EUROGRAPHICS 2016 / J. Jorge and M. Lin
(Guest Editors)

Volume 35 (2016), Number 2

Geometric Flows of Curves in Shape Space for Processing Motion of Deformable Objects

Christopher Brandt[1] Christoph von Tycowicz[2] Klaus Hildebrandt[1]

[1]Delft University of Technology [2]Zuse Institute Berlin

Abstract

We introduce techniques for the processing of motion and animations of non-rigid shapes. The idea is to regard animations of deformable objects as curves in shape space. Then, we use the geometric structure on shape space to transfer concepts from curve processing in $\mathbb{R}^n$ to the processing of motion of non-rigid shapes. Following this principle, we introduce a discrete geometric flow for curves in shape space. The flow iteratively replaces every shape with a weighted average shape of a local neighborhood and thereby globally decreases an energy whose minimizers are discrete geodesics in shape space. Based on the flow, we devise a novel smoothing filter for motions and animations of deformable shapes. By shortening the length in shape space of an animation, it systematically regularizes the deformations between consecutive frames of the animation. The scheme can be used for smoothing and noise removal, e.g., for reducing jittering artifacts in motion capture data. We introduce a reduced-order method for the computation of the flow. In addition to being efficient for the smoothing of curves, it is a novel scheme for computing geodesics in shape space. We use the scheme to construct non-linear "Bézier curves" by executing de Casteljau's algorithm in shape space.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

1. Introduction

Many problems in geometric modeling and more generally in graphics are dealing with deformable, flexible or non-rigid shapes. The idea of *geometric modeling in shape space*, introduced by Kilian et al. [KMP07], is to equip the manifold of shapes relevant for a problem with a Riemannian metric and to use the resulting geometric structure on such a shape space for modeling tasks. A Riemannian metric on a shape space provides a quantitative measure for the deformation of shapes and concepts from Riemannian geometry, like the Riemannian exponential map and parallel transport, have been applied for designing powerful tools for modeling tasks such as shape deformation and interpolation, shape space exploration, deformation transfer, shape correspondences, and the design of measures of shape similarity. Due to the non-linear nature of shape spaces, geometric modeling in shape space leads to high-dimensional non-linear problems that have to be solved. For example, evaluating the distance between two shapes requires computing a geodesic in shape space. Therefore, efficient solvers for these optimization problems are of central importance.

The main contributions of this work are twofold. The first contribution is a novel approach for processing motion and animations of non-rigid shapes. We regard sequences of deformations of shapes as curves in shape space and use the geometric structure on shape spaces to transfer concepts from curve processing in $\mathbb{R}^n$ to the processing of motion of deformable shapes. Following this principle, we introduce a geometric flow for curves in a shape space of meshes. The flow smoothes a curve by decreasing its length in shape space. Our analysis of the flow shows that the limits are discrete geodesics in shape space (as defined in [HRWW12]). The definition of the flow involves elastic shape averaging. In every iteration, every shape of the curve is replaced by a weighted average shape of the shape itself and its predecessor and successor. Since the limits are geodesics, the flow establishes a connection between shape averaging (or interpolation) and geodesics in shape space.

Based on the flow, we devise a scheme for the fairing of curves in shape space. The fairing scheme shortens the (shape space) length of the curve and thereby decreases the energy stored in the deformations between consecutive shapes. This means that the scheme is using knowledge of how elastic objects deform to faithfully filter the motion. For example, artifacts, like shrinkage of parts of an object, are avoided because the formation of such artifacts would require additional deformation energy. As of yet, no other temporal filter for mesh sequences with such properties has been introduced. We apply the scheme for removing jittering artifacts in motion capture data and for smoothing non-differentiable transitions that occur when concatenating different motions of an object.

The second main contribution is a reduced-order method for the efficient computation the flow. After a preprocess, the scheme has a computational cost that is independent of the (spatial) resolution of the meshes to be processed. Since the processing of sequences of meshes results in high-dimensional optimization problems, this method is essential for an efficient processing of curves in shape space. Moreover, the scheme provides a novel algorithm for the computation of geodesics in shape space. Compared to timings reported in previous work, this algorithm significantly accelerates the computation. Additionally, it allows for computing geodesics with much higher temporal resolution than previous approaches, which is due to the fact that our scheme performs only local operations in the temporal domain and the dimensional reduction in the spatial domain. Our experiments indicate that the combination of spatial reduction and higher temporal resolution yields a better approximation of the continuous geodesics.

We use the fast computation of geodesics for constructing nonlinear "Bézier curves" in shape space that are controlled by sets of poses. The curves are generated by applying de Casteljau's algorithm in shape space. This example follows the construction of Bézier curves in shape spaces of images that was recently introduced by Effland et al. [ERS*15].

Using Riemannian geometry on shape spaces for geometric modeling tasks is a powerful concept. Crucial for these methods is the efficient computation of geodesics. We are convinced that the proposed reduced-order method (as it allows for faster computation and for higher spatial and temporal resolutions) is a step forward in this development.

2. Related Work

Riemannian metrics on shape spaces of curves proved to be effective for various problems in Computer Vision. We refer to the textbook of Younes [You10] for an in-depth discussion.

Riemannian metrics on the shape space of triangular surface meshes (with a fixed connectivity) have been introduced by Kilian et al. [KMP07]. These metrics measure the stretching of the edges of the triangles, hence, the metric distortion of the surface. Heeren et al. [HRWW12, HRS*14] propose a metric that in addition to the metric distortion measures the change of bending of the surface. Their framework includes a model of elastic materials, which leads to Riemannian metrics on spaces of elastic shells. A Riemannian metric on the space of elastic solids was introduced by Wirth et al. [WBRS11]. Kurtek et al. [KKG*12, KSKL13] introduce Riemannian metrics on spaces of surfaces parametrized over the unit sphere. Berkels et al. [BFH*13] introduce an approach for computing geodesic regression curves in shape spaces. Important for the application is the efficient computation of the geodesics between pairs of points in these shape spaces. This requires solving non-linear optimization problems, which are high-dimensional as the search space consists of curves in shape space. Specialized multi-grid Newton's solvers have been developed for this problem in [KMP07, HRWW12].

The problem of shape interpolation (or averaging, morphing, blending) has many applications in graphics. Early work concerned the morphing of planar shapes [SGWM93, ACOL00]. For recent work on the interpolation of planar shapes, we refer to [CWKBC13] and references therein. Approaches for the interpolation of surface meshes are either based on linearized deformation models, like Poisson reconstruction [SP04, XZWB05] and linear rotation invariant coordinates [LSLCO05], or on non-linear deformation models [SK04, WDAH10, FB11, LG15, vTSSH15]. Linearized deformation models are limited to small deformations, see [BS08] for a detailed discussion. Fast approximation algorithms for the shape interpolation problem have been proposed. Fröhlich and Botsch [FB11] use a combination of mesh coarsening and deformation-transfer to avoid solving the shape interpolation problem for the fully-resolved surfaces. A model reduction approach that yields real-time computation times for shape interpolation has recently been introduced by von Tycowicz et al. [vTSSH15]. Compared to the Riemannian structure on the shape space, shape interpolation is a simpler concept. For example, elastic shape averaging does not lead to a distance measure that satisfies the triangle inequality [RW11]. A comparison of results of shape interpolation techniques and geodesics can be found in Section 7. Since our discrete flow is based on shape interpolation and has geodesics in shape space as its limits, this paper establishes a connection between shape interpolation and geodesics, which were separate concepts before.

