Discrete Combinatorial Laplacian Operators for Digital Geometry Processing

Hao Zhang

Abstract. Digital Geometry Processing (DGP) is concerned with the construction of signal processing style algorithms that operate on surface geometry, typically specified by an unstructured triangle mesh. An active subfield of study involves the utilization of discrete mesh Laplacian operators for eigenvalue decomposition, mimicking the effect of discrete Fourier analysis on mesh geometry. In this paper, we investigate matrix-theoretic properties, e.g., symmetry, stochasticity, and energy-compaction, of well-known combinatorial mesh Laplacians and examine how they would influence our choice of an appropriate operator or numerical method for DGP. We also propose two new symmetric combinatorial Laplacian operators for eigenanalysis of meshes and demonstrate their advantages over existing ones in several practical applications.

§1. Introduction

Frequency-domain characterization and processing of irregular triangle meshes has led to some promising developments in mesh filtering (especially smoothing [5, 22, 26]), geometry compression [13, 21], mesh watermarking [16, 17], and partitioning [10]. In this setting, the mesh geometry is represented by a 3D signal, i.e., the Cartesian (x, y, z) coordinates, defined over the vertices of the underlying graph. A mesh signal transform is given by a projection of the signal onto the eigenvectors of a suitably defined discrete Laplacian operator [22, 25].

One of the most frequently used operator for this purpose is the discrete uniform Laplacian, also known as the (normalized) *Tutte Laplacian* [9], or TL, for short. Taubin [22] points out that the eigenvectors of the TL represent the natural vibration modes of the mesh, while the corresponding eigenvalues capture its natural frequencies, resembling the scenario for

Geometric Modeling and Computing: Seattle 2003
M. Lucian and M. Neamtu (eds.), pp. 575–592.

ISBN 0-9728482-3-1

classical discrete Fourier Transform (DFT). However, the eigenvectors of the TL possess no analytical form in general and there are no fast methods, analogous to the Fast Fourier Transform, to compute the corresponding mesh signal transform.

In addition to the TL, its variants, the *Kirchhoff operator* (KL) and the *normalized graph Laplacian* (GL), have also been used for eigenvalue decomposition. While these operators are all *combinatorial*, as they depend on mesh connectivity only, *geometry-driven* Laplacians account for measures such as edge lengths and face angles. Operators of this type include the (edge-length based) scale-dependent Laplacian, or SDL, the mean curvature flow operator, suggested by Desbrun et al. [5] for implicit mesh fairing, as well as operators derived from Floater's shape-preserving weights [7] and mean-value coordinates [8], designed to generalize Tutte embedding [24] for minimizing parametric distortion. The mean curvature flow operator was actually derived much earlier by Pinkall and Polthier [18] in their study of discrete minimal surfaces. It is generally regarded as the discrete Laplacian-Beltrami operator for triangle meshes [15].

Traditionally, TL, KL, and GL have been used in spectral graph theory and their graph-theoretic properties have been studied extensively [2]. Recently, these operators have been applied to digital geometry processing, e.g., for mesh compression by Karni and Gotsman [13] and Sorkine et al. [21], mesh smoothing by Taubin [22], Desbrun et al. [5], and Zhang and Fiume [26], mesh parameterization by Floater [7] and Gotsman et al. [9], and spectral mesh watermarking by Ohbuchi et al. [16, 17]. In particular, we mention the nonlinear extension of Tutte embedding by Gotsman et al. [9] for spherical mesh parameterization, since it is of some relevance to our work. They restrict their discussion to first-order symmetric Laplacians, noting that such symmetric systems can be viewed as a mass-spring network, where vertices are point masses joined by springs of varying strengths along the edges. Symmetry also plays an important role in their proof of the validity of their spherical triangulations.

Examining these developments closely, we see that a number of important properties of the discrete Laplacian operators, e.g., symmetry and unit row sum (commonly viewed simply as the result of normalization), have often become necessary but this is not followed by further analyses. In general, there still lacks a formal and systematic study of the various properties of these operators and the subsequent theoretical or practical implications, especially in the context of geometry processing, where the emphases are often quite different from those in graph theory. For example, in transform coding, the main concerns include processing times, quality of the coded mesh, and the ability of a transform to compact signal energy. While for implicit mesh fairing [5], a sparse linear system defined by a discrete Laplacian needs to be solved iteratively, thus the convergence rate and behavior of the iterative solver becomes the central issue.

In this paper, we start with a formal treatment of linear systems for mesh signal processing and examine desirable matrix-theoretic properties, e.g., symmetry and stochasticity, of such a system. Several important implications, e.g., convergence, will be discussed and detailed proofs of these results can be found in [25]. We then establish a novel connection between the linear operators involved and the rather abstract notion of *smoothing matrices* [4]. From this, we define the class of *generalized shrinking mesh Laplacians* (GSML), which allows for a unified treatment of a larger class of mesh Laplacian operators than before. We show that the TL, KL, and the two symmetric operators we propose, the *symmetric quasi Laplacian* (*SQL*) and the *second-order symmetric Tutte Laplacian* (*SSTL*), are all GSMLs. Finally, from a practical point of view, we consider various applications of eigenvalue decomposition for digital geometry processing and demonstrate several advantages given by the two new operators.

Although we focus on combinatorial mesh Laplacians only in this paper, the type of analyses presented here can also be applied to geometry-driven operators. Note however that for transform mesh coding [13, 16], geometric operators are unsuitable to use since geometric information about the mesh is unknown prior to decoding. Also, as a mesh evolves geometrically, e.g., in iterative mesh smoothing, a geometric Laplacian would need to be recomputed, which results in more expensive computations [5]. Combinatorial operators can provide the necessary remedies, but there is good reason to believe that their dependence on mesh connectivity would make them less robust, e.g., against remeshing and mesh decimation, in characterizing 3D shapes. A focused study of the robustness of mesh Laplacians for shape characterization will be presented elsewhere.

§2. Linear mesh processing and generalized mesh Laplacians

2.1. Notations

In this paper, we focus on irregular triangle meshes and the processing of surface geometry. Thus we assume that the mesh surface is always a manifold. Mesh vertices are indexed by i, j, k, …, edges by (i, j), (j, k), …, and the graph formed is referred to as the mesh graph; its adjacency matrix $\mathcal{A}$ is defined as usual. The set of vertices adjacent to a vertex i, denoted by $N_1(i)$, are the one-ring or first-order neighbors of i. Higher-order neighbors may be defined recursively. The degree of i is denoted by d_i, and the diagonal matrix of $1/d_i$'s, $i = 1, \dots, n$, is denoted by $\mathcal{R}$.

