Spin Transformations of Discrete Surfaces

Keenan Crane
Caltech

Ulrich Pinkall
TU Berlin

Peter Schröder
Caltech

Abstract

We introduce a new method for computing conformal transformations of triangle meshes in $\mathbb{R}^3$. Conformal maps are desirable in digital geometry processing because they do not exhibit *shear*, and therefore preserve texture fidelity as well as the quality of the mesh itself. Traditional discretizations consider maps into the complex plane, which are useful only for problems such as surface parameterization and planar shape deformation where the target surface is flat. We instead consider maps into the *quaternions* $\mathbb{H}$, which allows us to work directly with surfaces sitting in $\mathbb{R}^3$. In particular, we introduce a *quaternionic Dirac operator* and use it to develop a novel integrability condition on conformal deformations. Our discretization of this condition results in a sparse linear system that is simple to build and can be used to efficiently edit surfaces by manipulating curvature and boundary data, as demonstrated via several mesh processing applications.

Keywords: digital geometry processing, discrete differential geometry, geometric modeling, geometric editing, shape space deformation, conformal geometry, quaternions, spin geometry, Dirac operator

Links: 🌐DL 📄PDF 🌐WEB

1 Introduction

How does one compute conformal deformations of a surface in $\mathbb{R}^3$? Discretizing the well-known Cauchy-Riemann equations does not help because these equations apply only to maps into the plane. We instead describe the geometry of a surface M via an immersion $f : M \rightarrow \mathrm{Im}\,\mathbb{H}$ into the imaginary part of the quaternions $\mathrm{Im}\,\mathbb{H}$ (Section 3). In this setting two surfaces f and $\tilde{f}$ are conformally equivalent as long as their differentials df and $d\tilde{f}$ are related by a scaling and rotation $\lambda \in \mathbb{H}$ at each point:

$$d\tilde{f} = \bar{\lambda} df \lambda.$$

The surface $\tilde{f}$ is called a *spin transformation* of f [Kamberov et al. 1998]. However, for *arbitrary* λ, this equation may have no solution. We introduce a linear integrability condition

$$\boxed{(D - \rho)\lambda = 0}$$

that characterizes all valid transformations λ in terms of a prescribed change ρ in *mean curvature half-density* (Section 4.1) and a first-order differential operator D called the *Dirac operator* (Section 4). The corresponding discrete operator is remarkably simple, involving only triangle areas and edge vectors (Section 5.2).

Figure 1: *Using spin transformations, a model that has already been carefully detailed* (left) *can be further altered without compromising texture or geometric detail* (middle), *whereas standard mesh editing tools may not respect these features* (right).

Practically speaking, this setup allows us to "paint" a change in curvature on a surface and produce the corresponding conformal deformation (Figures 2 and 13). As a result, we can edit surfaces without degrading texture (Figures 1 and 20) or mesh quality (Figure 3). The linearity of our formulation is most unusual: if one instead wants to prescribe standard quantities (such as the induced metric, principal curvatures, *etc.*) more difficult *non-linear* problems must be solved [Eigensatz et al. 2008]. We can also perform standard mesh processing tasks such as flattening and curvature flow (Section 7.3), and find conformal approximations of arbitrary deformations (Section 7.2).

Contributions We introduce the *quaternionic Dirac operator*, leading to a beautifully simple integrability condition for spin transformations of smooth surfaces (Section 4). This condition is readily discretized, enabling robust and accurate computation of conformal deformations (Section 5). Deformations are computed by solving an eigenvalue problem and a Poisson equation; matrices have the same sparsity as the standard cotangent discretization of the Laplace-Beltrami operator [Duffin 1959]. Our discrete Dirac operator faithfully captures the spectrum of its smooth counterpart, and can be used to produce (for the first time) surface deformations that are perfectly conformal in the limit of refinement, as validated by numerical experiment (Section 6). We also show how changes in mean-curvature half density can be used to express a variety of mesh processing tasks (Section 7).
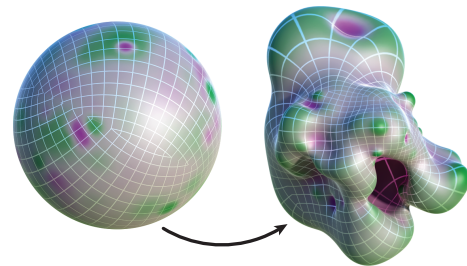

Figure 2: *Given desired change in curvature* (left) *we construct a new conformally equivalent surface* (right). *Green and purple indicate a positive and negative change in curvature, respectively.*
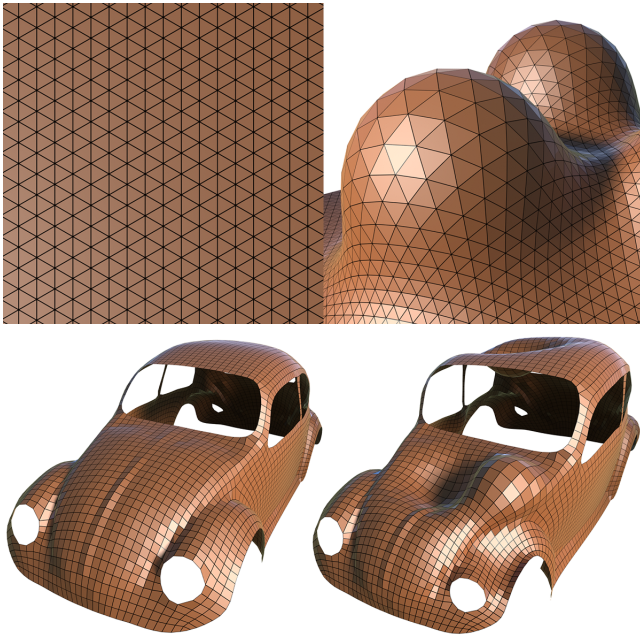
Figure 3: *Spin transformations also preserve mesh quality. Top: a grid of equilateral triangles* (left) *is substantially deformed while preserving triangle shape almost perfectly* (right). *Bottom: A carefully quadrangulated beetle* (left) *becomes an old beat up car* (right) *without degrading element quality.*

2 Related Work

Differential representations are well-studied in digital geometry processing [Botsch and Sorkine 2008] but ignore the question of *integrability*: under what condition does a prescribed differential quantity come from an actual surface? The standard practice is to forgo this question and instead look for a surface that best approximates prescribed data in the least-squares sense, as proposed by Yu *et al.* [2004]. For conformal maps we need to be more careful since the closest surface may be far from conformal, even when the new differential comes from a similarity transformation at each point (see Figure 22 for a comparison).

Later work expressed edge vectors in local coordinate frames at vertices to ensure rigid motion invariance [Lipman et al. 2005]. In this case reconstruction is non-linear since coordinate frames depend non-linearly on vertex positions. Paries *et al.* [2007] also seek conformal deformations and encode local frames as quaternions λ at vertices. Reconstruction entails a sequence of alternating least-squares problems: first to minimize the frame differences $d\lambda$, then to find the closest surface $\tilde{f}$; ultimately, distortion remains (see Figure 19). In [Lipman et al. 2007] frame differences are also minimized, with the additional non-linear constraint $|\lambda| = 1$ at each vertex to enforce *isometric* deformation. As before the non-linear relationship between positions and frames is resolved with an iterative scheme and no integrability condition is considered.

Conformal deformations have also been studied in the context of *cage-based editing*, where coordinate functions on a coarse volumetric *cage* induce transformations of an enclosed surface [Lipman et al. 2008; Ben-Chen et al. 2009]. The main difficulty is that the only conformal maps from $\mathbb{R}^3$ to itself are *Möbius transformations*, which are far too rigid for surface editing. As a result, cage-based methods are inherently limited to so-called

quasi-conformal maps, which can still introduce significant distortion (see Figures 4 and 18). On the other hand, these methods are highly efficient since they need only evaluate simple (often closed-form) coordinate expressions. It may therefore make sense to use spin transformations in *conjunction* with such methods to produce a final, high-quality surface (see Section 7.2).

