GEOMETRIC MODELING IN SHAPE SPACE [1]

Liu Gang
(*Department of Mathematics, Zhejiang University, Hangzhou 310027*)

April 1, 2008

[1] Report for the course of Digital Geometry Processing

Abstract

In this report, we conclude some of these papers which involve solving the trajectory problem in boundary representation based shape interpolation as one of their main contributions. These three main methods are: Computing intrinsic solution, Poisson shape interpolation and Shape space method. We compare these methods from theoretical views and their applications. In this report, we take the main emphasis on shape space method and pay more attention to 2-D method construction and 3-D promotion in this method.

Contents

Chapter 1

Introduction and past works

It is well-understood that there are two major issues in Boundary representation (B-rep) based shape interpolation. The first is how to find a feature preserving correspondence map between the given models, known as *correspondence problem.* The second is how to interpolate the positions for each pair of corresponding vertices along predetermined paths, known as *trajectory problem.*

1.1 Introduction

Shape interpolation, also known as shape blending or morphing, has been widely applied to various aspects of computer graphics industry, e.g. illustration and animation. Given two input models, shape interpolation can generate a sequence of intermediate shapes which gradually changes from the source shape to the target one.

This report considers three of these novel approaches based on geometric intrinsic variable, Poisson equation and Shape space theory.

1.2 Past Works

In 1992, Sederberg and Greenwood decomposed the deformation of in-between shapes into stretch and bend. And later Sederberg et al. propose a geometric algorithm with global optimization to ensure these blended polylines are close without local self-intersection. Liu Ligang and Wang Guojin generalize this idea to 3D meshes. But all the final morphing results are dependent of the computation order of dihedral angels and edge length. To handle large scale rotation in boundary representation based morphing, a lot methods which give various constraints have been proposed. Poisson shape interpolation is representative to these kind methods and takes both vertex coordinates and surface orientation into account. Since this method consider preserving rigid (as rigid as possible) as the main aim morphing, the result can not radically avoid shrinkage problem and the interpolation shapes might be unstable in variation of pose orientations.

In 2004, Klassen et al. presented a computational approach to spaces of curves to interpolate the given two models successfully and gave more application in shape analysis. But it has no natural extension to surface. The paper Geometric modeling in shape space proposed a general 3D computational modeling framework and give algorithms to shape deformations that satisfy various user given constraints.

The main body of this report is divided as follows.
Chap.2 analyzes the main steps of Computing intrinsic solution and Poisson shape interpolation.
Chap.3 discuss the idea using geodesic paths on shape space to shape analysis and geometric modeling.
Chap.4, list the main contributions of these methods in shape interpolation and compare latter two methods in examples and applications.

Chapter 2

Computing intrinsic solution and Poisson shape interpolation

In this chapter, we analyze two classical methods Computing intrinsic solution and Poisson shape interpolation in the simple theoretical view and applications.

2.1 Computing intrinsic solution

In computing intrinsic solution, the intrinsic variables consisting of edge length and the angels between edges in 2-D or edge length and the spherical coordinates in the moving coordinate system constructed by the previous points in 3-D should be computed first. These variables can be determined by the polylines uniquely and generate the same polylines differed by a rigid body motion. Interpolating these two intrinsic variables determined by the given two polylines linearly can naturally generate the in-between intrinsic variables and polylines can be reconstructed soon. To preserve the intermediate polylies closed when the reference and target polylines are closed is the main problem in this method. By using Lagrange multipliers method, we find the adjustment to make the polylines closed under changing the edge length only. Since almost any surface representation (spline, implicit surface, volumetric) can be converted with arbitrary accuracy to a quadrilateral mesh by polygonization processes, this geometric method should be generalized to quadrilateral meshes. The intrinsic variables of quadrilateral can be established by iteratively defining the spherical coordinates from the boundary coordinates. The interpolation implementation is similar. The picture Fig.1 showed below is the transformation from a 3-D digitalized sculpture into an "S" shaped subject.

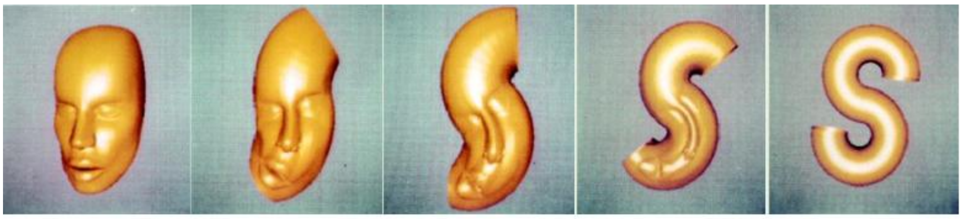


Fig 1. The transformation from a 3-D digitized sculpture into an "S" shaped subject, The intermediate frames are interpolated at time $t = 0.0, 0.25, 0.5, 0.75, 1.0$

2.2 Poisson shape interpolation

In the Poisson shape interpolation approach, we formulate the trajectory problem of shape interpolation as solving Poisson equations defined on a domain mesh. By Constructing the non-linear gradient field which involves both vertex coordinates and surface orientation. There are three steps to accomplish this program.

1. Compute correspondence map and generate compatible meshes from two input 3D meshes.

This method requires that the source model and the target one should be represented by compatible meshes, i.e. meshes with the same connectivity. These two models can be interpreted as three scalar fields (vertex positions) defined on a common domain that is actually an abstract mesh structure called domain mesh. In this report, we assumes the input models and domain meshes are all single-connected and 2-manifold triangular meshes throughout this report. We uses various feature-preserving remeshing methods. In the implementation, base domain is constructed after manually selected several pairs of corresponding feature vertices. Then, both source model and target model are parameterized onto the common base domain and relaxation is performed to reduce the parameterization distortion. Finally, the target model is remeshed using the connectivity of the source model. Iterative error-driven vertex relaxation and edge splitting are performed until approximation error is under user-specified threshold.

2. Compute and decompose the local transformation

For each corresponding triangle pair of compatible meshes, determine the local transform from source triangle to target one and decompose the transform into rigid rotation and stretch part.

