

Nonlinear Deformation Synthesis via Sparse Principal Geodesic Analysis


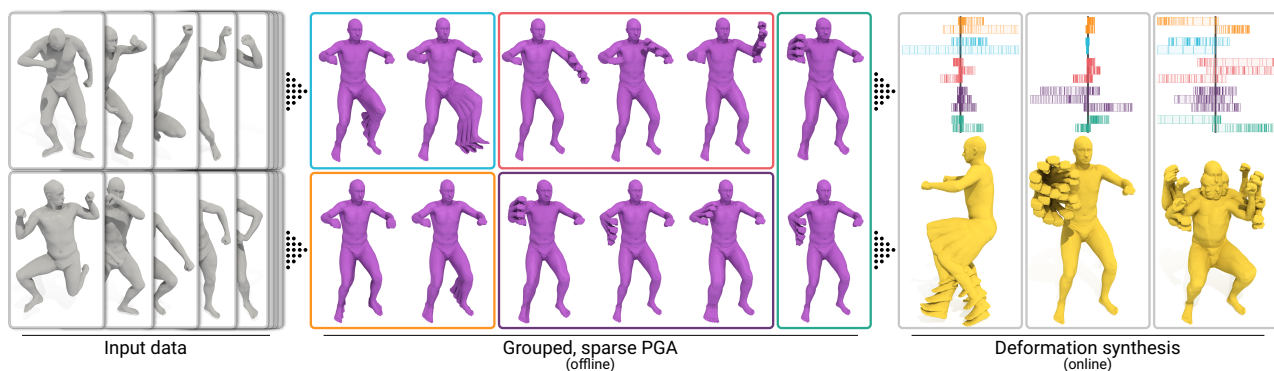
Josua Sassen¹ Klaus Hildebrandt²Martin Rumpf¹¹Institute for Numerical Simulation, University of Bonn, Germany²EEMCS – Dept. Intelligent Systems, Delft University of Technology, The Netherlands

Figure 1: Sparse principal geodesic analysis for data-driven nonlinear deformation synthesis. In an offline phase, sparse nonlinear deformation modes (middle) are extracted from a data set of poses of a non-rigid shape (left) using the shape space of discrete shells and Nonlinear Rotation-Invariant Coordinates. Then, the modes are grouped by overlap of the regions that are distorted (colored frames) and samples of nonlinear deformations for each group are computed and stored in lookup tables. In the online phase, nonlinear deformations are synthesized (right) at near-realtime rates.

Abstract

This paper introduces the construction of a low-dimensional nonlinear space capturing the variability of a non-rigid shape from a data set of example poses. The core of the approach is a Sparse Principal Geodesic Analysis (SPGA) on the Riemannian manifold of discrete shells, in which a pose of a non-rigid shape is a point. The SPGA is invariant to rigid body motions of the poses and supports large deformation. Since the Riemannian metric measures the membrane and bending distortions of the shells, the sparsity term forces the modes to describe largely decoupled and localized deformations. This property facilitates the analysis of articulated shapes. The modes often represent characteristic articulations of the shape and usually come with a decomposing of the spanned subspace into low-dimensional widely decoupled subspaces. For example, for human models, one expects distinct, localized modes for the bending of elbow or knee whereas some more modes are required to represent shoulder articulation. The decoupling property can be used to construct useful starting points for the computation of the nonlinear deformations via a superposition of shape submanifolds resulting from the decoupling. In a preprocessing stage, samples of the individual subspaces are computed, and, in an online phase, these are interpolated multilinearly. This accelerates the construction of nonlinear deformations and makes the method applicable for interactive applications. The method is compared to alternative approaches and the benefits are demonstrated on different kinds of input data.

CCS Concepts

• **Computing methodologies** → **Shape modeling**;

1. Introduction

This paper is concerned with the construction of low-dimensional models describing the variability of a non-rigid shape from a set

of example poses. Such models are used for templates-based surface reconstruction from incomplete data, data-driven shape editing, character animation, deformation transfer, and shape reconstruction from 2-dimensional data.