Discrete conformal maps have been studied extensively in the case of *surface parameterization* where the target surface has constant curvature (such as the plane or the sphere). A common approach is to compute a pair of discrete harmonic functions that describe a map into the plane, requiring the solution of a linear system [Mercat 2001; Desbrun et al. 2002; Lévy et al. 2002; Gu and Yau 2003], or an eigenvalue problem [Mullen et al. 2008]. Conformal transformations can also be constructed by prescribing values at vertices that directly control the rescaling of the metric [Ben-Chen et al. 2008; Yang et al. 2008; Springborn et al. 2008]. However, a metric alone is not sufficient to describe an embedding and so the restriction to constant curvature is needed to realize the final surface. We instead store values $\lambda \in \mathbb{H}$ at vertices that encode both local rescaling and *rotation*, which is sufficient to describe the extrinsic geometry of a deformation.

Several methods allow prescription of extrinsic curvatures [Eigensatz et al. 2008] and metric properties such as length [Eigensatz and Pauly 2009]. However, the resulting energies are expressed in terms of vertex positions, making them highly non-linear. Using higher-order differential quantities (such as our function ρ) as primary degrees of freedom may prove useful in this setting.

In contrast to previous approaches our setup ensures that transformations are conformal *by construction* – any failure to achieve integrability on a mesh is due purely to discretization error (Section 6.3). In contrast, the *discrete theory* of conformal equivalence developed by Springborn *et al.* [2008] ensures exact reconstruction for target surfaces of constant curvature.
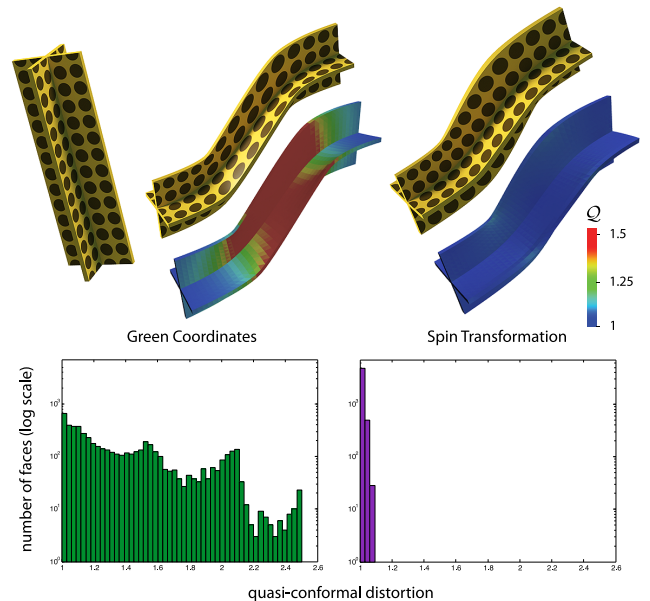

Figure 4: *Following [Lipman et al. 2008, Fig. 13], comparison of a cross-shaped bar (top) deformed using green coordinates (left) and the closest spin transformation (right), computed as in Section 7.2. Notice that the spin transformation preserves circles perfectly. See [Lipman et al. 2008] for comparison with other methods.*
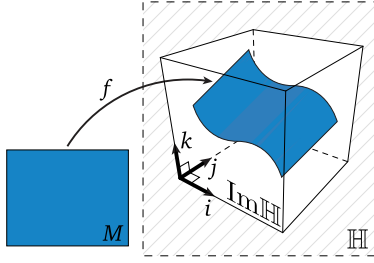
Figure 5: *We represent the geometry of a surface in $\mathbb{R}^3$ using a map f that associates each point on a surface M with a point in the imaginary quaternions Im $\mathbb{H}$.*

3 Background

We first recall facts about quaternions and differential forms that are needed to describe our smooth setup. The reader may also benefit from the discussion in [Desbrun et al. 2008]; the role of the Dirac operator in the geometry of surfaces is discussed in [Pedit and Pinkall 1998; Friedrich 1998; Taimanov 2006].

Quaternions The quaternions $\mathbb{H}$ can be viewed as a 4D real vector space with basis $\{1, i, j, k\}$ along with the non-commutative *Hamilton product*, which satisfies the relationships $i^2 = j^2 = k^2 = ijk = -1$. The *imaginary quaternions* Im $\mathbb{H}$ are elements of the 3D subspace spanned by $\{i, j, k\}$. The *conjugate* of a quaternion $q = a + bi + cj + dk$ is given by $\bar{q} = a - bi - cj - dk$, hence $\bar{q} = -q$ when $q \in$ Im $\mathbb{H}$. The squared length of q is given by $|q|^2 = q\bar{q}$, and finally $\overline{qp} = \bar{p}\bar{q}$ for all $q, p \in \mathbb{H}$.

Rotation of a vector $v \in$ Im $\mathbb{H}$ is expressed via conjugation by a unit element $|q| = 1$, *i.e.*, $\tilde{v} = \bar{q}vq$ is a rotated version of v. More generally, for $|q| \neq 0$, $\bar{q}vq$ represents a *similarity transformation*, *i.e.*, rotation and *uniform scaling* by $|q|^2$. Explicitly, $q = a(\cos(\theta/2) + \sin(\theta/2)u)$ yields a scaling by a^2 and a rotation by an angle θ around the axis u.

Quaternion Calculus Just as complex analysis identifies $\mathbb{R}^2$ with the complex plane $\mathbb{C}$, we identify $\mathbb{R}^3$ with Im $\mathbb{H}$: points $(x, y, z) \in \mathbb{R}^3$ become points $xi + yj + zk \in$ Im $\mathbb{H}$. A surface in $\mathbb{R}^3$ is then specified by a smooth map $f : M \to$ Im $\mathbb{H}$ from an abstract surface M to its realization in 3-space (Figure 5). More generally we will consider functions that take values anywhere in $\mathbb{H}$.

The differential $df : TM \to$ Im $\mathbb{H}$ maps tangent vectors on M to vectors in 3-space, *i.e.*, $X \mapsto df(X)$ – in coordinates, df is the *Jacobian matrix*, though we do not use that representation here. If df is non-degenerate everywhere, then f is an *immersion* (note that immersions still permit self-intersections). If $|df(X)|$ depends only on the *length* (and not the direction) of X, then f is a *conformal immersion* – in other words, stretching is equal in all directions. In this case the squared length element $|df|^2$ gives us the (unsigned) area element, or the induced *Riemannian metric*.

4 Spin Transformations

Consider two immersions $f, \tilde{f} : M \to$ Im $\mathbb{H}$ of a given surface M (see Figure 6). These surfaces are said to be *spin equivalent* if their differentials are related by a similarity transformation, *i.e.*, if there exists some function $\lambda : M \to \mathbb{H}$ such that

$$d\tilde{f}(X) = \bar{\lambda}df(X)\lambda \qquad (1)$$

for all tangent vectors X. In this case the induced Riemannian metrics $|df|^2$ and $|d\tilde{f}|^2 = |\lambda|^4|df|^2$ are related by a positive scaling $|\lambda|^4$, hence the two immersions are also *conformally equivalent*.

A more typical scenario is that a surface f is given and we wish to find a spin transformation $\tilde{f}$. An arbitrary λ will *not*, in general, produce a 1-form $d\tilde{f}$ that is the differential of some surface $\tilde{f}$, *i.e.*, $\bar{\lambda}df\lambda$ may not be *integrable*. Hence, even in the smooth setting we cannot solve Eq. (1) unless λ satisfies very special criteria.