Smoothing filters for mesh sequences are typically applied directly to the trajectories of the individual vertices. Vlasic et al. [VBMP08] use bilateral filter in the temporal domain for each of the vertex trajectories. Li et al. [LLV*12] smooth the frames of an animation using a mix of constraints from points on the current, next and previous frames. These filters smooth the motions of the individual vertices, but neglect the shape formed by the set of vertices. Thus they are unable to prevent unnatural deformations of the

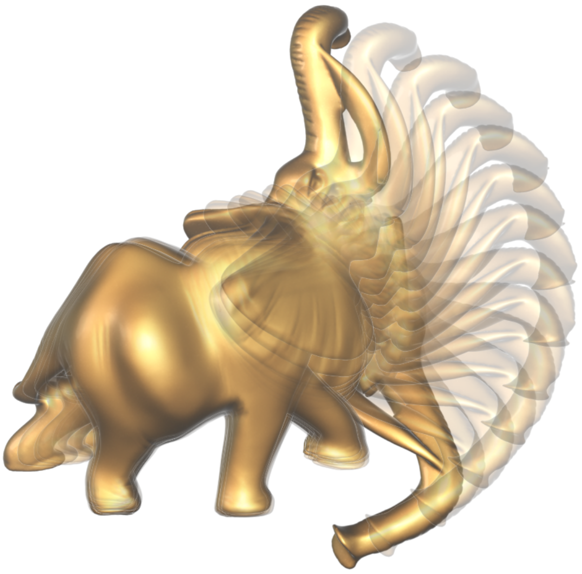


Figure 1: An illustration of a discrete geodesic in shape space consisting of 256 poses of a shape with 40k vertices is shown.

shape. An example is shown in the supplementary video as well as in Section 7.

Related to shape interpolation and geodesics in shape space is the problem of keyframe interpolation in computer animation. The spacetime constraints paradigm, introduced by Witkin and Kass [WK88], provides a variational framework for physically-based keyframe interpolation. The goal is to help animators in creating plausible motion by combining physical simulation with keyframe interpolation. Spacetime optimization of the motion of deformable objects has been consider in [BdSP09, HSvTP12]. Recently, schemes for interactive editing of simulations and animations [BSG12, LHdG*14] and for creating motion using sets of partial keyframes [SvTSH14] have been proposed.

3. Background: Deformation Energies and Shape Averaging

In this section, we briefly introduce deformation energies and the elastic shape averaging. Our presentation restricts to the discrete case. For an introduction to elasticity, we refer to [MH94] and for a background on elastic shape averaging to [RW09, vTSSH15].

Discrete deformation energies We consider a deformable object consisting of a hyperelastic material. A material is elastic if the (inner) forces acting on the object depend only on the current configuration and are independent of the deformation path and speed. This means that the forces can be described by a vector field on the space of configurations of the object. The material is hyperelastic if this field is conservative. Then, the function whose negative gradient equals the force field, is the deformation energy. This function is only determined up to a constant. The constant is chosen such that the neutral configuration stores no deformation energy.

In this paper, we restrict our attention to the discrete setting and consider triangle meshes for representing elastic shells and tetrahedral meshes for elastic solids. After the choice of materials and a discretization, the discrete deformation energy is a function

$$E : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}.$$

Here n is the number of degrees of freedom of the discrete object. In our setting, we keep the connectivity of the meshes fixed and n is three times the number of vertices. The first entry specifies the neutral configuration and the second the deformed configuration. The value $E(\bar{x},x)$ measures the energy stored in the deformable object when it is deformed from the neutral configuration $\bar{x}$ to the configuration x. In our experiments, we are using *Discrete Shells* [GHDS03]. However, other deformation energies like *PriMo* [BPGK06], *As-Rigid-As-Possible* [SA07, CPSS10], or finite elements discretizations of elastic solids or shells could be used as well.

Elastic shape averaging We consider a set of $\mu+1$ example configurations $\{y_0, y_1, ..., y_\mu\}$ and positive weights $\{\omega_0, \omega_1, ..., \omega_\mu\}$. Elastic shape averaging, introduced by Rumpf and Wirth [RW09], provides a way to compute weighted averages of the examples. The weighted average shape is defined as the configuration that minimizes the weighted sum of the energies $E(y_i,y)$ and $E(y,y_i)$

$$A(\omega_0, y_0, \ldots, \omega_\mu, y_\mu) = \underset{y\in\mathbb{R}^n}{\arg\min} \sum_{i=0}^{\mu} \omega_i \left(E(y_i,y) + E(y,y_i)\right). \quad (1)$$

This is a non-linear and elasticity-based approach for shape averaging that has a number of desirable properties. For example, the scheme can deal with larger deformations and the weighted average shape does not change if the example shapes are rigidly transformed.

4. Discrete Curve Flow in Shape Space

In this section, we introduce a curve smoothing flow in shape space and discuss its application to the fairing of curves in shape space.

Curve smoothing flow in shape space We consider a discrete curve in shape space given by a sequence of $m+1$ configurations $(x_0, x_1, ..., x_m)$. The curve can either be closed or have a boundary. In the latter case, we fix the first and last configurations. In the case of a closed curve, the indices are to be read modulo $m+1$.

The discrete curve smoothing flow is defined by the iterative procedure

$$\begin{aligned} x_i^0 &= x_i \\ x_i^{k+1} &= A(\frac{\tau}{2}, x_{i-1}^k, 1-\tau, x_i^k, \frac{\tau}{2}, x_{i+1}^k). \end{aligned} \quad (2)$$

The parameter $\tau \in (0,1]$ controls the size of the steps. It can be a fixed value or varied in every step. As we will discuss in the next section, controlling the stepwidth allows to guarantee that every step decreases an energy whose minimizers are discrete geodesics in shape space. In every iteration, the flow deforms every shape of the curve towards an average of the shape itself and its two neighbors and thereby smoothes the deformations between successive shapes of the curve.

Curve fairing in shape space The smoothing flow combines two smoothing effects. It decreases the (shape space) length of the curve (this is discussed in the following section) and it regularizes the parametrization by equalizing the lengths of the individual segments. Decreasing the shape space length smoothes the curve in a way that avoids the formation of artifacts (like linearization artifacts or shrinkage of parts of the shape) because this would require additional deformation energy and hence make the curve longer. For example, the limit of a closed curve is a static “mean” shape (Figure 4 illustrates this). The second effect means that the curves evolve towards a more uniform motion in which the deformations between successive shapes store the same amount of energy. If this effect is not desired, it can be reduced by altering the weights for the shapes x_{i-1}^k and x_{i+1}^k in (2). For example, one can use the weigths

$$\frac{\tau\, l_{i-1}}{l_{i-1}+l_i} \quad \text{and} \quad \frac{\tau\, l_i}{l_{i-1}+l_i}$$

where $l_i = \sqrt{E(x_i, x_{i+1})}$, in order to better preserve the original proportions of the energy stored in the deformations between successive shapes.

