

Spectral Conformal Parameterization

Patrick Mullen, Yiying Tong, Pierre Alliez, Mathieu Desbrun

▶ **To cite this version:**

Patrick Mullen, Yiying Tong, Pierre Alliez, Mathieu Desbrun. Spectral Conformal Parameterization. Computer Graphics Forum, 2008, Eurographics Symposium on Geometry Processing, 27 (5), pp.1487–1494. inria-00334477

HAL Id: inria-00334477
https://inria.hal.science/inria-00334477

Submitted on 5 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Eurographics Symposium on Geometry Processing 2008
Pierre Alliez and Szymon Rusinkiewicz
(Guest Editors)

Volume 27 (2008), Number 5

Spectral Conformal Parameterization

Patrick Mullen[1] Yiying Tong[2] Pierre Alliez[3] Mathieu Desbrun[1]

[1]Caltech [2]Michigan State University [3]INRIA Sophia-Antipolis

Abstract

We present a spectral approach to automatically and efficiently obtain discrete free-boundary conformal parameterizations of triangle mesh patches, without the common artifacts due to positional constraints on vertices and without undue bias introduced by sampling irregularity. High-quality parameterizations are computed through a constrained minimization of a discrete weighted conformal energy by finding the largest eigenvalue/eigenvector of a generalized eigenvalue problem involving sparse, symmetric matrices. We demonstrate that this novel and robust approach improves on previous linear techniques both quantitatively and qualitatively.

1 Introduction

Due to its central importance in geometry processing, the subject of mesh parameterization has been researched for a number of years. "Parameterizing" a triangle mesh traditionally means computing a correspondence between a discrete, triangulated surface patch (possibly with holes) and a homeomorphic planar mesh through a piecewise linear map. Finding this piecewise linear mapping amounts to assigning each mesh node a pair of coordinates (u,v) referring to its position in the planar region. Such (ideally one-to-one) mappings provide a flat parametric space, allowing complex mesh processing operations such as surface fitting and remeshing to be performed directly on a flat domain rather than on the curved, original surface patch. Planar coordinates are also particularly useful to dramatically enhance the visual richness of a 3D surface through texture mapping, both for overly simplified character meshes in game engines and for incredibly detailed surfaces in computer-generated feature films. Consequently, fast methods generating less distortion than current tools are still in high demand.

1.1 Previous Work

Since we are trying to flatten an intrinsically non-flat triangulated surface from 3D down to 2D, there is in general no perfect way to perform this flattening without introducing some form of distortion. Due to the ubiquitous need for local and global parameterizations in geometry processing applications, a number of distortion measures have been defined, and various numerical approaches have been proposed to efficiently provide (u,v) parameter values to meshes minimizing one of these measures. Although thorough surveys are available [FH05, SPR06, HLS07], we go over the methods most related to our contribution to motivate our approach.

Existing methods generally fall into one of three categories: linear, non-linear, and hybrid methods. Non-linear methods, based on discrete or differential-geometric non-linear distortion measures [HG00, SdS01, SLMB05, ZKK02, SGSH02, ZSGS04, KSS06, JKG07, SSP08], offer strong theoretical guarantees on the resulting maps, at the cost of a generally higher computational effort. Recently, hybrid techniques linearizing these non-linear measures have been proposed to emulate non-linear methods at the cost of only a few linear solves [ZLS07, BCGB08], potentially saving execution time—although the trade-off between the effect of the linearization and the computational saving may be very application-dependent.

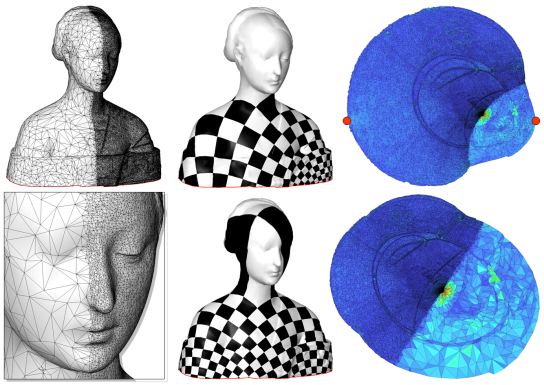


Figure 1: ***Sforza****. On this mesh (50K vertices) with varying sampling rates (left), previous linear methods (top, LSCM/DCP) fail to capture the symmetry of the mesh in the parameterization (solved in 4s). In contrast, our spectral approach (bottom) automatically computes a low-distortion conformal map (solved in 5.2s).*

Linear methods, instead, require solving a *single* linear system. These methods date back to the use of the connectivity-based graph Laplacian in Tutte's embedding theorem [Tut63] and its generalizations [GGT06]. A significant improvement, involving a linear system not only dependent on the connectivity but also on the geometry of the original mesh (through the "cotangent weights"), was introduced

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.
Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

in [EDD*95] (see [HAT*00] as well). Methods arising from strictly convex combinations were also proposed [Flo97] for increased robustness on arbitrary meshes. All these initial linear methods required pinning down the coordinates of the whole patch boundary to a given (generally convex) shape in the parameter plane in order to render the linear system full ranked. Extensions to free boundaries (requiring only two vertices to be fixed) were later proposed using two very different derivations under the respective names of *least square conformal maps (LSCM)* [LPRM02] and *discrete conformal parameterizations (DCP)* [DMA02]. Those were shown to achieve significantly lower angle distortion than previous results. Since then, alternatives offering either less flexibility in the boundary [LKL02] or requiring more than just one linear solve [ZRS05, ZLS07] have appeared. Although we will not consider altering the topology of the mesh in this paper, note that much reduced distortion can be obtained if cuts [GGH02,LPRM02,SH02,SCOGL02] or cone singularities [TACSD06,BCGB08] are judiciously placed in the original mesh before parameterization.

