An Investigation of the Spectral Robustness of Mesh Laplacians

Ramsay Dyer, Richard (Hao) Zhang, Torsten Möller, Andrew Clements
Simon Fraser University, GrUVi Lab, Burnaby, Canada
{rhdyer,haoz,torsten,aclement}@cs.sfu.ca

Abstract

This paper presents an empirical study of the spectral properties of mesh Laplacian operators and how they are affected by changes in the the connectivity or number of vertices of the mesh. Specifically, we investigate to what extent the eigenvalues of an operator differ when evaluated on different meshes representing the same object. Our experiments compare two different types of operators, one being purely connectivity based and the other based on discrete approximations to the Laplace-Beltrami operator. The spectrum of the latter operators generally displayed more robustness.

1. Introduction

One of the motivating interests in the spectral properties of mesh Laplacian operators comes from a desire to gain an insight into how traditional signal processing theory, especially Fourier analysis, can be applied to study the geometry of surfaces.

Taubin [24] pointed out that the eigenfunctions of the Tutte Laplacian operator can serve as basis functions which allow signals defined on a 2D-manifold to be expanded in a series akin to the Fourier series. Without any formal justification, these basis functions are simply ordered by increasing eigenvalues of the Laplacian. A common belief is that these eigenvalues capture the frequencies of the various modes of vibration of the underlying mesh graph [24, 15]

In differential geometry, the Laplacian (or Laplace-Beltrami operator) is defined on Riemannian manifolds and its eigenfunctions are considered analogous to the Fourier basis functions [19]. Discrete approximations to the differential Laplacian also exist [16, 7], but their spectral properties have not been explicitly exploited for digital geometry processing or analysis.

Our current work is motivated by the potential of using spectral properties of mesh Laplacians to characterise 3D shapes. This would allow us to extend the use of Fourier descriptors [27] and related measures to the surface setting. The spectral properties of interest are derived from the eigenvalues and eigenvectors of a mesh Laplacian, as well as projections of the mesh embedding function along the eigenbases, i.e., the spectral transforms.

We are interested in the extent to which the underlying continuous shape associated with a mesh is captured by these spectral properties. In order to qualify as shape descriptors, such properties must be sufficiently robust against up/down sampling, remeshing, and other mesh transformations that keep the geometry intact. This robustness would also allow us to compute spectral shape descriptors using a low mesh resolution, resulting in a significant reduction in computational cost.

The experiments conducted in this work focus on the eigenvalues of the Laplacians. We explain this choice in section 3, where the operators considered and some relevant theory are also presented. We test the robustness of the Laplacian spectrum against changes in connectivity and changes in sampling density, following procedures outlined in section 4. Experimental results are given in section 5 and their implications and future work are discussed in section 6.

2. Related Work

In the digital geometry processing literature, mesh Laplacian operators have mostly been employed in the design of smoothing filters [6, 25, 29]. Efficient means of computing their spectral transforms, e.g., via mesh partitioning, has led to the development of spectral compression [15, 22] and watermarking [17] schemes for mesh geometry. The operator favored here is a graph Laplacian, well-known from spectral graph theory [5], which we will refer to as the Kirchoff operator [17] (see section 3.1.1). It is interesting to note that both [17] and [22] exploit the fact that small perturbations to the *low* frequency components of a geometric signal are less perceptible than corresponding perturbations to high-frequency components.

One of the operators that is featured prominently in our current work was presented cogently by Meyer et al. [16], although a few of its variants had appeared previously, e.g., in [6] and [18]. It is a discrete approximation to the Laplace-Beltrami operator from differential geometry. Several other discretisations of the Laplace-Beltrami operator are not considered in this paper. These include an operator developed for valid planar mesh embedding [7] and those considered for their convergence properties [26, 8].

Although mesh Laplacian operators have been used quite extensively so far, there has been no comparative study of their spectral properties. In fact, most work to date, on the various properties of mesh Laplacians, have focused on the connectivity-based operators [1, 12, 11, 28], which are simply variants of the Kirchhoff operator. The discrete differential Laplacian of Meyer et al. [16] has only been used to derive surface flows [6, 18]. Neither its spectral characteristics nor the related robustness issue has been addressed before.

In the vision and machine learning communities however, some attention has been paid to spectral robustness [3]. There the focus has been on spectral embeddings formed by selected eigenvectors, ordered by eigenvalue magnitude, of an affinity matrix. The affinity matrix encodes pairwise relationships between points in a feature space and can be seen as a generalization of a weighted adjacency matrix, the latter being closely related to our mesh Laplacians. These spectral embeddings have been used for shape correspondence, graph matching, and cluster analysis [2, 21, 3, 4]. Although it has been noted that the embeddings can be quite sensitive to point jitters and outliers [4], the precise nature of these phenomena has not been explored further.

3. Robustness Analysis

A *mesh* $M = (\mathcal{K}, X)$ is a simplicial complex $\mathcal{K}$ together with an embedding $X : \mathcal{K} \hookrightarrow \mathbb{R}^3$ (see for example [13] for more details). As is typical, we will assume the embedding function to be piecewise linear and defined by its values at the vertices. Also we will only consider manifold meshes without boundaries.

In practice a Laplacian operator is defined over a mesh that represents some continuous surface. We are interested in the extent to which the geometry represented by the mesh is captured by the spectral properties of the operator. It is thus necessary to identify operators whose spectral properties are robust when evaluated on different meshes that represent the same object. In this paper we focus specifically on the eigenvalues.