Note that as a convention, we use calligraphic letters to denote special operators, e.g., $\mathcal{I}$ is reserved for identity matrices. General-purpose matrices and vectors are usually denoted by Q,R, …, and $\mathbf{x}, \mathbf{y}$, …, respectively. The transpose of a matrix or vector $\mathbf{p}$ is denoted by $\mathbf{p}^T$, and its L_2 norm by $||p||$. We use $\mathbf{e}$ and σ, with subscripts, to denote the eigenvectors and eigenvalues of a matrix H. The spectral radius of H, that is, the maximum magnitude of H's eigenvalues, is denoted by $\rho(H)$.

While the adjacency matrix $\mathcal{A}$ characterizes the connectivity of a mesh, its geometry is defined by the $n \times 3$ matrix $\mathbf{x}$, called the coordinate vector — it identifies the mesh. The i-th row $\mathbf{x}_i$ of $\mathbf{x}$ specifies the coordinates of vertex i. In this way, we treat the mesh as a 3D signal defined over the vertices of the mesh graph. This is similar to the vector-space representation of a 2D image [19]. In our subsequent formulation of linear systems for mesh processing, it is our intent to follow the standard treatment of the corresponding topics in image processing, e.g., as given by Jain [12].

2.2. Linear system for mesh processing and impulse response

Consider a mesh $M = (\mathcal{A}, \mathbf{x})$ with n vertices, and a connectivity-preserving linear mesh signal processing (LMSP) system $\mathbf{y} = H\mathbf{x}$ with input $\mathbf{x}$ and output $\mathbf{y}$, where $H \in \mathbf{R}^{n\times n}$. In functional form, we have $\mathbf{y}(k) = H[\mathbf{x}(k)] = H \cdot \mathbf{x}(k)$, where $\mathbf{x}(k)$ and $\mathbf{y}(k)$ denote the mesh signals indexed by k. Without loss of generality, let us work with the x-coordinates only, i.e., $\mathbf{x} \in \mathbf{R}^n$. We shall treat mesh signal processing within a similar framework as for a standard linear imaging system [12], where values at image grids denote light intensities or energy.

We view $\mathbf{x}_i$ as an amount of *signed potential energy*, relative to the origin of the coordinate space and along the x direction, at vertex i of the mesh. The impulse response of the system is then seen to mimic the result of an energy dispersion. When the input mesh is given by the discrete 1D Kronecker delta function at vertex k', i.e., $\delta(k - k') = 1$, $k, k' \in \{1, 2, \ldots, n\}$, if $k = k'$ and 0 otherwise, the output at location k is

$$h(k; k') = H[\delta(k - k')] = H \cdot \delta(k - k') = H_{k,k'}, \tag{1}$$

and is called the *impulse response* of the LMSP system. For a fixed k', $h(k; k')$ models the distribution of the unit amount of energy at vertex k' over the mesh grid. As we can see, the impulse response $h(k; k')$ is completely characterized by the matrix H. We now discuss various properties of the impulse response and the LMSP system it defines.

Nonnegativity: A matrix H is nonnegative if $H_{ij} \geq 0$ for all i and j. This is desirable for an impulse response since from a physical standpoint, it says that the weights characterizing the energy dispersion are positive. Mathematically, nonnegativity of the linear operator H offers many results from the theory of nonnegative matrices [14] at our disposal.

Irreducibility: A matrix Q is irreducible if its corresponding graph $G(Q)$ is connected [20]. Here, we use the 0-1 pattern of the matrix Q to define its corresponding graph in the obvious way: (i, j) is an edge if and only if $Q_{ij} \neq 0$. Typically, as long as the mesh is connected, its impulse response would be irreducible.

Symmetry: Symmetry of an impulse response has a nice physical interpretation: for any pair of mesh vertices i and j, the weight of energy that i receives from j is the same as the weight of energy that j receives from i. Symmetric matrices possess many desirable properties, e.g., real eigenvalues, orthogonality of eigenvectors (e.g., Parseval's theorem holds for such mesh signal transforms [25]), and potentially less costly computations of eigenstructures and decomposition (e.g., Cholesky), among others.

Constant stable states — unit row sums: Many mesh processing and analysis tasks we are interested in are carried out in an iterative manner. One particular notion of interest for all iterative systems is that of a *stable state*, which represents an equilibrium of a system or a *fixed point*, i.e., $\mathbf{x}$ for which $H\mathbf{x} = \mathbf{x}$. A natural choice for a stable is the constant mesh, defined as any mesh $\mathbf{x}$ with $\mathbf{x}_i = \mathbf{x}_j$ for all i and j. It is not hard to show [25] that a system $\mathbf{y} = H\mathbf{x}$ has a constant stable state if and only if the sum of each row of the impulse response H is 1.

Energy conservation — unit column sums: Our interpretation of the impulse response in terms of energy distribution also raises the issue of energy conservation. Consider again the LMSP system $\mathbf{y} = H\mathbf{x}$. We know that the distribution of unit energy at a vertex k' is given by the impulse response $h(k; k')$ for fixed k', which, in turn, is just the k'-th column of the matrix H. Thus the LMSP system is said to be *energy-conserving* if H has unit column sum. An immediate implication is that the LMSP system preserves the *centroid*, or *DC value* in signal processing terms, of the mesh signal, if and only if its impulse response is energy-conserving.

Stochasticity and double stochasticity: A real matrix is said to be row-(column)-stochastic if it is nonnegative and has a constant row (column) sum of 1. Double stochasticity requires both row- and column-stochasticity. These matrices have been widely used in statistics and numerical analysis and there are numerous results [6, 14] to be utilized, e.g., a useful one for row-stochastic matrices is that their spectral radius is 1.

Variance diminishing property: We define the variance of a mesh $\mathbf{x}$ with n vertices by $\sigma^2(\mathbf{x}) = [\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})^2]/n$, where $\overline{\mathbf{x}}$ is the mean of $[\mathbf{x}_1, \ldots, \mathbf{x}_n]$. We say that the LMSP system $\mathbf{y} = H\mathbf{x}$ has the *variance diminishing property* if repeated application of H to $\mathbf{x}$ cannot increase the variance of $\mathbf{x}$. Using Birkhoff's characterization of doubly stochastic matrices [6], it can be shown that doubly stochastic matrices do have the variance diminishing property [25]. This however, does not hold in general for matrices that are only row- or column-stochastic [25].

