Geometric Modeling in Shape Space

Martin Kilian Niloy J. Mitra Helmut Pottmann
Vienna University of Technology

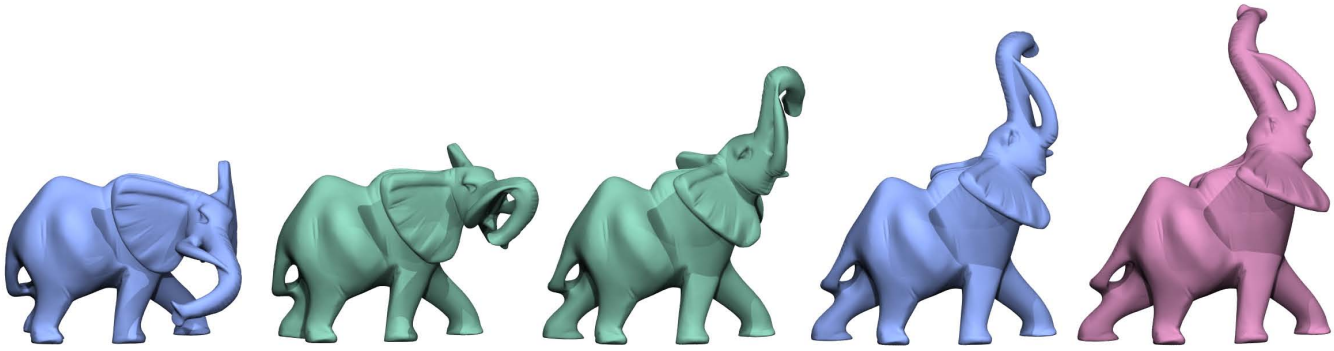

Figure 1: *Geodesic interpolation and extrapolation. The blue input poses of the elephant are geodesically interpolated in an as-isometric-as-possible fashion (shown in green), and the resulting path is geodesically continued (shown in purple) to naturally extend the sequence. No semantic information, segmentation, or knowledge of articulated components is used.*

Abstract

We present a novel framework to treat shapes in the setting of Riemannian geometry. Shapes – triangular meshes or more generally straight line graphs in Euclidean space – are treated as points in a shape space. We introduce useful Riemannian metrics in this space to aid the user in design and modeling tasks, especially to explore the space of (approximately) isometric deformations of a given shape. Much of the work relies on an efficient algorithm to compute geodesics in shape spaces; to this end, we present a multiresolution framework to solve the interpolation problem – which amounts to solving a boundary value problem – as well as the extrapolation problem – an initial value problem – in shape space. Based on these two operations, several classical concepts like parallel transport and the exponential map can be used in shape space to solve various geometric modeling and geometry processing tasks. Applications include shape morphing, shape deformation, deformation transfer, and intuitive shape exploration.

Keywords: Riemannian geometry, shape space, geodesic, isometric deformation, parallel transport, shape exploration.

1 Introduction

Computing with geometric shapes lies at the core of geometric modeling and processing. Typically a shape is viewed as a set of points and represented according to the available data, and the intended application. Geometry does not necessarily take this perspective: Projective geometry views hyperplanes as points in a dual space, line geometry interprets straight lines as points on a quadratic surface [Berger 1987], and the various types of sphere geometries model spheres as points in higher dimensional space [Cecil 1992]. Other examples concern kinematic spaces and Lie groups which are convenient for handling congruent shapes and motion design. These examples show that it is often beneficial and insightful to endow the set of objects under consideration with additional structure and to work in more abstract spaces. We will show that many geometry processing tasks can be solved by endowing the set of closed orientable surfaces – called *shapes* henceforth – with a Riemannian structure. Originally pioneered by [Kendall 1984], *shape spaces* are an active topic of interest in the mathematical research community. We focus our attention on the computational aspects of shape spaces and point to recent work of Michor and Mumford [2006], which provides a theoretical background for our research.

Our modeling and design paradigm is based on *geodesic curves* – locally shortest curves with respect to some metric. During interpolation, extrapolation (see Figure 1), and more general shape deformations (see Figure 10) shapes move along geodesics. Our approach is entirely geometric. Therefore the same method can be applied to a large class of problems with different underlying physical models, without knowing these models. Our algorithm does not need any segmentation of the model or external advise about the mesh structure. Working in a Riemannian manifold gives nice properties. For example geodesics from a shape M to each of a set of other shapes form a tree, thus generating globally consistent morphs. Such properties are harder to enforce with methods that do not consider a global space of deformations.

Related Work

To the best of our knowledge, there are only a few contributions treating shape spaces and related topics from a *computational perspective*. Cheng at al. [1998] realized the intimate connection between shape spaces and deformations, but neither discussed the critical choice of a metric, nor investigated essential geometric concepts such as geodesics. A computational approach to spaces of curves was presented in [Klassen et al. 2004] but has no natural extension to surfaces.

The gradient of a function on shape space depends on the met-

ric. Hence the choice of metric is crucial for any shape evolution driven by a gradient descent algorithm of a geometric energy. This is the driving motivation behind recent efforts in the computer vision community [Charpiat et al. 2005a; Charpiat et al. 2005b; Yezzi and Mennucci 2005] which discuss metrics for variational problems involving active contours.

The geometry of shape spaces is characterized by curves. Viewed from the original object space, a curve in shape space describes a *shape deformation*. Recently a number of different methods have been proposed to compute shape deformations. Such methods include maximum local rigidity criteria based on mesh representations [Alexa et al. 2000; Igarashi et al. 2005; Botsch et al. 2006], Laplacian coordinates [Sorkine et al. 2004], coordinates with respect to superposed control structures [Ju et al. 2005], the Poisson equation [Xu et al. 2005], divergence free vector fields [Funck et al. 2006], gradient domain techniques [Huang et al. 2006], and moving frame manipulation [Lipman et al. 2007]. In contrast to our approach all of the previously mentioned methods work directly on an object. For example [Lipman et al. 2007] directly aim at preserving first and second fundamental form, whereas we achieve this indirectly by computing in the space of all admissible shapes. Our work complements the above methods in the following sense: While almost all direct methods compute a diffeomorphism between a reference object and some target object given by certain constraints, we compute a smooth family of diffeomorphisms joining these two objects, i.e., all the missing in-between steps that direct methods cannot provide. This additional information can help to improve robustness and consistency of direct methods.