The convergence properties of an operator under mesh refinement are interesting [26], but they do not directly reflect robustness. If the spectrum of an operator converges under subdivision for example, then we are able to assign eigenvalues associated with that operator on the limit surface. Then, any fixed number of leading eigenvalues can be made to approximate the ideal ones as closely as we like, provided we carry out enough levels of subdivison.

When we talk about robustness on the other hand we are refering to a resistance to change in the eigenvalues with respect to different numbers of samples or different connectivity on the same idealised surface. Thus although an operator displays convergence under appropriate refinement, it may not be particularly robust when compared over different coarse samplings. Conversely, an operator may show good robustness properties and yet not exhibit convergence.

Next, we will first discuss what we mean by a Laplacian operator, and why we have chosen to focus our research on these operators. Then we will introduce the operators we have used in our experiments and the various quantities that we have studied.

3.1. Laplacian Operators

Mesh Laplacian operators are linear operators that act on functions defined on a mesh. Functions are specified by their values at the vertices, thus if M has n vertices, functions on M will be represented by vectors with n components and a Laplacian operator will be described by an $n \times n$ matrix.

Loosely speaking, a 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 neighbour nodes. Specifically, for our purposes a Laplacian will have a local form given by

$$(Lf)_i = {a_i}^{-1} \sum_{j \in N(i)} w_{ij} \left(f_i - f_j\right), \qquad (1)$$

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

For our purposes weights are not required to be positive, but we will impose restrictions on the allowed weights in terms of two global properties of the operator as will be explained shortly.

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

$$L = A^{-1}Q, \qquad (2)$$

where A^{-1} is a diagonal matrix whose diagonal entries are the ${a_i}^{-1}$ and Q is a symmetric matrix whose diagonal entries are given by $Q_{ii} = \sum_{j \in N(i)} w_{ij}$ and whose off diagonal entries are $-w_{ij}$. Although L itself is not symmetric, it is similar to the symmetric matrix $O = A^{-1/2} Q A^{-1/2}$ and so it has the same real eigenvalues. It is easy to see that if ξ_i is an eigenvector of O with eigenvalue λ_i, then $e_i = A^{-1/2}\xi_i$ is an eigenvector of L with eigenvalue λ_i.

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

$$\langle f, g \rangle_A = f^T A g, \tag{3}$$

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

$$\langle e_i, e_j \rangle_A = {e_i}^T A e_j = {\xi_i}^T \xi_j = \delta_{ij}.$$

For some operators, such scalar products have natural interpretation as an approximation to integration.

Zhang [28] characterises a generalised Laplacian operator, L on a connected mesh M by two essential properties:

Prop. 1 The nullspace of L is the constant vectors (vectors all of whose coefficients are the same).

Prop. 2 All eigenvalues of L are real and non-negative.

Operators that are described by equation (1) are called *first order* Laplacians because they only involve the one-ring neighbours at each vertex. If the weights are all positive, such operators satisfiy the above two properties. We do not require that the weights be positive, but we do require that the above two properties be satisfied in order for an operator to be a Laplacian.

More general Laplacian operators may be defined which satisfy these properties but cannot be described by equation (1). In fact we will be considering one such operator in this paper: the operator corresponding to the linear finite element method.

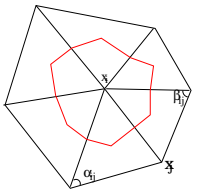
3.1.1. Types of Laplacian Operators For the subsequent discussion it is convenient to identify the graph $\mathcal{G} = (V, E)$ formed by the vertices and edges of the simplicial complex $\mathcal{K}$ associated with a mesh M.

An operator L on M is usually defined by specifying its local form, the a_i and w_{ij} in equation (1). If the definition of these quantities involves only the connectivity information contained in $\mathcal{G}$ then we say that L is a *topological operator*. If the embedding X is also used in defining the operator, then we say it is a *geometric operator*.

The primary example of a topological operator that we will use is the Kirchoff operator, defined by setting $a_i = 1$ and $w_{ij} = 1$ in equation (1):

$$(L_K f)_i = \sum_{j \in N(i)} (f_i - f_j).$$

Another popular topological operator, the one that was used in [24] and [15] for example, is defined by $L_T = D^{-1} L_K$, where D is the diagonal matrix whose i^{th} diagonal entry is d_i, the valence of the i^{th} vertex. We refer to L_T as the Tutte Laplacian. In terms of equation (1) it is obtained by setting $a_i = d_i$ and $w_{ij} = 1$.



Geometric operators are usually modeled on the differential Laplacian operator (Laplace-Beltrami operator) that is defined on a smooth surface (a 2D Riemannian manifold). The one on which we have concentrated our studies has been described in [16]. It is defined only on manifold triangle meshes and its local form is

$$[L_b(f)]_i = \frac{1}{|\Omega_i|} \sum_{j \in N(i)} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})(f_i - f_j), \tag{4}$$

where the angles α_{ij} and β_{ij} are shown in the inset to the left. $|\Omega_i|$ refers to the size of the cell created by joining the midpoint of each edge adjacent to $\boldsymbol{x}_i$ with the barycentre the adjacent triangles (barycentre cell containing $\boldsymbol{x}_i$). It is one third the area defined by the one ring [16]. Thus for this operator the diagonal matrix A of equation (2) has entries that are the areas of the barycentre cells of the associated vertex and the trace of this matrix is the surface area of the mesh. We will refer to this operator as the bary-cot operator.