5. Analysis of the Flow and the Computation of Geodesics

In this section, we analyze the relation of the smoothing flow and geodesics in shape space. First, we show that the stationary points of the flow are discrete geodesics. Then, we prove that the flow decreases an energy whose minimizers are discrete geodesics (as defined in [HRWW12]). As a consequence, the discrete flow can be used for the computation of discrete geodesics.

Stationary points of the flow As a first step, we characterize the stationary points of the flow in the following lemma. By $\partial_1 E(x,y)$ and $\partial_2 E(x,y)$ we denote the derivatives of the energy E with respect to the first and the second argument.

Lemma 1 A stationary point $(x_0,x_1,\ldots,x_m)$ of the discrete flow (2) satisfies

$$\partial_1 E(x_i,x_{i-1}) + \partial_2 E(x_{i-1},x_i) + \partial_1 E(x_i,x_{i+1}) + \partial_2 E(x_{i+1},x_i) = 0 \tag{3}$$

for all interior shapes x_i.

Proof Assume $(x_0,x_1,\ldots,x_m)$ is a stationary curve. This means

$$x_i = A(\frac{\tau}{2},x_{i-1},1-\tau,x_i,\frac{\tau}{2},x_{i+1})$$

for all (interior) i. Using (1), we see that x_i has to satisfy

$$\frac{\tau}{2}\left(\partial_1 E(x_i,x_{i-1}) + \partial_2 E(x_{i-1},x_i)\right) + (1-\tau)\left(\partial_1 E(x_i,x_i) + \partial_2 E(x_i,x_i)\right)$$
$$+\frac{\tau}{2}\left(\partial_1 E(x_i,x_{i+1}) + \partial_2 E(x_{i+1},x_i)\right) = 0.$$

This implies (3) since $E(x,x) = 0$ for all configurations x. □

Limits are discrete geodesics in shape space Different Riemannian metrics on the spaces of shapes have been defined. We consider the physically-based metric introduced by Heeren et al. [HRWW12]. It uses viscous dissipation required to deform physical objects for measuring the distance of shapes. After a spatial discretization (which is the setting considered here) the discrete geodesics $(x_0,x_1,\ldots,x_m)$ are defined as the minimizers of the functional

$$\sum_{i=1}^{m} \left(E(x_{i-1},x_i) + E(x_i,x_{i-1})\right) \tag{4}$$

for fixed configurations x_0 and x_m and with respect to variations of the other configurations x_i. The relation of the energy (4) to the Riemannian distance on the (continuous) shape space is that $E(x_i,x_{i+1})$ is a second-order approximation of the squared Riemannian distance between x_i and x_{i+1}, see [HRWW12]. The Euler-Lagrange equation satisfied by the minimizers of (4) is exactly equation (3), which is satisfied by the stationary points of the discrete flow.

This shows that discrete geodesics are stationary points of the flow. However, this does (in general) not guarantee that curves evolve towards geodesics. The following lemma shows that for every configuration of the curve, a small enough step of the flow de-

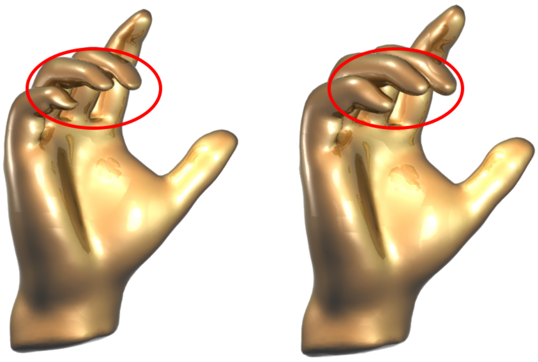


Figure 2: Applying our flow to animations exhibiting artifacts can remove them: The hand animation from the supplementary video contains frames with strong artifacts (left, visible on the fingers), after smoothing the animation, the shape looks artifact-free again, while the motion is kept intact due to the restoring force.

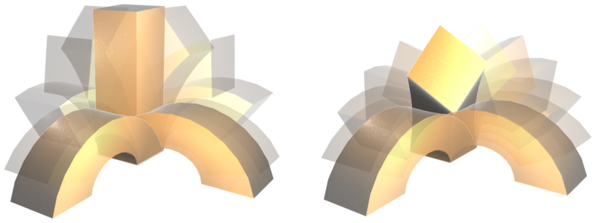


Figure 3: The bending block sequence (exhibiting a C^1-discontinuity) before smoothing (left) and after 50 smoothing iterations (right).

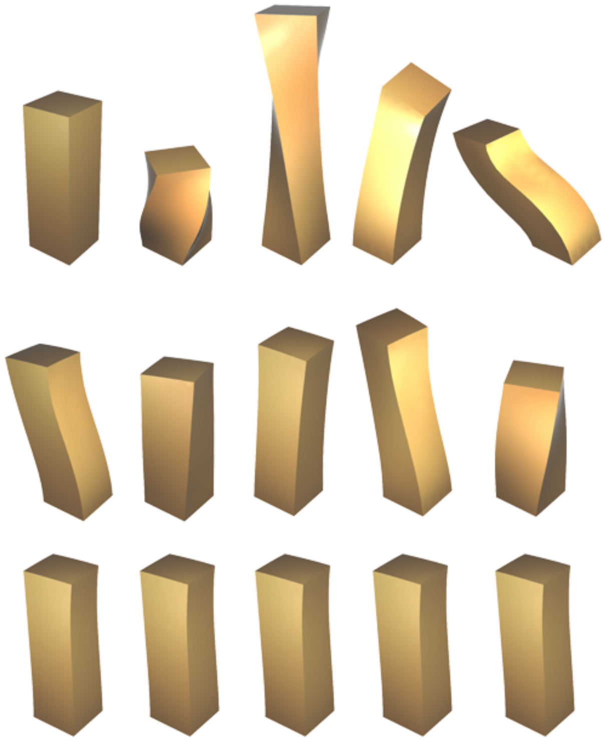


Figure 4: A periodic curve evolves to a constant curve. Top: Original sequence, middle: after 1 smoothing step, bottom: after 10 smoothing steps.

creases the energy (4). As a consequence, if we control the stepwidth, the limits of the flow are discrete geodesics in shape space.

Lemma 2 For any non-stationary curve $(x_0, x_1, \ldots, x_m)$ and small enough $\tau > 0$, an iteration of the flow (2) decreases the energy (4).