One approach to preserve surface characteristics is to choose a shape representation such that certain characteristics are invariant under a basic set of operations defined on this representation [Lipman et al. 2005; Kraevoy and Sheffer 2007]. In the shape space framework we can easily encode shape deformations. Desirable properties of such deformations can be automatically enforced by choosing a suitable metric in the space. We present one such metric based on *isometric deformations*. The importance of isometry for a number of applications is well known [Mémoli and Sapiro 2004; Bronstein et al. 2005].

A crucial component of our work deals with the computation of *geodesics* in shape spaces. Several efficient algorithms for computing geodesics in manifolds exist ([Kimmel and Sethian 1998; Mémoli and Sapiro 2004]). However, most of them rely on distance fields, making them unsuitable for high-dimensional manifolds such as shape spaces. We demonstrate how the special geometry of shape spaces simplifies this task allowing us to compute geodesics efficiently.

Contributions and Overview

- We provide a general computational framework for shape spaces designed for geometric modeling and geometry processing. In Section 2, we discuss the choice of proper metrics and how to deal with fibers consisting of shapes with vanishing distance.
- Section 3 presents a multi-resolution algorithm for computing geodesics in shape space. We address both the shape interpolation problem and the extrapolation problem for geodesics. The latter usually amounts to the solution of a second order ODE. In shape space we can reduce it to a first order ODE by integrating once.
- In our shape space framework many concepts from classical differential geometry can be applied to a wide variety of geometry processing tasks: As examples, Section 4 applies geodesics in shape spaces to shape morphing, parallel transport to deformation transfer, and the exponential map to shape exploration. We also demonstrate the efficient handling of isometric deformations by applying our algorithms to shape deformations that satisfy user given constraints.

2 Computing in Shape Space

In the following, we present the essentials for computing in spaces of shapes. By shapes we refer to triangular meshes in Euclidean 3-space. A mesh can be separated into its *connectivity*, i.e., the mesh structure, and its *geometry* which encodes the assignment of vertices to points in 3D. For the time being we require all meshes to be 2-dimensional manifolds – boundaries are allowed. That means that any two triangles share a common edge, or share a common vertex, or have no points in common. Furthermore the star of any non-boundary vertex is a topological disk, and is a half disk for a boundary vertex. In what follows, we keep the connectivity fixed, and change only the vertex positions.

Shape Space. Given a fixed simplicial complex, we consider the space $\mathfrak{S}$ of all immersions of this connectivity in Euclidean 3-space. Such an immersion is seen as a point in shape space, and represented by one large vector $P \in \mathbb{R}^{3m}$ concatenating the m vertices of the complex. The tangent space $T_M\mathfrak{S}$ of $\mathfrak{S}$ at the shape M is given by all discrete vector fields on M. That means a tangent vector $X \in T_M\mathfrak{S}$ assigns a vector $X_p \in \mathbb{R}^3$ to each vertex p of the mesh M. A smooth deformation of M is a mapping $\varphi : [0,1] \times M \to \mathbb{R}^3$, such that each vertex path $p(t) := \varphi(t,p)$ is smooth; equivalently the mapping $P : [0,1] \to \mathbb{R}^{3m}$ is smooth. Given a deformation of M,

$$X(t) := \left(\frac{d}{dt}p(t)\right)_{p\in M}$$

is called the *deformation field* at time t. A deformation of M may have special properties. For example, we can express an affine deformation as $p(t) = A(t) \cdot p + a(t)$ with smoothly varying matrix $A(t)$ and translation vector $a(t)$. Further, if the matrix $A(t)$ is orthogonal at every instant of time, the deformation is a smooth rigid body motion. Such a deformation preserves inter-point Euclidean distances. We can classify the type of deformation by studying the vector fields $X(t)$.

Lemma 1 (Rigid Deformations) *A deformation of a shape M is rigid, if and only if each component of the deformation field $X(t)$ can be expressed as*

$$X_p(t) = \bar{c}(t) + c(t) \times p(t) \tag{1}$$

with smoothly varying vectors $\bar{c}, c \in \mathbb{R}^3$.

See e.g. [Bottema and Roth 1990] for a proof. Note that $\bar{c}$ and c are constant for each shape position $M(t)$, but vary over time.

Recall that a mesh is deformed *isometrically* if distances measured on the mesh are preserved during deformation. Equivalently the length of each edge in the triangulation has to be preserved. Differentiating the squared length $\|p(t) - q(t)\|^2$ of each edge yields:

Lemma 2 (Isometric Deformations) *A deformation of a shape M is isometric if and only if*

$$\langle X_p(t) - X_q(t), p(t) - q(t) \rangle = 0 \tag{2}$$

holds for each edge $(p(t), q(t))$ of the mesh $M(t)$, where $\langle\ ,\ \rangle$ denotes the canonical inner product in $\mathbb{R}^3$.

Vector fields corresponding to isometric deformations are called *Killing vector fields* [do Carmo 1992]. The set of Killing fields on M is a linear subspace of $T_M\mathfrak{S}$. Vector fields of rigid motions as described in Lemma 1 are a special case of Killing fields.

Designing Metrics. We use the following design paradigm when it comes to defining a metric, i.e., an inner product: Given a property of a shape to be preserved during deformation, we translate this property to an equivalent condition on deformation fields. The norm of a deformation field is derived from this condition.

If we want to preserve pairwise Euclidean distances between vertices of a shape M, the only allowable deformations are rigid body motions. Deformation fields of rigid deformations are characterized by Lemma 1. In a first step, we extract the rigid component of a deformation field. This means we project a given deformation field onto the subspace of instantaneous motions: Given X, the residue of this projection is computed in terms of $\bar{c}, c \in \mathbb{R}^3$ as

$$\pi_X(c,\bar{c}) = \sum_{p \in M} \langle X_p - \bar{c} - c \times p, X_p - \bar{c} - c \times p \rangle.$$

This yields a quadratic function in the variables c and $\bar{c}$ which can be minimized easily for any given X. If c^* and $\bar{c}^*$ are minimizers of π_X, the projection X' is given by $X'_p = \bar{c}^* + c^* \times p$. Given deformation fields X and Y, we define,

$$\langle\!\langle X, Y \rangle\!\rangle^R_M := \sum_{p \in M} \langle X_p - X'_p, Y_p - Y'_p \rangle. \tag{3}$$

This expression is linear and symmetric in both of its arguments. Moreover, it depends on the shape M, i.e., the point of $\mathfrak{S}$ at which it is evaluated. Therefore, (3) defines a (semi-) Riemannian metric on the space $\mathfrak{S}$.