It should be noted that Meyer et al. [16] expressed a preference for using the Voronoi cells of each vertex rather than the barycentre cells. An argument was given for the optimality of this choice, based on the assumption that the approximating functions are piecewise constant. In practice we have not noticed any significant difference in the spectral properties of the operator resulting from using one choice of area cell over the other.

As a discrete approximation to the differential Laplacian, this bary-cot operator can be derived by treating the discrete functions on M as piecewise linear and using the divergence theorem on a barycentre cell.

In this piecewise linear context a natural choice of scalar product in the function space might be integration of piecewise linear functions. However, the eigen-

functions are not orthogonal with respect to this inner product. The scalar product (3) which renders L_b self adjoint is given by a naïve approximation to the integral:

$$\langle f, g \rangle_A = \sum_{i=0}^{n} f_i g_i \, |\Omega_i| \simeq \int_M f g \, ds, \tag{5}$$

and so this is the one which must be employed to compute transform coefficients for example. Such a scalar product is attractive because it will tend to compensate for nonuniform sampling densities.

However, we can consider a refinement of this operator where the associated scalar product on the function space is exactly integration of piecewise linear functions. This requires a refinement of the matrix A in order to obtain the appropriate bilinear form. The matrix Q remains exactly as for the bary-cot operator.

We represent the piecewise linear functions in terms of the nodal basis (hat) functions $\{\varphi_i\}$:

$$f = \sum_{i=0}^{n} f_i \varphi_i$$

so that the scalar product induced by integration is given by

$$\int_M f g \, ds = \sum_{i=0}^{n} \sum_{j=0}^{n} f_i g_j \int_M \varphi_i \varphi_j \, ds = \langle f, g \rangle_A$$

The matrix

$$A = [a_{ij}] = \int_M \varphi_i \varphi_j \, ds \tag{6}$$

is sparse: only diagonal enteries and entries corresponding to first order neighbours are non-null. The entries on the diagonal are $\frac{1}{6}$ the area of the corresponding one-ring and the off diagonal entries a_{ij} are $\frac{1}{12}$ the area of the two triangles adjacent to the edge $[i, j]$ (or zero if there is no such edge). This can be easily verified by considering the integral (6) over a single right triangle. A general triangle can be transformed into a right triangle by an area preserving shear map. Note that the resulting matrix A is symmetric positive definite and so corresponds to a valid bilinear form.

To compute this new operator $L_f = A^{-1}Q$ would require inverting the matrix A. However, for our purposes this is not necessary because ARPACK++ is able to solve the generalised eigenvalue problem $Qe = \lambda Ae$. We recognise this equation as the one that results from the finite element method [23] with piecewise linear elements and so we will refer to it as the fem operator.

As a third variation of an operator based on the cot formula (4) we will look at the properties of the operator given by setting $\Omega_i = 1$ in equation (4). Thus the associated bilinear form is just the identity matrix and the operator is just the symmetric matrix Q. This operator is featured in the work of Polthier, see [18] for example. We will refer it as the sym-cot operator.

Many of the properties of the differential Laplacian operator are also enjoyed by the sym-cot, bary-cot and fem operators in the piecewise linear setting [18]. In particular, the formula

$$\langle f, \mathrm{L}\, f \rangle_A = \int_M \|\boldsymbol{\nabla} f\|^2 \, ds \tag{7}$$

applies when f is considered as a piecewise linear function. This guarantees that the **Prop. 1** and **Prop. 2** are satisfied.

3.2. Measuring Robustness

The spectral robustness of an operator is reflected in both its eigenvalues and its eigenvectors. If the eigenvectors of an operator are to be used as the basis for a Fourier-type transform, then it is certainly desirable that they be robust so as to ensure consistent values for the transform coefficients.

However, it is difficult to quantify directly the robustness of eigenvectors. A starting point is to consider the eigenvectors as discrete eigenfunctions on a continuous surface represented by the mesh. These functions can be visualised by colouring the surface according to the value of the function at each point. Qualitatively, robustness would be reflected by a similar appearance of the eigenfunctions over different discretisations of the same surface.

A more quantitative, although somewhat indirect, measurement can be made by considering the coefficients obtained by performing a spectral transformation on the embedding function X (typically the transform coefficients of each of the three coordinate functions are computed independently).

Thus if a surface S is represented by meshes M and M' with embedding functions X and X' respectively, we compare the coefficients $\hat{X}_i$ and $\hat{X}'_i$, arising from the expansions $X = \sum_i \hat{X}_i e_i$ and $X' = \sum_i \hat{X}'_i e'_i$ where e_i and e'_i are the eigenfunctions of an operator L on the respective meshes.

A number of factors influence the consistency of such measurements. The eigenfunctions must line up with the geometry of the object in a consistent fashion. In other words if the zero crossing curves are plotted on the surface, they should lie in roughly the same place regardless of the underlying mesh.

For topological operators it is easy to see that such a criterion can never be guaranteed. Such operators have no direct geometric information so their eigenfunctions can only align with the connectivity pattern

of the mesh. If a mesh M is viewed as an embedding of a graph $\mathcal{G}$ on a surface S, then we can construct a mesh M' by using a different embedding of the same underlying graph on the same surface. The orientation of the eigenfunctions relative to the surface will be different in the two cases.