2.3. Eigenvalue decomposition

Given a LMSP system defined by the operator H whose eigenvectors

$\mathbf{e}_1, \ldots, \mathbf{e}_n$ are linearly independent, any mesh $\mathbf{x}$ can be written as,

$$\mathbf{x} = \mathbf{e}_1 X_1 + \mathbf{e}_2 X_2 + \ldots + \mathbf{e}_n X_n = EX. \quad (2)$$

Typically, H is derived from the connectivity of the mesh. We call (2) the *eigenvalue decomposition* of $\mathbf{x}$ with respect to H, and X, having the same dimension as $\mathbf{x}$, the *ED-transform* of $\mathbf{x}$. The sequence of 3-D vectors $X_1, \ldots, X_n$ are referred to as the spectral coefficients, and E is the basis matrix whose columns are the eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$.

Convergence of LMSP: Basic eigenanalyses show that a necessary condition for the above LMSP system to converge, e.g., for $\lim_{k\to\infty} H^k\mathbf{x}$ to exist, is $\rho(H) \le 1$. Following the Perron-Frobenius Theorem [6], we can show [25] that if H is irreducible and doubly stochastic, then the limit given above is the centroid of the vertices of the mesh $\mathbf{x}$.

Note that it is often believed [23] that such a convergence result also holds for Laplacian smoothing, which repeatedly moves each mesh vertex towards the centroid of its one-ring neighbors. This is not true however, as Laplacian smoothing is defined by a *row-stochastic* operator, namely, the TL, and its limit is really a *valence-weighted* centroid of the vertices [25], i.e., $\lim_{k\to\infty} H^k\mathbf{x} = \sum_{i=1}^n t_i\mathbf{x}_i$, where $t_i = d_i / \sum_{j=1}^n d_j$.

Discrete Laplacians in the sense of Taubin [22]: It is well-known that the 1D DFT bases coincide with the orthonormal eigenvectors of the 1D uniform Laplacian [12]. Thus we are motivated to generalize the 1D Laplacian to irregular triangle meshes and use the corresponding ED-transform to carry out DFT-type mesh analysis. The TL has been chosen for this purpose [5, 13, 22, 26]. In general, Taubin [22] defines the (first-order) discrete Laplacian at a mesh vertex i to be

$$\Delta\mathbf{x}_i = \sum_{j\in N_1(i)} w_{ij}(\mathbf{x}_j - \mathbf{x}_i), \quad (3)$$

where w_{ij} are positive weights that sum up to 1. The corresponding Laplacian operator is given by $L = \mathcal{I} - W$, where $W_{ij} = w_{ij}$, while the Laplacian (measure) $\Delta\mathbf{x} = -L\mathbf{x}$. Next, we generalize this to allow for a unified treatment of a larger class of mesh Laplacian operators.

2.4. Smoothing matrices

The TL operator $\mathcal{T}$ is closely linked with the notion of smoothing in the sense of *low-pass filtering*. For instance, Laplacian smoothing uses an operator of the form $(\mathcal{I} - \lambda\mathcal{T})^N$, where $0 \le \lambda \le 1/2$ and N is the number of smoothing steps applied. Butterworth filters can also be defined and efficiently implemented for mesh smoothing [26]. It is quite natural then to ask what properties of a matrix H would be required for it to be "smoothing." The only reference we are aware of is due to Greville [4], who stipulates, rather abstractly, that a matrix H is "smoothing" if

1. H has $\sigma = 1$ as an eigenvalue, and
2. $H^\infty = \lim_{p\to\infty} H^p$ exists.

The rationale behind this definition is as follows. Let E_1 be the eigenspace corresponding to $\sigma = 1$. If $\mathbf{u} \in E_1$, then $H\mathbf{u} = \mathbf{u}$. One can view E_1 as the space of "*infinitely smooth vectors*", which cannot be smoothed further in the sense of Greville. If we smooth a given vector $\mathbf{v}$ using H, then in the limit we have $H^\infty\mathbf{v}$. Since $H(H^\infty\mathbf{v}) = H^\infty\mathbf{v}$, $H^\infty\mathbf{v} \in E_1$, i.e., it is "infinitely smooth," in an abstract sense.

As we can see, there is really no direct correlation between Greville's definition and our intuitive notion of de-noising or fairing of a geometric shape. However, independent of any topological relationship, e.g., connectivity, among the vertices of a mesh, the obvious choice for defining geometrically "infinitely smooth vectors" would be to insist that they all have zero variation, i.e., they are constant vectors. It follows that the smoothing operator must have unit row sum. Such an infinitely smooth mesh degenerates to a point, so the smoothing operator has to cause *shrinkage*.

Combining the row sum property and a necessary and sufficient condition for H to be smoothing in the sense of Greville, we restrict our definition of smoothing matrices and say that H is smoothing if:

1. H has constant unit row sum
2. The eigenvalue 1 of H is unique (has *multiplicity* 1) and if $\sigma \neq 1$ is another eigenvalue of H, then $|\sigma| < 1$.

Note that constant unit row sum implies that H does have an eigenvalue 1, which possesses a constant eigenvector.

2.5. Generalized shrinking mesh Laplacian (GSML)

We define an operator F to be a *generalized shrinking mesh Laplacian*, or *GSML*, if for some positive scalar k, the operator $H = \mathcal{I} - F/k$ is smoothing according to our definition above. Intuitively, for a mesh $\mathbf{x}$, the vector $F\mathbf{x} = k(\mathcal{I} - H)\mathbf{x}$ gives a measure of the deviation of the mesh $\mathbf{x}$ from a smoothed version of itself. Specifically, if we let $\mathbf{y}_i = [F\mathbf{x}]_i$ for vertex i, then the direction of $\mathbf{y}_i$ gives an estimate of the normal of $\mathbf{x}$ at i and its magnitude estimates the discrete curvature. The mesh may be smoothed by a *vertex flow*: $\mathbf{x}' = H\mathbf{x} = (\mathcal{I} - F/k)\mathbf{x}$. The following gives a sufficient condition for F to be a GSML:

1. F has constant zero row sum
2. F has real eigenvalues and the smallest one is zero and is unique.

It turns out that if m is the largest eigenvalue of F, then as long as the scalar k satisfies $k \geq m/2$, the $H = \mathcal{I} - F/k$ would be smoothing.