1.2 Contributions

We propose a novel linear-algebra-based conformal parameterization technique to parameterize triangle mesh patches. Unlike previous free-boundary linear methods we do not require point constraints to be given to the solver, thus drastically reducing distortion at the relatively small additional cost of finding an eigenvector rather than solving a linear system directly. While Laplacian eigenvectors have been proposed as a constraint-free approach to least-distorted maps in the context of manifold learning [BN03] and graph drawing [Kor05], we demonstrate that a better conformal parameterization can be found through a *generalized* eigenvalue problem as it minimizes a weighted conformal energy mostly insensitive to sampling irregularity of the original mesh. We discuss the similarities and differences between our approach and previous work (in particular, the work of [LPRM02] and [DMA02]), and demonstrate numerical advantages of our spectral method on small and large meshes alike.

1.3 Notations

Throughout this paper, the number of vertices will be denoted V, while E will refer to the number of edges. V_b will denote the number of vertices on the boundary of the mesh patch (i.e., not including other possible internal boundaries).

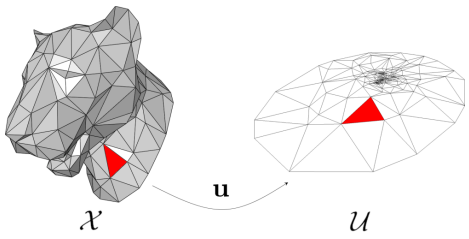


Figure 2: ***Parameterization.*** *a piecewise-linear map* $\mathbf{u}$ *creates a correspondence between a 3D mesh* $\mathcal{X}$ *with boundary and a 2D mesh* $\mathcal{U}$ *of same topology, mapping each triangle from* $\mathbb{R}^3$ *to* $\mathbb{R}^2$.

We will also denote $\mathbf{x}_i = (x_i, y_i, z_i)$ the 3D position of the i-th node of a mesh patch $\mathcal{X}$, and $\mathbf{u}_i = (u_i, v_i)$ the 2D position (parameter value) of the corresponding node in the 2D mesh $\mathcal{U}$. The vector $\mathbf{u}$ will thus denote the column vector $(u_1, v_1, u_2, v_2, \ldots, u_V, v_V)^T$. Finally, we will use the notation e_{ij} to denote the edge in $\mathcal{U}$ between vertex $\mathbf{u}_i$ and $\mathbf{u}_j$.

2 Background on Discrete Conformal Maps

Before presenting our contribution, we point out some of the important results known on linear methods to conformal maps, including the treatment of boundary conditions that are often glanced over in parameterization surveys.

2.1 Relevant Continuous Definitions

A number of differential geometric notions are quite relevant to the problem of conformal parameterizations, as we briefly review next. For simplicity, we will restrict our explanations to disk-like surface patches, i.e., simply connected 2-manifolds with boundary.

Dirichlet Energy. For a smooth map $\mathbf{u}$ from a differential surface patch $\mathcal{X}$ to its image $\mathcal{U}$, its Dirichlet energy is defined as the $\mathcal{L}^2$ norm of its gradient, i.e.:

$$E_D = \frac{1}{2}\int_{\mathcal{X}} |\nabla \mathbf{u}|^2 dA. \qquad (1)$$

This energy is, by definition, always positive, and measures a form of distortion that the map $\mathbf{u}$ creates. Additionally, if the area of the image of the map $\mathbf{u}$ is denoted by $\mathcal{A}(\mathbf{u}) = \int_{\mathcal{X}} \det(\mathbf{u}) dA$, it is well known that for any map $\mathbf{u}$, the Dirichlet energy is bounded from below by the map area: $E_D \geq \mathcal{A}(\mathbf{u})$ [PP93].

Conformality and Harmonicity A map that minimizes the Dirichlet energy given fixed boundary conditions is *harmonic*, as it satisfies $\Delta \mathbf{u} = 0$ (where Δ is the Laplace-Beltrami operator, i.e., the Laplacian on the original surface $\mathcal{X}$). If the minimal possible value of the Dirichlet energy (i.e., the area of the image of the map) is attained, the resulting map is *conformal* (also referred to as angle-preserving), as one can show that the angle between any two tangent vectors at a point on the original surface $\mathcal{X}$ is mapped to the exact same angle between the two mapped tangent vectors on the image $\mathcal{U}$. It is therefore convenient to define another energy, called the *conformal energy* E_C as:

$$E_C(\mathbf{u}) = E_D(\mathbf{u}) - \mathcal{A}(\mathbf{u}), \qquad (2)$$

so that now a map $\mathbf{u}$ is conformal iff this energy is zero. In this continuous setting, the Riemann mapping theorem states that for any given shape of the boundary of the image $\mathcal{U}$, a conformal parameterization always exists. However, if we also fix the map of the boundary of the domain (i.e., if we fix the map $\mathbf{u}$ along the boundary), conformality is often impossible: minimizing the conformal energy will reduce the angle distortion of the map as much as possible given this boundary map, but will not suppress it completely.

2.2 Discrete Setting

When dealing with triangle meshes instead of smooth manifolds, the previous definitions need to be adapted. In fact, only the map $\mathbf{u}$ itself needs to be discretized: the rest of the

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

definitions can be used as is, resulting in particularly simple expressions for piecewise-linear maps.

Discretization of the Map As computer graphics applications often use triangle meshes, an obvious discretization of the smooth map $\mathbf{u}$ is to now consider it *piecewise-linear*, i.e., mapping each triangle on $\mathcal{X}$ to a triangle on $\mathcal{U}$ linearly (see Fig. 2). It was shown in [PP93] that the Dirichlet energy of such a piecewise linear map can be expressed in a particularly simple expression:

$$E_{\mathrm{D}}(\mathbf{u}) = \sum_{e_{ij}} \frac{1}{4}\big(\cot(\theta_{ij}) + \cot(\theta_{ji})\big)(\mathbf{u}_i - \mathbf{u}_j)^2, \qquad (3)$$

where θ_{ij} and θ_{ji} are the two angles opposite to the edge linking $\mathbf{x}_i$ and $\mathbf{x}_j$ on mesh $\mathcal{X}$. Notice that this expression is a quadratic form in the coordinates of $\mathcal{U}$, with coefficients computed on the original 3D mesh. Furthermore, the Dirichlet energy can be expressed in matrix form as:

$$E_{\mathrm{D}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^t L_{\mathrm{D}}\mathbf{u},$$

where $L_{\mathbf{D}}$ is a $2V \times 2V$ sparse, symmetric matrix containing only the cotangent coefficients computed on $\mathcal{X}$. Since a map minimizing the Dirichlet energy should have its Laplacian being zero, the matrix L_{D} is sometimes interpreted as a *discrete Laplacian*. It was later noticed that this expression can be also derived through discrete exterior calculus [DKT05]: the Laplacian matrix L_{D} is formulated as $d^t \star d$, where d is the edge-vertex adjacency matrix, and $\star$ is an $E \times E$ diagonal matrix called the *discrete Hodge star*, whose non-zero entries are the cotan coefficients discussed above (altering these coefficients allows for the design of quasi-conformal maps, as we will mention in Section 4.3).