To devise a metric in the case of isometric deformations, we could proceed along the same lines as for rigid deformations, i.e., subtracting the part that preserves isometry. Unfortunately Killing fields are harder to express explicitly. In addition, the given mesh might not be flexible at all. In such cases, there are no Killing fields except instantaneous motions as characterized by Lemma 1. To overcome this we do not try to be isometric in the strict sense, but deform a shape as isometrically as possible. To achieve this we take

$$\langle\!\langle X, Y \rangle\!\rangle^I_M := \sum_{(p,q) \in M} \langle X_p - X_q, p - q \rangle \langle Y_p - Y_q, p - q \rangle \tag{4}$$

as the inner product of two deformation fields. This expression is symmetric and bilinear and hence defines a (semi-) Riemannian metric. The corresponding quadratic form reduces to the square of (2) and effectively penalizes non-isometric deformations.

We now consider geodesics in shape space with the above metrics (3) and (4). Recall that a geodesic is a locally shortest curve, i.e., given two points on a geodesic the part between those points is a local minimum of the length functional with respect to small perturbations of the curve. For the metric (3) this means that the length of a deformation decreases as the deformation becomes more rigid. Analogously for metric (4), the length of the deformation decreases as the deformation becomes more isometric. See Figure 2 for a comparison of metrics.

Deformation Fields of Vanishing Length. The introduced metrics are only semi-Riemannian since there are non-vanishing deformation fields X with $\langle\!\langle X, X \rangle\!\rangle_M = 0$. This means that any velocity vector field (1) of a rigid body motion has vanishing norm in the metric (3). All shapes which are congruent to a given shape M form points of a *fiber* in $\mathfrak{S}$. Any smooth curve in a fiber has length zero and corresponds to a smooth rigid body motion of M. Likewise any Killing field has vanishing norm with respect to (4), and the fibers of $\mathfrak{S}$ are formed by the isometric shapes to a given one. This observation also shows that shortest paths (geodesics) in the described metrics are only unique up to changes within the fibers.

Figure 2: *Comparing the as-rigid-as-possible shape metric (left) with the as-isometric-as-possible shape metric (right).*

To overcome this problem, we add a small regularization term to the length which is minimized by geodesics. The obvious choice for this term is the L^2 shape metric: Given vector fields X, Y on a shape M, let

$$\langle\!\langle X, Y \rangle\!\rangle^{L^2}_M := \sum_{p \in M} \langle X_p, Y_p \rangle A_p, \tag{5}$$

where A_p is one-third of the area of all triangles adjacent to the vertex p. Blending the L^2 inner product with the metrics (3) or (4) yields Riemannian metrics

$$\langle\!\langle X, Y \rangle\!\rangle_{M,\lambda} := \langle\!\langle X, Y \rangle\!\rangle_M + \lambda \langle\!\langle X, Y \rangle\!\rangle^{L^2}_M$$

that have the same visual behavior as their semi-Riemannian counterparts if λ is chosen appropriately. We set λ to be 0.001.

Remark. The proposed metrics work in the general setting of straight line graphs embedded in Euclidean 3-space. Further, by adding additional edges, we can selectively stiffen parts of objects which we want to deform rigidly. For example, adding the second diagonal of a quad as *stiffener* allows us to apply all metrics to quad meshes.

3 Algorithms for Geodesics in Shape Space

We now describe how to solve the following problems: (a) *Boundary Value Problem*: Given two isomorphic shapes, how to compute a geodesic path joining them; (b) *Initial Value Problem*: Given a shape and a deformation field, how to compute a geodesic that originates at this point, and moves in the direction of the deformation field. We solve both problems using a multi-resolution framework. Propagating the solution from coarser to finer resolutions not only leads to faster convergence, but also makes the approach more robust.

The Boundary Value Problem. The input to the boundary value problem are two compatible meshes M and N, i.e., the underlying simplicial complexes are isomorphic. Such meshes can be obtained using techniques from [Kraevoy and Sheffer 2004], [Schreiner et al. 2004] or [Sumner and Popovič 2004]. In all the examples shown, the isomorphism was either explicitly available (given by vertex enumeration), or computed (as in example 11) using a marker assisted correspondence algorithm as proposed in [Sumner and Popovič 2004]. Combining methods from [Hoppe 1996] and [Garland and Heckbert 1997], the input meshes are concurrently decimated to preserve correspondences across all resolutions of the resulting progressive meshes. Edges in the two meshes are paired according to the underlying isomorphism. In each iteration, the edge pair with minimal combined cost is collapsed. The output of this stage are two mesh hierarchies $(M^0, M^1, \ldots, M^l = M)$ and $(N^0, N^1, \ldots, N^l = N)$. To get an initial estimate of a geodesic

path we linearly blend the meshes M^0 and N^0. We call the resulting polyline in shape space a *path*.

In Euclidean space one estimates the length of a curve by inscribing a polyline, and accumulating the length of its individual segments. It is guaranteed that this converges to the length of the curve as the polyline is refined. We take a similar approach in shape space but in a Riemannian fashion, i.e., the metric that measures the length of a segment may *change* along the path. Assume the vertices of the polyline are given by shapes $P_0, P_1, \ldots, P_{n+1}$ (we drop the superscripts indicating mesh resolution), and the segments are given as $X_0, X_1, \ldots, X_n$. For a given metric, we define the symmetric energy of the polyline P as

$$E(P) := \sum_{i=0}^{n} \left(\langle\!\langle X_i, X_i \rangle\!\rangle_{P_i} + \langle\!\langle X_i, X_i \rangle\!\rangle_{P_{i+1}} \right). \tag{6}$$

This approximates the sum of squared lengths of the polygon edges X_i, and is a discretization of the energy $\int \langle\!\langle X, X \rangle\!\rangle_{P(t)} dt$ of a curve $P(t)$ in shape space. It is well known that the local minimizers of this energy are geodesics in a scaled arc length parametrization [do Carmo 1992]. We allow the vertices of the intermediate shapes P_i, $i = 1, \ldots, n$, to vary in order to minimize the energy of the polyline. A quasi-Newton method is used to minimize (6). After attaining a local minimum of the energy at a given resolution we perform refinement which comes in two flavors: (a) *Space Refinement*: increase the resolution of the meshes and, (b) *Time Refinement*: refine the path by inserting more vertices in the polyline. These steps are mutually independent, and can be applied in any order. After refinement, we repeat the optimization on the new path, see Figure 3 for an illustration.