Let T_0 and T_1 be a pair of corresponding triangles in source model mesh S_0 and target mesh S_1 which has been remeshed, respectively. we denote ν_0^i and ν_1^i $(i = 1, 2, 3)$ as corresponding vertices, n_0 and n_1 as corresponding unit face normals. For gradients are translation invariant, we choose the first vertex ν_j^1 $(j = 0, 1)$ of each triangle as the origin, and the three axes of affine frames F_j are $\nu_j^2 - \nu_j^1$, $\nu_j^3 - \nu_j^1$ and n_j. Thus we have constructed the local reference system. The unique transform matrix H can be determined such that $F_1 = H(F_0)$, that is the matrix H can be regard as the deformation gradient relating the reference configuration F_0 and the present configuration F_1. We factorizes the deformation gradient tensor into the rigid rotation part and the pure stretch part with the polar decomposition. That is, $H = RS$, where R is the closest rotation matrix to H in Frobenius norm, and S is a symmetric, positive definite matrix(See the Fig2). We define the local continuous transform function H_t in the time $t(t \in [0, 1])$ for a given transition state $h(\nu, t) = t$ as $H_t = R_t((1 - t)I + tS)$, where The function $h(\nu, t)$ is named transition state function whose value also lies in $[0, 1]$, and satisfies $h(\cdot, 0) = 0$ and $h(\cdot, 1) = 1$, The transition state function is used to provide flexible non-uniform controls. Since there is no prior knowledge of movements in morphing process we choose the transition state function in the simplest form $h(\nu, t) = t$; R_h is the rotation matrix defined by linearly interpolating the rotation angle of R using quaternion, and I is the identity matrix. Now we have the general detail transformation H_t then compute in the each triangle .We apply H_t to three source gradient vectors simultaneously for generating interpolated gradient vectors. That is,

$$G_t^i = H_t(G_0^i)(i = x, y, z) : \tag{2.1}$$

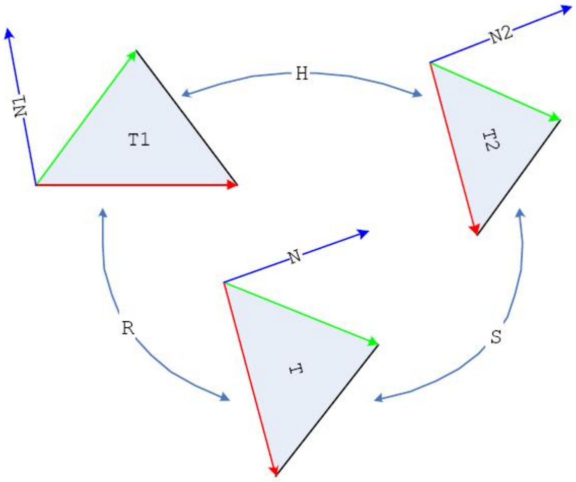


Fig2. The deformation gradient (matrix) H is factorized into the rigid rotation part R and pure stretch part S

After gradient fields interpolation, each triangle is locally transformed by the transformation H_t . The triangles of target mesh become disconnected, i.e., yielding a triangular soup. The Poisson equation stitches together the triangles in the final step.

3. Reconstruct the intermediate shapes by Poisson equation solver

The Poisson equation with Direchlet boundary conditions is formulated as:

$$\begin{cases} \Delta f = \nabla \cdot w \\ f|_{\partial\omega} = f^*|_{\partial\omega} \end{cases} \tag{2.2}$$

Where the divergence operator $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^T$ over a a vector field $\omega = (\omega_x, \omega_y, \omega_z)$ is $\nabla \cdot \omega = \frac{\partial \omega_x}{\partial x} + \frac{\partial \omega_y}{\partial y} + \frac{\partial \omega_z}{\partial z}$; $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the Laplacian operator; f is an umknown scalar function and f^* provides the desirable value on the boundary $\partial\omega$.

In 2003, Tong et al. proposed well-defined discrete differential operators of scalar and vector fields on irregular domains. Based on their result, the discrete Poisson equation on triangular meshes is formulated as follows.

A mesh scalar field f on M is defined to be a piecewise linear combination $f(\nu) = \sum_i f_i \phi_i(\nu)(\nu$ is a point on $M)$. The function $\phi(\nu)$ defined as: $\phi_i(\nu_j) = 1$, if $j = i$ and $\phi_i(\nu_j) = 0$, if $j \neq i$. The discrete gradient of mesh scalar function f on the domain mesh M is expressed as: $\nabla f(\nu) = \sum_i f_i \nabla \phi_i(\nu)$. Given a piecewise constant vector field ω, which has constant value in each triangle of M, the discrete divergence of ω at vertex ν_i is defined as $(div\omega)(\nu_i) := \sum_{T \in N_T(\nu_i)} \omega(T) \cdot \nabla \phi_i|_T A_T$ where A_T denotes the area of triangle T. Therefore, the discrete Laplacian operator on domain mesh M is:

$$\Delta f(\nu_i) = \sum_{\nu_j \in N_\nu(\nu_i)} \frac{1}{2}(\cot \alpha_j + \cot \beta_j)(f_i - f_j), \tag{2.3}$$

where α_j and β_j are two angles opposite to the edge (v_i, v_j).
Finally, the discrete Poisson equation is expressed as $\Delta f \equiv div(\nabla f) = div\omega$. With specified boundary conditions, the above equation can be reformulated as a sparse linear system:

$$Ax = b(t). \tag{2.4}$$

where the unknown vector x represents coordinates to be reconstructed in the intermediate shape. The coefficient matrix A is determined by the (2.3) and the fixed domain mesh; $b(t)$ comes from the smoothly changing gradients (2.1) and boundary conditions.

The Fig3 demonstrated below shows that this method can be using in human pose interpolation. More examples will be given in Chap 4.