The area of the parameterization can also be expressed simply: summing up the area $\mathcal{A}_{\mathrm{T}}$ of each mapped triangle T in $\mathcal{U}$ (with $\mathcal{A}_{\mathrm{T}}(\mathbf{u}) = \sum_{e_{ij}\in T} \frac{1}{2}(u_i v_j - u_j v_i)$), the total area is computed using only the coordinates of the boundary vertices (as contributions from internal edges cancel out) through:

$$\mathcal{A}(\mathbf{u}) = \sum_{e_{ij}\in\partial\mathcal{U}} \frac{1}{2}(u_i v_j - u_j v_i),$$

where e_{ij} is along the boundary $\partial\mathcal{U}$. Therefore, we can define a matrix A such that $\mathcal{A}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^t A\mathbf{u}$. This $2V \times 2V$ matrix is symmetric and extremely sparse, as only the entries corresponding to boundary vertices are non zero. Consequently, the conformal energy $E_{\mathrm{C}} = E_{\mathrm{D}} - \mathcal{A}$ can be expressed as a quadratic form:

$$E_{\mathrm{C}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^t L_{\mathrm{C}}\mathbf{u},$$

where $L_{\mathrm{C}} = L_{\mathrm{D}} - A$ is a sparse, symmetric matrix that was explicitly given in [DMA02](Eq. 7).

Minimization of Conformal Energy A natural way to get a *discrete* notion of a conformal map is presented in [DMA02]: the piecewise-linear mapping is called discrete conformal when it minimizes the discrete quadratic energy $E_{\mathrm{C}}(\mathbf{u})$—thus only requiring a linear system to be solved. LSCM [LPRM02], instead, asks that the gradient of the u coordinate and the gradient of the v coordinate be as orthogonal as possible per triangle in the parameterization. In other words, a piecewise-linear map is least-square conformal when the energy $E_{\mathrm{LSCM}}(\mathbf{u}) = \int_{\mathcal{X}} \frac{1}{2}|\nabla u^{\perp} - \nabla v|^2 dA$ is minimized, where $^{\perp}$ denotes a counterclockwise 90° rotation in $\mathcal{X}$. However, it is easy to see that:

$$\begin{aligned} E_{\mathrm{LSCM}}(\mathbf{u}) &= \int_{\mathcal{X}} \frac{1}{2}(\nabla u^{\perp}\cdot\nabla u^{\perp} + \nabla v\cdot\nabla v - 2\nabla u^{\perp}\cdot\nabla v)\, dA \\ &= \int_{\mathcal{X}} \frac{1}{2}(\nabla u\cdot\nabla u + \nabla v\cdot\nabla v - 2\nabla u\times\nabla v)\, dA \\ &= E_D(\mathbf{u}) - \mathcal{A}(\mathbf{u}). \end{aligned}$$

(The same derivation can be expressed almost verbatim using discrete differential forms, by substituting d for ∇, and the Hodge star $\star$ for $(.)^{\perp}$; this common discrete-calculus view nicely ties up LSCM/DCP with the other discrete conformal methods based on linear algebra [GY03, TACSD06].) Consequently, and as noted in [CSD02], LSCM and DCP are identical, down to the way they constrain two vertices to guarantee a unique solution—notice that both resulting matrices are symmetric if the constraints are moved to the right-hand side of the linear system. The only differences that can be found in practice are due to different numerical approximations of the cotangent weights, and according to our tests, the resulting maps are visually undistinguishable. Due to its simplicity, LCSM/DCP has since been used in many 3D geometric tools (Blender3D, CGAL, and plugins for Autodesk 3ds Max® and Maya®) as a quick way to provide a good parameterization for surface patches.

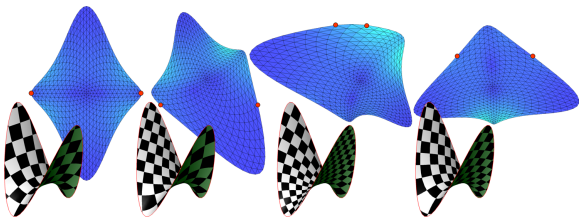


Figure 3: ***Saddle mesh****. LSCM/DCP with different boundary vertices chosen as constraints. The resulting parameterizations (*top*) highly depend on the constraints, and impact texture mapping.*

3 Spectral Parameterizations

We now present our spectral approach to discrete conformal parameterization through a sparse, symmetric generalized eigenvalue problem—a fast numerical procedure to solve a weakly-constrained minimization of the conformal energy.

3.1 Motivation

The conventional approach to find a free-boundary parameterization from the quadratic form L_{C} of the conformal energy described above is to fix two (u,v) coordinates (thus pinning down two vertices in the parameter plane in order to suppress rank deficiency and avoid the degenerate solution $\mathbf{u} = \text{constant}$), and solve the linear system $L_{\mathrm{C}}\mathbf{u} = \mathbf{0}$: this sets the translation, rotation, and scale of the solution, and leads to what can be called a *discrete* or *least-square* conformal map [DMA02, LPRM02]. However, the choice of vertices to pin down drastically influences the results as illustrated

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

in Fig. 3, a property specific to the discrete setting since, in the continuous case, a conformal map exists for any boundary shape homeomorphic to a circle. It was independently found by the authors of LSCM and DCP that picking two boundary vertices the *farthest* from each other (in terms of their geodesic distance) seems to give good results in general. However, this particular choice is far from being devoid of artifacts on non-trivial models: global distortion can ensue and a (sometimes significant) degradation in conformality is often witnessed around the pinned vertices as shown in Figures 3, 4, and 6 (note that this dependency on pinned points has sparked the development of hierarchical methods, see [RL03]). This issue only worsens if more vertices are fixed (unless carefully chosen by hand), as it adds more point constraints to the minimization. As we now show, a lesser-distorted parameterization can be efficiently obtained via spectral analysis of the Laplacian-like matrix L_C.