We describe the refinement step in more detail. Refinement with respect to time – inserting shapes on the path – is easy. We linearly blend neighboring meshes to refine the path. Space refinement – increasing mesh resolution – is more subtle. The boundary meshes P_0^k and P_{n+1}^k are the easy ones. We simply increase the resolution of the progressive meshes to get refined boundary meshes P_0^{k+1} and P_{n+1}^{k+1}. Then we propagate the added detail to the intermediate meshes $P_1^k, \ldots, P_n^k$ on the path. We construct these refined meshes in the following way. The process is described for detail transfer from P_0^{k+1} to the intermediate meshes. Each new vertex p that was added during mesh refinement is projected onto P_0^k. We store the index of the face f that carries the projection p', and compute barycentric coordinates of the projected vertex with respect to the face f. If N_f is the normal of the face f, we store an additional coordinate given by $\langle p - p', N_f \rangle$. This set of local coordinates allows us to compute the spatial coordinates of each added vertex p relative to any of the intermediate meshes. Depending on the location of P_i^k inside the polyline the transferred details are blended: Assume that $P_{i,j}^{k+1}$ is the mesh with detail transferred from P_j^{k+1}, $j = 0, n+1$. The refined mesh at position i of the path is given by

$$P_i^{k+1} = \frac{n+1-i}{n+1} P_{i,0}^{k+1} + \frac{i}{n+1} P_{i,n+1}^{k+1}.$$

This equation also holds for the boundary meshes.

The Initial Value Problem. We solved the boundary value problem via the variational property of geodesics, i.e., minimization of the energy (6). Alternately, geodesics can be described by the Euler Lagrange equation of the energy. This so called *geodesic equation* expresses vanishing geodesic curvature. We need this equation to formulate and solve the initial value problem. Let us fix a mesh M and assume that all its vertices move along smooth

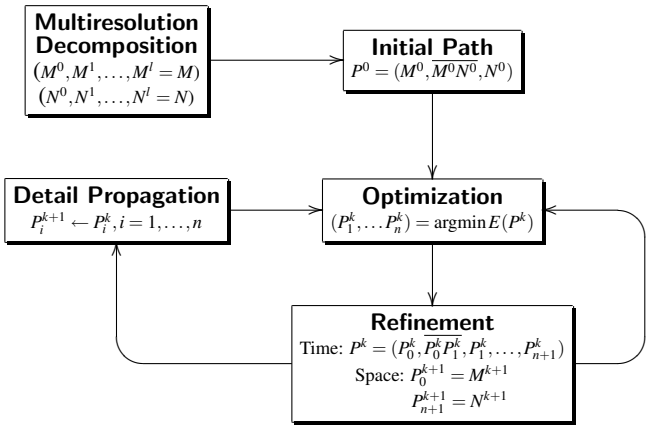

Figure 3: *Processing pipeline to solve the boundary value problem for geodesics in shape space. A path on resolution k is a polyline in shape space whose vertices are compatible meshes. The initial path consists of the meshes M^0, N^0, and their linear blend $\overline{M^0 N^0}$.*

curves. Concatenate all vertices of the mesh to one big vector $P \in \mathbb{R}^{3m}$, where m denotes the number of vertices of M. Let

$$F\left(t, P, \dot{P}\right) := \langle\!\langle \dot{P}(t), \dot{P}(t) \rangle\!\rangle_{M(t)},$$

where $\langle\!\langle \, , \, \rangle\!\rangle_M$ refers to our choice of metric as explained before. We have to minimize $\int F dt$. Since F does not explicitly depend on t, the Euler-Lagrange equation of the problem $\int F dt \to min$ can be integrated once [Gelfand and Fomin 1963]. Thus, similar to the well-known relations of Clairaut and Sauer for geodesics on rotational or helical surfaces (see e.g. [Pottmann and Wallner 2001]), the geodesic equation in shape space is a system of first order ODEs,

$$F - \dot{p}_i^k \frac{\partial F}{\partial \dot{p}_i^k} = C_{i,k}, \quad i = 1, \ldots, m, \; k = 1, 2, 3, \tag{7}$$

where $p_i = (p_i^1, p_i^2, p_i^3)$ denotes the i-th vertex of the mesh M.

Given a shape M_0 and an initial vector X_0, we can determine the constants $C_{i,k}$ using (7). These values are preserved along the geodesic path starting from M_0 in direction X_0. Starting with this data, we build a polyline in shape space by explicit Euler steps. We choose a step size Δt and set

$$M_{i+1} := M_i + \Delta t X_i, \quad i \geq 0.$$

Now we have to solve the non-linear system (7) to determine the vector X_{i+1}. We do this by minimizing the function

$$f(p) = \sum_{i=1}^{m} \sum_{k=1}^{3} \left(F - \dot{p}_i^k \frac{\partial F}{\partial \dot{p}_i^k} - C_{i,k} \right)^2.$$

We use X_i as an initial guess for X_{i+1}. This scheme easily integrates into our multiresolution approach. On the coarsest level we use the scheme just described. After increasing the resolution of M_0 we can transfer the detail to the previously computed path, and take the difference vectors of this path as initial guesses for the optimization. Figure 4 shows a schematic illustration of the steps.