However, in practice we have observed that the first few eigenfunctions of the topological operators do seem to line up with the geometry in a consistent manner. This brings to mind the observation that a considerable amount of geometric information is inherent in the connectivity of most meshes [14]. Nonetheless, the lack of any hope of a guarantee makes the use of topological operators less appealing.

This problem is not inherent to the geometric operators. For the discrete differential operator (4), the eigenfunctions do seem to consistently orient themselves with the geometry. However here we have observed a different complication in the relationship between geometry and the eigenfunctions.

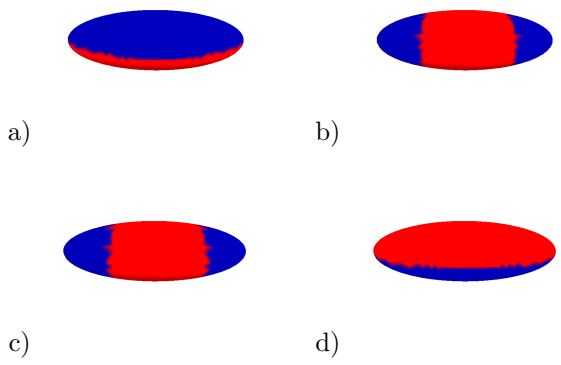


Figure 1. Eigenfunctions are visualised by colouring the surface according to the sign of the function value at each point, i.e. red is positive and blue is negative. The top row shows an ellipsoid with a slightly shorter major axis than the one shown in the bottom row. Both ellipsoids have a minor axis of length 0.5 and a semi-major axis of length 0.7. For the top row, the length of the major axis is 1.59 (a,c) depict eigenvector number three and (b,d) show eigenvector number four. For the bottom row, the major axis is 1.65. The relative ordering of qualitatively identical eigenvectors has changed.

If two objects are very similar in shape, we expect their eigenfunctions to be correspondingly similar. However, we have observed that the *order* of the eigenfunctions, as defined by the magnitude of the corresponding eigenvalue, can be sensitive to small changes in the shape. An example of this is shown in figure 1, where we have compared ellipsoids that are almost identical.

This behaviour complicates any attempts to use the spectral coefficients as part of a shape indexing scheme.

Note that this problem exhibited by figure 1 can arise when we are considering different meshings of the same ellipsoid if it has a principle axis of the critical length where the fourth and fifth eigenfunctions 'swap places'. In this case the fourth and fifth eigenvalues are practically the same; we have a degenerate eigenvalue.

If the eigenspace $\mathcal{E}_\lambda$ associated with an eigenvalue λ has multiple dimensions, then any basis for $\mathcal{E}_\lambda$ can represent the orthogonal eigenfunctions associated with λ. Thus the spectral coefficients associated with each specific eigenfunction are not well defined. Even if $\mathcal{E}_\lambda$ has a single dimension, the normalised eigenfunction is only defined up to sign.

To get around these problems we can define our coefficients to be the magnitude of the projection of the function into each eigenspace. This leads to the problem of deciding when two eigenvalues are close enough to be considered the same. There will be sensitivity of the coefficients in the neighbourhood of this threshold. These algorithmic issues tend to eclipse the focus of our study which is the spectral robustness of the operators themselves.

The other approach, the one we take in this paper, is to measure the robustness of the eigenvalues. This avoids many of the complications associated with measuring eigenvectors. If robustness of the eigenvalues is not manifest, we do not expect robustness of the eigenvectors. If e is a normalised eigenfunction of L with eigenvalue λ then we have

$$\lambda = \langle e, \mathrm{L}\, e \rangle_A. \tag{8}$$

In particular, for the operators based on the cotangent formula (4), we see from equation (7) that the eigenvalues are given by the integral of the squared norm of the gradient of the corresponding eigenfunctions. Thus, at least for the discrete differential operators, the robustness of the eigenfunctions implies a robustness of the eigenvalues.

In order to be able to compare eigenvalues resulting from different meshes, we need to apply a normalisation factor. The normalisation factor we use is $\langle \mathbb{1}, \mathbb{1} \rangle_A$, where $\mathbb{1}$ is the constant vector of all ones. In other words we use the sum of all elements in the associated bilinear form. For the bary-cot and fem operators this means that we multiply the eigenvalues by the surface

area of the mesh. This makes sense because if these operators are multiplied by the surface area of the mesh then they are invariant with respect to changes of scale.

For the Kirchoff operator, A is the identity, so we multiply the eigenvalues by the number of nodes in the mesh. This normalisation is the same as arises in the Nyström approximation [2], but this theory does not directly apply to our case.

4. Experimental setup

We are interested in the effects of changes in the sampling density, as well as connectivity pattern among mesh vertices on the eigenvalues of the Laplacian operators. An appropriate eigensolver also needs to be chosen to compute the Laplacian spectrum.

Numerical eigensolver: As we are primarily interested in the leading eigenvalues and eigenfunctions of the mesh Laplacians, ARPACK++ [10], a C++ interface to ARPACK, proves to be a good choice. ARPACK is a collection of Fortran77 subroutines designed to compute a few eigenvalues and corresponding eigenvectors of a large, sparse matrix, making use of an algorithmic variant of the Arnoldi process called the Implicitly Restarted Arnoldi Method.