Fortunately, a simple, linear condition on λ guarantees integrability. This condition was originally derived by Kamberov *et al.* [1998, Lemma 2.2], but we propose a form that is more suitable for discretization. In particular, we define the *quaternionic Dirac operator*

$$D\lambda = -\frac{df \wedge d\lambda}{|df|^2}. \qquad (2)$$

Roughly speaking, D is a generalization of the gradient operator which expresses first-order derivatives of both scalar- and vector-valued functions on M (see Section 8), and in local coordinates is equivalent to the standard Dirac operator from physics (see Appendix A). Readers familiar with exterior calculus should note that division by the area element $|df|^2$ simply indicates the canonical identification of 2-forms with 0-forms, as is usually expressed via the Hodge star.

Using D, one can easily show that $\bar{\lambda}df\lambda$ is integrable if and only if λ satisfies the integrability condition

$$(D - \rho)\lambda = 0 \qquad (3)$$

for some real-valued function $\rho : M \to \mathbb{R}$ (see Appendix B).

At first glance this problem seems difficult to solve: we have to *simultaneously* find a function ρ and a similarity transformation λ satisfying Eq. (3). However, suppose that ρ is any scalar function whatsoever. If we solve the eigenvalue problem

$$(D - \rho)\lambda = \gamma\lambda \qquad (4)$$

for the pair (γ, λ) with $\gamma \in \mathbb{R}$, it follows that $(D - (\rho + \gamma))\lambda = 0$, *i.e.*, we get an exact solution for Eq. (3) by simply adding a constant shift γ to ρ.

Overall, then, we have the following procedure:

1. pick a scalar function ρ on M;

2. solve an eigenvalue problem (Eq. (4)) for the similarity transformation λ;

3. solve a linear system (Eq. (1)) for the new surface $\tilde{f}$.

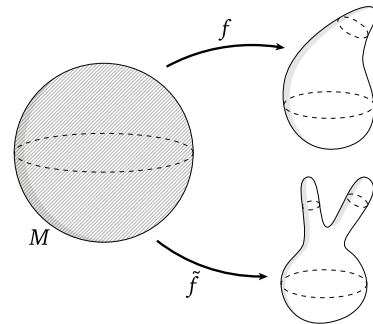A discrete version of this procedure is developed in Section 5.

Figure 6: *Given an initial immersion f of a surface, a new immersion $\tilde{f}$ is a conformal deformation or* spin transformation *if the differentials of these maps are related by scaling and rotation.*

4.1 Mean Curvature Half-Density

In the procedure above deformations are specified via the scalar function ρ. What is the geometric meaning of ρ?

Let H denote the mean curvature of an immersion f and let $|df|$ denote the induced length element. Then the quantity $H|df|$ is called the *mean curvature half-density* of f, and the mean-curvature-half density of a spin transformation $\tilde{f}$ is given by

$$\tilde{H}|d\tilde{f}| = H|df| + \rho|df|$$

(see Kamberov *et al.* [1998, Lemma 2.3]). Thus ρ controls the *change* in mean curvature half-density – its effect is illustrated in Figures 2 and 13. Note, then, that the effect of applying a constant shift to ρ is akin to adding a constant value to a grayscale image – the resulting appearance is largely the same.

Working with $H|df|$ is often more natural than manipulating mean curvature itself. For instance, an increase in H is often thought of as an increase in bending, but could also indicate uniform shrinking and a simultaneous *decrease* in bending. On the other hand $H|df|$ is scale-independent (since H is inversely proportional to length) and the meaning is unambiguous: increasing $H|df|$ exaggerates bending; decreasing $H|df|$ mitigates it.

4.2 Non Trivial Topology

On surfaces that are not simply-connected the transformed differential $\omega = \bar{\lambda} df \lambda$ still describes a conformal immersion of M, but the immersion may fail to "close up" globally – this failure is encoded by the harmonic component of ω. Our reconstruction step (Section 5.6) effectively removes the harmonic part by seeking the closest exact form $d\tilde{f}$, which can introduce a low-frequency shear in the solution that does not vanish under refinement. However, in our experience such distortion is negligible, producing excellent results in practice (see Figures 3 and 16).

4.3 Boundary Conditions

For surfaces with boundary the condition $(D - \rho)\lambda = 0$ must be supplemented by appropriate boundary conditions. As with Cauchy-Riemann, prescribing the exact *position* of boundary points is too strong in the sense that we may not end up with a conformal map. We instead prescribe the *tangent directions* of the new boundary curve arbitrarily at each point (see for example Figure 7). An implementation of these boundary conditions is detailed in section 5.7. Note that when the target mesh is planar our boundary condition amounts to prescribing the *boundary curvature* [Ben-Chen et al. 2008; Springborn et al. 2008] and we get a unique conformal solution.
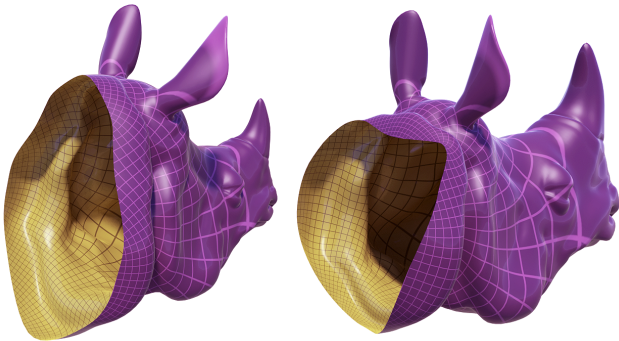

Figure 7: *The boundary of a surface* (left) *can be modified without disrupting texture or geometric detail* (right, $\mathcal{Q}_{mean} = 1.016$).
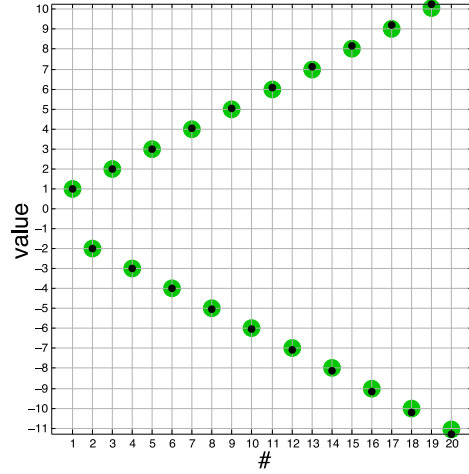

Figure 8: *Even on a coarse mesh of 8k triangles the spectrum of our discrete operator (black dots) matches the predicted spectrum (green dots) exceptionally well. Here we plot the first twenty distinct eigenvalues (out of 120 total) for the unit sphere.*

5 Discretization

In this section we discretize the operators used in Eq. (4). We use sans serif characters to denote the matrix of a smooth linear operator – for instance, L approximates L.

5.1 Quaternionic Matrices

Matrices $\mathsf{X} \in \mathbb{H}^{m \times n}$ with quaternion-valued entries provide a convenient representation for the operators used in our discretization. Although standard packages for numerical linear algebra do not support quaternionic matrices, they can easily be implemented by constructing a real matrix $\hat{\mathsf{X}} \in \mathbb{R}^{4m \times 4n}$ where each entry $q = a + bi + cj + dk$ is replaced by a 4×4 block

$$\begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix}.$$

(If performance or storage are an issue, one can alternatively provide a callback routine that applies the Hamilton product directly.) Note that the real transpose of such a matrix corresponds to the conjugate transpose of the original quaternionic matrix, *i.e.,* $\hat{\mathsf{X}}^T = \mathsf{X}^H$.

5.2 Discrete Dirac Operator

To compute spin transformations of meshes, we need a discrete Dirac operator D. Assume that $\mathcal{K} = \{V, E, F\}$ is an oriented 2-manifold triangle mesh with a value $f_i \in \operatorname{Im} \mathbb{H}$ and $\lambda_i \in \mathbb{H}$ at each vertex. Then $\mathsf{D} \in \mathbb{H}^{|F| \times |V|}$ is a sparse (3 entries per row) rectangular matrix whose nonzero entries are
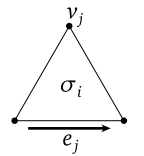
$$\mathsf{D}_{ij} = -\frac{1}{2\mathcal{A}_i} e_j, \qquad (5)$$

where $\mathcal{A}_i$ is the area of triangle σ_i, j is the index of any vertex v_j of σ_i, and e_j is the opposing edge vector (see insert). Note that edge vectors should be oriented counter-clockwise around each triangle.