© 2020 The Author(s).

Computer Graphics Forum published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

One approach for constructing such a model is to apply Principal Component Analysis (PCA) to the vertex positions of the example poses. The resulting PCA modes span a linear subspace of deformations that approximate the examples. Still, this is not an adequate shape model since the linear structure of the PCA cannot accurately represent shape deformations, which are usually highly nonlinear in the vertex positions. Furthermore, shapes do not deform under rigid transformations of \mathbb{R}^3 , which can also not be captured by PCA. We consider shape manifolds that provide a shape representation invariant to rigid transformations. To this end, a point in a shape manifold corresponds to an equivalence class of poses, which differ only in rigid deformations. A Principal Geodesic Analysis (PGA) on such shape manifolds [HZRS18] can be used to obtain nonlinear models of shape variability that are invariant to rigid transformations of the example poses. A property of PCA and PGA is that they result in modes that involve all variables. In particular, they are not expected to be localized or decoupled. An alternative is the Sparse Localized Deformation Components (SPLOCS), introduced in [NVW*13]. The approach modifies the optimization problem underlying PCA by adding a sparsity inducing term to the objective. The resulting modes encode localized deformations that are intuitively meaningful.

In this paper, we introduce a Sparse Principal Geodesic Analysis (SPGA), which aims at combining the advantages of PGA with those of the sparse modes (cf. Figure 1). We consider the Riemannian manifold of discrete shells, which is a shape manifold of meshes equipped with a physically-based metric. The meshes are treated as the midsurfaces of thin shells and the metric measures the membrane and bending distortions of the shells. Our approach constructs a submanifold of the shell manifold and is invariant to rigid transformations of the example poses. Analogous to SPLOCS, we use a discrete L^1 term to enforce sparsity of the modes. However, in the nonlinear setting, the sparsity term has a different effect than in the linear setting. Since the metric measures membrane and bending distortions of the shells, the sparsity term forces the modes to describe localized deformations. For example, sparse modes can describe the movement of a human shoulder. Although the mode moves the whole arm, the distortions are active around the shoulder, hence, the distortion is localized (cf. Figure 2). In contrast, SPLOCS would not consider such a mode as sparse, because the sparsity refers to the local displacement instead of distortions. This property makes SPGA applicable to articulated shapes, which is a limitation of SPLOCS stated in [NVW*13]. Our approach builds on Nonlinear Rotation-Invariant Coordinates (NRIC), which represent a mesh by the vector that stacks the edge lengths and dihedral angles of all edges. The manifold of discrete shells corresponds to the set of such vectors, from which a mesh can be reconstructed. The SPGA modes are tangent vectors to the shell manifold at the mean shape. By applying the Riemannian exponential map to the space spanned by these modes, one obtains the submanifold of the shell manifold which approximates the input data.

The SPGA modes represent infinitesimal characteristic articulations of the mean and usually can be decomposed into low-dimensional almost completely decoupled subspaces, which is not possible for PGA modes. For example, for human models, one expects distinct, localized modes for the bending of elbow or knee whereas some more modes are required to represent shoulder ar-

ticulation. By grouping the modes via their coupling and a simple summation of nonlinear coordinates corresponding to these groups, a practical approximation of the submanifold can be defined. Then, in an offline phase, one precomputes the exponential map on lattices in the low-dimensional decoupled subspaces. For given weights on the principal modes, they are then used in an online phase together with a multilinear interpolation and a very small number of Gauß–Newton iterations for the back projection onto the manifold. This yields a highly efficient discrete shell reconstruction and enables near real-time shape animation.

Contribution. We introduce an SPGA in the Riemannian manifold of discrete shells that combines the benefits of the PGA [HZRS18] and SPLOCS [NVW*13] in one model. Our main contributions are:

- We use NRIC for PGA in the shell manifold resulting in a compact model for shape variability that is invariant to rigid body motions of the example poses and can handle nonlinear shape deformation.
- By using a sparsity inducing term which reflects the shell metric, modes that describe localized deformations are obtained. This makes the analysis applicable to articulated shapes.
- To solve the SPGA problem, we use a scheme that iteratively solves quadratic problems. In our experiments, this scheme outperforms the ADMM schemes used in prior work.
- Based on the decoupling property of the sparse modes, we introduce a fast approximation scheme for the construction of a nonlinear deformation via superposition of deformations resulting from the decoupled groups of modes. This makes it possible to use our approach for interactive applications, which is not feasible for PGA.