Proof Let us consider τ as a variable and denote the next iterate by $(x_0^+(\tau), x_1^+(\tau), \ldots, x_m^+(\tau))$. To prove the claim, we show that the derivative

$$\frac{\partial}{\partial \tau}(x_0^+(\tau), x_1^+(\tau), \ldots, x_m^+(\tau)) \tag{5}$$

at $\tau = 0$ points into a descent direction of the energy (4). From the definition of the flow, (2), it follows that $(x_0^+(\tau), x_1^+(\tau), \ldots, x_m^+(\tau))$ satisfies

$$\frac{\tau}{2}(\partial_1 E(x_i^+(\tau), x_{i-1}) + \partial_2 E(x_{i-1}, x_i^+(\tau))) + (1-\tau)(\partial_1 E(x_i^+(\tau), x_i) \tag{6}$$

$$+\partial_2 E(x_i, x_i^+(\tau))) + \frac{\tau}{2}(\partial_1 E(x_i^+(\tau), x_{i+1}) + \partial_2 E(x_{i+1}, x_i^+(\tau))) = 0$$

We use these equations as an implicit function that determines $x_i^+(\tau)$. To compute (5), we need the derivatives of the left-hand side of (6) at $\tau = 0$ with respect to τ and x_i^+. The former is

$$\frac{1}{2}(\partial_1 E(x_i^+(0), x_{i-1}) + \partial_2 E(x_{i-1}, x_i^+(0)) \tag{7}$$
$$+\partial_1 E(x_i^+(0), x_{i+1}) + \partial_2 E(x_{i+1}, x_i^+(0)))$$

and the latter is

$$\partial_1\partial_1 E(x_i^+(0), x_i) + \partial_2\partial_2 E(x_i, x_i^+(0)). \tag{8}$$

The configuration $x_i^+(0)$ equals x_i. Plugging this into (7) and (8), we see that (7) is the gradient direction of (4) at x_i. Furthermore, the matrices $\partial_1\partial_1 E(x_i, x_i)$ and $\partial_2\partial_2 E(x_i, x_i)$ are positive definite (modulo rigid transformation of the shape) because $E(x_i, x_i)$ is a minimum of E for variations of the first and of the second argument. This implies that (8) is positive definite. The implicit function theorem implies that the derivatives in (5) satisfy

$$(\partial_1\partial_1 E(x_i, x_i) + \partial_2\partial_2 E(x_i, x_i))\,\frac{\partial}{\partial \tau} x_i^+(0) \tag{9}$$
$$= \frac{1}{2}\left(\partial_1 E(x_i, x_{i-1}) + \partial_2 E(x_{i-1}, x_i) + \partial_1 E(x_i, x_{i+1}) + \partial_2 E(x_{i+1}, x_i)\right).$$

Since the right-hand side is the gradient direction of (4) and the matrix $\partial_1\partial_1 E(x_i, x_i) + \partial_2\partial_2 E(x_i, x_i)$ on the left-hand side is positive definite, $\frac{\partial}{\partial \tau} x_i^+(0)$ points into a descent direction of (4). This implies that for small enough τ an iteration of the flow will decrease the energy (4). □

6. Efficient Computation of the Flow

Integrating the discrete flow requires solving a number of shape averaging problems. We use a reduced-order technique for this, which combines dimensional reduction in the spatial domain and a scheme for the efficient evaluation of the reduced deformation energy and its gradient. Before we introduce the reduction strategy, we first discuss an asymmetry in the elastic potential and its effect on the definitions of elastic shape averaging and geodesics in shape space.

Remark on shape averaging and geodesics The elastic potentials are not symmetric in their two arguments, *i.e.*, in general we have $E(x, y) \neq E(y, x)$. This means the energy stored in an object with rest shape y that is deformed into configuration x is not the same as the energy stored in an object with rest shape x that is deformed into configuration y. Because of this asymmetry, we have defined the elastic shape averaging using both terms $E(y, y_i)$ and $E(y_i, y)$ in (1). As an alternative, one can use only one of the terms to define the averaging. For example, in [RW09, vTSSH15] only the terms $E(y_i, y)$ are used. Then the averaging is

$$A(\omega_0, y_0, \ldots, \omega_\mu, y_\mu) = \underset{y \in \mathbb{R}^n}{\arg\min} \sum_{i=0}^{\mu} \omega_i E(y_i, y). \tag{10}$$

In the same spirit, we define the geodesics in shape space using $E(x_{i-1}, x_i)$ and $E(x_i, x_{i-1})$ in (4). In [HRWW12], only the terms $E(x_{i-1}, x_i)$ were used to define geodesics. This introduces a slight asymmetry in the definition. The discrete geodesic from shape x to y is not exactly the same as that from shape y to x. However, the difference is small and reduces with temporal refinement of the geodesic.

We have used the symmetric definitions involving both of the energy terms for shape averaging and geodesics in shape space in Sections 4 and 5 because in this case the connection between averaging and geodesics in shape space established by the proposed discrete smoothing flow is exact. The geodesics are exactly the limits of the flow. This makes the presentation simpler and easier accessible.

In our experiments, we have not noticed significant differences between the results obtained with the different definitions, which matches the observations reported in [RW09]. Experiments con-

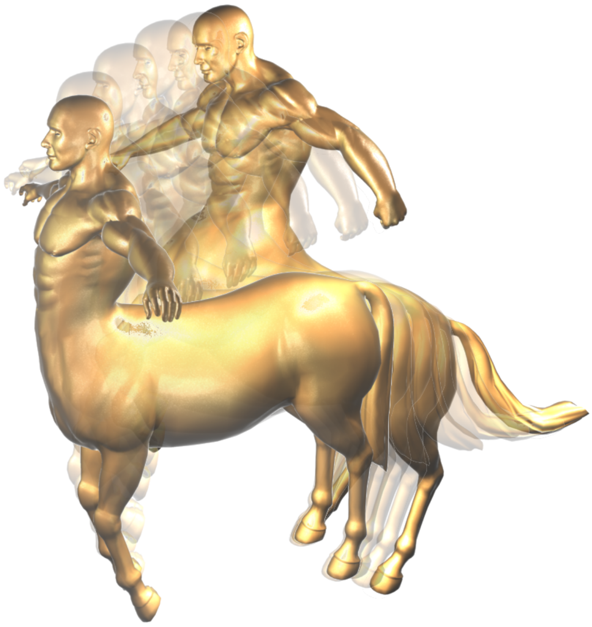


Figure 5: Applying our flow to concatenated animations (like the centaur animation shown in the supplementary video) results in a visually smooth animation without visible “corners” in the motion.

cerning the issue can be found in Table 2 in Section 7. The computation, of course, is faster if only one of the energy terms is used. Therefore, we used (10) for shape averaging in most of our experiments.

Dimensional reduction For the dimensional reduction, we are restricting the variations of every shape to a low-dimensional affine subspace of $\mathbb{R}^n$. We have used two subspace constructions for our experiments. If the input is a curve in shape space, a good candidate is the affine span of all shapes of the curve. This space can be represented as a linear space attached to one of the shapes. To further reduce the dimension, we select one shape x_σ and compute a principle component analysis (PCA) of the displacement vectors to all other shapes. The new affine space is the d-dimensional linear subspace spanned by the left singular vectors (of the matrix formed by the displacement vectors) with the highest singular value attached to x_σ. This affine space does not contain all shapes anymore. Therefore, we use for every shape x_i the affine span of this space and the vector representing x_i itself. To represent these spaces, we only need to store one subspace basis, which is augmented with the missing vector (the difference of the shape x_i and x_σ) for each shape at runtime. The additional vectors can be orthonormalized against the basis in the preprocess.