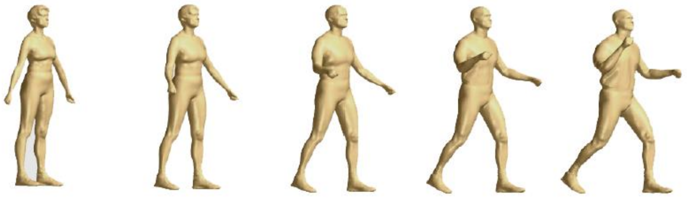


Fig3. Human pose morphing

Chapter 3

Geometric modeling in Shape space

In this chapter, we consider geometric modeling especially shape interpolation by using geometric paths on shape spaces. In the first section we analyze the planar shapes. In the second section we take our emphases on the 3-D surface.

3.1 Analysis of planar shapes using geodesic paths on shape space

Historically, there have been many exemplary efforts in characterization and quantification of shapes. From the view of elegant statistical theory of shapes, Shapes are represented using a *finite* number of salient points or landmarks. Shapes invariant transformation includes rigid rotations, translations and non-rigid uniform scaling, the resulting quotient space is a finite-dimensional Riemannian manifold, called a *shape manifold*. Different shapes correspond to these elements of this space and quantification of shape differences is achieved via a Riemannian metric on this space. But from the Grenander's formulation, shapes are considered as points on some infinite-dimension differentiable manifold. The variations between the shapes are modeled by the action of Lie group (deformations) on the manifold. Low dimensional groups, such as rotation, translation, and scaling, keep the shapes unchanged, while high dimensional groups smoothly change the object shapes, a central idea behind deformation template theory. The action of diffeomorphisms group suffers from a high cost in computing.

This part of research makes contribution over the past works in several facets. We consider each contour as a continuous curves avoiding to finding landmarks and deal more intrinsically with the shape spaces to avoid the diffeomorphism of Euclidean spaces. The main idea proposed here is the use of computational differential geometry, i.e., a computational analysis of shapes using differential geometry of both curves and spaces of curves. In the rest of this chapter, we present the framework in details.

1. Geometric representation of closed, planar curves

In this section, we assume that curves boundaries are closed curves with a single component views as closed immersed curves in the plane. We want to find the shape representation that is invariant to these transformations, such as rigid motions (including rigid rotation and translation in $\mathbb{R}^2$) and uniform scaling of $\mathbb{R}^2$ because they represent the same shapes.

We assume that curves parameterized by arclength: $\alpha(s) = (\alpha_1(s), \alpha_2(s)) : \mathbb{R} \to \mathbb{R}^2$ satisfy two conditions: 1. periodic: $\alpha(2\pi + s) = \alpha(s)$; 2. Tangent unit length $|\alpha'(s)| = 1$. Associated with each α, there is a tangent indicatrix $\nu : \mathbb{R} \to \mathbb{S}^1 \subsetneq \mathbb{R}^2$ given by $\nu(s) = \alpha'(s) = e^{j\theta(s)}(j = \sqrt{-i})$, where $\theta(s)$ is the *direction function.*

From the next paragraph, we will construct the direction function preshape space $\mathcal{C}$ and shape space S

By analyzing the function $\theta(s)$ in the unit circle: $\theta_0(s) = s$, we know any closed curves of rotation index 1 have the direction function taking the forms $\theta = \theta_0 + f, f \in \mathbb{L}^2$. For any two elements: $\alpha, \beta \in \theta_0 + \mathbb{L}^2$, we have $\alpha - \beta \in \mathbb{L}^2$. So the space $\theta_0 + \mathbb{L}^2$.is an affine space but not vector space. In the meaning time, it is obvious that the tangent space at any point is naturally identified with $\mathbb{L}^2$ because $\theta' = f' \in \mathbb{L}^2$

Now we will give more restricts: Because addition of a constant to the direction function θ results in a rotation of the corresponding curve in the plane. That is: $\nu_1(s) = \alpha_1'(s) = e^{j\theta(s)+c} = e^{j\theta(s)}e^c$ we want to mod out by this group action to make shapes invariant to rotation and give the first restriction:

$$\frac{1}{2\pi}\int_0^{2\pi} \theta(s)ds = \pi \tag{3.1}$$

Although any constant can be used instead of π, we choose it to include the identity function in the restricted set.

In the meaning time, $\theta(s)$ must be satisfy the closure condition and this is the second restriction:

$$\int_0^{2\pi} e^{j\theta(s)}ds = 0 \tag{3.2}$$

Consequently, we define 'the set' as a subset of $\theta_0 + \mathbb{L}^2$ satisfying the two conditions mentioned above.

If we define the map $\phi = (\phi_1, \phi_2, \phi_3) : (\theta_0 + \mathbb{L}^2) \to \mathbb{R}^3$satisfying:

$$\begin{aligned} \phi_1 &= \frac{1}{2\pi}\int_0^{2\pi} \theta(s)ds, & (3.3)\\ \phi_2 &= \frac{1}{2\pi}\int_0^{2\pi} \cos(\theta(s))ds, & (3.4)\\ \phi_3 &= \frac{1}{2\pi}\int_0^{2\pi} \sin(\theta(s))ds & (3.5)\end{aligned}$$

Then, $\mathcal{C}$ can be written as $\phi^{-1}(\pi, 0, 0)$, and is a complete submanifold of $\theta_0 + \mathbb{L}^2$ of condimension three. By restricting the $\mathbb{L}^2$-inner product to the tangent space of $\mathcal{C}$, it becomes a Hilbert manifold. But it is a preshape space since it is possible to have multiple elements of $\mathcal{C}$ denoting the same shape. This variability is due to the choice of the reference point($s = 0$) along the curve. For $x \in \mathbb{R}, \theta \in \mathcal{C}$, if we define $(x \cdot \theta)(s) = \theta(s - x) + x$, This operation corresponds to change the initial point($s = 0$) on the closed curve by a distance of x along the curve. By defining the action on the $\mathbb{R}$: $\mathbb{S}^1 = \mathbb{R}/2\pi\mathbb{Z}$($\mathbb{S}^1$ is called *reparametrization group*), $S = \mathcal{C}/\mathbb{S}^1$ is the space of planar shapes under θ representations. S is also a manifold except the set of shapes with rotational symmetries, a negligible set.