Robustness, Accuracy, and Complexity. To conclude this section, we examine the numerical properties of the proposed algorithms. Since we are using techniques from optimization theory, the resulting geodesic will depend on the initial choice of the path. This dependence is roughly proportional to the number of vertices of the reduced meshes with which we start. Starting with coarse

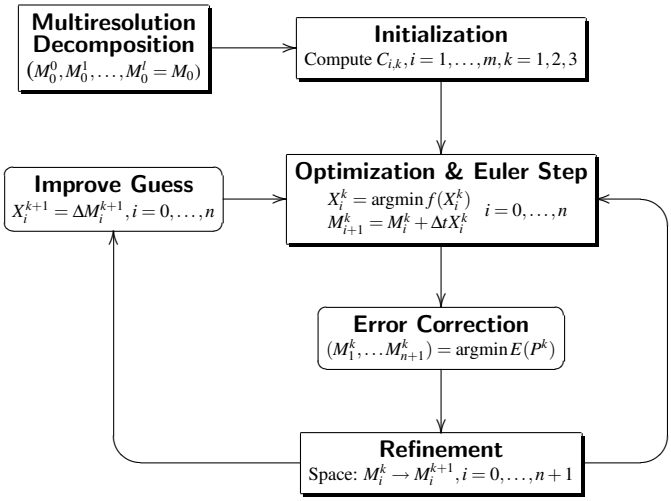

Figure 4: *Processing pipeline to solve the initial value problem for geodesics in shape space. Error correction and guess improvement are optional steps and may be skipped.*

meshes reduces the degrees of freedom, and helps to find a visually pleasing geodesic almost independent of the initial path

Concerning the error distribution, the following behavior is expected: If the shapes are isometric the error should be within some ε bound. If the shapes are not isometric the error should concentrate at the right spots, i.e., at joints if we consider articulated motions, see Figure 5. Recall that we do not use any information about articulated components. The yellow regions show high relative deviation in edge length. These regions can be used to extract individual finger bones. The maximum error along the geodesic is bounded by the deviation in edge lengths between the input meshes. This is a typical behavior exhibited by all examples we considered.

To reduce the error made during extrapolation, we provide an automatic error correction tool. During extrapolation we want to stay as close as possible to the fiber of M_0. If we did not leave this fiber during extrapolation the length of the resulting path will vanish. Otherwise we will have a positive length. We can shorten the polyline by not only allowing the intermediate meshes to vary, but also the final shape M_{n+1}. It is important to use the semi-Riemannian metrics (3) or (4) during error correction. Otherwise the fiber of M_0 collapses to a point and the geodesic will contract to M_0.

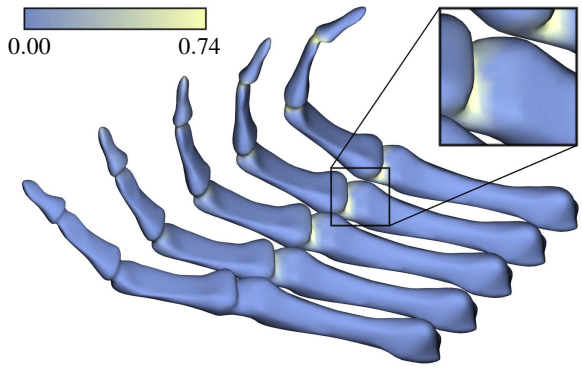

Figure 5: *Error distribution along geodesic that joins non isometric shapes. The maximum relative deviation in edge length relative to the reference pose shown at the bottom is 74%. Maximum stretch occurs at the joints and was introduced during modeling the bent finger from the pose shown at the bottom. Stretch along the geodesic is bounded by this a-priori error. Computation time: 2sec.*

Typically we reduce the input meshes to about 100 triangles for shapes of genus zero. For higher genus more triangles may be required. After every optimization step we alternate between space and time refinement, doubling the number of triangles or shapes on the path, respectively. The elephant sequence of Figure 1 with a final resolution of 85K triangles per shape was created in this manner and took 274 seconds to finish on an AMD64 4000+ with 2 GB main memory.

Evaluating metric (4) has complexity $O(m)$. The energy (6), and its gradient can be evaluated in $O(mn)$ time. Hence the algorithm to solve the boundary value problem has complexity $O(mn)$. Evaluating each function from (7) is linear in the number of vertices of a mesh, but since we have one function for each vertex the algorithm to solve the initial value problem has quadratic complexity. Since evaluating the energy and its gradient is cheap we use a limited memory BFGS method for optimization [Liu and Nocedal 1989].

4 Applications

Morphing Using Geodesics. Our method, as currently implemented, applies to meshes of identical connectivity. Different poses of a model of course have this property. For different shapes, various researchers have proposed methods for generating compatible meshes; see e.g. [Schreiner et al. 2004; Kraevoy and Sheffer 2004]. The mesh for the models in Figure 11 was produced using the method of [Sumner and Popovič 2004]. It would also be possible to integrate a multiresolution method as proposed by [Schreiner et al. 2004] into our pipeline in Figure 3. To do this, one could replace the initialization stage with the decimation step from [Schreiner et al. 2004] and add a new stage between Space Refinement and Detail Propagation to generate correspondences for added details. After compatible remeshing we can compute a morph which is simply a geodesic path between M and N in shape space using as-rigid-as-possible (3) or as-isometric-as-possible (4) metrics. Figure 1 and 6 show interpolation with subsequent extrapolation using the as-isometric-as-possible shape metric.

The intermediates generated by linear blending may be too degenerate to serve as starting configuration for metric (4). In this case metric (3) can be used beneficially during the first round of optimization since the as-rigid-as-possible metric tries to align shapes globally.

Shape Exploration. Given a set of compatible meshes, our goal is to provide an intuitive and interactive metaphor for exploring the space spanned by the input models. Other approaches for more specific tasks e.g. for parametrization and animation of human body poses [Allen et al. 2003; Anguelov et al. 2005] as well as example based methods [Sloan et al. 2001] can be found in the literature.