If the curve to be processed is very coarse, *e.g.*, less than 20 frames, using on the affine span of the shapes would provide enough flexibility. In such a case, we use the flow tangent directions, $\frac{\partial}{\partial\tau}x_i^+(0)$ in equation (9) for every shape as an additional input for constructing the space. In the case, that only two shapes x and y are given and we want to compute a geodesic joining them, we follow the subspace construction in [vTSSH15]. The starting point is the affine space spanned by the two shapes. This space is enriched with additional vectors. First, two vectors, v_1 and v_2, obtained from linearizing the deformation from x to y and from y to x are computed. Then, further vectors are generated using a Krylov sequences that involves the Hessians of the elastic potentials, the mass matrices and v_1 and v_2. For details, we refer to the original work.

Energy and force approximation In addition to dimensional reduction, we are using a scheme for the efficient approximation of the reduced energy and force. Since we are working in a reduced space, the deformations of the individual vertices are correlated. The approximation schemes aim at exploiting this structure. Different schemes have been proposed in the literature.

We adapt the mesh coarsening technique introduced in [HSvTP11] to our setting. The idea is to create a coarse mesh and a subspace $\bar{V}$ for the coarse mesh that is isomorphic to the subspace V for the fine mesh. To approximate the energy at a point in V, the energy of the coarse mesh at the corresponding point in $\bar{V}$ is evaluated. To evaluate the gradient of the energy, the gradient of the coarse mesh is projected onto $\bar{V}$. The corresponding vector in V is the vector that has the same reduced coordinates (however this vector is not explicitly computed, as we only work with the reduced coordinates). Explicitly, we use an edge-collapse scheme to generate a coarse version $\bar{x_\sigma}$ of the selected shape x_σ. Edge-collapse schemes implicitly generate a map from the vertices of the fine to the vertices of the coarse mesh, see [HSvTP11] for details. We use this map to get subspace basis vectors for the coarse mesh.

An alternative approach for reduced force approximation is the optimized cubature [AKJ08, vTSSH13]. A second alternative is polynomial restriction [BJ05], which allows for exact evaluation of a finite elements discretization of St. Venant–Kirchhoff materials for elastic solids at costs depending only on the subspace dimension. By combining the dimensional reduction and the reduced energy and force approximation, we obtain a computational cost for the integration of the flow that is independent of the resolution of the meshes.

Solving the reduced problem To solve the reduced problems, we use the BFGS scheme (see [NW06]). This is a quasi-Newton scheme that maintains an approximation of the inverse Hessian of the objective functional. Approximating the inverse Hessian avoids costly solving of a linear system to get the Newton direction. It is efficient to initialize the BFGS scheme with an inverse Hessian approximation to get a warm start. In the preprocess, we once compute the inverse Hessian (of the energy of the ghost mesh) of the mean shape of the predecessor and successor (on the initial curve) for every shape.

For the computation of the geodesic between two shapes, we use a coarse-to-fine strategy in the temporal domain. Starting with the two boundary shapes, we perform a two-step procedure. First the temporal domain is refined by inserting a fixed number (two or three in our experiments) of new shapes between every pair of successive shapes x_i and x_{i+1}. These shapes are initialized as interpolating shapes between x_i and x_{i+1}. Secondly, the geometric flow (2) is iterated until the squared norm of the reduced gradient of (4) is below 1^{-8} times the degrees of freedom.

7. Applications and Experiments

First, we want to point to the supplementary video that shows curves in shape space computed with our method. Details of the experiments and computation times are shown in Table 1. The implementation was done in Java and the experiments were performed on a custom laptop (Intel Core i7-4600U, 2.1GHz). The precomputation times shown in the table (T_p) include the construction of the subspace, the initialization of the deformation energies and the initialization of the minimizer (computing initial Hessian approximations as a warm start for the BFGS minimizer). The precomputation time is significantly lower when the subspace can be constructed from a PCA on the shapes of the input curve, and not via the more involved subspace construction from [vTSSH15]. For our experiments, we used the Discrete Shells energy [GHDS03] as the elastic energy E, where we set the parameters to $k_B = 1$ and $k_L = k_A = 1/2$ (following the notation from that paper).

Basic examples The twisting block sequence (Figure 4) is meant as a simple demonstration of how a periodic sequence converges to a single point (or constant curve) when we perform several smoothing steps, akin to the contraction of closed curves to single points under the curve shortening flow. Since the original sequence consists of few shapes, the limit is reached after a few iterations (explicitly after 10 iterations with $\tau = 0.6$).

Animation	#verts.	#verts. ghost	#shapes	$\|V\|$	#steps	T_s	T_p
Twisting Block	450	(not used)	6	24	10	0.26	3.44
Bending Block	450	(not used)	155	20	10/50/200	1.02	0.16
Centaur	15768	1252	138	15	50	2.13	12.08
Finger Geodesic (short)	2046	(not used)	5	14	214	0.03	3.51
Finger Geodesic (long)	2046	1252	81	7	115	0.17	3.48
Elephant Geodesic	39969	1246	256	14	270	1.09	82.18
Hand Linear Artifacts	6094	1252	49	14	100	0.92/0.94*	0.56
Hand Temporal Noise	6094	1252	49	14	100	0.91/0.94*	0.54
Motion Capture	2502	1252	91	20	50	1.81	0.34

Table 1: Data for the experiments. $|V|$ = size of subspace, T_s = seconds per step on average, T_p = seconds for precomputation, *: flow with restoring force.

We demonstrate the ability of our smoothing technique to get rid of sudden jumps in the object's velocity (*i.e.* C^1-discontinuities in the temporal domain): the bending block sequence shows a block starting from a bent-over position, getting into an upright position and back into a bent-over position, this time bending to a different direction (cf. Figure 3). The animation has a visible jump in the velocity around the frame where the block stands upright, since there is a sudden change in the direction of the motion. Applying smoothing steps to this animation leads to a smoother motion: the block does not get into a fully upright position anymore, and the direction of the motion evenly changes across the full animation.

The usefulness to this type of animation smoothing becomes clear when used on a more complex animation: the centaur animation (cf. Figure 5) was created by concatenating interpolation curves to 6 successive poses of the centaur shape. This results in an animation with visible jumps in the moving direction at the input poses. After some steps of the discrete flow, to the animation is visibly smoother and without discontinuities. This can be made precise in the following sense: we can look at the part of the gradient of the energy functional (4) of the animation (as a curve in shape space) that corresponds to a certain frame of the animation, in particular at the norm of the l.h.s. of (3). In case of the centaur sequence, we can make the observation that the norms of these individual parts of the gradient are much larger at the discontinuities than at the other shapes. This leads to large spikes in the norm of the gradient parts corresponding to these frames. In Figure 7, we plot the norm of these individual parts of the gradients after various iterations of smoothing. One can observe how the spikes become less sharp while smoothing the animation.

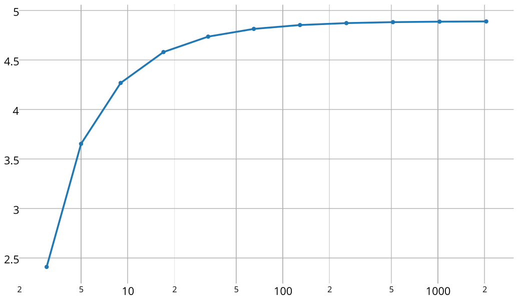


Figure 6: Plot of the length of the bending block geodesic (y-axis) with a varying number of intermediate shapes (x-axis).