The SPGA is evaluated on different data sets and compared to alternative approaches.

Organization. In Section 2, we discuss related work. The background on Nonlinear Rotation-Invariant Coordinates, geodesic calculus, and its variational discretization is revisited in Section 3. Based on this, in Section 4 the discrete PGA is introduced. Then, Section 5 presents the SPGA and Section 6 its computation via an efficient quadratic matrix programming approach. The superposition of large deformation submanifolds generated via grouping the SPGA modes is discussed in Section 7, whereas the lookup table based fast deformation synthesis is presented in Section 8. Experimental results are given in Section 9 and conclusions are finally drawn in Section 10.

2. Related work

Rotation invariant coordinates. For various tasks in geometry processing, it is beneficial to switch from the usual mesh representation in terms of vertex positions to an alternative representation that is adapted to the structure of the problem considered. Different representations have been developed, such as gradient-based representations [SP04; XZWB05] and Laplace coordinates [LSC*04; SCL*04]. In this paper, we are interested in rotation-invariant mesh representations. Linear rotation-invariant coordinates were introduced in [LSLC05]. Each vertex is associated with a coordinate

frame and differences between adjacent frames are used to represent the mesh. This representation was further developed in [KG08; BVGP09; HSS*09; GLL*16].

An alternative to working with frame differences is to use the lengths of all edges and the dihedral angles between adjacent triangles [WDAH10]. In analogy to the classical differential geometry in which a surface is described by the first fundamental form, representing the metric of the surface, and the second fundamental form, characterizing curvature; edge lengths describe the metric of the mesh and the angles describe its curvature. Moreover, two meshes have identical edge lengths and angles if and only if they differ by a rigid transformation. Therefore, we call this representation the Nonlinear Rotation-Invariant Coordinates (NRIC). Integrability conditions, which guarantee that there are vertex positions realizing a given list of edge lengths and angles have been established in [WLT12]. The set of integrable NRICs forms a manifold and the tangential spaces at points in the NRIC manifold can be computed from the derivatives of the integrability conditions [SHHR20].

Since vertex positions do not exist for all combinations of edge lengths and angles, the question of how to project given lengths and angles to integrable NRIC coordinates arises. In [FB11] discrete shells, a model of elastic thin shells [GHDS03], is used for this purpose. Considering the mesh as the midsurface of a thin shell, membrane and bending distortions can be defined based on the edge lengths and angles of the mesh. To determine a mesh for a given list of edge lengths and angles, these are used to specify a rest configuration of the shell. The mesh is determined as a minimizer of the resulting discrete shell energy in the absence of constraints and external loads.

Shape spaces. Riemannian shape spaces are shape manifolds equipped with a Riemannian metric. In such spaces, concepts from Riemannian geometry can be used to derive tools for shape analysis and processing. This idea has been applied in computer vision, computational anatomy, and medical imaging and geometry processing. For a general overview of the topic, we refer to the textbook of Younes [You10]. A Riemannian metric on the space of triangle meshes and algorithms for computing the shortest geodesics, the Riemannian exponential, and parallel transport in this space were introduced and used for deformation transfer, shape interpolation and extrapolation in [KMP07]. A physically-based Riemannian metric was defined in [HRWW12; HRS*14]. The meshes are regarded to be discrete shells and the metric measures the viscous dissipation required to deform the shells.