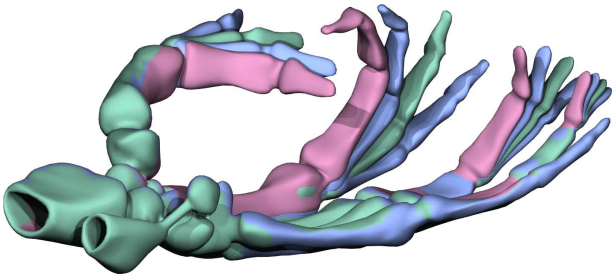

Figure 6: *Geodesic interpolation and extrapolation using the as-isometric-as-possible shape metric. The blue shapes serve as input poses. Green poses are interpolated and purple ones are extrapolated. Each mesh comprises 26K triangles. The final path consists of 128 shapes and was computed in 20 seconds.*

Figure 7: *Result of shape exploration. Sampling the curve shown in the design domain of Figure 8 yields the above sequence of poses. The 2D design domain has been subdivided twice to minimize metric distortion. Interaction with the design domain produces results in real-time.*

In the preprocessing phase, we build a distance matrix relating the input shapes, where distance between any pair of meshes is measured as their geodesic distance. Applying metric multidimensional scaling [Cox and Cox 2001] to the distance matrix, we map the input shapes to a set of 2D points, and construct a Delaunay triangulation of this point set. Each edge of this triangulation is assigned two scalar values: One is the geodesic distance between meshes associated to the endpoints of the edge; the other is the length of the curve obtained by linear interpolation of those meshes. If the values are close to one another we can safely use the linear blend of the endpoint meshes instead of computing a more expensive geodesic path – the differences between both curves are negligible if the threshold is small. Using this observation, a triangle is refined if the difference of the geodesic distance between any pair of its vertices and the length of the corresponding linear blend is greater than a user defined threshold.

Each triangle selected for refinement is split into 4 triangles by introducing midpoints for each edge. The midpoint of an edge is assigned the shape which is halfway along the geodesic connecting the edge's vertex shapes. Notice that while we use the connectivity from the 2D triangulation, all distance computations are done in the original shape space.

To map a point from the 2D design plane to shape space we determine its barycentric coordinates with respect to the triangle containing the point. These coordinates are used to blend the meshes associated to the triangle's vertices. In this way we get a piecewise linear approximation of shape space. In the exploration phase, the user interactively traverses shape space by drawing curves on the 2D projection. A corresponding path in shape space is computed in *real-time* using the above mapping procedure. Figure 8 shows the planar triangulation resulting from 5 input poses of the Armadillo model after two refinement steps. We show a curve drawn in the exploration phase – a few poses along the curve are shown in Figure 7. The precomputation step for this example took about an hour with a final resolution of 100K triangles of the Armadillo.

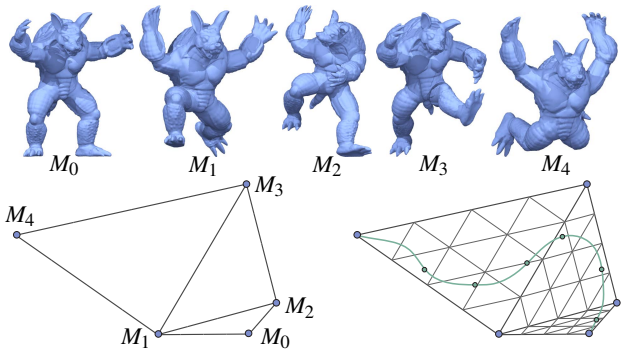

Figure 8: *Shape Exploration. The top row shows the input poses for shape exploration. The bottom row shows the 2D design plane with which the user interacts by drawing curves. All nodes of the planar triangulation shown on the right are computed in an independent precomputation step and stored for later use. Figure 7 shows a sampling of the curve displayed on top of the domain on the right.*

Constraint Shape Deformation. Our framework allows us to deform a shape in an intuitive way. The user can select any vertex or part of the shape, and modify its position and orientation. The deformation induced on the remaining part of the mesh is guided by the metric used. In contrast to the shape interpolation problem where the final location of each vertex is known, now only some of these locations are known.

Assume we are given a subset $\{p_i\}_{i\in I}$ of vertices of M. Each of these vertices has some desired location p_i' as assigned by the user. We do not move the vertices p_i to the location p_i' in one step, but rather minimize the squared distances $\|p_i - p_i'\|^2$ to iteratively move the vertices. Moving p_i induces distortion on the shape M. We measure these distortions by storing an unmodified copy M_0 of M and a path P that joins M_0 with the modified mesh M. Minimizing distortion while moving the vertices to their desired locations means minimizing the energy (6) of the path P and the sum of squared distances simultaneously. It may not be possible to move each vertex p_i exactly to its desired location if we want the distortion of the shape to be minimal. To account for this tradeoff we consider

$$f_\lambda(P) := \lambda E(P) + (1-\lambda)\sum_{i\in I}\|p_i - p_i'\|^2$$

with $\lambda \in [0,1]$. Minimizing this function for small values of λ we will get each vertex p_i close to its desired location, but in the process may incur large distortions of the mesh. Increasing the value of λ will reduce distortion while approximating the desired locations of the p_i as good as possible. Integrating this way of measuring distortion in our algorithms we get a multi-resolution mesh modeling and correction tool. This tool can be adapted to the needs of the application by replacing the metric with an application specific one. See Figure 9 and 10 for examples. The deformation of Figure 10 (d) was obtained by equally blending metric (4) and a metric whose geodesics are volume preserving as presented in [Kilian 2007].

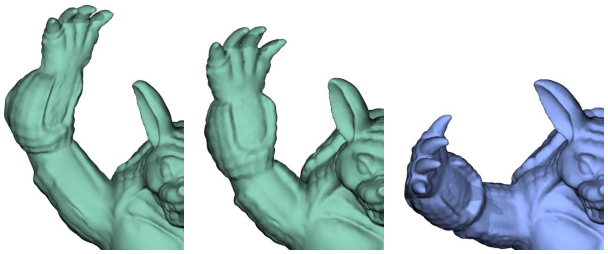

Figure 9: *Distortion correction. Using constraint freeform deformation we are able to fix errors the designer makes during modeling. Compared to the reference model (right) the deformed model (left) shows a change in volume of 25%, whereas the corrected model (center) shows only a change of 3.6%.*