Computing geodesics As explained in Section 5, we can use our technique to compute geodesics. The advantage of only having to solve optimization problems involving one unknown shape at a time and the application of the model reduction techniques lead to a significant speed-up when comparing to the times stated in [HRWW12]: when computing the discrete geodesic of the finger mesh on 8 shapes, Heeren et al. report a computation time of 628s for a multilevel optimization. For the same setting, our technique needed 6,42s, including the precomputation time (subspace construction and computation of initial Hessians). To justify our reduced-order modeling, we computed the same geodesic using the full-order model in the space $\mathbb{R}^n$ and compared the lengths of both curves, which differed by less than 0.1% of the length of the full geodesic, as well as the L_2-distances of all shapes, which differed by less than 1^{-3} % of the summed lengths of the shape vectors.

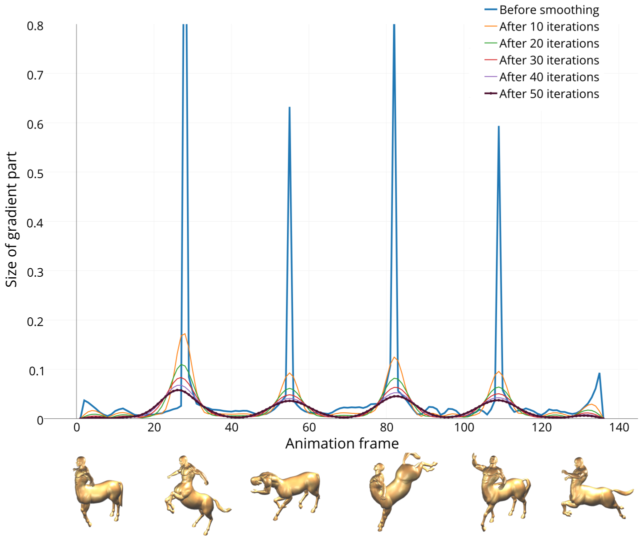


Figure 7: The norm of the energy gradients of the individual shapes (left-hand side of (3)) for the centaur sequence after various smoothing iterations.

The dimensional reduction of the spatial domain and the fast computation times allow for computing discrete geodesics with finer time discretizations: the finger geodesic on 64 shapes leads to a computation time of 19.55s, the elephant geodesic (cf. Figure 1) on 256 shapes (40k vertices per shape) leads to a computation time of 376s.

To demonstrate the convergence of our flow to a geodesic, we plot the squared length of the curve and the length of the gradient of functional (4) in Figure 9 for the computation of the elephant geodesic on 100 shapes, where we initialize the geodesic with 100 interpolation shapes right away (instead of using the adaptive scheme). The plot shows that the length of the curve decreases monotonically.

While the loss of precision due to model reduction is very low, we actually gain accuracy by being able to compute discrete geodesics with more shapes. In Figure 6, we plot the length of the geodesic between the bent over and upright block on a varying number of shapes. The plot demonstrates that the length differs significantly when comparing a geodesic with 10 to a geodesic with 100 shapes. This shows the benefit of computing geodesics on many shapes.

De Casteljau algorithm in shape space The fast computation of geodesics makes it possible to evaluate geometric constructions that require repeated computation of geodesics. As one example, we construct nonlinear “Bézier curves” in shape space by executing the de Casteljau algorithm with respect to the shape space metric. This means the straight lines used in Euclidean space are replaced by geodesics in shape space. In [ERS*15], Effland et al. applied the de Casteljau algorithm in a shape space of images to obtain smooth curves from a few input images, we refer to this paper for further details on the algorithm.

To compute a point of the “Bézier curves” controlled by four shapes, we need to compute the three geodesics between each successive pair of boundary shapes once and three additional geodesics per shape of the final curve. The supplementary video and Figure 10 show an example of such a “Bézier curves” with four control poses of a block shape. The computed curve consists of 32 shapes and each auxiliary geodesic was also computed on 32 shapes. The computation time for the whole procedure (which requires the computation of almost 100 geodesics) was 241s. This could be drastically reduced by using coarser time discretizations for the auxiliary geodesics.

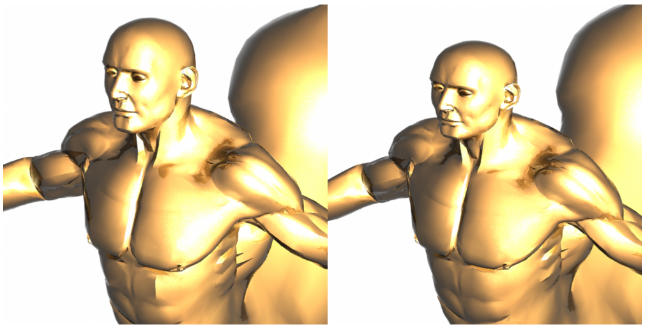


Figure 8: A frame of the smoothed centaur sequence. Left: Our technique. Right: bilateral filtering of the vertex trajectories [VBMP08], where the head appears to be shrunken.

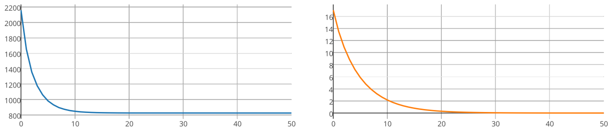


Figure 9: Plot of the squared curve length (blue) and size of the gradient of functional (4) (orange) while smoothing a sequence of 12 elephant shapes (initialized as the interpolation curve). The x-axis denotes the number of smoothing iterations in both plots.

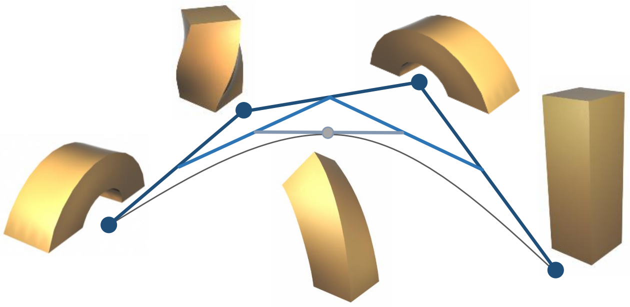


Figure 10: Visualization of the de Casteljau algorithm in shape space: the straight lines are geodesics between shapes.

Denoising and animation repair Our smoothing technique is also able to enhance and repair noisy data: an animation of a hand bending its fingers with temporal noise becomes completely denoised after 100 smoothing steps. The shape of the fingers remains plausible even after applying a large amount of smoothing. However, the motion itself has also changed: before smoothing, the index finger and thumb touched, whereas after the smoothing, the fingers stay far away from each other throughout the whole animation.