Change of sampling density via mesh decimation: In order to obtain a family of meshes for the same object at different resolutions, we used qslim, an existing implementation of the quadric-based decimation algorithm of Garland and Heckbert [9], as well as our own implementation of the vertex removal algorithm of Schroeder et al. [20]. In the latter case, vertices are chosen randomly for removal and they are removed only if its discrete curvature is below an appropriately chosen threshold. The two decimation algorithms produce rather different mesh tessellations thus reducing the risk of interpreting artifacts of a decimation as a property of an operator. In particular, since each mesh resolution is obtained from decimating the base mesh directly, the randomised algorithm produces a family of meshes whose underlying graphs cannot be described by a tower of subgraphs.

Connectivity alteration via edge flips: For meshes, even when the locations of the sample points (vertices) are fixed, the tessellation can be altered by changing the underlying graph. From a given triangle mesh M, we create a series of related meshes by performing successive, random edge flips. An edge will not be flipped if certain criteria are met. Specifically, we require that the scalar product of the normals of the faces adjacent to an edge be larger than some threshold. We also require that the angle subtended by the newly flipped edge not be too large, so that the resulting triangles are not close to being degenerate. Since the same edge is never flipped twice, the connectivity of the meshes become progressively 'more different' from the original model as the number of edge flips increases.

5. Experimental results

Our results are divided into three subsections. The first two sections correspond to the two principle types of experiments we performed. In section 5.1 we present some of the results we obtained from experiments involving mesh decimation. In section 5.2 we present the results of experiments which changed the connectivity of the mesh while holding the sampling points fixed. Finally in section 5.3 we make observations about the times required to solve the different eigenvalue systems with ARPACK++.

The errors are presented in contour plots. In all of these plots the errors are relative to the finest resolution mesh and the relative error scale ranges from 0 to 0.1. Contour lines are drawn at intervals of 0.01.

5.1. Changes in sample density

The difference between the error plots of the Kirchoff operator and the Tutte operator is barely discernable. The Tutte operator can be viewed as the Kirchoff operator multiplied by a diagonal matrix that is a perturbation of the identity matrix times 1/6. The effects of these perturbations do not manifest themselves in the low eigenvalues. Even in the case of 4-8 meshed spheres produced by the smfsphere utility provided with qslim [9], the differences were barely noticable. We have therefore decided to focus our attention on the simpler Kirchoff operator.

In figure 2 we present the results of four different operators applied to the sphere. Meshes of varying resolutions were produced by the smfsphere utility and we plotted the eigenvalue curves of each mesh side by side so that they form a surface. In this presentation the eigenvalue curves of a robust operator will all be approximately the same so the resulting surface will resemble a kind of wide staircase.

These plots are shown on the left in figure 2. On the right we show the relative errors of the eigenvalues as compared with the highest resolution mesh. These errors are displayed as a contour plot where the eigenvalue number is varied on the x-axis and the number of vertices in the mesh is varied on the y-axis.

Each row corresponds to a different operator. On the bottom row is the fem operator. The error plot for this operator is not surprising: an increase in error is pro-

duced by a decrease in mesh resolution or an increase in the eigenvalue index. This could also be claimed as a trend in the error data for the bary-cot operator on the sphere (third row). However the errors for the other two operators display a more complicated structure. It is interesting to note that the sym-cot operator (second row) has an error structure that resembles more the Kirchoff operator than it does the other discrete differential operators.

In figure 2 the fem operator has the cleanest looking error plots. However, for general meshes that aren't derived from an analytically defined surface, the fem operator often displays error behaviour at course resolutions that is much worse than that of the bary-cot operator. An example of this is shown in figure 3. We have not observed the fem operator to significantly out perform the bary-cot operator on any model, however the converse does occur often.

Another interesting observation regarding the bary-cot and the fem operators is the fact that the low resolution eigenvalues of the fem operator are consistently greater than the corresponding high resolution values (as predicted by theory [23]), but for the bary-cot operator they are lower. This can be observed in the left hand plots of the last two rows of figure 2 for example.

The sym-cot and fem operators generally have inferior performance when compared with the bary-cot operator. We therefor focus on this latter operator as the best representative of the geometric operators that we have investigated. In figure 4 we plot the average eigenvalue errors of the Kirchoff and bary-cot operators resulting from different datasets. There is a clear trend of superior performance of the bary-cot operator over the Kirchoff operator.

An unexpected phenomenom is also apparent in figure 4. The errors of the Kirchoff operator are often greatest for the eigenvalues with low indices. This behaviour is also evidenced in our connectivity experiment example of figure 5 described in section 5.2. Although this behaviour is much less prevalent in the bary-cot operator, the latter is not immune from it as can be seen in figure 3(a).

The error plots for both operators often show an interesting behaviour: certain eigenvalues exhibit much greater or much lesser stability than others so that ridges and valleys appear in the error plots. This can also be seen in figure 3(a). In the sphere plots of figure 2 ridges of the Kirchoff operator seem to align with jumps between successive eigenspaces. However, we have not observed this as a general trend.