There are a number of ways to derive D, but the simplest is perhaps to observe that $D\lambda$ can be written as

$$D\lambda = \frac{d(df\,\lambda)}{|df|^2}.$$

Then if f and λ are piecewise linear functions interpolating values at vertices, integrating the function $D\lambda$ over a triangle σ_l with vertices (i,j,k) gives us

$$\int_{\sigma_l} D\lambda |df|^2 = \int_{\partial\sigma_l} df\,\lambda = \sum_{\sigma_{ij}\in\partial\sigma_l} \int_{\sigma_{ij}} df\,\lambda = \sum_{\sigma_{ij}\in\partial\sigma_l} (f_j - f_i)\frac{\lambda_i+\lambda_j}{2}.$$

Simplifying this sum with $e_k = f_j - f_i$ (and similarly for the other edges) and noting that $D\lambda$ is constant in each face, we can divide by area to get the density

$$-\frac{1}{2\mathcal{A}_l}(e_i\lambda_i + e_j\lambda_j + e_k\lambda_k),$$

which is precisely the quantity computed by our matrix above.

5.3 Scalar Multiplication

To solve our eigenvalue problem (Eq. (4)) we also need a discretization of ρ, which represents scalar multiplication by a real-valued function. We proceed as above by integrating over triangles – in particular, let ρ be a piecewise constant function with values $\rho_i \in \mathbb{R}$ on each face, and let λ be piecewise linear as before. Then we have

$$\frac{1}{\mathcal{A}_i} \int_{\sigma_i} \rho\lambda |df|^2 = \rho_i \left(\frac{1}{3} \sum_{v_j\in\sigma_i} \lambda_j \right)$$

over any triangle σ_i. For our purposes it will be convenient to express the corresponding discrete operator $\mathsf{R} \in \mathbb{H}^{|F|\times|V|}$ as

$$\mathsf{R} = \mathsf{PB}$$

where $\mathsf{P} \in \mathbb{H}^{|F|\times|F|}$ is a diagonal matrix with entries $\mathsf{P}_{ii} = \rho_i$, and $\mathsf{B} \in \mathbb{H}^{|F|\times|V|}$ amounts to a simple triangle-vertex incidence matrix with nonzero entries $\mathsf{B}_{ij} = 1/3$ for each vertex v_j of triangle σ_i. Finally, we let $\mathsf{A} = \mathsf{D} - \mathsf{R}$ be the discretization of the operator $A = (D - \rho)$ appearing in our eigenvalue problem (Eq. (4)).

5.4 Adjoint Matrices

We will also need the *adjoint* matrices D^*, A^*, *etc.* (with respect to the usual L^2 inner product). Consider the diagonal *mass matrices* $\mathsf{M}_F \in \mathbb{H}^{|F|\times|F|}$ of triangle areas and $\mathsf{M}_V \in \mathbb{H}^{|V|\times|V|}$ containing one-third the area of triangles incident on each vertex. Then the adjoint of any matrix $\mathsf{X} \in \mathbb{H}^{|F|\times|V|}$ is given by $\mathsf{M}_V^{-1}\mathsf{X}^H\mathsf{M}_F$. In the systems we consider the term M_V^{-1} appears on both the left- and right-hand side – for convenience we omit this term and adopt the notational convention

$$\mathsf{X}^* = \mathsf{X}^H\mathsf{M}_F.$$

5.5 Discrete Eigenvalue Problem

Consider the eigenvalue problem $(D - \rho)\lambda = \gamma\lambda$ from Section 4. If we try to discretize this equation as simply $\mathsf{A}\lambda = \gamma\mathsf{B}\lambda$ we get a *rectangular* eigenvalue problem, which is difficult to solve directly. One idea is to make the system square by averaging the values $\mathsf{A}\lambda$ and $\mathsf{B}\lambda$ from faces back to vertices, but this local averaging introduces spurious modes to the null space that severely corrupt solutions.

We can instead obtain a square system in the following way. Consider that any solution (γ, λ) to the problem $A\lambda = \gamma\lambda$ also yields

a solution to the problem $A^2\lambda = \gamma^2\lambda$. However, the opposite is not true since positive and negative eigenspaces can "mix" when we square A. In other words, if (γ, λ^+) and $(-\gamma, \lambda^-)$ are solutions to the original problem then linear combinations of λ^+ and λ^- are eigenfunctions of A^2 but not of A. The solution is to use the *generalized* eigenvalue problem $A^2\lambda = \gamma A\lambda$, where the additional factor of A on the right hand side now distinguishes between eigenvalues of different sign. This leads us to the system

$$\mathsf{A}^*\mathsf{A}\lambda = \gamma\mathsf{B}^*\mathsf{A}\lambda, \tag{6}$$

which we solve for the smallest eigenvalue γ and its corresponding eigenvector $\lambda \in \mathbb{H}^{|V|}$. Solutions to this problem exhibit excellent agreement with smooth solutions (see Section 6).

For surface editing tasks we are usually interested in the eigenvalue closest to zero and do not have to worry as much about mixing eigenspaces. In practice, then, we can often solve the problem $\mathsf{A}^*\mathsf{A}\lambda = \gamma\mathsf{M}_V\lambda$, or equivalently the *standard* eigenvalue problem

$$(\mathsf{AV})^*\mathsf{AV}\hat{\lambda} = \gamma\hat{\lambda}, \tag{7}$$

where $\mathsf{V} = \mathsf{M}_V^{-1/2}$ and $\lambda = \mathsf{V}\hat{\lambda}$ recovers the final solution. We use this problem for most of the applications in Section 7. Note that all matrices in systems (6) and (7) – including $\mathsf{B}^*\mathsf{D}$ – are sparse and symmetric, with positive semi-definite operators on the left-hand side. It is easy to verify that these operators involve values only in the 1-ring of a given vertex. Finally, in practice we observe improved numerics if the entries of V are rescaled to have mean value 1.

5.6 Discretized Spin Transformations

After solving for the similarity transformations λ, we still need to determine the final vertex positions $\tilde{f}_i$ by solving $d\tilde{f} = \bar{\lambda}df\lambda$ (Eq. (1)). In the discrete setting this equation says that each new
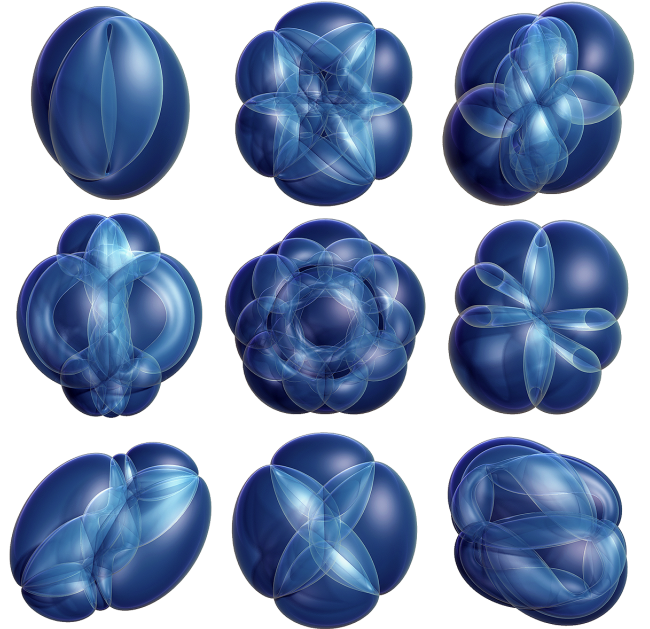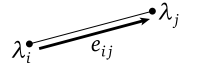