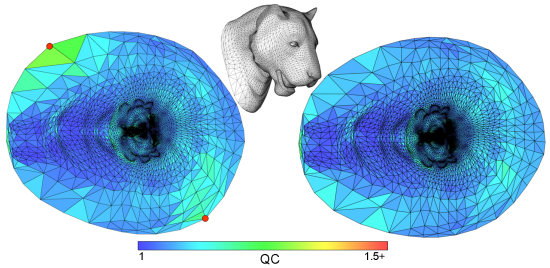


Figure 4: ***Lion-head****. Left: LSCM/DCP with two constrained vertices shown in red. Note the conformality distortion near the constraints, despite the nearly uniform and circular boundary. Right: Spectral conformal parameterization, with no visual bias. The color scheme indicates quasi-conformality (QC) distortion per triangle.*

3.2 Fiedler Vector of L_C

A solution to avoid adding *vertex constraints* to the rank deficient linear system $L_C\mathbf{u} = 0$ is through recourse to spectral theory. Digital geometry processing is no stranger to spectral graph theory [Got03, ZvKD07, KSO04, Ló6, ACSTD07]: graph theoretical results have even been used for parameterization purposes [GGS03, ZSGS04]—albeit in the context of *non-linear* methods. Particularly relevant to our approach, a well-known graph-theoretical result is a variational characterization of the eigenvectors of sparse symmetric positive semi-definite Laplacian matrices: the eigenvector $\mathbf{u}^*$ associated to the *first non-zero eigenvalue* of an $n \times n$ Laplacian matrix L (i.e., the vector satisfying $L\mathbf{u}^* = \lambda\mathbf{u}^*$, λ being the smallest non-zero eigenvalue) of rank $n-k$ is the *solution to a constrained minimization* of a simple quadratic form:

$$\mathbf{u}^* = \underset{\substack{\mathbf{u} \\ \mathbf{u}^t\mathbf{e}=\mathbf{0} \\ \mathbf{u}^t\mathbf{u}=1}}{\operatorname{argmin}} \ \mathbf{u}^t L\,\mathbf{u},$$

where $\mathbf{e}$ is an $n \times k$ matrix whose columns span the kernel of L. This eigenvector $\mathbf{u}^*$ is called the *Fiedler vector of L* as an homage to M. Fiedler, who first stressed the importance of its variational nature [Fie73]. In our context of mesh parameterization, the Fiedler vector of L_C has a simple, intuitive interpretation: it is the closest solution to the linear system defined in LSCM/DCP under the *constraint* that the barycenter of the solution must be at $\mathbf{0}$ ($\mathbf{u}^{*t}\mathbf{e} = \mathbf{0}$) *and* its moment of inertia (i.e., sum of squared distances to the barycenter) must be unit ($\mathbf{u}^{*t}\mathbf{u}^* = 1$). This spectral approach thus gives us a parameterization *without singling out two vertices*; the constraints are, in effect, spread evenly throughout the mesh.

While exploiting Fiedler vectors of Laplacian-like matrices has been successfully proposed in the context of manifold learning [RS00], graph drawing [Kor05], dimensionality reduction [BN03], image segmentation [SM00], and spherical parameterization [GGS03], we argue that this is *not* a good alternative to LSCM/DCP. Indeed, the Fiedler vector minimizes the so-called Rayleigh quotient:

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} \left| \frac{\mathbf{u}^t L_C \mathbf{u}}{\mathbf{u}^t\mathbf{u}} \right|, \qquad (4)$$

thus striking a balance between discrete conformality (to best minimize the numerator) and $\mathcal{L}^2$-distance to the origin (to best maximize denominator). The latter $\mathcal{L}^2$ norm, tantamount to the energy of springs between *each* vertex and the origin, is however not appropriate in our context: minimizing the Rayleigh quotient may come at the price of severely sacrificing conformality locally, as illustrated in Fig. 5.

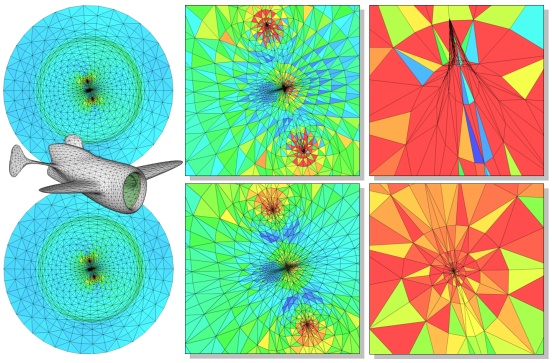


Figure 5: ***Airplane****. A conformal parameterization of the airplane model (where the boundary is in the front) results in a mesh with wildly varying triangle sizes (left); while the Fiedler vector of L_C (*top*, computed at machine accuracy) creates flips and tent-like distortion on the back stabilizers (upper right), our robust spectral parameterization (*bottom*) produces no flip (lower right).*

3.3 Spectral Conformal Map

Instead of directly using an eigenvector of the conformal-based Laplacian matrix, we propose to define the discrete spectral conformal parameterization $\mathbf{u}^*$ as a *generalized eigenvector* satisfying:

$$L_C\mathbf{u} = \lambda B\mathbf{u}, \qquad (5)$$

where L_C is the matrix associated to the conformal quadratic form defined in Section 2.2, and B is a $2V \times 2V$ diagonal matrix with 1 at each diagonal element corresponding to *boundary* vertices (not including any of internal boundaries of the patch) and 0 everywhere else. As L_C is (theoretically) positive semi-definite, and both matrices are symmetric, the generalized eigenvalues and eigenvectors are real. This new