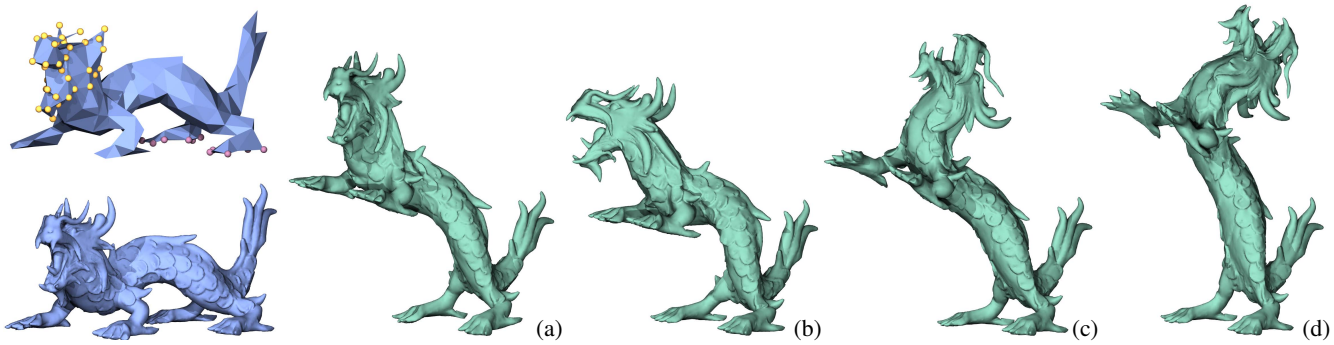

Figure 10: *Freeform deformation by moving and reorienting the dragon's head. Distortion is controlled by metric (4). The model comprises 65K faces on the shown resolution. Vertex constraints are specified on the initial resolution of 400 faces (see top left for constraint vertices). Computation time for each deformation is 120 seconds without user interaction. Volume distortion: (a)* 0.4%, *(b)* 1.4%, *(c)* 0.4%, *(d)* 1.8%.

Deformation Transfer Based on Parallel Transport. Deformation transfer refers to the problem of transferring motion or a deformation sequence from a source model onto a target model [Sumner and Popovič 2004]. In the shape space setting, deformation transfer is about how to meaningfully transfer a curve from one part of shape space to another. To solve this problem, we use the concept of parallel transport which is a natural way to transport geometric data along smooth curves in a manifold.

Given a geodesic $\gamma : [0,1] \to \mathfrak{S}$ and a vector field $X(t)$ along γ, a vector field is called *parallel* if the angle $\langle\!\langle X(t), \dot{\gamma}(t) \rangle\!\rangle_{\gamma(t)} = C$ is constant with the additional property that $X(t)$ rotates minimally around $\dot{\gamma}(t)$. Given the vector $X(0)$ there is a unique parallel vector field $X(t)$ along γ that extends $X(0)$. Given γ and $X(0)$ we can compute the constant C. To get a discrete sampling of the parallel vector field that extends $X(0)$ we sample the geodesic γ at times t_i, starting with $t_0 = 0$. We iteratively minimize $\|Y - X(t_i)\|^2$ with respect to Y under the constraint $\langle\!\langle Y, \dot{\gamma}(t_{i+1}) \rangle\!\rangle_{\gamma(t_{i+1})} = C$ to get the vector $X(t_{i+1})$. This ensures that the computed vector minimally deviates from $X(t_i)$, maintaining the minimal rotation property.

Now we are ready to transfer deformations. Suppose we are given a deformation $\varphi : [0,1] \times M \to \mathfrak{S}$ of some shape M (if we are only given a reference pose M and an additional pose of M, as in [Sumner and Popovič 2004], we join them by a geodesic to get a deformation). This deformation is a curve in shape space. Given another shape N we want to transfer the deformation of M onto N. From an abstract point of view that means we want to attach the curve starting in M to the shape N. This is easy in Euclidean space but difficult in curved spaces. We solve this problem in the following way: We sample the curve φ at n equidistant points M_i, and compute a geodesic γ_0 that joins $M = M_0$ and N. Let $X_i = \Delta M_i$ be a segment (i.e., a discrete tangent vector) of the polyline $(M_0, M_1, \ldots, M_n)$. We transport the vector X_0 along the geodesic γ_0 to get Y_0 and let $N_1 := N_0 + Y_0$. We repeat the above steps with γ_1, M_1, X_1 and N_1 until we reach M_n. Figure 11 shows the result of transferring a deformation from a cat to a lion model that was constructed in this way.

Limitations. The examples show that the deformations obtained using metric (4) generate very natural deformation sequences. However, we do not make any specific attempts to ensure that intermediate poses are free from self-collisions, or do not penetrate surrounding objects. In case of widely varying poses, we may get such penetrations. While user intervention can definitely help to fix such problems, in the future we plan to extend our algorithm to automatically compute collision free geodesics.

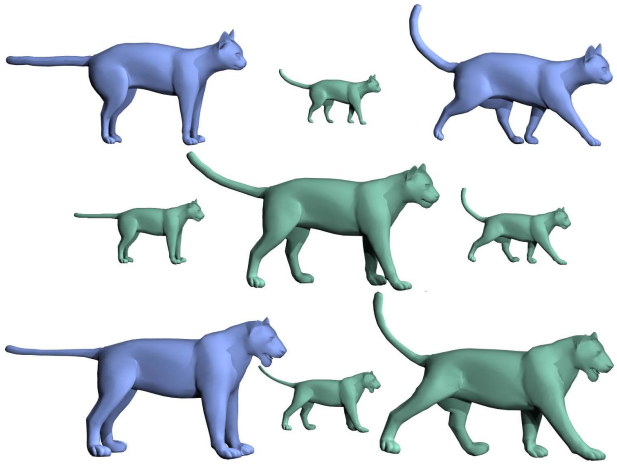

Figure 11: *Deformation transfer. The blue input shapes of the cat (top row) are joined by a geodesic to get a deformation. This deformation is transferred to the blue lion model (bottom row). The middle row shows hybrids generated during deformation transfer.*

5 Conclusion and Future Work

We presented a Riemannian geometry based framework for defining and exploring shape spaces. A key ingredient of such spaces is the associated metric: We introduced metrics that push geodesics towards rigid body motions or towards isometric deformations. We solved the general shape interpolation and extrapolation problems. For meshes which are flexible, the computed shapes along a geodesic are isometric (modulo numerical errors) to the original meshes. For other objects, our results are approximately isometric. This allows us to traverse and explore shape space, with applications to shape design. Additionally we illustrate the generality of our approach by solving several geometry processing tasks in our framework using tools from classical differential geometry. Our algorithms are designed for meshes in correspondence. An interesting future direction is to study geodesics between given meshes over all possible correspondence assignments – the shortest geodesic solves the correspondence problem. Since a combinatorial explosion prevents us from taking a brute force approach, we are looking for an efficient and elegant solution.