Figure 9: *On the sphere, eigenfunctions of the Dirac operator correspond to relativistic wave functions of an electron orbiting an atomic nucleus, here visualized for the first time.*

edge vector $\tilde{e}_{ij} = \tilde{f}_j - \tilde{f}_i$ should equal the edge vector $e_{ij} = f_j - f_i$ from the original mesh scaled and rotated by λ. This transformation is discretized by integrating $\bar{\lambda} df \lambda$ along each edge, yielding

$$\tilde{e}_{ij} = \frac{1}{3}\bar{\lambda}_i e_{ij}\lambda_i + \frac{1}{6}\bar{\lambda}_i e_{ij}\lambda_j + \frac{1}{6}\bar{\lambda}_j e_{ij}\lambda_i + \frac{1}{3}\bar{\lambda}_j e_{ij}\lambda_j. \quad (8)$$

At this point we can solve the linear system $d\tilde{f} = \tilde{e}$ for vertex positions $\tilde{f}$. To do so we minimize the residual $r = |d\tilde{f} - \tilde{e}|^2$, which amounts to a standard Poisson problem $\Delta\tilde{f} = \nabla \cdot \tilde{e}$. Discretization of this problem yields (using notation from [Desbrun et al. 2008]) the standard cotangent-Laplace matrix $\Delta = d_0^H \star_1 d_0 \in \mathbb{H}^{|V| \times |V|}$ and divergence operator $\nabla \cdot = d_0^H \star_1 \in \mathbb{H}^{|V| \times |E|}$ built as quaternionic matrices with purely real entries.

One might wonder why we need to solve this system in the least-squares sense – after all, in the smooth setting our integrability condition (Eq. (3)) guarantees that Eq. (1) has an exact solution. As with linear discretizations of Cauchy-Riemann, however, this exactness does not carry over to the discrete setting. Nonetheless we find that, empirically, the magnitude of the residual r depends only on mesh resolution and vanishes under refinement. More importantly, even at a coarse level we see substantial improvement over methods that do not consider integrability, which may exhibit significant conformal distortion even under refinement (see Figures 4, 18, 19 and 22).

5.7 Discrete Boundary Conditions

Suppose that T and $\tilde{T}$ are unit tangent vectors that specify initial and target directions at each boundary point. Furthermore, let θ be the angle between T and $\tilde{T}$ and let $w = T \times \tilde{T}/\sin\theta$ be the unit vector orthogonal to both tangents. (For $\tilde{T} = \pm T$ we use $\theta = 0$ and any vector w.) Then λ must have the form
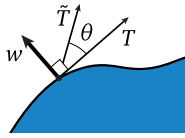
$$\lambda = \left(\cos\frac{\theta}{2} + w\sin\frac{\theta}{2}\right)(a + b\tilde{T})$$

for any $a, b \in \mathbb{R}$. The first term rotates from T to $\tilde{T}$ and the second term describes subsequent scaling and rotation around $\tilde{T}$.

In the discrete setting we store a pair of values $a, b \in \mathbb{R}$ at each boundary vertex (in lieu of a value $\lambda \in \mathbb{H}$) and build the block diagonal matrix $C \in \mathbb{R}^{4|\partial V| \times 2|\partial V|}$ where ∂V is the set of boundary vertices and each block has the form

$$\begin{bmatrix} 1 & 0 \\ 0 & \tilde{T}^x \\ 0 & \tilde{T}^y \\ 0 & \tilde{T}^z \end{bmatrix},$$

where $\tilde{T}^x, \tilde{T}^y, \tilde{T}^z$ are the components of $\tilde{T}$. We also build the diagonal matrix $W \in \mathbb{H}^{|V| \times |V|}$ where W_{ii} equals 1 for interior vertices and $\cos(\theta_i/2) + w_i\sin(\theta_i/2)$ for boundary vertices v_i. Finally, indexing interior vertices before boundary vertices we build

$$U = W\begin{bmatrix} I & 0 \\ 0 & C \end{bmatrix},$$

which maps both interior and boundary degrees of freedom to the values of λ. Hence, Eq. (7) becomes $(AVU)^*AVU\hat{\lambda} = \gamma\hat{\lambda}$ and the final solution is recovered by evaluating $\lambda = VU\hat{\lambda}$.

6 Validation

In this section we validate our discrete D against known properties of the smooth operator and inspect convergence behavior.
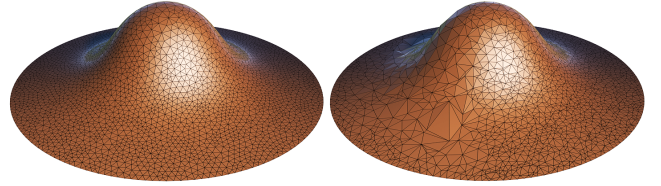

Figure 10: *Because we discretize a smooth energy, similar results are achieved on a uniform* (left) *and nonuniform sampling of the domain*(right). *Here a bump is added to an initially flat disk.*

6.1 Quasi-Conformal Error

Throughout the paper we measure the quality of any map g between immersed surfaces in terms of *quasi-conformal error* $\mathcal{Q} \in \mathbb{R}^{|F|}$, which is the ratio of largest to smallest singular value of dg in each face [Sander et al. 2001]. An error of $\mathcal{Q}_i = 1$ in each face is ideal, indicating that no shearing takes place (only uniform scaling and rotation). We use $\mathcal{Q}_{mean}$ to denote the (area-weighted) mean quasi-conformal error over the entire surface.

6.2 Spectrum of Discrete Dirac Operator

On the unit sphere the eigenvalues of D are the integers $n \in \mathbb{Z}$, which appear with multiplicity $n+1$ (note that this means $n = -1$ does not appear at all). Figure 8 indicates that the spectrum of D is in excellent agreement with the smooth spectrum – these eigenvalues also appear with the correct multiplicity $n + 1$.

Eigenfunctions of D on the unit sphere are called the *spinor spherical harmonics*, which can be used to compute fundamental solutions to the *Dirac equation* for the configuration of a spin-1/2 particle such as an electron in a spherically symmetric potential (this relationship motivates the "spin equivalence" terminology). The corresponding spin transformations $\tilde{f}$ are called the *Dirac spheres*, which provide another basis for validation since closed-form expressions are known and have previously been visualized for small values of n [Richter 1997] (see Figure 11). We are also able to visualize for the first time Dirac spheres corresponding to higher energy states (Figure 9). Note that here we must remove the constant component from each eigenvector, since constant solutions are in the null space of the system $D^*D\lambda = \gamma B^*D\lambda$.

6.3 Convergence

Numerical tests such as the one in Figure 12 indicate that surfaces produced by our method converge to perfectly conformal deformations in the limit of refinement. Note that D is a self-adjoint elliptic operator, which means that on a compact surface it has a discrete spectrum. Hence, asking for the *smallest* eigenvalue is a well-posed problem, and linear convergence is precisely what we expect from a piecewise linear discretization. Figure 10 suggests that results do not depend heavily on element quality. These tests used the standard eigenvalue problem (Eq. (7)).
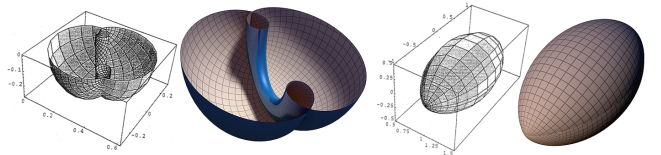