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

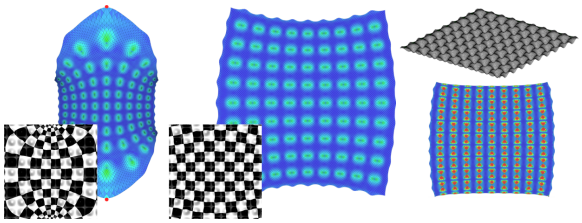


Figure 6: ***Egg crate****. While point constraints in LSCM/DCP can introduce significant distortion (left), our spectral approach (middle) better respects the original symmetry with lower distortion (solved in 3.2s, QC = 1.0629), producing a result similar to Circle Patterns [KSS06] (right) in a tenth of the time (QC= 1.1518).*

eigenvalue problem corresponds to a slightly-altered constrained minimization:

$$\mathbf{u}^* = \underset{\substack{\mathbf{u} \\ \mathbf{u}^t B\,\mathbf{e}=\mathbf{0} \\ \mathbf{u}^t B\,\mathbf{u}=1}}{\operatorname{argmin}} \; \mathbf{u}^t L_C\,\mathbf{u}, \tag{6}$$

where the $2V \times 2$ matrix $\mathbf{e}$ is such that $\mathbf{e}_{i,1}$ (resp., $\mathbf{e}_{i,2}$) is 1 for each u-coordinate (resp., v-coordinate) and 0 otherwise. Equivalently, Eq. (5) corresponds to a modified Rayleigh quotient $\mathbf{u}^t L_C\mathbf{u}/\mathbf{u}^t B\,\mathbf{u}$, where now the denominator depends only on the patch boundary vertices. Consequently, the optimal eigenvector will only balance conformality for the quadratic form defined by B *at the boundary*, removing the artifacts seen in internal vertices of the original Fiedler vector (see comparison in Fig. 5): our spectral parameterization maximizes the squared distance from boundary vertices to their barycenter over the unit ball defined by the conformal energy. Note that our approach differs from other generalized eigenproblems (such as Normalized Cuts [SM00], which picks $B_{ii} = (L_D)_{ii}$ to alter the $\mathcal{L}^2$ norm) in that we pick a degenerate matrix B. As demonstrated in Figures 8 and 4, this approach removes the distortion typically arising from vertex constraints or appearing in the Fiedler vector of the conformal quadratic form. A natural alternative is to choose $B = A$, the parameterization area; however, fold-overs tend to appear more often with this formulation.

3.4 Numerical Implementation

There are very efficient eigensolvers designed to find the smallest eigenvalues of the *sparse symmetric generalized eigenvalue problem* we need to solve. They usually proceed through Choleski decomposition (to turn the problem into a conventional eigenvalue problem) and Lanczos iterations, particularly fast in our context since our matrices are sparse. We can thus obtain a spectral free-boundary parameterization very efficiently. However, we propose a customized approach to provide even faster results as follows.

We first transform (at no cost) the minimization (5) into a maximization by solving for the *largest eigenvalue* $\mu = 1/\lambda$ of the eigenvalue problem where the left- and right-hand sides have been switched, as it makes the solve more robust and efficient. Robustness to large meshes with extremely degenerate triangles is further improved by adding $\varepsilon\mathrm{Id}$ to L_C to guarantee that numerical inaccuracies in the coefficients of the Laplacian-like matrix will not alter the positive semi-definiteness of the matrix (we use $\varepsilon = 10^{-8}$). Finally, we remove the need to compute multiple eigenvalues by directly removing the known kernel of L_C through the following modified generalized eigenvalue problem:

$$\boxed{\left[B - \frac{1}{V_b}\mathbf{e}_b\mathbf{e}_b^t\right]\mathbf{u} = \mu\, L_C\mathbf{u}} \tag{7}$$

where $\mathbf{e}_b = B\mathbf{e}$ (i.e., a copy of the matrix $\mathbf{e}$ for which the coefficients corresponding to internal vertices have been zeroed out), while V_b is the number of boundary vertices (not including any of internal boundaries of the patch). Note that the pair of eigenvectors associated with the maximum eigenvalue of this generalized eigenvalue problem coincides exactly with the pair of eigenvectors $\mathbf{u}^*$ associated with the first non-zero eigenvalue of Eq. (5): since the eigenvectors of the latter eigenproblem are B-orthogonal, the added outer product is zero for all eigenvectors except the two in the columns of $\mathbf{e}$. This slightly modified system thus requires the computation of a *single* eigenvalue/eigenvector pair, speeding up the solver by 5 to 15% in our tests. Furthermore, the case of a developable/flat mesh would have required a special test with the previous expression as the eigenvector $\mathbf{u}^*$ still has an eigenvalue 0. Instead, Eq. (7) will *always* return the optimal solution for any non-degenerate input mesh.

In our `C++` implementation, we use a call to the `eigs` function from the Matlab library to find the largest eigenvalue of Eq. (7) and its associated eigenvector $\mathbf{u}^*$. We use a callback function to perform the matrix-vector product needed on the left-hand side of the eigenvalue problem, as it does not require any matrix storage and can be efficiently computed in $O(V_b)$ time. Again, no vertex needs to be fixed since the solver picks directly the first eigenvector associated with the maximum eigenvalue, with an arbitrary rotation as our problem is rotation-invariant. Due to the sparsity and symmetry of L_C, the resulting timings are on the order of a few seconds for any meshes with $V < 10k$. For instance, the lion head in Fig. 4 (8.3K vertices) solves in .7 seconds, while Fig. 1 (50K vertices) solves in 5.2 seconds. According to our tests, the Matlab call to `eigs` takes 5 to 10 times *less* than the OpenNL solve used in LSCM [LPRM02]; however, a Matlab linear solve of the LSCM/DCP equation is between two and three times faster than the eigenvalue solve we require. A faster Cholesky decomposition and/or a hierarchical eigensolver may improve execution times on very large meshes, such as the head on the right of Fig. 9 (150K vertices), which solved in 31 seconds.