Acknowledgements

This work is supported by the Austrian Science Fund (FWF) under grant S9206. We thank J. Wallner and L. J. Guibas for stimulat-

ing discussions, B. Sumner for providing the cat and lion model, M. Botsch and M. Pauly for the Goblin poses, and the anonymous reviewers for their comments and suggestions. Elephant and Armadillo poses are taken from the AIM@SHAPE shape repository, the dragon model is courtesy of the Stanford 3D Scanning Repository and the skeleton hand is available at the Large Geometric Models Archive from the Georgia Institute of Technology.

References

ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH '00*, 157–164.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graphics 22*, 3, 587–594.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. SCAPE: shape completion and animation of people. *ACM Trans. Graphics 24*, 3, 408–416.

BERGER, M. 1987. *Geometry I, II*. Springer.

BOTSCH, M., PAULY, M., GROSS, M., AND KOBBELT, L. 2006. Primo: coupled prisms for intuitive surface modeling. In *Symp. Geom. Processing*, 11–20.

BOTTEMA, O., AND ROTH, B. 1990. *Theoretical kinematics*. Dover Publ.

BRONSTEIN, A., BRONSTEIN, M., AND KIMMEL, R. 2005. Isometric embedding of facial surfaces into S^3. In *Proc. of Scale-Space*, 622–631.

CECIL, T. 1992. *Lie Sphere Geometry*. Springer.

CHARPIAT, G., FAUGERAS, O., AND KERIVEN, R. 2005. Approximations of shape metrics and application to shape warping and empirical statistics. *Foundations of Comp. Math.*, 5, 1–58.

CHARPIAT, G., KERIVEN, R., PONS, J.-P., AND FAUGERAS, O. 2005. Designing spatially coherent minimizing flows for variational problems based on active contours. In *Proc. ICCV 2005*, vol. 2, 1403–1408.

CHENG, H.-L., EDELSBRUNNER, H., AND FU, P. 1998. Shape space from deformation. In *Proc. Pacific Graphics*, 104–113.

COX, T., AND COX, M. 2001. *Multidimensional Scaling*. CRC/Chapman and Hall.

DO CARMO, M. P. 1992. *Riemannian Geometry*. Birkhäuser.

FUNCK, W., THEISEL, H., AND SEIDEL, H. 2006. Vector field based shape deformations. *ACM Trans. Graphics 25*, 3, 1118–1125.

GARLAND, M., AND HECKBERT, P. 1997. Surface simplification using quadric error metrics. In *ACM SIGGRAPH*, 209–216.

GELFAND, I. M., AND FOMIN, S. V. 1963. *Calculus of Variations*. Prentice Hall.

HOPPE, H. 1996. Progressive meshes. In *ACM SIGGRAPH*, 99–108.

HUANG, J., SHI, X., LIU, X., ZHOU, K., WEI, L., TENG, S., BAO, H., GUO, B., AND SHUM, H. 2006. Subspace gradient domain mesh deformation. *ACM Trans. Graphics 25*, 3, 1126–1134.

IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graphics 24*, 3, 1134–1141.

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graphics 24*, 3, 561–566.

KENDALL, D. G. 1984. Shape manifolds, procrustean metrics and complex projective spaces. *Bull. London Math. Soc. 18*, 81–121.

KILIAN, M. 2007. Shapes, metrics, and their geodesics. Tech. Rep. 178, Vienna University of Technology.

KIMMEL, R., AND SETHIAN, J. 1998. Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci. 95*, 8431–8435.

KLASSEN, E., SRIVASTAVA, A., MIO, W., AND JOSHI, S. H. 2004. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE PAMI 26*, 3, 372–383.

KRAEVOY, V., AND SHEFFER, A. 2004. Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graphics 23*, 3, 861–869.

KRAEVOY, V., AND SHEFFER, A. 2007. Mean-value geometry encoding. *International Journal of Shape Modeling 12*, 1.

LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. *ACM Trans. Graphics 24*, 3, 479–487.

LIPMAN, Y., COHEN-OR, D., GAL, R., AND LEVIN, D. 2007. Volume and shape preservation vai moving frame manipulation. *ACM Trans. Graphics 26*, 1.

LIU, D. C., AND NOCEDAL, J. 1989. On the limited memory BFGS method for large scale optimization. *Math. Program. 45*, 3, 503–528.

MEMOLI, F., AND SAPIRO, G. 2001. Fast computation of weighted distance functions and geodesics on implicit hypersurfaces. *J. Comput. Phys. 173*, 730–764.

MÉMOLI, F., AND SAPIRO, G. 2004. Comparing point clouds. In *Symp. Geometry Processing*, 32–40.

MICHOR, P. W., AND MUMFORD, D. 2006. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc. 8*, 1–48.

POTTMANN, H., AND WALLNER, J. 2001. *Computational Line Geometry*. Springer.

SCHREINER, J., ASIRVATHAM, A., PRAUN, E., AND HOPPE, H. 2004. Inter-surface mapping. *ACM Trans. Graphics 23*, 3, 870–877.

SLOAN, P.-P. J., ROSE, C. F., AND COHEN, M. F. 2001. Shape by example. In *Proc. of the 2001 symposium on interactive 3D graphics*, 135–143.

SORKINE, O., LIPMAN, Y., COHEN-OR, D., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Symp. Geom. Processing*, 179–188.

SUMNER, R. W., AND POPOVIČ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graphics 23*, 3, 399–405.

XU, D., ZHANG, H., WANG, Q., AND BAO, H. 2005. Poisson shape interpolation. In *SPM '05: Proc. ACM Symp. on Solid and Physical Modeling*, 267–274.

YEZZI, A., AND MENNUCCI, A. 2005. Conformal metrics and true "gradient flows" for curves. In *Proc. ICCV '05*, 913–919.