The GSML can be viewed as a generalization of the Laplacian operators in the sense of Taubin (3). To see this, note that the generalized mesh Laplacian (measure) at vertex i, given by $F = k(\mathcal{I} - H)$, is of the form

$$\Delta \mathbf{x}_i = (-F\mathbf{x})_i = k \sum_{j=1}^{n} H_{ij}(\mathbf{x}_j - \mathbf{x}_i), \tag{4}$$

where the weights H_{ij}'s may be negative and take on nonzero values outside the one-ring of i, but they still sum up to 1. The unit sum property is a consequence of the zero row sum property of F.

2.6. Frequencies, mesh fairness, and spectral processing

In general, the degree of oscillation of a signal, determined by its frequency contents, corresponds approximately to its fairness. Fairness is a measure of the total variation or curvature over a mesh surface, which can be defined by a GSML operator. This motivates the use of ED-transforms derived from a GSML to mimic the effect of discrete Fourier analysis.

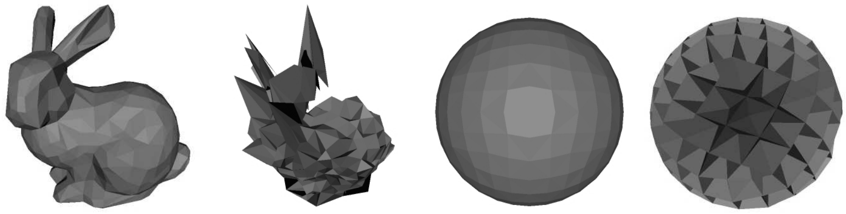
Consider an ED-transform (2) derived from a GSML operator F defined for mesh $\mathbf{x}$. Denote by $\mathbf{x}^{(m)}$ the projection of $\mathbf{x}$ onto the subspace spanned by the first $m \leq n$ eigenvectors of F. That is

$$\mathbf{x}^{(m)} = \mathbf{e}_1 X_1 + \ldots + \mathbf{e}_m X_m.$$

Roughly, the sequence $\mathbf{x}^{(n)}, \ldots, \mathbf{x}^{(1)}$ give progressively smoother and distorted (in the L_2 sense) versions of the original $\mathbf{x}$, ending at the (infinitely smooth) point $\mathbf{x}^{(1)}$. It is this analogy to signal transforms such as the DFT that has inspired many to develop a variety of signal processing style algorithms for irregular meshes [5, 13, 16, 17, 21, 22, 23, 26, 27]. For example, a JPEG-like compression scheme [13] for mesh geometry can truncate the ED-transform to $\{X_1, X_2, \ldots, X_m\}$ for $m \ll n$ while still retaining most of the mesh signal energy.

§3. Discrete combinatorial mesh Laplacian operators

The Kirchhoff operator (KL) $\mathcal{K}$ of a mesh is given by $\mathcal{K} = \mathcal{R}^{-1} - \mathcal{A}$, where we recall that $\mathcal{A}$ denotes the adjacency matrix of the mesh graph and $\mathcal{R}$ is a diagonal matrix of $1/d_i$'s, $i = 1, \ldots, n$, and d_i is the degree of vertex i. Thus, $\mathcal{K}_{ij} = d_i$, if $i = j$, $\mathcal{K}_{ij} = -1$ if $i \neq j$ and (i, j) is an edge, and $\mathcal{K}_{ij} = 0$ otherwise. Clearly, the KL is symmetric and it has constant zero row sum. By the Gerschgorin's Theorem [20], the eigenvalues of $\mathcal{K}$ are within $[0, 2d_{max}]$, where d_{max} is the maximum vertex degree. Also, it is well-known [2] that the number of zero eigenvalues of $\mathcal{K}$ is precisely the number of components in the mesh graph. Thus for a connected mesh, the smallest eigenvalue of the KL is 0 and it is unique. It follows that the KL is indeed a GSML.



Original bunny. GL compression. Original sphere. GL compression.

Fig. 1. Results of JPEG-like mesh compression using GL transforms (ED-transforms with respect to the GL operator) illustrates that they are unsuitable to use in DGP.

The normalized graph Laplacian (GL) $\mathcal{G} = \mathcal{I} - Q$, where $Q_{ij} = Q_{ji} = \mathcal{A}_{ij}/\sqrt{d_i d_j}$. Clearly, $\mathcal{G}$ is symmetric. It is also known [2] that the smallest eigenvalue of $\mathcal{G}$ is 0 and it has multiplicity 1. However, the GL is not a GSML since it does not have constant zero row sum. In general, the eigenvectors of $\mathcal{G}$ corresponding to the zero eigenvalue are not constant, and these supposedly "infinitely smooth vectors" are not really smooth.

Consequently, the sequence of eigensubspace projections (4), when derived from an ED-transform with respect to the GL, does not give progressively smoother versions of the original mesh, as shown in Figure 1, where we show the result of JPEG-like compression of a bunny and sphere model by truncating the GL spectrum. Thus we can conclude that even though the GL has proven to be quite useful in analyzing topological properties of graphs, it is unsuitable to use in DGP, such as for mesh smoothing or spectral mesh compression.

The Tutte Laplacian (TL) $\mathcal{T} = \mathcal{R}\mathcal{K} = \mathcal{I} - \mathcal{R}\mathcal{A} = \mathcal{I} - \mathcal{C}$, where $\mathcal{C}$ is the *centroid matrix*: $\mathcal{C}_{ij} = 1/d_i$ if and only if (i, j) is an edge. Although $\mathcal{C}$ has the same zero-nonzero structure as the adjacency matrix $\mathcal{A}$, it is not symmetric in general, and neither is it doubly stochastic, as such the variance diminishing property does not always hold [25].