Figure 11: *Comparison with closed-form solutions.* Left: *constant-curvature spin transformations of the unit sphere (from [Richter 1997]).* Right: *numerical solutions produced by our method.*
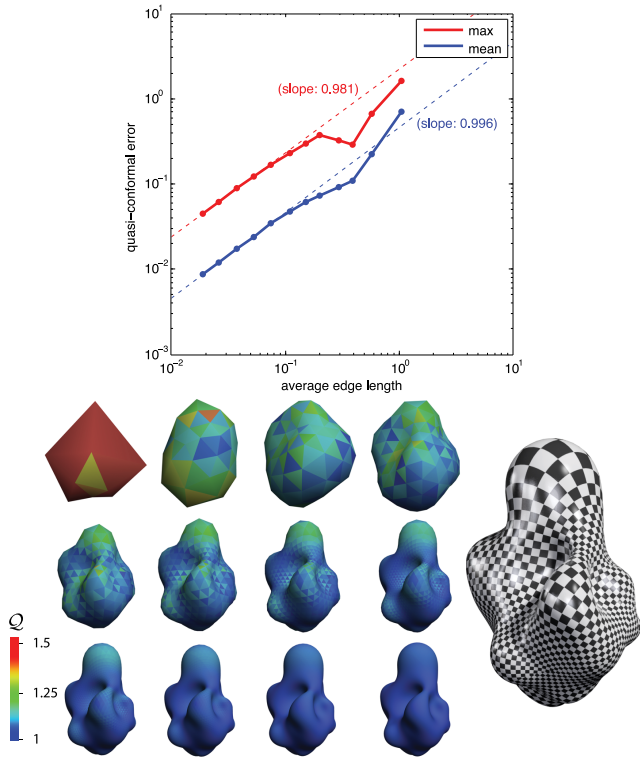
Figure 12: *Convergence of sphere with bumps added – the limit surface is a perfect conformal deformation* (bottom right). *Top: quasi-conformal error with respect to mean edge length (lines fit to first few data points).* *Bottom left: intermediate surfaces showing spatial distribution of quasi-conformal distortion (1 is ideal).*

7 Applications

Many mesh processing tasks can be expressed in terms of a change in mean curvature half-density – this section takes a quick look at some of these applications. We used the `eigs` function in MATLAB to solve systems (6) and (7); not surprisingly, performance was very similar to spectral methods for mesh parameterization [Mullen et al. 2008]. For instance, computation on a mesh of ∼33k faces (Figure 13) took about 7 seconds on a 2.4 GHz Core 2 Duo laptop. Overall performance scaling was roughly linear in the number of faces.

7.1 Painting Curvature

The most straightforward way to specify a deformation is to "paint" a function ρ on a surface, altering its curvature (Figure 2). Since ρ is a scalar quantity, we can easily process surfaces by applying standard image filters (Figure 13). In Figure 1 we add detail to an already carefully-textured model. Figure 3 illustrates that adding bumps does not damage element quality (for the car example we temporarily split each quad into two triangles).

Generally speaking, if g is a bump function used to displace a surface in the *normal* direction, setting $\rho = \Delta g$ produces a similar but conformal bump (see Figure 20 for example). This idea is based on the observation that for a *flat* surface undergoing a normal deformation $\tilde{f}(t) = f + tgN$, the change in $\tilde{H}|d\tilde{f}|$ at time $t = 0$ equals Δg. Additionally, ρ should not contain any large *spikes* since it is difficult to significantly deform a small number of triangles while remaining conformal (consider trying to add a large amount of curvature to a single vertex 1-ring).

7.2 Arbitrary Deformations

More generally, we can take a mesh that has been deformed *arbitrarily* and find a nearby spin transformation. We first compute "best-fit" similarity transformations in each face – in particular, let A_1, A_2 be rotation matrices bringing the source and target face into a common plane, and let B be the (orientation-preserving) map between the resulting planar triangles. If $B = RY$ is the polar decomposition of B, then the best-fit similarity transformation is

$$S = \sqrt{\det(Y)}A_2^{-1}RA_1,$$

i.e., the geometric average of principal stretches times the rotational component of the map between triangles. These transformations are expressed as quaternions and averaged from faces to vertices to get a value μ_i at each vertex.

In general μ will not satisfy our integrability condition, but we can still find a change in curvature that closely approximates the target surface by minimizing $|(D - R)\mu|^2$ with respect to the values ρ_i. The optimal values are given explicitly by

$$\rho_i = \frac{\mathrm{Re}((\overline{B\mu})_i (D\mu)_i)}{|(B\mu)_i|^2}.$$

We then solve for λ and the new surface $\tilde{f}$ as usual. In Figure 14 we edit a mesh using commercial software and project the result onto a nearby spin transformation, preserving texture; see also Figures 4, 18, and 19. We found that the generalized eigenvalue problem (Eq. (6)) proved more robust for this application.

Deformations with a large amount of *shear* may not be well-approximated by *any* conformal map, even in the smooth setting. Here we can make a tradeoff between approximation quality and conformality by interpolating between μ and λ (Figure 21).

Figure 13: Top left: *the man in the moon is sculpted by "painting" a scalar function (inset) onto a disk. Applying standard image filters to this function achieves a variety of effects while preserving a conformal map to the original disk.* Top right: *low-pass filter.* Bottom left: *high-pass filter.* Bottom right: *unsharp mask.*

Figure 14: *Giraffe attempts ballet: a textured mesh* (left) *can be modified arbitrarily and projected onto the nearest conformally equivalent surface* (middle, $\mathcal{Q}_{mean} = 1.015$), *preserving texture fidelity. We can also explicitly modify scale without disturbing texture – on the right we ask for a much larger head.*
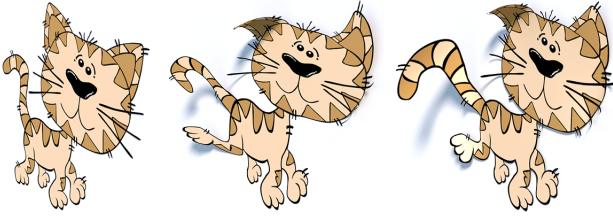

Figure 15: *By controlling the curvature and the direction of tangent vectors at the boundary we can lift a planar cartoon* (left) *into the third dimension* (middle) *and recover the closest conformally equivalent shape* (right). *Note that the initially squashed leg, tail, and head, recover their initial shape.* ($\mathcal{Q}_{mean} = 1.011$)

7.3 Willmore Flow

The L^2 norm of the mean curvature half-density $H|df|$ (Section 4.1) conveniently yields the *Willmore functional*

$$\mathcal{W} = \int_M H^2|df|^2,$$

which is used in digital geometry processing for surface fairing and the construction of minimal surfaces [Bobenko and Schröder 2005]. Using our framework we can compute what we call the *conformal Willmore flow*

$$\tfrac{d}{dt}\rho = -\nabla\mathcal{W}(\rho)$$

which reduces mean curvature as quickly as possible while preserving a conformal map at each point in time. Of course, this flow is subject to our usual constraint $(D-\rho)\lambda = 0$. On a disk this constraint can be satisfied for any ρ (without a shift) and hence the flow has the simple closed form expression $\rho(t) = -tH$, where H is the initial mean curvature. The situation on a sphere is not as simple, and will be investigated in future work.

At an extreme, flow on a (possibly punctured) disk removes *all* mean curvature, yielding a *minimal surface* (Figure 16, *top*) – note that this "flow" can actually be computed in a single time step by setting $\rho = -H$. Prescribing a planar boundary progressively flattens the surface, ultimately achieving a conformal parameterization (Figure 17). In these experiments we computed H via the magnitude of the mean curvature normal $HN = \Delta f$.

7.4 Boundary Prescription

We can also edit surfaces by manipulating boundary data. In Figure 7 we modify the boundary shape while maintaining the mean curvature half density ($\rho = 0$), thereby preserving geometric features. Notice that some features change scale, which is permitted by our conformal energy. This same setup provides us with a novel way to compute minimal surfaces: start with a flat surface and prescribe a new boundary without changing $H|df|$ (Figure 16, *bottom*). Finally, we can prescribe a new boundary *and* new curvature, as in Figure 15 where we compute ρ as described in Section 7.2. In these applications we simply set T and $\tilde{T}$ to the (unit) outgoing edge vector at each boundary vertex of the source and target mesh, respectively.