2. Geometries of the resulting shape spaces and Computing geodesic on this shape space.

An important tool in the process which analyze shapes and perform statistical inferences on the shape space S is a technique for computing geodesic paths between arbitrary points on the preshape space $\mathcal{C}$. Firstly we will draw infinitesimal tangent lines in the affine spaces $\theta_0 + \mathbb{L}_2$; then project them onto the preshape spaces $\mathbb{C}$. So we need specify the tangent spaces or equivalent the normal spaces on these manifolds and construct a mechanism for projecting points from $\theta_0 + \mathbb{L}_2$ to $\mathcal{C}$

Tangents and Normals to preshape space Rather than specifying the tangent spaces on these manifold, it is easier to describe the spaces of normals to $\mathcal{C}$, inside $\mathbb{L}^2$. The direction derivative $d\phi$, at a point $\theta \in \theta_0 + \mathbb{L}_2$ in the direction of an $f \in \mathbb{L}_2$ (means $\theta'(s) = f(s)$) is defined:

$$\begin{aligned} d\phi_1(f) &= \frac{1}{2\pi}\int_0^{2\pi} f(s)ds = \langle f, \frac{1}{2\pi}\rangle & (3.6)\\ d\phi_2(f) &= -\frac{1}{2\pi}\int_0^{2\pi} \sin(\theta(s))f(s)ds = -\langle f, \sin(\theta)\rangle & (3.7)\\ d\phi_3(f) &= \frac{1}{2\pi}\int_0^{2\pi} \cos(\theta(s))f(s)ds = \langle f, \cos(\theta)\rangle & (3.8)\end{aligned}$$

These functions mean that $d\phi$ is a surjective from $\mathbb{L}^2$ to $\mathbb{R}^3$ and $f \in \mathbb{L}^2$ is tangent to $\mathcal{C}$ at θ if and only if f is orthogonal to the subspace spanned by $\{1, \cos(\theta), \sin(\theta)\}$. So the basis $\{1, \cos(\theta), \sin(\theta)\}$ span the normal space at θ, the tangent space is given as:

$$T_\theta(\mathcal{C}_1) = \{f \in \mathbb{L}^2 | f \perp span\{1, \cos(\theta), \sin(\theta)\}\}$$

Theoretical idea is to move in direction perpendicular to the level set such that their images under ϕ form a straight line in $\mathbb{R}^3$. we will construct the level set as follow. For any $b \in \mathbb{R}^3$, set $\phi^{-1}(b) = \{\theta \in \theta_0 + \mathbb{L}^2 | \phi(\theta) = b\}$ is the level set of ϕ and the level set for $b = (\pi, 0, 0)$ is the preshape space.

For the reason $\phi : \mathbb{L}^2 \to \mathbb{R}^3$, there will be $d\phi : T_\theta(\mathbb{L}^2) \to T_\theta(\mathbb{R}^3) = \mathbb{R}^3$. So for any point in this level set $\theta \in \phi^{-1}(b)$, we need find the nearest point and the straight line connecting these two point perpendicular to the normal vector of this level set. That means give a displacement $d\theta$ moving in $\mathcal{C}$ and orthogonal to this level set. $d\theta$ is the normal vector at θ ($\phi(\theta) = b$) which takes $\phi(\theta + d\theta)$ to the desire point $b_1 \in \mathbb{R}^3$. We can define a Jacobian matrix to approximate the first order.
Iteration method to find the superior trajectory as follow.

Define the residual vector of approximation of b_1 is $r_1(\theta) = b_1 - \phi(\theta)$. Then the desire tangent vector is given by:

$$d\theta = J(\theta)^{-1} r_1(\theta)(1, \sin(\theta), \cos(\theta))^T \tag{3.9}$$

The Jacobian matrix is:

$$J(\theta) = \begin{pmatrix} \langle \frac{1}{2\pi}, 1 \rangle & \langle \frac{1}{2\pi}, \sin(\theta) \rangle & \langle \frac{1}{2\pi}, \cos(\theta) \rangle \\ -\langle \sin(\theta), 1 \rangle & -\langle \sin(\theta), \sin(\theta) \rangle & -\langle \sin(\theta), \cos(\theta) \rangle \\ \langle \cos(\theta), 1 \rangle & \langle \cos(\theta), \sin(\theta) \rangle & \langle \cos(\theta), \cos(\theta) \rangle \end{pmatrix}$$

Then update the curve using $\theta = \theta + d\theta$, and iterate till the norm $|r_1(\theta)|$ converges to zero. Thus we have this projection: $\mathbb{P} : \mathbb{L}^2 \to \mathcal{C}$
Geodesic on preshape space We consider the "Most efficient deformation" as to construct the shortest path between the corresponding points in the preshape space with respect to the Riemannian metric given by the $\mathbb{L}^2$ inner product on the tangent space, i.e., a length-minimizing geodesic. A geodesic on a manifold embedded in a Euclidean space is defined to be a constant speed curve on the manifold, whose acceleration vector is always perpendicular to the manifold.

We now construct geodesic paths on the preshape space and the corresponding algorithm will be proposed later. Let $\theta \in \mathcal{C}$ and $f \in T_\theta(\mathcal{C})$, we want to generate a geodesic path or a parameter flow starting from θ and with tangent vector f at θ; denote this flow by $\Psi(\theta, t, f)$ where t is the time parameter. We will evaluate this flow for discrete times $t = \Delta, 2\Delta, ...$, for a small $\Delta > 0$. Setting $\Psi(\theta, 0, f) = \theta$, take the first increment to reach $\theta + \Delta f$ in $\mathbb{L}^2$ and apply the projection $\mathbb{P}$ to this point. Set $\Psi(\theta, \Delta, f) = \mathbb{P}(\theta + \Delta f)$

to get the next point along the geodesic. Iterating this process provides successive points along the geodesic Ψ in $\mathcal{C}$. To transport the tangent vector f to the tangent space at the next point, we normalize it to keep the "speed" of the geodesic constant.