4 Extensions

While our spectral method can be used as is as a fast and robust parameterization approach, we also provide a few extensions that are easy to implement, but particularly relevant for practical purposes.

4.1 Enforcing Insensitivity to Sampling Irregularity

The sampling of an input mesh can be highly irregular: flat regions may have only a few large triangles, while curved parts can contain many small triangles. To better preserve

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

this irregularity in our resulting parameterizations, we use a simple and particularly effective modification of the conformal energy. We simply *weight the area functional and Dirichlet energy from each triangle T by the inverse of its original area $|T|$ in $\mathcal{X}$*. The initial sampling is thus taken into account in the parameterization, removing unnatural distortion due to irregular sampling as Figures 1 and 7 demonstrates. This weighting can be interpreted in two different ways: either it can be seen as an alteration of the area form dA on the original mesh, or, in the context of LSCM, as an inverse-area-weighted least-square enforcement of orthogonality. This simple change makes the matrix A less sparse than mentioned in Section 2.2, as now the internal edges no longer cancel each other out: the coefficients of this matrix are easily computed on a per edge (i, j) basis as:

$$A_{u_i,v_j} = \frac{1}{2}\left[1/|T_{ijk}| - 1/|T_{ijl}|\right], A_{v_i,u_j} = \frac{1}{2}\left[-1/|T_{ijk}| + 1/|T_{ijl}|\right].$$

Each cotangent coefficients in L_D are simply divided by the area of the triangle they were computed on. However, the resulting inverse-area-weighted L_C has exactly the same sparsity as before, thus no additional computational cost for our eigensolver is incurred by this modification.

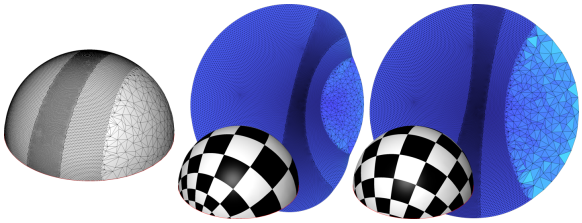
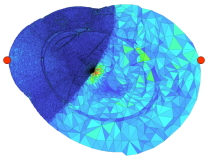


Figure 7: ***Half-sphere****, discretized with variable density; Spectral conformal parameterization w/o (left) and w/ (right) area weighting.*

We note that this area-weighting of the conformal energy can be directly (and independently of our spectral treatment) applied to the LSCM/DCP method too. Considerable improvements over the initial method are often obtained as illustrated in the inset (compare to Fig. 1). However, the distortion due to point constraints still remains—confirming the need for our spectral approach.



4.2 User Control

Our spectral method can be further modified to allow for more user control. First, one can use previous methods directly to add cuts in the mesh based on the initial distortion of our results to help unfold the mesh further—and rerun the eigenvalue solver on the cut mesh. One can also let the user "interact" with the parameterization by adding "extenders" to repulse two vertices $\mathbf{u}_i$ and $\mathbf{u}_j$: adding a quadratic term of the form $\alpha(\mathbf{u}_i - \mathbf{u}_j)^2$ to the quadratic form represented by matrix B will add a force between these two vertices, where $|\alpha|$ indicates how much the conformality can be sacrificed to accommodate this user-specified force. Thanks to the global nature of our minimization procedure, we witness no local distortion around the extended vertices, and

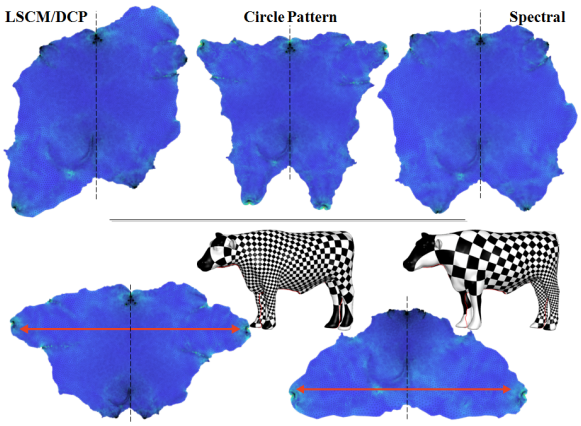


Figure 8: ***Cow****. Top: LSCM/DCP with two farthest vertices pinned (left) results in asymmetric, deformed legs. Our spectral method (right) does not require any constraints, yielding a parameterization which automatically respects the symmetry of the mesh. Bottom: User-specified extenders (red arrows) modify the parameterization without undue distortion, offering texture density control.*

multiple extenders are trivially handled. This simple modification, akin to the multiple-pins option used in LSCM/DCP, makes it possible for a user to tailor the parameterization at low cost (see Fig. 8, right). Note that the same idea can be used to prevent holes in a mesh from collapsing in the parameterization: we can add extenders between an additional "fake" vertex (added to the system) and all the vertices of an internal boundary to add rigidity to the hole if necessary. While we have found this force-based method of control to behave in a straightforward manner, the development of a tool to translate potentially more intuitive control gestures into this framework is left for future work.

4.3 Extensions to Quasi-Conformal Parameterizations

We finally point out that our framework could also be adapted to accommodate *quasi-conformality* by allowing a bit of slack in the orthogonality of ∇u and ∇v. This can be made through a change of the Hodge star as proposed in [ACSTD07] and [Gli07], for instance based on a tensor field derived from two orthogonal tangent vector fields as proposed in [FSDH07] or on the Cauchy tensor of existing parameterizations [ZRS05]. This will in turn change the expression of the discrete Laplacian L_D, but the spectral approach remains unchanged. While we foresee no obstacles in integrating these techniques into our method, we have left further exploration for future work.