It can be shown however that the eigenvalues of $\mathcal{T}$ are all real and lie in the interval $[0, 2]$. Also, $\mathcal{T}$ has constant zero row sum. It turns out that the TL and GL are similar and thus they share the same set of eigenvalues. To see this, note that since $\mathcal{C} = \mathcal{R}\mathcal{A}$, $\mathcal{R}^{-1/2}\mathcal{C}\mathcal{R}^{1/2} = \mathcal{R}^{1/2}\mathcal{A}\mathcal{R}^{1/2}$. Note that $Q = \mathcal{R}^{1/2}\mathcal{A}\mathcal{R}^{1/2} = \mathcal{I} - \mathcal{G}$, therefore

$$\mathcal{R}^{-1/2}\mathcal{T}\mathcal{R}^{1/2} = \mathcal{R}^{-1/2}(\mathcal{I} - \mathcal{C})\mathcal{R}^{1/2} = I - \mathcal{R}^{-1/2}\mathcal{C}\mathcal{R}^{1/2} = \mathcal{I} - \mathcal{Q} = \mathcal{G}.$$

Therefore, the zero eigenvalue of $\mathcal{T}$ also has multiplicity 1, and $\mathcal{T}$ is a GSML. In general however, the eigenvectors of $\mathcal{T}$ are not orthogonal, since $\mathcal{T}$ is not symmetric. Next, we propose a new operator which can be seen as a symmetric approximation of $\mathcal{T}$.

3.7. The symmetric quasi-Laplacian (SQL)

We define the *symmetric quasi-Laplacian*, or *SQL*, operator $\mathcal{S}$, of a mesh $\mathbf{x}$ as $\mathcal{S} = D - W$, where D is a *positive* diagonal matrix and W is a matrix of weights with $W_{ij} = 0$ if (i, j) is not an edge.

First let us suppose that i is an interior vertex. Then we set $D_{ii} = 1$. If j, k, $l \in N_1(i)$ are as shown in Figure 2(a), where k and l can be uniquely identified in a manifold mesh, then we set

$$W_{ij} = W_{ji} = \frac{1}{d_i} + \frac{1}{d_j} - \frac{1}{2}(\frac{1}{d_k} + \frac{1}{d_l}).$$

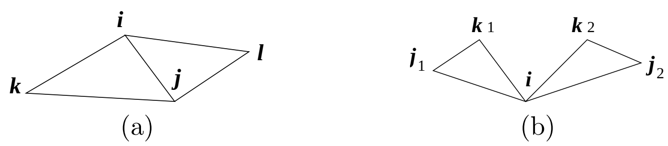


Fig. 2. Relevant vertices for defining the SQL operator. (a) For an interior vertex i. (b) For a boundary vertex i.

Now consider the case where i is a boundary vertex. Let its neighbors be as shown in Figure 2(b). Then for $m = 1, 2$, we set

$$W_{ij_m} = \frac{1}{d_i} + \frac{1}{d_{j_m}} - \frac{1}{d_{k_m}} \quad \text{and} \quad D_{ii} = 1 + \frac{1}{2}\sum_{m=1}^{2}\left(\frac{1}{d_{k_m}} - \frac{1}{d_{j_m}}\right).$$

Note that the TL and the SQL become identical over any region of a mesh with regular connectivity. We can view the SQL as derived from the TL with some perturbations added to achieve symmetry and desired row sum. To see this, consider an interior vertex i. Without loss of generality, let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be the neighbors of $\mathbf{x}_i$ in order. It is not hard to show that

$$[W\mathbf{x}]_i = \frac{1}{d_i}\sum_{j=1}^{m}\mathbf{x}_j + \sum_{j=1}^{m}\frac{1}{d_j}\left(\mathbf{x}_j - \frac{\mathbf{x}_{j-1} + \mathbf{x}_{j+1}}{2}\right) = [\mathcal{C}\mathbf{x}]_i + \sum_{j=1}^{m}\frac{\mathbf{v}_j}{d_j}.$$

We can see that the weighted average $[W\mathbf{x}]_i$ for the SQL operator is simply the centroid $[\mathcal{C}\mathbf{x}]_i$ perturbed by a weighted average of the vector displacements $\mathbf{v}_j$. The situation for boundary vertices is similar. Geometrically, the displacement $\mathbf{v}_j$ is the discrete uniform 2D Laplacian at the vertex j of the polygon formed by the neighbors of i. We expect the total perturbation $\mathbf{v}_1/d_1 + \mathbf{v}_2/d_2 + \ldots + \mathbf{v}_m/d_m$ to be small in most cases.

Symmetry, row sum, and negative weights: Unlike the TL, $\mathcal{S}$ is nonuniform. But it has constant zero row sum, as one can easily verify.

The main advantage of $\mathcal{S}$ over the TL is its symmetry, while its disadvantage is the possible *negative weights* in W — this happens when the degrees of nearby vertices differ significantly. In practice, we find such large degree discrepancies to occur rarely and the negative weights appear to have no noticeable negative effect on smoothing or other applications.

Observe that the effect of a negative weight is expected to be small as it would only occur around a vertex i with a large degree and the influence from other neighbors of i tends to "correct" the situation. But from a theoretical point of view, having a nonnegative W is highly desirable as it would ensure that $\mathcal{S}$ is a GSML and furthermore, all the nice properties listed in Section 2.2 for an LMSP system will be satisfied by the smoothing operator H corresponding to $\mathcal{S}$, where $H = \mathcal{I} - \mathcal{S}/k = (\mathcal{I} - D/k) + W/k$, $k \geq \max(D)$. In particular, H would be doubly stochastic.

An effective heuristic to eliminate negative weights is via edge swapping. That is, when $W_{ij} < 0$, implying that there is a relatively large discrepancy between the degrees of i, j and the degrees of k, l (see Figure 2(a)), then swapping (i, j) with (k, l) tends the correct the situation. However, new negative weights may be introduced as a result. A greedy approach, where edge swapping order is determined by the extent of the negative weights, has worked well in practice. Another approach to eliminate high-degree vertices is by splitting them as done in the "vertex-split" phase of progressive mesh construction [11]. But so far we cannot yet prove that either heuristic is guaranteed to eliminate all negative weights.

Positive semi-definiteness: In our subsequent analyses, let us assume that the weight matrix W is nonnegative. Then it is not hard to show that $\mathcal{S}$ is positive semi-definite, since for any vector $\mathbf{u}$, the quadratic form

$$\mathbf{u}^T \mathcal{S} \mathbf{u} = \sum_{(i,j) \text{is an edge}} W_{ij} (\mathbf{u}_i - \mathbf{u}_j)^2 \geq 0. \tag{5}$$

Note that a similar argument holds for the TL, GL, and KL as well.

Eigenvalue range: Since $\mathcal{S}$ is positive semi-definite, its eigenvalues are all nonnegative. If there are no boundary vertices, then by the Gershgorin's Theorem [20], an upper bound for the eigenvalues of $\mathcal{S}$ is 2. With boundary vertices, the upper bound could be slightly larger than 2.