Statistics in shape space. Principal Geodesic Analysis (PGA) [FLPJ04] is a generalization of PCA to data given in a Riemannian manifold. First, the Riemannian center of mass [Kar77] of the data points is computed and the data points are mapped to the tangent space at the center using the logarithmic map. Then a PCA in the tangent space is computed. Finally, the exponential map is used to map superpositions of the PCA modes back to the manifold. To perform nonlinear statistics for the analysis of shape variability, a PGA on the Riemannian manifold of discrete shells was introduced in [HZRS18]. In this paper, we extend this work by formulating the Shell PGA in the NRIC coordinates. These coordinates are well-

sued for this purpose because the invariance under rigid transformations matches the structure of the Shell PGA.

Sparse PCA. PCA allows for analyzing and compressing sets of meshes representing animation sequences [AM00]. Varimax rotations can be used to alter the basis of the space spanned by the lowest principal components. For animation sequences, Varimax rotations were applied to obtain sparse components that are meaningful and could be used to support animators [MA07]. In [NVW*13], a sparsity inducing term is included in the computation of the principal components resulting in Sparse Localized Deformation Components (SPLOCS). Since the SPLOCS span linear spaces, larger deformations are not supported. To counteract linearization artifacts, a warping strategy, which represents the components using the deformation gradient and then treats rotational components nonlinearly, was introduced in [HYZ*14]. SPLOCS can also be formulated in the NRIC representation of meshes [WLZH17; LLW*19], which results in an improved analysis for articulated shapes. In contrast to our approach, the NRIC representation is treated as a linear space. We demonstrate that the PGA, which uses the NRIC manifold, its tangent spaces, the physically-based metric, and the Riemannian exponential and logarithmic maps, leads to improved results.

Data-driven shape editing. Statistics of shape variation are useful for a variety of applications in geometry processing and related fields. In this paper, we focus on data-driven shape editing.

Elasticity-based energies, such as As-Rigid-As-Possible [SA07] or discrete shells [GHDS03], are commonly used for mesh editing [SA07; HsvTP11; JBK*12]. These measure the energy stored in a deformation of a rest configuration. The choice of a rest configuration has a major influence on the deformation behavior, still, this choice is ambiguous. Data-driven shape editing uses a variable rest configuration, which is built from example poses, instead of a fixed one. In [FB11] the NRIC coordinates of example poses are linearly interpolated to obtain a model of a variable rest configuration for the discrete shells energy. While this works for a low number of example poses, statistical shape models, such as a PCA in appropriate coordinates [GLL*16; WLZH17; LLW*19], PGA [HZRS18], or SPLOCS [GLY*20], are needed when dealing with larger numbers of example poses. Our experiments demonstrate the advantages of the proposed model for data-driven shape editing.

3. Background

We start with the necessary preliminaries to introduce our SPGA. First, we review NRIC as a useful representation for the geometry of a triangular surface to define the shape manifold of interest. Afterwards, we discuss the Riemannian structure and geodesic calculus on this manifold.

Nonlinear Rotation-Invariant Coordinates. In this paragraph, we briefly review the work by [WLT12] who introduced a discrete version of the fundamental theorem of surfaces. We consider a simply connected, triangular surface with a set of vertices \mathcal{V} , edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, and faces $\mathcal{F} \subset \mathcal{V} \times \mathcal{V} \times \mathcal{V}$. For nodal positions $X \in \mathbb{R}^{3|\mathcal{V}|}$, we denote by $l(X) = (l_e(X))_{e \in \mathcal{E}}$ the vector of edge lengths and by $\theta(X) = (\theta_e(X))_{e \in \mathcal{E}}$ the vector of dihedral angles.

To ensure that $z = (l, \theta) \in \mathbb{R}^{2|\mathcal{E}|}$ corresponds to a two-dimensional triangular surface immersed in \mathbb{R}^3 , it has to fulfill two admissibility conditions. The first condition is the triangle inequality, i.e. $\mathcal{T}_f(l) > 0$ for all $f \in \mathcal{F}$ where $\mathcal{T}_f(l) = (l_i + l_j - l_k \quad l_i - l_j + l_k \quad -l_i + l_j + l_k)$ for a face $f \in \mathcal{F}$ with edge lengths l_i, l_j, l_k and the above inequality is to be understood componentwise. Furthermore, traversing from face to face along the fan of triangles at each vertex v in the set of interior vertices \mathcal{V}_0 , one can reconstruct from l and θ the geometry of the fan up to rigid body motions. From this, one obtains a closing condition for the fan, which can be expressed as the admissibility condition $\mathcal{Q}_v(l, \theta) = 0$, where \mathcal{Q}_v can be robustly and efficiently computed using quaternions. For details on this, we refer to [SHHR20].