5 Results and Conclusions

We presented a simple and efficient spectral approach to discrete conformal parameterization which is robust to sampling irregularity and does not require distortion-prone positional constraints on vertices. The numerical procedure required to obtain the parameterization involves a sparse, symmetric generalized eigenvalue problem which, although not as simple as a single linear solve, proved efficient and robust in our tests. In terms of computation speed, solving the eigenvalue problem necessary to alleviate the need for con-

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

straints is only two to three times slower than the linear solve required by LSCM/DCP, thus requiring only a second or two for moderate meshes. We also tried the quasi-conformal (QC) distortion as a measure of the deviation from perfect conformality: it is computed (and rendered in our figures) per triangle as the ratio of larger to smaller eigenvalues of the Jacobian of the map [SGSH02]. This quotient is 1 iff the mapping is conformal, and above one otherwise. An average quasi-conformal distortion is computed as the area-weighted average of each QC distortion, where the area is computed in the original mesh. Our method achieves parameterizations that, in our tests, result in lower average QC than LSCM/DCP (with or without area weighting), even for meshes with multiple boundaries (see Fig. 10) or many nearly-degenerate triangles. It can also result in lower QCs than non-linear methods like [KSS06], even if the resulting parameterizations are sometimes less visually appealing (see the cow mesh example in Fig. 8).

Limitations. It bears repeating here that our linear-algebra solution to conformal parameterization of arbitrary surface patches does *not* guarantee fold-free results, similar to LSCM/DCP and any other linear methods. While there are *no folds or triangle flips in the examples presented in this paper*, we have experienced some flips in models with very degenerate triangles as well as in the three-triangle example presented in [Flo98]. We also mention that we have occasionally experienced non-injectivity due to the intersection of separate parts of the mesh in regions where the boundary has large, sharp concavities. The sheer efficiency of our linear-algebra based method, however, can *quickly and robustly* yield an initial parameterization that can be further improved to achieve whatever goal the user may have. This should not be considered as an alternative to more involved methods which, with proper non-linear solvers and higher computational cost, can provide better results.

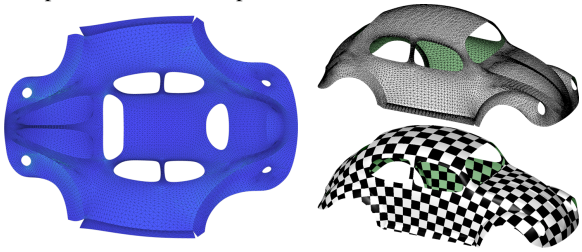


Figure 10: ***Beetle****. Our spectral method can also handle multiple boundaries without modification (solved in 1.69s, QC=1.007874).*

As for future work, we want to explore the consequences of our unbiased treatment of parameterization for other known weights (mean value coordinates, Wachpress, etc), and in the context of manifold learning [BN03] as well: our modification to the eigenvalue problem should carry over to higher dimensions, potentially providing less distortion than current Laplacian-based eigenmethods. For this type of application, however, developing multigrid eigensolvers may prove crucial to handling large multidimensional datasets. We would also like to further investigate the consequences of changing the B matrix in the generalized eigenvalue problem to obtain better control and predictability of resulting parameterizations. Using our results to develop *seamless*, global parameterizations is another a valuable goal for the future. Finally, understanding the condition needed to guarantee one-to-one parameterizations remains an outstanding problem worthy of further work.

Acknowledgements: We wish to express our thanks to the reviewers for providing meshes and advice, and to Lily Kharevych for providing the Circle Patterns examples used in this paper.

References

[ACSTD07] ALLIEZ P., COHEN-STEINER D., TONG Y., DESBRUN M.: Voronoi-based variational reconstruction of unoriented point sets. In *Symposium on Geometry Processing* (July 2007), pp. 39–48.

[BCGB08] BEN-CHEN M., GOTSMAN C., BUNIN G.: Conformal flattening by curvature prescription and metric scaling. In *Computer Graphics Forum, 27(2)* (2008).

[BN03] BELKIN M., NIYOGI P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput. 15*, 6 (2003), 1373–1396.

[CSD02] COHEN-STEINER D., DESBRUN M.: Hindsight: LSCM and DCP are one and the same. In *unpublished note found at:* www.geometry.caltech.edu/pubs/CD02.pdf (2002).

[DKT05] DESBRUN M., KANSO E., TONG Y.: Discrete differential forms for computational modeling. In *Discrete Differential Geometry*. ACM SIGGRAPH Course Notes, 2005.

[DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. In *Proceedings of Eurographics* (2002), pp. 209–218.

[EDD*95] ECK M., DEROSE T., DUCHAMP T., HOPPE H., LOUNSBERY M., STUETZLE W.: Multiresolution analysis of arbitrary meshes. In *ACM SIGGRAPH* (1995), pp. 173–182.

[FH05] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*. Springer, 2005, pp. 157–186.

[Fie73] FIEDLER M.: Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal 23*, 98 (1973), 298–305.

[Flo97] FLOATER M. S.: Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design 14*, 4 (1997), 231–250.

[Flo98] FLOATER M. S.: Parametric tilings and scattered data approximation. *IJSM 4*, 3-4 (1998), 165–182.

[FSDH07] FISHER M., SCHRÖDER P., DESBRUN M., HOPPE H.Design of tangent vector fields. *ACM TOG 26(3)* (2007), 56.

[GGH02] GU X., GORTLER S. J., HOPPE H.: Geometry images. In *ACM SIGGRAPH* (2002), pp. 355–361.

[GGS03] GOTSMAN C., GU X., SHEFFER A.: Fundamentals of spherical parameterization for 3d meshes. In *ACM SIGGRAPH* (2003), pp. 358–363.

[GGT06] GORTLER S. J., GOTSMAN C., THURSTON D.: Discrete one-forms on meshes and applications to 3d mesh parameterization. *Comput. Aided Geom. Des. 23*, 2 (2006), 83–112.

[Gli07] GLICKENSTEIN D.: A monotonicity property for weighted delaunay triangulations. *Journal Discrete and Computational Geometry (to appear) 38*, 4 (2007), 651–664.