If the flow is used for noise removal, curve shortening can lead to over-smoothing. To reduce this effect, we use a restoring force that pushes the curve towards its initial state. To achieve this, we modify the flow equation (2):

$$y_i^0 = x_i$$
$$y_i^{k+1} = A\left((1-\rho)\frac{\tau}{2}, y_{i-1}^k, (1-\rho)(1-\tau), y_i^k, (1-\rho)\frac{\tau}{2}, y_{i+1}^k, \rho, x_i\right),$$

that is, in addition to the current shape and its neighbors, we take the original shape into the local averaging processes. The parameter $\rho \in [0,1]$ controls the strength of the restoring force. The restoring force can be used for all shapes of the curve or just for some selected shapes.

With this force, the animation is denoised, while articulation of the motion remains intact. We perform a similar experiment with an animation obtained by linear blending of a coarse set of keyframes (cf. Figure 2). By adding a restoring force to the keyframes, we

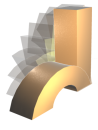
	Bending Block	Centaur	Elephant
			
Length of (10)-geodesic	0.3680	142.1023	368.5427
Length of (1)-geodesic	0.3648	141.7774	348.3585
Length of interpolation curve	37.4360	155.9907	613.1286
Rel. L_2-error of geodesics	5.2375^{-5}	1.4173^{-5}	4.2512^{-5}

Table 2: Geodesics computed using the symmetric (1) and the asymmetric averaging operator (10).

are able to remove the linear-blending artifacts, while keeping the overall motion intact.

In addition, we tested our method on motion capturing data from Gall et al. [GSA*09], which exhibits temporal and spatial noise. Again, using our smoothing technique with restoring force, we are able to acquire a smooth animation, which at the same time keeps its main characteristics. An advantage of using our smoothing flow to denoise motion capturing data is that each smoothing step regularizes the motion of the whole mesh instead of individual vertex positions. This implies that unwanted deformations and artifacts due to strong noise can be filtered without smoothing the mesh itself.

Averaging operators As discussed in Section 6, we used the formulation (10) of the shape averaging in most of our experiments to enhance computation speed. We observe that the choice of the operator used for shape averaging, (1) or (10), does not lead to significantly different geodesics. Table 2 shows the results of an experiment performed in this regard: We computed three different geodesics using both formulations (1) and (10). We state the energy (4) of these geodesics (with the energy of the interpolation curve as a comparison) as well as their relative L_2-error. This error was computed by first registering each pair of corresponding shapes via a best rigid fit, and then taking the ratio of the norm of the difference of both geodesics (interpreted as vectors in $\mathbb{R}^{n \cdot m}$) and the norm of the (1)-geodesic.

Comparisons

For comparisons to the timings and accuracy of computing geodesics using our technique as opposed to [HRWW12] see the paragraph above.

To prevent confusion, we first want to stress that our flow formulation is based on shape averaging, which is closely related to shape interpolation. This being the case, every interpolation technique can be used to define a smoothing flow using our formulation. However, not every interpolation scheme exhibits the properties of elastic shape averaging required to prove existence of a metric and convergence of our flow to geodesics in shape space.

Nonlinear shape interpolation While our model reduction and flow formulation allow for a fast computation of geodesics, interpolation curves can be produced even faster. The reason is that each interpolating shapes can be computed directly (without the need to compute all interpolating shapes). This cannot be done for geodesics, we always have to compute the whole curve. Hence, the optimization problem for geodesics is more involved. On the other hand, the more complex problem couples the shapes of a geodesic to each other, which is not the case for shape interpolation. For example, if the computations of interpolating shapes end in different local minima, we can get a stuttering interpolation curve. Due to the coupling of the shapes such effects are smoothed out during the computation of geodesics. We demonstrate this in the supplementary video. To produce the shown examples, we used the same elastic deformation energies, reduction techniques and minimization schemes, for both the geodesic and the interpolation curve.

Poisson-type shape interpolation We also compare to a Poisson interpolation curve (see supplementary video, cf. [XZWB05]). We show that our technique does not suffer from problems that arise in Poisson interpolation when elements are rotated by more than 180 degrees. Also, as can be seen in our video, while offering a smooth deformation over time, linear interpolation of the vertex positions leads to strong artifacts, even for very small deformations.

Comparison to other smoothing and denoising techniques Temporal filtering of the vertex positions, as proposed in [VBMP08], also leads to a smoother motion, but since a lot of filtering is required, the shape undergoes unnatural deformations, similar to linearization artifacts, as can be seen in Figure 8. Additionally, in the supplementary video, we compare our results for denoising the motion capture data to applying a temporal filter to the vertex trajectories. As can be seen, with the latter type of smoothing, the noise is converted to a wiggling surface, since it does not include a physical model of deformations of objects.

8. Conclusion

In this paper, we are proposing techniques for the processing of curves in shape space. In particular, we introduce a discrete geometric flow for curves in shape space. The flow iteratively computes local weighted average shapes and thereby decrease the magnitude of the deformation between consecutive shapes of the curve. Based on the flow, we design a novel type of smoothing filter for motions and animations of shapes. In contrast to previous work, the filter only smoothes the deformations between the shapes and thereby minimizes the distortion of the shapes themselves. One application of this filter is the smoothing of motions and animations of objects. We use the filter for reducing jittering artifacts in motion captured data and for smoothing transitions that appear when different motions are concatenated.

Our analysis shows that the flow converges to geodesics in shape space. To compute the flow, we propose a reduced-order scheme that combines a dimensional reduction with a scheme for reduced energy and force approximation. The approach significantly accelerates the computation of geodesics in shape space. In addition, it allows for finer temporal discretizations, which improves the approximation quality. We think that these two benefits are important for an effective processing of curves in shape space. We demonstrate results obtained with our scheme for the computation of geodesics that blend between two shapes as well as the computation of nonlinear "Bézier curves" in shape space that are controlled by a coarse polygon in shape space.

Future work We introduce techniques for the processing of curves in shape space. We think that this is a promising direction for processing motion and animation and we expect to see more algorithms that transfer techniques from the processing of curves in $\mathbb{R}^n$ to curves in shape space. For example, analogous to the example of De Casteljau's algorithm, curve subdivision schemes like corner cutting could be transferred to shape space. Another example is the fairing of curves in shape space. The proposed smoothing filter is the first of this kind and we expect that more filtering techniques for curves in Euclidean space will be transferred to filters for motion and animation of deformable shapes.

Furthermore, we think that reduced-order modeling has a great potential for geometry processing in shape space and other applications using Riemannian metrics on shape spaces. Fast approximation algorithms for shape space computations can be designed and larger data sets can be processed.

With a growing market for devices which are able to directly capture deforming geometry, processing of motion and animation becomes more and more important. The concept of curves in shape space provides powerful and theoretical sound tools for processing this data (in particular for template-based approaches).

Acknowledgements

We would like to thank the anonymous reviewers for their comments and suggestions. This work was partially supported by the DFG Project *KneeLaxity: Dynamic multi-modal knee joint registration* (EH 422/1-1).