We do not have a good explanation for this phenomenum of robust eigenvalues. It is possible that it is

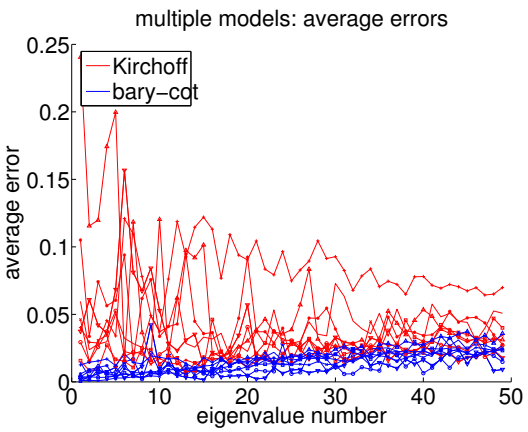


Figure 4. Average eigenvalue errors for eight different datasets. These datasets are from six different models. Four of them were decimated with qslim and four with the random vertex removal program.

an artifact of the decimation algorithms, but we have noticed this pattern with both algorithms. It can even be observed to a certain extent in the connectivity perturbation test shown in figure 5.

5.2. Connectivity perturbations via edge flips

The bary-cot operator consistantly and significantly outperforms the Kirchoff operator. Figure 5 shows a representative example. We have taken a model of a hand with 2500 vertices and performed a series of edge flips subject to the constraints described in section 4. The Kirchoff operator exhibits errors an order of magnitude greater than the bary-cot operator. The fem and sym-cot operators both perform quite well in these tests, however the bary-cot operator consistently outperforms them.

5.3. Timing

Any algorithm that is based on an eigendecomposition is faced with the problem that it is an extremely costly calculation. Generally the smaller the mesh, the faster the system can be solved but the reliability of the resulting eigenvalues is compromised if the mesh is too coarse.

In table 6 we show the time required to solve each of the five operators on three different meshes. The first mesh is a hand model with 6191 vertices. The second mesh is a sphere with 20000 vertices produced by the smfsphere ustility [9]. The final model is a fish with

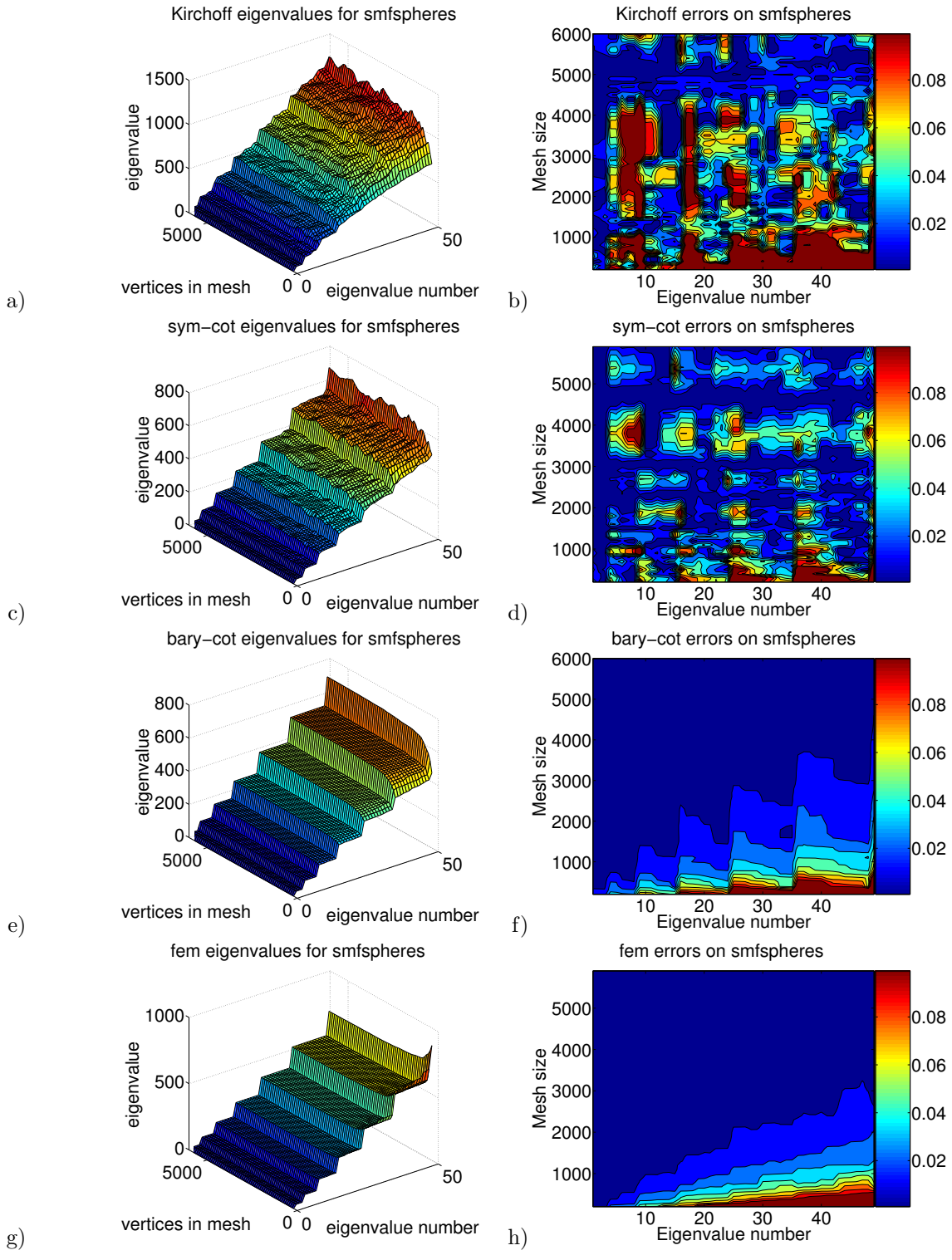


Figure 2. Eigenvalues and their corresponding errors for four different operators on the sphere. On the left are the eigenvalue curves plotted side by side for varying mesh resolutions. On the right are contour plots of the errors relative to a 20000 vertex reference mesh. See the text body for discussion.