GSML: To show that $\mathcal{S}$ is a GSML, it only remains to show that the zero eigenvalue of $\mathcal{S}$ has multiplicity 1. Let $\mathbf{u}$ be an eigenvector of $\mathcal{S}$ with corresponding eigenvalue σ. Then $\mathcal{S}\mathbf{u} = \sigma\mathbf{u}$ and it follows that $\mathbf{u}^T \mathcal{S} \mathbf{u} = \sigma ||\mathbf{u}||^2$. For the eigenvalue $\sigma = 0$, we have $\mathbf{u}^T \mathcal{S} \mathbf{u} = 0$. Examining (5), we see that since $W_{ij} \geq 0$, we must have $\mathbf{u}_i = \mathbf{u}_j$ for all edges (i, j). Therefore, as long as the mesh is connected, $\mathbf{u}$ must be a constant vector, implying that the zero eigenvalue is unique. Hence, if W is irreducible and nonnegative, then the SQL is a GSML.

3.8. Second-order symmetric Tutte Laplacian (SSTL)

One unsatisfying aspect of the SQL, at least from a theoretical point of view, is that for it to be a GSML, the negative weights in W need to be eliminated. We now propose another symmetric operator, SSTL, given by $\mathcal{J} = \mathcal{T}^T\mathcal{T}$, where $\mathcal{T}$ is the TL. As a direct consequence of the zero row sum property of $\mathcal{T}$, the SSTL also has constant zero row sum. It is also positive semi-definite. This can be verified trivially by noting that for any vector $\mathbf{v}$, $\mathbf{v}^T\mathcal{J}\mathbf{v} = \mathbf{v}^T\mathcal{T}^T\mathcal{T}\mathbf{v} = (\mathcal{T}\mathbf{v})^T(\mathcal{T}\mathbf{v}) = ||\mathcal{T}\mathbf{v}||^2 \geq 0$.

GSML: It remains to show that the zero eigenvalue of $\mathcal{J}$ has multiplicity 1. Let $\mathbf{e}$ be an eigenvector of $\mathcal{J}$ corresponding to the zero eigenvalue. Then $\mathcal{T}^T\mathcal{T}\mathbf{e} = 0$, but $e \neq 0$. It follows that $\mathbf{e}^T\mathcal{T}^T\mathcal{T}\mathbf{e} = 0$, so $||\mathcal{T}\mathbf{e}||^2 = 0$. Thus $\mathcal{T}\mathbf{e} = 0$ and $\mathbf{e}$ is an eigenvector of $\mathcal{T}$ corresponding to 0, so it has to be the constant one, as long as the mesh is connected. In this case, the multiplicity of the zero eigenvalue of $\mathcal{J}$ is one, and $\mathcal{J}$ is a GSML.

Approximation to $\mathcal{T}^2$: Unlike the TL, KL, and SQL, the SSTL extends weights to the second-order neighbors of a vertex. One may view it as a symmetric approximation of the second order TL $\mathcal{T}^2$. It has been shown [5] that second-order fairing operators tend to achieve a good balance between smoothing and having less shape distortion. Thus we expect SSTL to perform well in spectral mesh processing.

§4. GSMLs for digital geometry processing

4.9. Direct spectral geometry processing

We first compare the performances of the TL, KL, SQL, and SSTL operators where the corresponding ED-transform has to be constructed for spectral geometry processing. As we have explained before, the GL is unsuitable to use here. The mesh models used in our experiments include the well-known Stanford bunny, the horse mesh, the Igea, the Isis, etc., and patches generated from them. These models have been decimated down to having a couple of hundred vertices so that the ED-transforms can be computed in reasonable time in Matlab. We have also included some tessellated cubes, spheres, etc.

Computation of the ED-transforms: Karni and Gotsman [13] propose spectral compression of mesh geometry. The vertex positions of a mesh are transformed into the spectral domain via a TL transform. Spectral coefficients corresponding to the highest eigenvalues are neglected as they represent high-frequency information. Ohbuchi et al. [16, 17] embed a watermark bit stream into the leading spectral coefficients so that the resulting watermarked model can resist attacks such as smoothing. The biggest bottleneck of both of these schemes is the computation of the eigenvectors of the Laplacian operator. As this is prohibitively expensive

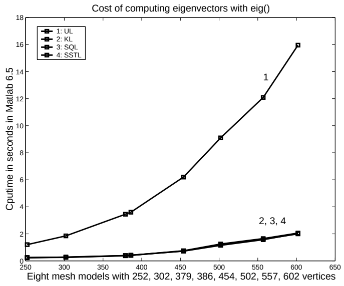
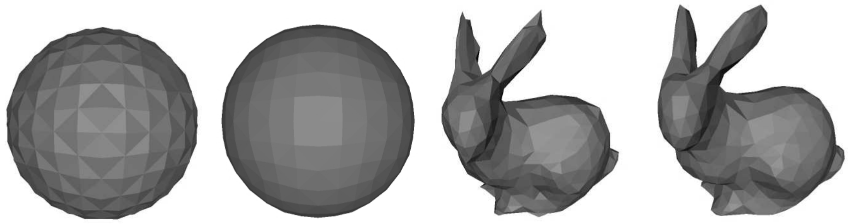


Fig. 3. Cost of computing the eigenvectors of different combinatorial Laplacians. UL is the most expensive to eigen-decompose. Note that although the GL is symmetric, its eigenvectors are as expensive to compute as those of the UL using Matlab's routines.

for very large matrices, the original mesh must first be partitioned into smaller pieces [10]. Subsequent processing is carried out in a piece-wise fashion. Typically, each piece contains several hundred vertices and the number of pieces could be several hundred for very large meshes.

We examine the cost of computing the eigenvectors of the various Laplacian operators. The eig() function from Matlab 6.5 is used, which invokes routines from the LAPACK. Plots in Figure 3 shows a rather consistent trend. Evidently, both the KL and our new symmetric GSMLs would save us considerable amount of time in computing the ED-transforms, needed by both the encoder and the decoder. It is also worth noting that since the eigenvectors of the TL are not orthogonal, computing the spectral coefficients requires us to solve a dense linear system $EX = \mathbf{x}$. The symmetric GSMLs allow us to obtain these coefficients via projection, $X_i = \mathbf{e}_i^T\mathbf{x}$. Although the relative saving in computation is quite significant, the absolute gain becomes insignificant in comparison to the time required for eigenvector computations. This does make a difference however when the basis vectors have been precomputed.

