectra for comparing graphs. In *Proc. of British Machine Vision Conference* (2005).