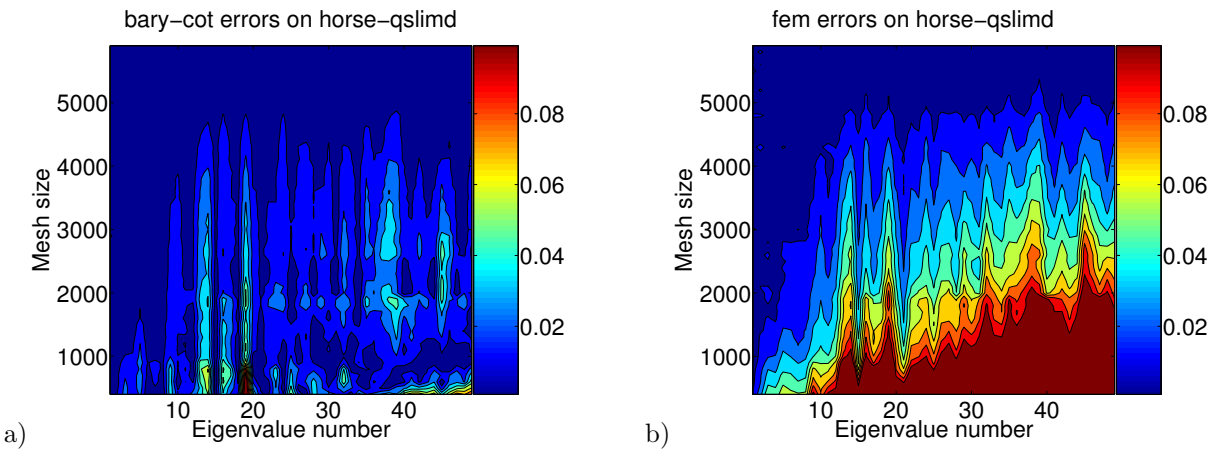


Figure 3. Shown here are the eigenvalue errors for the bary-cot operator (a) and the fem operator (b) on a horse model. The bary-cot operator shows greater robustness. The meshes of different resolutions were produced with qslim.

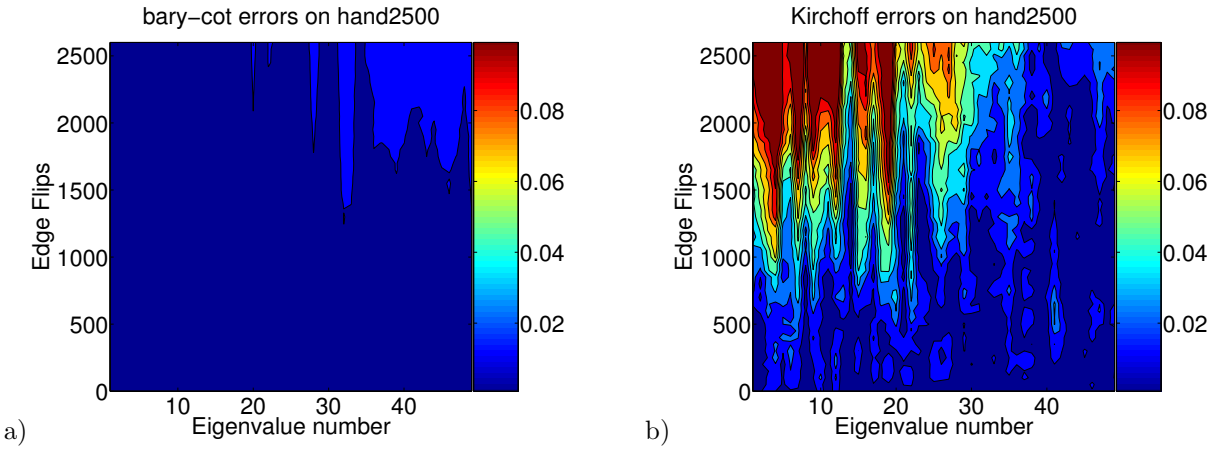


Figure 5. Eigenvalue errors for the bary-cot operator (a) and the Kirchoff operator (b) on a hand model submitted to progressive edge flips. The bary-cot operator shows errors an order of magnitude smaller than the Kirchoff operator.

6000 vertices. The fish has been decimated from a much larger mesh using qslim [9].

For the topological operators the size mesh seems to be the most important factor influencing the time required to solve the system. However for the operators based on equation (4), the shape of the triangles can be much more important. qslim can produce very thin triangles which presumably result in ill conditioned matrices when the cot operators are employed. Fifty FEM eigenvalues for the fish of 6000 vertices took 11 hours to compute whereas those of the sphere with 20000 vertices were computed in 45 minutes.

One of the goals of this paper was to gain insight into how much a mesh can be decimated before too much of the essential spectral information of the original mesh is lost. It seems that the quality of the decimated mesh is actually at least as important as its size when a mesh for fast spectral decomposition is desired.