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

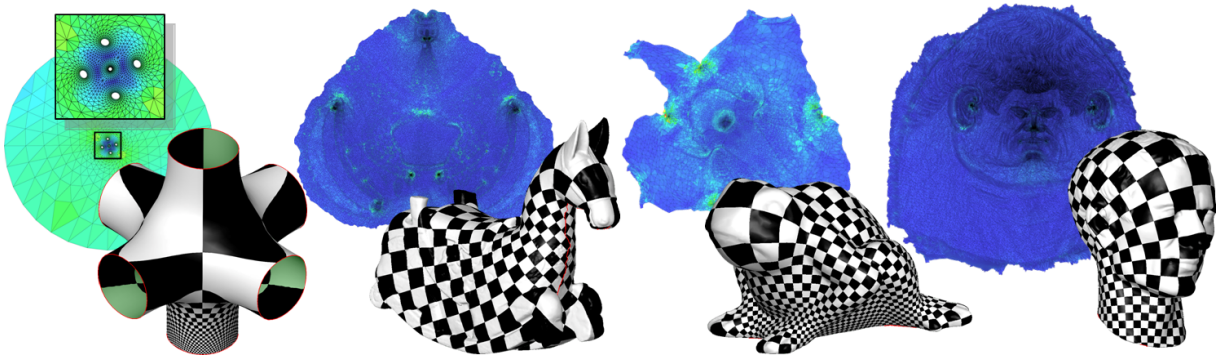


Figure 9: ***Zoo****. We demonstrate the results of our spectral conformal parameterization scheme on a variety of meshes, with multiple boundaries (left) and high vertex count (150K vertices, right). No triangle flips or folds are present in the results.*

[Got03] GOTSMAN C.: On graph partitioning, spectral analysis, and digital mesh processing. In *Solid Modeling International* (2003), pp. 209–218.

[GY03] GU X., YAU S.-T.: Global conformal parameterization. In *Symposium on Geometry Processing* (2003), pp. 127–137.

[HAT*00] HAKER S., ANGENENT S., TANNENBAUM A., KIKINIS R., SAPIRO G., HALLE M.: Conformal surface parameterization for texture mapping. *IEEE TVCG 6*, 2 (2000), 181–189.

[HG00] HORMANN K., GREINER G.: MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*. Vanderbilt University Press, 2000, pp. 153–162.

[HLS07] HORMANN K., LÉVY B., SHEFFER A.: Mesh parameterization: Theory and practice. In *SIGGRAPH 2007 Course Notes* (Aug. 2007), vol. 2, pp. 1–122.

[JKG07] JIN M., KIM J., GU X. D.: Discrete surface Ricci flow: Theory and applications. In *Lecture Notes in Computer Science (Mathematics of Surfaces XII)* (2007), vol. 4647.

[Kor05] KOREN Y.: Drawing graphs by eigenvectors: theory and practice. *C&MA 49*, 11-12 (2005), 1867–1888.

[KSO04] KOLLURI R., SHEWCHUK J. R., O'BRIEN J. F.: Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing* (July 2004), pp. 11–22.

[KSS06] KHAREVYCH L., SPRINGBORN B., SCHRÖDER P.: Discrete conformal mappings via circle patterns. *ACM Trans. Graph. 25*, 2 (2006), 412–438.

[LÓ6] LÉVY B.: Laplace-Beltrami eigenfunctions: Towards an algorithm that understands geometry. In *IEEE Shape Modeling and Applications (invited talk)* (2006).

[LKL02] LEE Y., KIM H., LEE S.: Mesh parameterization with a virtual boundary. *Computers & Graphics 26*, 5 (2002), 677–686.

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph. 21*, 3 (2002), 362–371.

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Exp. Math. 2(1)* (1993), 15–36.

[RL03] RAY N., LEVY B.: Hierarchical least squares conformal map. In *Pacific Graphics* (2003), pp. 263–270.

[RS00] ROWEIS S., SAUL L.: Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (2000), 2323–2326.

[SCOGL02] SORKINE O., COHEN-OR D., GOLDENTHAL R., LISCHINSKI D.: Bounded-distortion piecewise mesh parameterization. In *IEEE Visualization* (2002), pp. 355–362.

[SdS01] SHEFFER A., DE STURLER E.: Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers 17*, 3 (2001), 326–337.

[SGSH02] SANDER P. V., GORTLER S. J., SNYDER J., HOPPE H.: Signal-specialized parametrization. In *Eurographics Workshop on Rendering* (2002), pp. 87–98.

[SH02] SHEFFER A., HART J. C.: Seamster: Inconspicuous low-distortion texture seam layout. In *IEEE Visualization* (2002), pp. 291–298.

[SLMB05] SHEFFER A., LÉVY B., MOGILNITSKY M., BOGOMYAKOV A.: Abf++: Fast and robust angle based flattening. *ACM Trans. Graph. 24*, 2 (2005), 311–330.

[SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *IEEE Trans. PAMI 22*, 8 (2000), 888–905.

[SPR06] SHEFFER A., PRAUN E., ROSE K.: Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision 2*, 2 (2006), 105–171.

[SSP08] SPRINGBORN B., SCHRÖDER P., PINKALL U.: Conformal equivalence of triangle meshes. *ACM TOG 27*, 3 (2008).

[TACSD06] TONG Y., ALLIEZ P., COHEN-STEINER D., DESBRUN M.: Designing quadrangulations with discrete harmonic forms. In *Symposium on Geometry Processing* (2006), pp. 201–210.

[Tut63] TUTTE W. T.: How to draw a graph. *Proc. London Math 13* (1963), 743–768.

[ZKK02] ZIGELMAN G., KIMMEL R., KIRYATI N.: Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. on Visualization and Computer Graphics 8*, 2 (2002), 198–207.

[ZLS07] ZAYER R., LÉVY B., SEIDEL H.-P.: Linear angle based parameterization. In *Symposium on Geometry Processing* (2007), pp. 135–141.

[ZRS05] ZAYER R., RÖSSL C., SEIDEL H.-P.: Setting the boundary free: a composite approach to surface parameterization. In *Symposium on Geometry Processing* (2005), pp. 91–100.

[ZSGS04] ZHOU K., SYNDER J., GUO B., SHUM H.-Y.: Isocharts: stretch-driven mesh parameterization using spectral analysis. In *Symposium on Geometry Processing* (2004), pp. 45–54.

[ZvKD07] ZHANG H., VAN KAICK O., DYER R.: Spectral methods for mesh processing and analysis. In *Proc. of Eurographics State-of-the-art Report* (2007), pp. 1–22.

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.