8 Conclusion

Our discretization of the integrability condition $(D - \rho)\lambda = 0$ provides a principled, efficient way to construct conformal deformations of triangle meshes in $\mathbb{R}^3$. On the whole, it is remarkable that an operator as simple as D (Eq. (5)) is enough to discretize the wide variety of tasks described above, requiring very little additional machinery in each case. An even bigger surprise is that the discrete Dirac operator has many uses beyond constructing conformal maps: it enables a unified treatment of vector calculus on immersed surfaces, incorporating div, grad and curl into a single operator on quaternionic functions. Its square can also be used to recover the Laplace-Beltrami operator for both scalar functions and vector fields. Moreover, since our quaternionic D is expressed in terms of *extrinsic* geometry it can be used to compute normal information, mean curvature, and the shape operator. These properties and their applications will be explored in future work.
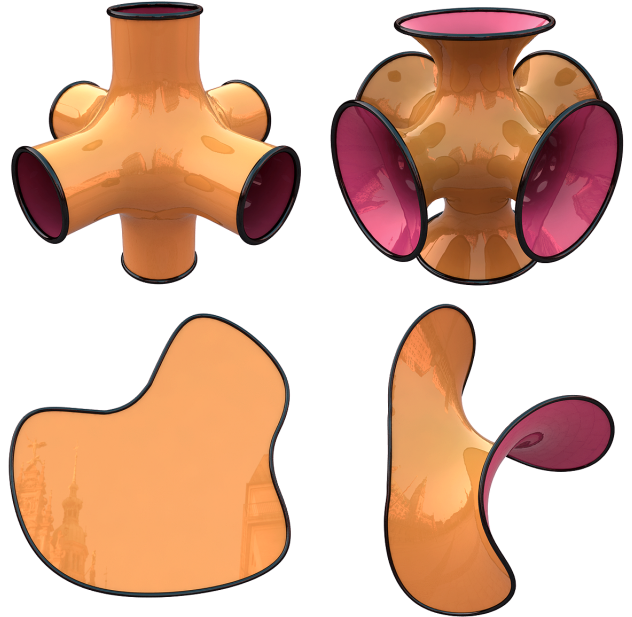

Figure 16: *Spin transformations can be used to compute minimal surfaces in two distinct ways.* Top: *starting with a curved surface* (left) *we remove all mean curvature* (right, $\mathcal{Q}_{mean} = 1.070$). Bottom: *starting with a flat surface* (left) *we prescribe new tangent directions along the boundary* without *changing mean curvature* (right, $\mathcal{Q}_{mean} = 1.054$). *In both cases the surface is computed directly without an iterative flow.*
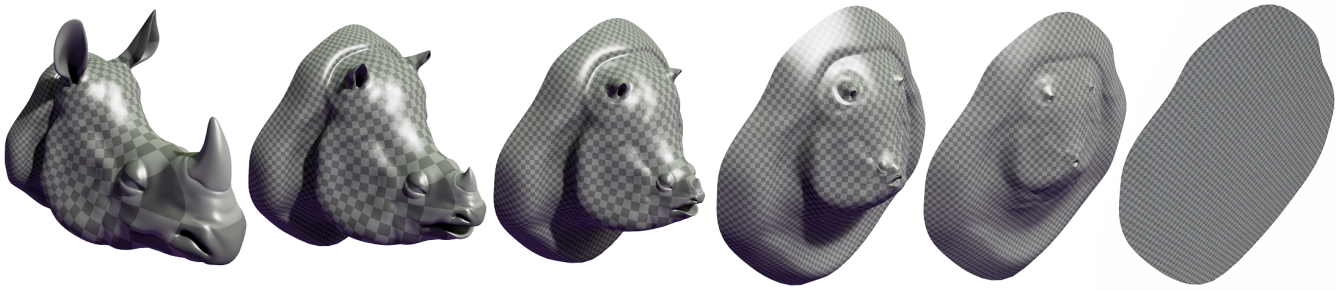
Figure 17: *Conformal Willmore flow on a rhino head: mean curvature shrinks as quickly as possible.*

References

Ben-Chen, M., Gotsman, C., and Bunin, G. 2008. Conformal Flattening by Curvature Prescription and Metric Scaling. *Comp. Graph. Forum 27*, 2, 449–458.

Ben-Chen, M., Weber, O., and Gotsman, C. 2009. Variational Harmonic Maps for Space Deformations. *ACM Trans. Graph. 28*, 3, 34:1–34:11.

Bobenko, A., and Schröder, P. 2005. Discrete Willmore Flow. In *Proc. Symp. Geom. Proc.*, 101–110.

Botsch, M., and Sorkine, O. 2008. On Linear Variational Surface Deformation Methods. *IEEE Trans. Vis. Comp. Graph. 14*, 1, 213–230.

Desbrun, M., Meyer, M., and Alliez, P. 2002. Intrinsic Parameterizations of Surface Meshes. *Comp. Graph. Forum 21*, 3, 209–218.

Desbrun, M., Kanso, E., and Tong, Y. 2008. Discrete Differential Forms for Computational Modeling. In *Discrete Differential Geometry*, A. I. Bobenko, P. Schröder, J. M. Sullivan, and G. M. Ziegler, Eds., Vol. 38 of *Oberwolfach Seminars*. Birkhäuser Verlag, 287–324.

Duffin, R. 1959. Distributed and Lumped Networks. *J. Math. Mech.* [continued as *Indiana Univ. Math. J.*] *8*, 793–826.

Eigensatz, M., and Pauly, M. 2009. Positional, Metric, and Curvature Control for Constraint-Based Surface Deformation. *Comp. Graph. Forum 28*, 2, 551–558.

Eigensatz, M., Sumner, R. W., and Pauly, M. 2008. Curvature-Domain Shape Processing. *Comp. Graph. Forum 27*, 2, 241–250.

Friedrich, T. 1998. On the Spinor Representation of Surfaces in Euclidean 3-Space. *J. Geom. and Phys. 28*, 1-2, 143–157.

Gu, X., and Yau, S.-T. 2003. Global Conformal Surface Parameterization. In *Proc. Symp. Geom. Proc.*, 127–137.

Kamberov, G., Pedit, F., and Pinkall, U. 1998. Bonnet Pairs and Isothermic Surfaces. *Duke Math. J. 92*, 3, 637–644.

Lévy, B., Petitjean, S., Ray, N., and Maillot, J. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Trans. Graph. 21*, 3, 362–371.

Lipman, Y., Sorkine, O., Levin, D., and Cohen-Or, D. 2005. Linear Rotation-Invariant Coordinates for Meshes. *ACM Trans. Graph. 24*, 3, 479–487.

Lipman, Y., Cohen-Or, D., Gal, R., and Levin, D. 2007. Volume and Shape Preservation via Moving Frame Manipulation. *ACM Trans. Graph. 26*, 1.

Lipman, Y., Levin, D., and Cohen-Or, D. 2008. Green Coordinates. *ACM Trans. Graph. 27*.

Mercat, C. 2001. Discrete Riemann Surfaces and the Ising Model. *Comm. Math. Physics 218*, 1, 177–216.

Mullen, P., Tong, Y., Alliez, P., and Desbrun, M. 2008. Spectral Conformal Parameterization. *Comp. Graph. Forum 27*, 5, 1487–1494.

Paries, N., Degener, P., and Klein, R. 2007. Simple and Efficient Mesh Editing with Consistent Local Frames. In *Proc. Pac. Graph.*, Comp. Graph. Appl., 461–464.

Pedit, F., and Pinkall, U. 1998. Quaternionic Analysis on Riemann Surfaces and Differential Geometry. In *Proc. Int. Congr. Math.*, Vol. 21, 389–400.

Richter, J. 1997. *Conformal Maps of a Riemann Surface into the Space of Quaternions*. PhD thesis, TU Berlin.

Sander, P. V., Snyder, J., Gortler, S. J., and Hoppe, H. 2001. Texture Mapping Progressive Meshes. In *Proc. ACM/SIGGRAPH Conf.*, 409–416.