6. Conclusions and Future Work

We have attempted to shed light on the extent to which the spectrum of a Laplacian operator is influ-

operator	hand	sphere	fish
Kirchoff	27.6	237.6	24.0
Tutte	26.1	171.8	19.1
sym-cot	43.8	237.3	5968
bary-cot	65.9	404.0	20675
fem	101.5	1590	39634

Figure 6. Times (in seconds) to compute fifty eigenvalues for various operators and models on a 2.8GHz Intel Xeon processor using ARPACK++.

enced by changes in the number of samples or connectivity of the samples of the mesh representing an object. As expected, the topological operators are much more sensitive to connectivity changes than the geometric operators. The geometric operators also generally outperformed the topological ones with respect to changes in sampling density.

Among the three geometric operators we investigated, the bary-cot operator had the best performance. Overall it is this operator that displayed the least sensitivity with respect to changes in sampling density and connectivity.

One striking phenomenon that we have observed is the occurence of isolated eigenvalues that exhibit much more robustness than the neighbouring eigenvalues. It appears unlikely that this is merely an artifact of our experimental procedures. It would be interesting to understand and exploit this phenomenon.

Another observation is that, especially for the topological operators, some of the small eigenvalues exhibit much less robustness than the larger ones.

The operators based on the cotangent formula (4) became extremely costly to solve when the quality of the triangles in the mesh became poor. Surprisingly, the eigenvalues, when they could be calculated, remained robust under these conditions. An essential tool for future spectral work using these operators will be a remeshing algorithm that produces clean meshes that allow the eigenvalue problem to be solved quickly. We have arbitrarily confined the number of calculated eigenvalues to 50. With a good remeshing algorithm we will be able to effectively explore greater numbers of eigenvalues.

Other future work will include experiments which involve perturbations of the geometry rather than the connectivity or number of samples. This would also be a step towards a possible spectral shape indexing system. Ultimately we would like to have a universal spectral transform that is resistant to the problem of degenerate eigenvalues.

References

[1] M. Ben-Chen and C. Gotsman. On the optimality of spectral compression of meshes. preprint, 2003.

[2] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[3] M. Carcassoni and E. R. Hancock. Point pattern matching with robust spectral correspondence. In *Proceedings of Computer Vision and Pattern Recognition*, pages 649–655. IEEE Computer Society, 2000.

[4] M. Carcassoni and E. R. Hancock. Alignment using spectral clusters. In *British Machine Vision Conference*, 2002.

[5] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. CBMS Regional Conference Series in Mathematics.

[6] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH 1999 Conference Proceedings*, pages 317–324. ACM SIGGRAPH, 1999.

[7] M. S. Floater. Mean value coordinates. *Comput. Aided Geom. Des.*, 20(1):19–27, 2003.

[8] K. Fujiwara. Eigenvalues of laplacians on a closed riemannian manifold and its nets. In *Proceedings of the AMS*, volume 123, pages 2585–2594, 1995.

[9] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. *SIGGRAPH*, 31(Annual Conference Series):209–216, 1997.

[10] F. M. Gomes and D. C. Sorensen. ARPACK++. http://www.ime.unicamp.br/~chico/arpack++/, 1997.

[11] C. Gotsman. On graph partitioning, spectral analysis, and digital mesh processing. In *Proceedings of the Shape Modeling International 2003*, page 165. IEEE Computer Society, 2003.

[12] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. In *SIGGRAPH 2003 Conference Proceedings*, pages 358–363. ACM SIGGRAPH, 2003.

[13] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Computer Graphics Proceedings (SIGGRAPH 99)*, pages 325–334. ACM Siggraph, 1999.

[14] M. Isenburg, S. Gumhold, and C. Gotsman. Connectivity shapes. In *Proc. of IEEE Visualization 2001*. IEEE Visualization, 2001.

[15] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 279–286. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

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

[17] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama. Watermarking 3d polygonal meshes in the mesh spectral domain. In *Graphics Interface*, pages 9–17. Canadian Information Processing Society, 2001.

[18] K. Polthier. Computational aspects of discrete minimal surfaces. In J. Hass, D. Hoffman, A. Jaffe, H. Rosenberg, R. Schoen, and M. Wolf, editors, *Proc. of the Clay Summer School on Global Theory of Minimal Surfaces*, to appear 2002.

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

[20] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. In *Proceedings of ACM SIGGRAPH*, pages 65–70. ACM Press, 1992.

[21] L. S. Shapiro and J. M. Brady. Feature-based correspondence: an eigenvector approach. *Image Vision Computing*, 10(5):283–288, 1992.

[22] O. Sorkine, D. Cohen-Or, and S. Toledo. High-pass quantization for mesh encoding. In *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 41–51, 2003.

[23] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.

[24] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH 95 Conference Proceedings*, pages 351–358. ACM SIGGRAPH, 1995.

[25] G. Taubin, T. Zhang, and G. H. Golub. Optimal surface smoothing as filter design. In *Proceedings of the 4th European Conference on Computer Vision*, volume 1, pages 283–292. Springer-Verlag, 1996.

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

[27] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21(3):269–281, March 1972.

[28] H. Zhang. Discrete combinatorial laplacian operators for digital geometry processing. In *Proceedings of SIAM Conference on Geometric Design and Computing*. Nashboro Press, 2004. to appear.

[29] H. Zhang and E. Fiume. Butterworth filtering and implicit fairing of irregular meshes. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, pages 502–506. IEEE Computer Society, 2003.