Let θ_1 be the point along the geodesic path; we want to find f_1 that is tangent to $\mathcal{C}_1$ at θ_1 and is parallel transport of f. This can be accomplished using:

$$f_1 = \|f\| \frac{g}{\|g\|}, where\ g = f - \sum_{k=1}^{3} \langle f, h_k \rangle h_k, \tag{3.10}$$

Here $h_k s$ form an orthogonal basis of the normal space: $span\{1, \sin(\theta_1), \cos(\theta_1)\}$ An algorithm summarizing the steps for constructing a geodesic path on $\mathcal{C}$ is as follows:

Algorithm Start with a point $\theta \in \mathcal{C}$ and a direction $f \in T_\theta(\mathcal{C})$. Set $l = 0$ and $\Psi(\theta, l\Delta, f) = 0$, and choose a small $\Delta = 0$.

1. Add increment $\Psi(\theta, l\Delta, f) + \Delta f$ and set $\Psi(\theta, (l+1)\Delta, f) = \mathbb{P}(\Psi(\theta, l\Delta, f) + \Delta f)$
2. Transport f to the new point by using $\Psi(\theta, (l+1)\Delta, f)$ for(3.10)
3. Set $l = l + 1$, Go to Step 1 with $f = f_1$

It can be shown that $\Delta \to 0$, Ψ converges to a geodesic path on $\mathcal{C}$.

Geodesic on shape space To find a geodesic in $\mathcal{C}$ which is orthogonal to the $\mathbb{S}^1$-orbit, we simply restrict the allowable tangent directions to be orthogonal to the $\mathbb{S}^1$-orbit, i.e., use only those $f \in T_\theta(\mathcal{C})$ which are perpendicular to $T_\theta(\mathbb{S}^1)$. It can be shown that this one-dimension space is spanned by $1 - \theta'$ and hence, f should be orthogonal to $1 - \theta'$. The algorithm for constructing geodesic in S_1 is identical to the previous algorithm in this chapter except that, in(3.10), the vector g is now given by:

$$g = f - \sum_{k=1}^{4} \langle f, h_k \rangle h_k \tag{3.11}$$

where $h_k s$ form an orthogonal basis of the space span $\{1, \sin(\theta_1), \cos(\theta_1), \theta'_1\}$

3. Application in shape interpolation.

Given $\theta, \vartheta \in S_1$, let Ψ be the desired one-parameter flow from θ to ϑ. we want to find that appropriate direction $f \in T_\theta(S_1)$ such that a geodesic in that direction passed through the $\mathbb{S}^1$-orbit of ϑ. That is to solve for an $f \in T_\theta(\mathcal{C})$ such that $\Psi(\theta, 0, f) = \theta$ and $\Psi(\theta, 1, f) = s \cdot \vartheta$, for some $s \in \mathbb{S}^1$

We can treat this problem as an optimization problem over the $T_\theta(S_1)$. The cost function for minimizing is given by the functional:

$$H[f] = \inf_{s \in \mathbb{S}^1} \|\Psi(\theta, 1, f) - s \cdot \vartheta\| \tag{3.12}$$

and we look for that $f \in T_\theta(S_1)$ satisfies that $H[f]$ is zero and $\|f\|$ is minimum among all such tangents.

Since the $T_\theta(S_1)$ is infinite dimensional, we solve it by using a finite-dimension approximation of the elements of $T_\theta(S_1)$ by the Fourier decomposition (since $f \in \mathbb{L}^2$) to find the optimal direction f. Here we have some examples to express this method(See Fig4, Fig5). In Fig4, the deformation from one halobios θ_1 to another θ_2 is constructed via geodesics. In Fig5, The resulting path not only gives intermediate shapes along the geodesic but also is geodesically continued to naturally extend the sequence.

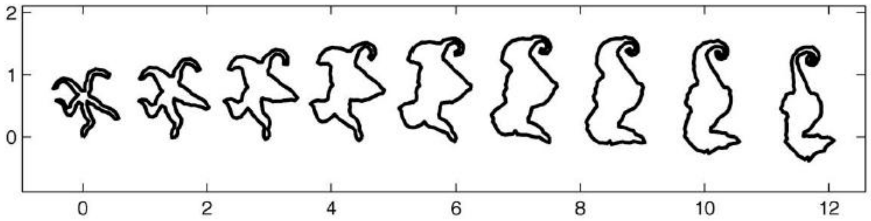


Fig4. Examples of evolving one shape into another via geodesics. Leftmost shape is θ_1, rightmost is θ_2, and intermediate shapes are equi-spaced points along the geodesic.

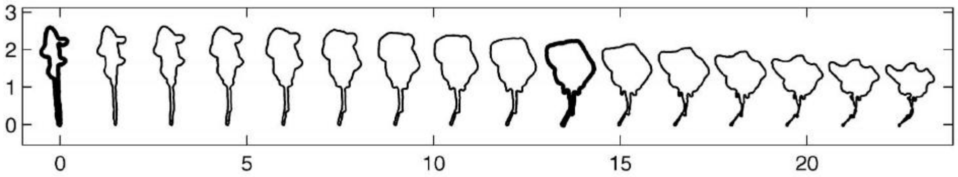


Fig5. Interpolation and extrapolation on the shape space given the two bold shapes.

3.2 Geometric modeling in shape space

In this section, we treat the 3-D shapes which are triangular meshes or more generally straight line graph in Euclidean space. We take the choice of proper metrics and computing geodesic on shape space under this Riemanian metric as our main work and contribution. This work of 3-D shape space approach is the extension of the work presented in the previous section, but this work give some new ideas and total new methods, algorithms.

Typically a shape is viewed as a set of points and represented according to the available data, and the intended application. Geometry always took these perspectives in the past: Projective geometry views hyperplanes as points in a dual space, line geometry interprets straight lines as points on a quadratic surface, and the various types of sphere geometries model spheres as points in higher dimensional space. Other examples concern kinematic spaces and Lie groups which are convenient for handling congruent shapes and motion design. We will show that many geometry processing tasks can be solved by endowing the set of closed orientable surfaces C called shapes henceforth C with a Riemannian structure.