References

[ACOL00] Alexa M., Cohen-Or D., Levin D.: As-rigid-as-possible shape interpolation. In *ACM SIGGRAPH* (2000), pp. 157–164. 2

[AKJ08] An S. S., Kim T., James D. L.: Optimizing cubature for efficient integration of subspace deformations. *ACM Transactions on Graphics 27*, 5 (2008), 1–10. 6

[BdSP09] Barbič J., da Silva M., Popović J.: Deformable object animation using reduced optimal control. *ACM Transactions on Graphics 28*, 3 (2009), 53:1–53:9. 3

[BFH*13] Berkels B., Fletcher P. T., Heeren B., Rumpf M., Wirth B.: Discrete geodesic regression in shape space. In *Energy Minimization Methods in Computer Vision and Pattern Recognition* (2013), Springer, pp. 108–122. 2

[BJ05] Barbič J., James D. L.: Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics 24*, 3 (2005), 982–990. 6

[BPGK06] Botsch M., Pauly M., Gross M., Kobbelt L.: PriMo: Coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing* (2006), pp. 11–20. 3

[BS08] Botsch M., Sorkine O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 213–230. 2

[BSG12] Barbič J., Sin F., Grinspun E.: Interactive editing of deformable simulations. *ACM Transactions on Graphics 31*, 4 (2012). 3

[CPSS10] Chao I., Pinkall U., Sanan P., Schröder P.: A simple geometric model for elastic deformations. *ACM Transactions on Graphics 29* (2010), 38:1–38:6. 3

[CWKBC13] Chen R., Weber O., Keren D., Ben-Chen M.: Planar shape interpolation with bounded distortion. *ACM Transactions on Graphics 32*, 4 (2013), 108:1–108:12. 2

[ERS*15] Effland A., Rumpf M., Simon S., Stahn K., Wirth B.: Bézier curves in the space of images. In *Scale Space and Variational Methods in Computer Vision*. Springer, 2015, pp. 372–384. 2, 8

[FB11] Fröhlich S., Botsch M.: Example-driven deformations based on discrete shells. *Computer Graphics Forum 30*, 8 (2011), 2246–2257. 2

[GHDS03] Grinspun E., Hirani A. N., Desbrun M., Schröder P.: Discrete Shells. *Symposium on Computer Animation* (2003), 62–67. 3, 6

[GSA*09] Gall J., Stoll C., Aguiar E. D., Theobalt C., Rosenhahn B., peter Seidel H.: Motion capture using joint skeleton tracking and surface estimation. In *IEEE CVPR* (2009). 9

[HRS*14] Heeren B., Rumpf M., Schröder P., Wardetzky M., Wirth B.: Exploring the geometry of the space of shells. *Computer Graphics Forum 33*, 5 (2014), 247–256. 2

[HRWW12] Heeren B., Rumpf M., Wardetzky M., Wirth B.: Time-discrete geodesics in the space of shells. *Computer Graphics Forum 31*, 5 (2012), 1755–1764. 1, 2, 4, 5, 7, 9

[HSvTP11] Hildebrandt K., Schulz C., von Tycowicz C., Polthier K.: Interactive surface modeling using modal analysis. *ACM Transactions on Graphics 30*, 5 (2011), 119:1–119:11. 6

[HSvTP12] Hildebrandt K., Schulz C., von Tycowicz C., Polthier K.: Interactive spacetime control of deformable objects. *ACM Transactions on Graphics 31*, 4 (2012), 71:1–71:8. 3

[KKG*12] Kurtek S., Klassen E., Gore J., Ding Z., Srivastava A.: Elastic geodesic paths in shape space of parameterized surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 9 (2012), 1717–1730. 2

[KMP07] Kilian M., Mitra N. J., Pottmann H.: Geometric modeling in shape space. *ACM Transactions on Graphics 26*, 3 (2007), 64:1–64:8. 1, 2

[KSKL13] Kurtek S., Srivastava A., Klassen E., Laga H.: Landmark-guided elastic shape analysis of spherically-parameterized surfaces. *Computer Graphics Forum 32*, 2 (2013), 429–438. 2

[LG15] Levi Z., Gotsman C.: Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE Transactions on Visualization and Computer Graphics 21*, 2 (2015), 264–277. 2

[LHdG*14] Li S., Huang J., de Goes F., Jin X., Bao H., Desbrun M.: Space-time editing of elastic motion through material optimization and reduction. *ACM Transactions on Graphics 33*, 4 (2014), 108:1–108:10. 3

[LLV*12] Li H., Luo L., Vlasic D., Peers P., Popović J., Pauly M., Rusinkiewicz S.: Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics 31*, 1 (2012), 2. 2

[LSLCO05] Lipman Y., Sorkine O., Levin D., Cohen-Or D.: Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics 24*, 3 (2005), 479–487. 2

[MH94] Marsden J. E., Hughes T. J. R.: *Mathematical Foundations of Elasticity*. Dover Publications, 1994. 3

[NW06] Nocedal J., Wright S. J.: *Numerical Optimization (2nd edition)*. Springer, 2006. 6

[RW09] Rumpf M., Wirth B.: A nonlinear elastic shape averaging approach. *SIAM Journal on Imaging Science 2*, 3 (2009), 800–833. 3, 5

[RW11] Rumpf M., Wirth B.: Variational methods in shape analysis. In *Handbook of Mathematical Methods in Imaging*. Springer, 2011, pp. 1363–1401. 2

[SA07] Sorkine O., Alexa M.: As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing* (2007), pp. 109–116. 3

[SGWM93] SEDERBERG T. W., GAO P., WANG G., MU H.: 2–d shape blending: An intrinsic solution to the vertex path problem. In *ACM SIGGRAPH* (1993), pp. 15–18. 2

[SK04] SHEFFER A., KRAEVOY V.: Pyramid coordinates for morphing and deformation. In *Proceedings of Symposium on 3D Data Processing, Visualization, and Transmission* (2004), pp. 68–75. 2

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on Graphics 23*, 3 (2004), 399–405. 2

[SvTSH14] SCHULZ C., VON TYCOWICZ C., SEIDEL H.-P., HILDEBRANDT K.: Animating deformable objects using sparse spacetime constraints. *ACM Transactions on Graphics 33*, 4 (2014), 109:1–109:10. 3

[VBMP08] VLASIC D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics* (2008), vol. 27, ACM, p. 97. 2, 8, 9

[vTSSH13] VON TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: An efficient construction of reduced deformable objects. *ACM Transactions on Graphics 32*, 6 (2013), 213:1–10. 6

[vTSSH15] VON TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: Real-time nonlinear shape interpolation. *ACM Transactions on Graphics 34*, 3 (2015), 34:1–34:10. 2, 3, 5, 6

[WBRS11] WIRTH B., BAR L., RUMPF M., SAPIRO G.: A continuum mechanical approach to geodesics in shape space. *Int. J. Comput. Vision 93*, 3 (2011), 293–318. 2

[WDAH10] WINKLER T., DRIESEBERG J., ALEXA M., HORMANN K.: Multi-scale geometry interpolation. *Computer Graphics Forum 29*, 2 (2010), 309–318. 2

[WK88] WITKIN A., KASS M.: Spacetime constraints. *ACM SIGGRAPH 22* (1988), 159–168. 3

[XZWB05] XU D., ZHANG H., WANG Q., BAO H.: Poisson shape interpolation. In *Symp. on Solid and Phys. Mod.* (2005), pp. 267–274. 2, 9

[You10] YOUNES L.: *Shapes and Diffeomorphisms*. Springer, 2010. 2