Springborn, B., Schröder, P., and Pinkall, U. 2008. Conformal Equivalence of Triangle Meshes. *ACM Trans. Graph. 27*, 3.

Taimanov, I. 2006. Two-Dimensional Dirac Operator and Surface Theory. *Russ. Math. Surveys 61*, 1, 79–159.

Yang, Y., Kim, J., Luo, F., Hu, S., and Gu, D. 2008. Optimal Surface Parameterization Using Inverse Curvature Map. *IEEE Trans. Vis. Comp. Graph. 14*, 5, 1054–1066.

Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., and Shum, H.-Y. 2004. Mesh Editing with Poisson-based Gradient Field Manipulation. *ACM Trans. Graph. 23*, 3, 644–651.

A Relationship with Standard Dirac Operator

The Dirac operator for a spin $1/2$ particle in the plane is given by $-i(\sigma_x \partial_x + \sigma_y \partial_y)$, where σ_x and σ_y are the *Pauli matrices*

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

Note that any function $\lambda : M \to \mathbb{H}$ can be written as $\lambda = \psi_1 + j\psi_2$ for some pair of $\mathbb{C}$-valued functions ψ_1, ψ_2. Define ∂ and $\bar{\partial}$ via

$$\partial \psi := \tfrac{1}{2}(\partial_x - i\partial_y)\psi, \quad \bar{\partial}\psi := \tfrac{1}{2}(\partial_x + i\partial_y)\psi.$$

If our surface f locally maps to the j, k-plane, *i.e.*, $f = jz$ for some $z : M \to \mathbb{C}$, then we have

$$df \wedge d\lambda = jdz \wedge ((\partial\psi_1 dz + \bar{\partial}\psi_1 d\bar{z}) + j(\partial\psi_2 dz + \bar{\partial}\psi_2 d\bar{z}))$$
$$= (j\bar{\partial}\psi_1 + \partial\psi_2)dz \wedge d\bar{z}$$
$$= -2(j\bar{\partial}\psi_1 + \partial\psi_2)(i|df|^2).$$

Dividing through by $-1/|df|^2$ we get $D\lambda = 2(j\bar{\partial}\psi_1 + \partial\psi_2)i$, which means the matrix of D is

$$D = 2\begin{bmatrix} 0 & \partial \\ \bar{\partial} & 0 \end{bmatrix} i = i(\sigma_x \partial_x + \sigma_y \partial_y),$$

i.e., our quaternionic Dirac operator is locally the same as the standard Dirac operator up to sign.

B Derivation of Integrability Condition

(This derivation follows [Kamberov et al. 1998].) The spin-equivalence condition (Eq. (1)) says that the 1-form $\bar{\lambda}df\lambda$ must be *exact*. On a simply-connected surface M, this is equivalent to saying that this 1-form is *closed*, from which we get

$$0 = d(\bar{\lambda}df\lambda) = d\bar{\lambda} \wedge df\lambda - \bar{\lambda}df \wedge d\lambda = \overline{\bar{\lambda}df \wedge d\lambda} - \bar{\lambda}df \wedge d\lambda.$$

But for any quaternion $q \in \mathbb{H}$, $q - \bar{q} = 2\,\mathrm{Im}\,q$. The above expression therefore says that the 2-form $\bar{\lambda}df \wedge d\lambda$ is purely real, or in other words, $\bar{\lambda}df \wedge d\lambda = \hat{\rho}|df|^2$ for some real function $\hat{\rho}$. Equivalently, we can require that $-df \wedge d\lambda = \rho\lambda|df|^2$ for a different real-valued function ρ which is related to $\hat{\rho}$ by $\rho = -\hat{\rho}/|\lambda|^2$. Substituting D into this final expression yields $(D - \rho)\lambda = 0$.
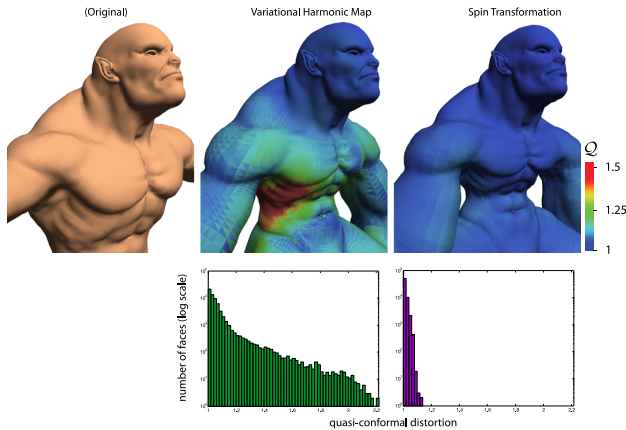

Figure 18: *Quasi-conformal deformation* (center) *produced via a variational harmonic map [Ben-Chen et al. 2008] and a nearby spin transformation computed as in Section 7.2* (right).
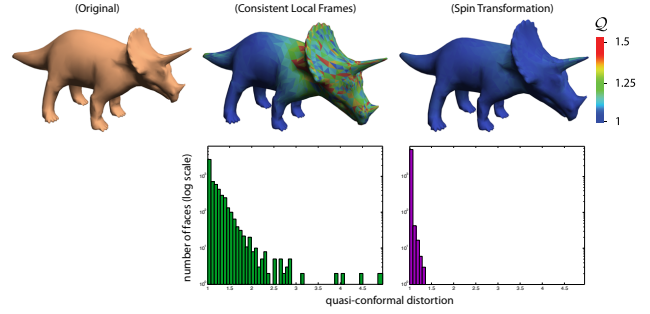

Figure 19: *Deformation produced via the method of consistent local frames [Paries et al. 2007]* (center) *and a nearby spin transformation computed as in Section 7.2* (right).
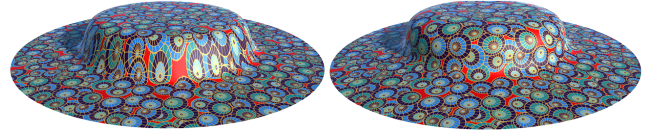

Figure 20: Left: *even for modest displacements, simple normal offsets can severely distort texture.* Right: *using a spin transformation to apply displacement prevents distortion.*
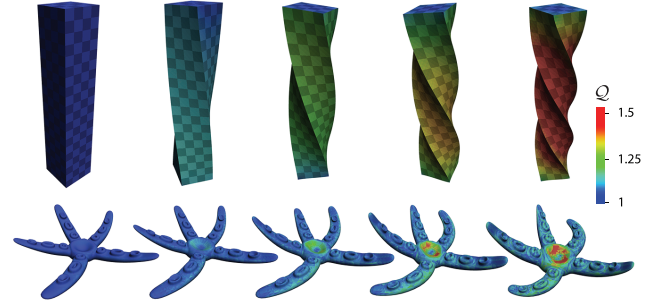

Figure 21: *Twist is inherently non-conformal, effectively a sequence of infinitesimal shears. Therefore, the conformal "deformation" closest to a twisted beam* (top right) *is the original beam* (top left). *We can trade off between conformality and approximation quality* (left to right).
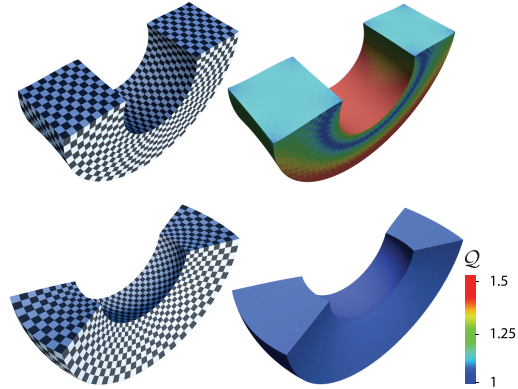

Figure 22: *Bent block reconstructed using least-squares* (top) *and our integrable approach* (bottom). *Colors indicate quasi-conformal error* $\mathcal{Q}$.