1. Choose proper metrics and Construct shape space

By shape-triangular meshes in Euclidean 3-space, shape can be separated into its connectivity. Here we have **Some Assumptions**: we require all meshes to be 2-dimensional manifolds C boundaries are allowed; any two triangles share a common edge, or share a common vertex, or have no points in common; the star of any non-boundary vertex is a topological disk and a boundary vertex is a half disk. In what follows, we keep the connectivity fixed, and change only the vertex positions.

Given a fixed simplicial complex, we consider the space $\mathbb{S}$ of all immersions of this connectivity in Euclidean 3- space. Such an immersion is seen as a point in shape space, that is $p \in \mathbb{R}^{3m}$(we assume this shape M involve m points). For any vertex $p \in M$, there will be a vector $X_p \in \mathbb{R}^3$ and assigned a tangent vector $X \in T_M(\mathbb{S})$. A smooth deformation of M is a mapping $\phi : [0,1] \times M \to \mathbb{R}^3$ such that all the shapes consisting a function $p(t) := \phi(t,p)$ is smooth. Given a deformation of M, $X(t) := (\frac{d}{dt}p(t))_{p \in M}$ defines the deformation field at time t. In this point, The affine deformation $p(t) = A(t) \cdot p + a(t)$ is a smooth rigid body motion if $A(t)$ is a smoothly varying matrix and orthogonal at every instant of time and $a(t)$ is a translation vector. A mesh is deformed isometrically if distances measured on the mesh are preserved during deformation. So a deformation of a shape M is isometric if and only if $\|p(t) - q(t)\|^2$ the length of each edge in the triangulation is preserved in the deformation, that is $\langle X_p(t) - X_q(t), p(t) - q(t) \rangle = 0$ for each edge $(p(t), q(t))$ of the mesh $M(t)$, where $\langle , \rangle$ denotes the canonical inner product in $\mathbb{R}^3$. The vector field corresponding to isometric deformations of M named Killing vector fields is a linear subspace of $T_M(\mathbb{S})$.

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. In this section, we choose preserving distances measured on the mesh rather than pairwise Euclidean distances between vertices(See Fig6. for a comparison of metrics.). In a first step, we extract 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, we choose to deform a shape as isometrically as possible. To achieve this we take:

$$\langle\langle X, Y\rangle\rangle_M^I := \sum_{(p,q)\in M} \langle X_p - X_q, p-q\rangle\langle Y_p - Y_q, p-q\rangle \tag{3.13}$$

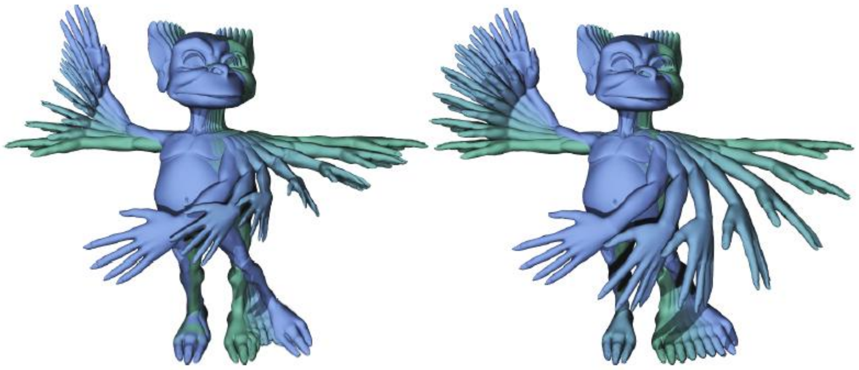


Fig6. Comparing the as-rigid-as-possible shape metric (left) with the as-isometric-as-possible shape metric (right).

as the inner product of two deformation fields. This expression is symmetric and bilinear and hence defines a semi-Riemannian metric. It is just a semi-Riemannian metric since there are non-vanishing deformation fields X with $\langle\langle X, X\rangle\rangle_M = 0$. Thus all the shapes which are congruent to a given shape M form points of a fiber in $\mathbb{S}$(like the reparametrization group $\mathcal{S}^1$ in the 2-D case). Any smooth curve in a fiber has length zero and corresponds to a smooth isometric motion of M. This observation also shows that shortest paths (geodesics) in the described metrics are only unique up to changes within the fibers.

To overcome this problem, we add a small regularization term to the length which is minimized by geodesics, rather than mod out this congruence relation. The obvious choice for this term is the $\mathbb{L}^2$ shape metric: Given vector fields X, Y on a shape M, let

$$\langle\langle X, Y\rangle\rangle_M^{\mathbb{L}^2} := \sum_{p\in M} \langle X_p, Y_p\rangle A_p, \tag{3.14}$$

where A_p is one-third of the area of all triangles adjacent to the vertex p. Blending the $\mathbb{L}^2$ inner product with the metrics (3.13) or (3.14) yields Riemannian metrics

$$\langle\langle X, Y\rangle\rangle_{M,\lambda} := \langle\langle X, Y\rangle\rangle_M + \lambda\langle\langle X, Y\rangle\rangle_M^{\mathbb{L}^2} \tag{3.15}$$

that have the same visual behavior as their semi-Riemannian counterparts if l is chosen appropriately small.

So till now we have constructed a Riemannian metric based shape space

$(\mathbb{S}, \langle\langle , \rangle\rangle)$. 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 (3.15) this means that the length of a deformation decreases as the deformation becomes more isometric. From the below figure, we know the isometric metric(as isometric as possible) is better than the rigid metric(as rigid as possible).

2. Computing geodesic on shape space

We now describe how to solve the following problem: *Interpolation problem*: Given two isomorphic shapes, how to compute a geodesic path joining them; We solve this 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.