Energy compaction: We measure the *transform efficiency* of the ED-transforms, which corresponds to the ratio between energy concentrated in the first m spectral coefficients and the total signal energy. We find that although the TL is not orthogonal, it can still achieve excellent energy compaction. Overall, we have only noticed slight differences between the energy compaction capabilities of the four GSML operators. Typically, when a mesh model is fairly smooth, e.g., the bunny or sphere in Figure 1, the order, from better energy compaction to worse, tends to be SSTL, SQL, TL, and KL. This order is often reversed for models that contain sharp transitions, such as tessellation of a step function or cube.



KL compression. SSTL compression. KL compression. SSTL compression.

Fig. 4. JPEG-like compression (note that the sphere mesh has a 4-8 connectivity). Artifacts at low-degree vertices are quite evident for KL.

Shape distortion and artifacts: A key measure in evaluating the quality of spectral compression or watermarking schemes is shape distortion. We realize that the Metro tool [3] has become a standard error measure for mesh approximation. But since we expect mesh vertices to vary only slightly in our experiments, a rough and simple-to-compute measure similar to the one used by Karni and Gotsman [13] has been adopted. We compute the sum of: (1) squared differences between the *scale-dependent Laplacian* (as vectors) at corresponding vertices, note that this geometric Laplacian measure is tailored to the local mesh geometry, and (2) squared vertex displacements projected along the direction of the SDL (as an estimate of the normal) at the vertex.

Our experiments show that the SSTL and SQL consistently give better results. In most cases, the SSTL gives the best visual results, as we had anticipated. The KL is prone to various artifacts, especially at vertices with small degrees (3 or 4), as shown in Figure 4. This does not come as a surprise since the expected local variance at such vertices is inversely proportional to the vertex degrees [27].

4.10. Implicit mesh fairing

The GSML operators can also be applied to very large meshes to perform filtering. But instead of manipulating the ED-transforms directly, e.g., using an ideal filter, polynomial or rational filters can be implemented [5, 22, 26] — this corresponds to performing vertex averaging in the spatial domain. In polynomial filtering, a polynomial of a GSML operator is applied to a mesh repeatedly [22], while for a rational filter, a large sparse linear system involving the GSML has to be solved [5, 26].

In practice, visual results generated using the SQL and TL are almost indistinguishable for large meshes at similar levels of smoothing. The same holds for SSTL vs. the second-order TL $\mathcal{T}^2$. However, we have found the symmetric positive semi-definiteness of the SQL operator to be beneficial in solving sparse linear systems for implicit fairing. In this section, we report the performance of conjugate gradient (CG) for the symmetric SQL systems vs. bi-conjugate gradient (BiCG) for the non-symmetric TL systems. The convergence rates for higher-order filters, such as the SSTL, are quite low, and we propose alternative numerical techniques in our other work [26].

Implicit fairing, CG, and BiCG: Implicit fairing of a mesh using a GSML operator F requires the solution of a large sparse linear system

$$B(F)\mathbf{x} = (\mathcal{I} + \lambda F)\mathbf{x} = \mathbf{b}, \tag{6}$$

where $\mathbf{b}$ represents the original mesh and $\lambda = 1/\gamma_{PB} > 0$ is related to the pass-band frequency γ_{PB} of the rational low-pass filter $(1 + \lambda\gamma)^{-1}$ — larger γ implies a higher level of smoothing. Note that the choice of γ_{PB}, and thus λ, depends on the eigenvalue range of F.

The BiCG method is applicable to general, sparse, non-symmetric systems. The convergence of BiCG is often observed for a variety of problems, but few theoretical results are known about the rate and behavior of its convergence [1]. In certain situations, BiCG can exhibit “plateaus” during the iterations, with the residual norm stagnating at some constant value for many iterations before decreasing again. The convergence of BiCG may also break down due to division by zero [1]. In many respects, the CG solver is more stable numerically and its convergence behavior is often more regular than that of the BiCG method [1]. However, CG is only applicable to symmetric positive definite systems.

SQL systems vs. TL systems: Assume that the negative weights in $\mathcal{S}$ has been eliminated, thus $\mathcal{S}$ becomes positive semi-definite. It is not hard to see that then the coefficient matrix of the SQL system $B(\mathcal{S}) = (\mathcal{I} + \lambda\mathcal{S})$ is symmetric positive definite and thus CG applies. For the TL system, which is positive definite but non-symmetric in general, we employ the BiCG method. Note that the TL system may be made symmetric positive definite by multiplying both sides of (6) by the diagonal matrix $\mathcal{R}^{-1}$ of vertex degrees. The resulting system becomes $(\mathcal{R}^{-1} + \lambda\mathcal{K})\mathbf{x} = \mathcal{R}^{-1}\mathbf{b}$, where $\mathcal{K} = \mathcal{R}^{-1}\mathcal{T}$ is the KL. Then CG can be applied to this system.

We have tested the CG and BiCG solvers on some large real-world mesh models. In Table 1, we show the execution times and number of iterations required for seven meshes: cow (3K vertices), horse (20K), cube (25K), Stanford bunny (36K), a teeth model (116K), the Igea (134K), and the Isis (187K). Note that “CG (TL)” refers to the symmetric system converted from the TL, as described above.

We have used a weak stopping criterion: $||\mathbf{x}_i^{(n)} - \mathbf{x}_i^{(n-1)}|| < \epsilon||\mathbf{x}_i^{(n-1)}||$ for all i, where $\mathbf{x}_i^{(n)}$ is the n-th iterate. That is, we stop the iteration when no apparent progress is being made. The error tolerance of $\epsilon = 10^{-5}$ is used in our experiments. We have found that the strong stopping criterion $||B\mathbf{x}_i^{(n)} - \mathbf{b}_i|| < \epsilon||\mathbf{b}_i||$ often results in many more iterations with little improvement in mesh quality; this stopping criterion is also more expensive to test. The weak stopping criterion applied appears to be quite effective for CG, since for all our test cases, good convergence results are obtained when the iterations are stopped. For BiCG however, we have experienced the "plateau" problem described earlier, as we explain below.

As shown in Table 1, CG and BiCG require about the same number of iterations, while the real execution time for CG is about $50\% - 60\%$ of that of BiCG. This is because the per-iteration cost of BiCG is about twice as much as that of CG [1]. We have neglected the execution times for $\lambda = 100$ to save space. Note that the symmetric system converted from a TL system has a slower convergence. The use of a diagonal preconditioner yields the same results as BiCG on the original TL system.

Mesh	**CG**	**CG (TL)**	**BiCG**	**CG**	**BiCG**
Cow	0.56 [15]	1.38 [37]	0.99 [15]	[24]	[30]
Horse	0.97 [15]	2.21 [34]	1.81 [15]	[28]	[30]
Cube	1.04 [12]	1.78 [21]	1.33 [9]	[17]	[17]
Bunny	1.67 [14]	3.66 [31]	3.75 [18]	[24]	[28]
Teeth	4.60 [12]	12.65 [31]	9.94 [14]	[21]	[24]
Igea	4.91 [11]	13.07 [29]	9.18 [12]	[16]	**[14]**
Isis	9.03 [15]	23.56 [33]	11.27 [10]	[22]	**[18]**
	$\lambda = 30$	$\lambda = 30$	$\lambda = 30$	$\lambda = 100$	$\lambda = 100$

Tab. 1. Execution time in seconds [iteration count].

The "plateau" problem shows up for the Igea and Isis models when $\lambda = 100$, as highlighted in Table 1. In these cases, the models obtained after the iterations are stopped are not close to the true solution at all. Note that CG has not suffered from this problem in our experiments.

§5. Summary and future work

We have conducted a careful study of various discrete combinatorial Laplacian operators, their matrix-theoretic properties, and several theoretical and practical implications. The notion of smoothing matrices and GSMLs enables us to provide a unified treatment. We propose two new symmetric operators, the SSTL and the SQL, for eigenvalue decomposition. They are shown to provide better alternatives for digital geometry processing. In particular, the SSTL achieves the best quality in transform coding without any comprise in speed. This is followed by the SQL. On the other hand, the KL is prone to various artifacts, the ED-transforms of the TL are much harder to compute, and the GL is not even a GSML. Finally, we demonstrate through our experiments the numerical advantages of replacing the TL by the SQL for implicit mesh fairing. However, much work still remains to be done to study these important linear systems, especially for higher-order filters.

From a theoretical point of view, there is still a great deal we do not understand about the ED-transforms derived from these Laplacian operators. Questions related to the robustness of the spectral coefficients against alterations in mesh connectivity and precise characterization of the vibration patterns of the eigenvectors remain to be answered. On the practical side, we plan to test more sophisticated eigensolvers, such as the Arnoldi method, on our Laplacians operators. We are also interested in applying the type of analyses given here to geometric Laplacian operators.

§6. References

1. Barrett, B., et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods,* SIAM, 1994.
2. Chung, F. R. K., *Spectral Graph Theory,* CBMS Regional Conference Series in Mathematics, AMS, 1997.
3. Cignoni, P., C., Rochini, and R., Scopigno, Metro: measuring error on simplified surfaces, *Computer Graphics Forum,* **17**(2), 167–174, 1998.
4. Davis, P., *Circulant Matrices,* John Wiley & Sons, 1979.
5. Desbrun, M., M. Meyer, P. Schröder, and A. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, *Proceedings of SIGGRAPH,* 317–324, 1999.
6. Fiedler, M., *Special Matrices and Their Applications in Numerical Mathematics*, Martinus Nijhoff Publishers, 1986.
7. Floater, M. S., Parameterization and smooth approximation of surface triangulations, *Computer Aided Geometric Design,* **14**, 231–250, 1997.
8. Floater, M. S., Mean value coordinates, *Computer Aided Geometric Design,* **20**, 19–27, 2003.
9. Gotsman, C., X. Gu, and A. Sheffer, Fundamentals of spherical parameterization for 3D meshes, *ACM Transactions on Graphics,* **22**(3), 358–363, 2003.
10. Hendrickson, B. and R. Leland, A multilevel algorithm for partitioning graphs, *Supercomputing 1995*.
11. Hoppe, H., Progressive meshes, *ACM SIGGRAPH 1996,* 99-108.
12. Jain, A., *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

13. Karni, Z. and C. Gotsman, Spectral compression of mesh geometry, *Proceedings of SIGGRAPH,* 279–286, 2000.

14. Minc, H., *Nonnegative Matrices,* Wiley, 1998.

15. Meyer, M., Desbrun, M., Schröder, P., and A. Barr, Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, *Visualization and Mathematics III,* H-C. Hege and K. Polthier, editors, 35–57, 2003.

16. Ohbuchi, R., S. Takahashi, T. Miyazawa, and A. Mukaiyama, Watermarking 3-D polygonal meshes in the spectral domain, *Proceedings of Graphics Interface,* 9–18, 2001.

17. Ohbuchi, R., H. Ueda, and S. Endoh, Watermarking 2D vector maps in the mesh spectral domain, *Proceedings of Shape Modeling International,* 216–225, 2003.

18. Pinkall, U. and K. Polthier, Computing Discrete Minimal Surfaces and Their Conjugates, *Experimental Mathematics,* W. Kühnel, editor, 15–36, 1993.

19. Pratt, W. K., *Digital Image Processing,* Second Edition, Wiley, 1991.

20. Saad, Y., *Iterative Methods for Sparse Linear Systems,* PWS Publishing Company, 1996.

21. Sorkine, O., D. Cohen-Or, and S. Toledo, High-pass quantization for mesh encoding, *Symposium on Geometry Processing,* 42–51, 2003.

22. Taubin, G., A signal processing approach to fair surface design, *Proceedings of SIGGRAPH,* 351–358, 1995.

23. Taubin, G., Geometric signal processing on polygonal meshes, *STAR, Eurographics 2000*.

24. Tutte, W. T., How to draw a graph, *Proc. London Math. Soc.*, **13**, 743–768, 1963.

25. Zhang, H., *Signal Processing and Eigenvalue Decomposition of Polygonal Meshes and Applications,* Ph.D. dissertation, Dept. of Computer Science, University of Toronto, March 2003.

26. Zhang, H. and E. Fiume, Butterworth filtering and implicit fairing of irregular meshes," *Proceedings of Pacific Graphics,* 502- 506, 2003.

27. Zhang, H. and H. C. Blok, Optimal Mesh Signal Transforms, *Proceedings of Geometric Modeling and Processing, Theory and Applications*, 373-379, 2004.

Hao Zhang, School of Computing Science
Simon Fraser University, Burnaby, BC Canada
haoz@cs.sfu.ca