Interpolation problem Assumptions: The input shapes are two compatible meshes M and N, i.e., the underlying simplicial complexes are isomorphic; 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. These assumptions mean that our input meshes 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. So we get the initial path $P^0 = (M^0, \overline{M^0N^0}, N^0)$.

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 $X_0, X_1, \ldots, X_n (X_i = P_iP_{i+1})$. Since the local minimizers of this energy are geodesics in a scaled arc length parametrization, we discretize the energy $\int \langle\langle X, X \rangle\rangle_{P(t)} dt$ of a curve $P(t)$ as

$$E(P) := \sum_{i=0}^{n} (\langle\langle X_i, X_i \rangle\rangle_{p_i} + \langle\langle X_i, X_i \rangle\rangle_{p_{i+1}}) \tag{3.16}$$

A quasi-Newton method is used to minimize (3.16). 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, we linearly blend neighboring meshes to refine the path. (b) Time Refinement: refine the path by inserting more vertices in the polyline. Firstly we simply increase the resolution of the progressive meshes to get refined boundary meshes from P_0^k and P_{n+1}^k to P_0^{k+1} and P_{n+1}^{k+1}. Then we transfer the detail of P_0^{k+1}, P_{n+1}^{k+1} to the intermediate meshes. For any new vertex, P_i^k as a example, we project P_i^k onto P_0^{k+1}, P_{n+1}^{k+1}. Denote f_1, f_2 are the index of the faces that carry the projection p', p'', and N_{f_1}, N_{f_2} are the normals of the faces f_1, f_2, we compute barycentric coordinates of the projected vertices with respect to the face f_1, f_2 and denoted by $P_{i,0}^{k+1} = \langle P_i^k - p', N_{f_1}\rangle$ and $P_{i,n+1}^{k+1} = \langle P_i^k - p'', N_{f_2}\rangle$. The refined mesh at position i of the path is given by:

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

. These steps are mutually independent, and can be applied in any order. After re- finement, we repeat the optimization on the new path.

3. Some other applications

Similarly, we can solve *Extrapolation 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 by using a multi-resolution framework. From the theoretical view, geodesic equation expresses vanishing geodesic curvature. So geodesics can be described by the Euler Lagrange equation of the energy. Alternatively computing a first-order ODE and an optimization process under a multi-resolution framework, we solve this problem. In this shape space framework many concepts from classical differential geometry can be applied to a wide variety of geometry processing tasks: parallel transport to *deformation transfer*, and the exponential map to *shape exploration* (See the pictures below).

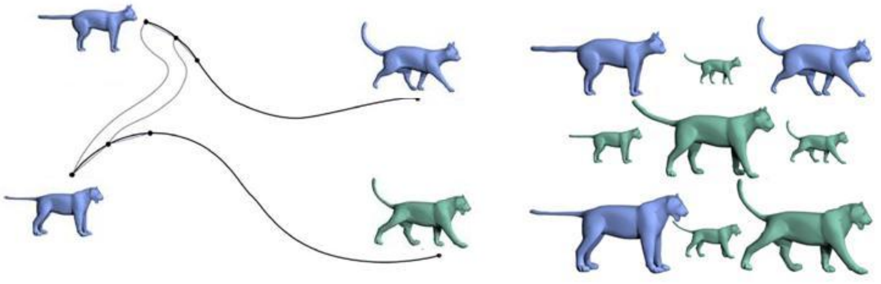


Fig7. Deformation transfer. The blue input shapes of the cat (top row) are joined by a geodesic to get a deformation. This deformation is transferred to the blue lion model (bottom row). The middle row (in right part of graph) shows hybrids generated during deformation transfer.

Chapter 4

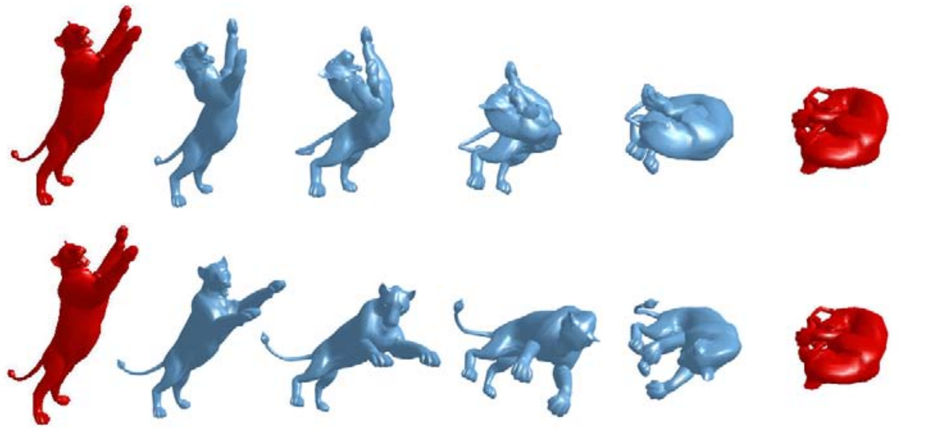
Comparisons

In this chapter, we demonstrate some 3-D experiment results to compare the methods Poisson shape interpolation and Shape space methods.

In all of the examples, the mesh reconstructions part use gradient-based method(presented in the paper 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.). Sometimes if we globally align poses of two models and then do shape interpolation, the result of the method Poisson shape interpolation can be improved(See Example 1). However, we present a example that it is useless to handle using a single global alignment (See Example 2).

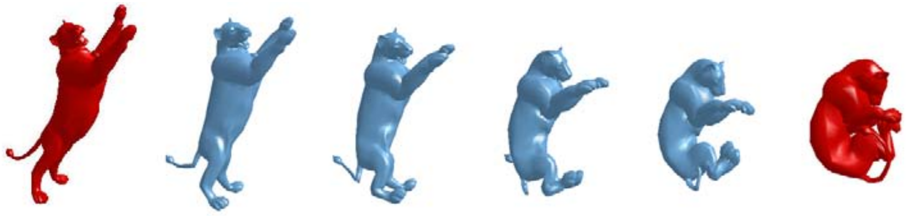
In all these examples, the source and target shapes are rendered in red color.

Example 1. A lion shape (9996 faces) from a stretched pose to a curled pose. Limbs and tail are bended naturally toward the target pose. Obviously, shape space method presents a smoother transformation than Poisson shape interpolation



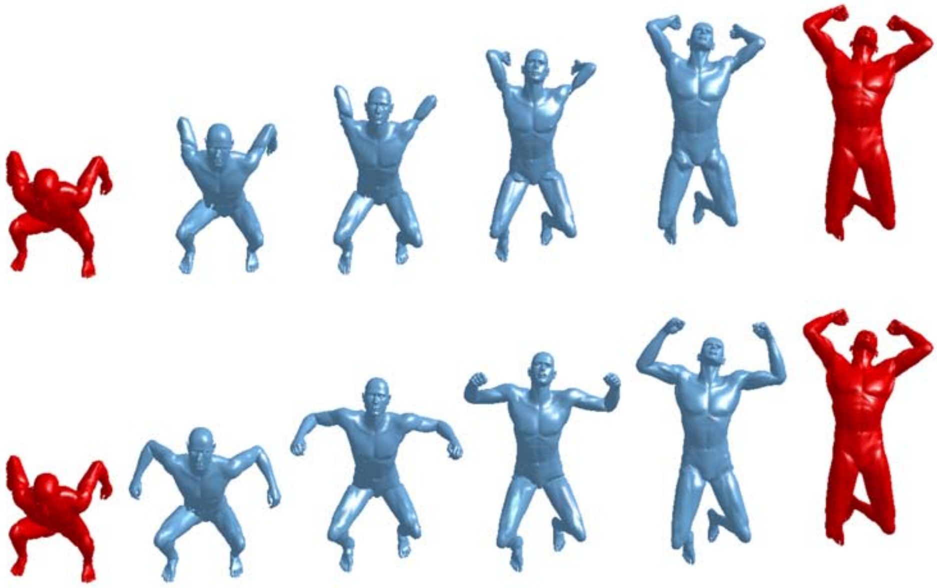
Lion shape transformation. *A lion shape transforms from a stretched* *pose to a curled pose by Poisson shape interpolation(above) and Shape space method(blow).*

Although Poisson shape interpolation can generate better result with rough global shape alignment, it still has some defects.

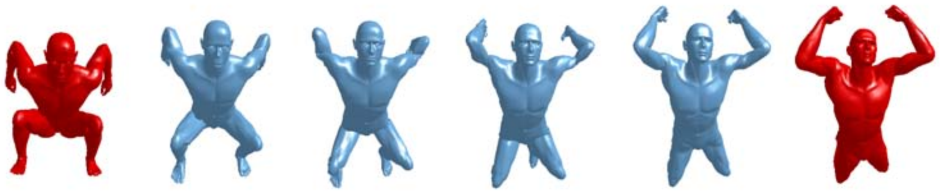


Lion shape transformation after global alignment. *The interpolation generated by Poisson shape interpolation still suffers from several fallacies: See the changes of tail in the transformation.*

Example 2.Shape interpolation of a male shape (34970 faces) from a crouched pose into a stretched pose. Notice the natural bending of the body, limbs and fingers, and the preservation of the local details (lines of the muscle) during the interpolation.



Male shape transformation. *A male shape transforms from a crouched pose to a stretched pose by Poisson shape interpolation(above) and Shape space method(blow). The latter shows superiority in the transformation*
Even with the aid of global shape alignment, the improvement to the result of the transformation based on Poisson shape interpolation is little.



Male shape transformation after global alignment *The global alignment can not avoid the critical shrinkage in the arms.*

The comparisons in the theoretical view

To the method Poisson shape interpolation, interpolating vector fields in gradient domain results in the morphing results depending on the quality of compatible meshes. Reconstruction of intermediate shape which gradually change both vertex coordinates and face normals by Poisson equation solver is the main contribution. In essence, Poisson shape interpolation is a as-rigid-as possible algorithm and it can not avoid shrinkage in variation of pose orientation. In this point, the shape space method give a significantly profound understanding of interpolation.

shape space promotion from planar shape analysis to 3-D geometric modeling

As-isometric-as possible shape interpolation is one of important contributions in the paper Geometric modeling in shape space. Except the choice of metrics, there are totally different solutions in dealing with reparametrization groups in 2-D or fiber consisting of shapes with vanishing distance in 3-D. The authors find the quotient space by moding that space in the 2-D case but the latter solve it dexterously by adding a small regularization term from the $\mathbb{L}^2$ shape metric to the length which is minimized by geodesic. Since the authors of these two papers focus different emphases on the concept shape space, the anterior presents more theoretical derivation and application in statistical analysis, while the latter propose a marvelous approach in geometric modeling

Acknowledgements

I thank for Dr Liu Ligang for his nice patience and the ease atmosphere at the CAGD Group.

References

[1] SEDERBERG. T. W., GAO, P., WANG, G., AND MU, H. 1993. 2-d shape blending: An intrinsic solution to the vertex path problem. In Computer Graphics (SIGGRAPH '93 Proceedings), vol. 27, 15-18.

[2] Liu, L.-G., AND WANG, G.-J. 1999. Three-dimensional shape blending: intrinsic solutions to spatial interpolation problems. Computers & Graphics 23, 4, 535-545.

[3] XU, D., ZHANG, H., WANG, Q., BAO, H. 2005. Poisson shape interpolation. In SPM '05: Proc. ACM Symp. On Solid and Physical Modeling, 267-274.

[4] KLASSEN, E., SRIVASTAVA, A., MIO, W., JOSHI, S. H. 2004. Analysis of planar shape using geodesic paths on shape spaces. IEEE transactions on Pattern analysis and machine intelligence. Volume 26, 3, 372-383.

[5] MARTIN, K., NILOY J. M., AND HELMUT P. 2007. Geometric Modeling in Shape Space. In Computer Graphics (SIGGRAPH '07 Proceedings), Volume 26, Issue 3