Then, the manifold of all $z \in \mathbb{R}^{2|\mathcal{E}|}$ which correspond to immersed triangular surfaces is given by

$$\mathcal{M} = \{z \in \mathbb{R}^{2|\mathcal{E}|} \mid \mathcal{T}(z) > 0, \mathcal{Q}(z) = 0\}, \quad (1)$$

where we collect all constraints in vector-valued functionals $\mathcal{T} = (\mathcal{T}_f)_{f \in \mathcal{F}}$ and $\mathcal{Q} = (\mathcal{Q}_v)_{v \in \mathcal{V}_0}$. Following [SHHR20], we call \mathcal{M} the NRIC manifold (Nonlinear Rotation-Invariant Coordinates). For $z \in \mathcal{M}$, its tangent space is given by

$$T_z \mathcal{M} = \ker D\mathcal{Q}(z) := \{w \in \mathbb{R}^{2|\mathcal{E}|} \mid D\mathcal{Q}(z)w = 0\}, \quad (2)$$

where $D\mathcal{Q}(z) \in \mathbb{R}^{3|\mathcal{V}_0| \times 2|\mathcal{E}|}$ is the Jacobian of \mathcal{Q} . The conditions can be extended to higher-genus surfaces by including integrability conditions along non-contractible paths that generate the fundamental group, see also [WLT12].

The NRIC z enable a local description of shell deformations based on the variation of edge lengths encoding membrane distortions and the variation of dihedral angles encoding bending distortions. Frequently, in applications, these distortions and thus also the associated modulation of z components are localized. For example, on human models, in the vicinity of joints and active muscles which is in striking difference to vertex-based mesh representations (cf. Figure 2).

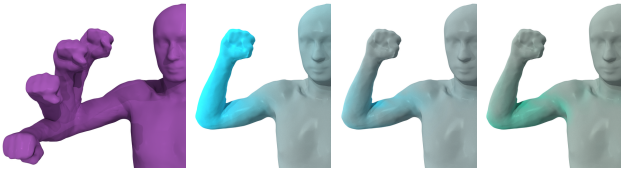


Figure 2: Support of deformation (left) in nodal positions (middle left), edge lengths (middle right) and dihedral angles (right). Change of nodal positions and absolute length change are shown on the same scale as color map 0 \rightarrow ≥ 0.05 , while change of dihedral angle is shown as color map 0 \rightarrow 1.3.

Geodesic calculus. As in [HRWW12] we equip the manifold \mathcal{M} with a Riemannian metric g reflecting the physical dissipation caused by the infinitesimal variation of a triangular surface interpreted as discrete shell. Following Rayleigh's paradigm, the metric acting on strain rates is related to the Hessian of the elastic energy, i.e. $g_z: \mathbb{R}^{2|\mathcal{E}|} \times \mathbb{R}^{2|\mathcal{E}|} \rightarrow \mathbb{R}$ with $g_z = \frac{1}{2} \text{Hess} \mathbf{W}[z, \cdot]$ restricted to $T_z \mathcal{M} \times T_z \mathcal{M}$. Here, $\mathbf{W}[z, \tilde{z}] = \mathbf{W}_{\text{mem}}[z, \tilde{z}] + \mathbf{W}_{\text{bend}}[z, \tilde{z}]$

is the elastic energy of a deformation of the discrete shell with coordinates z into the discrete shell with coordinates \tilde{z} . It is composed of a membrane and a bending contribution with $\mathbf{W}_{\text{bend}}[z, \tilde{z}] = \sum_{e \in \mathcal{E}} (\theta_e - \tilde{\theta}_e)^2 d_e^{-1} l_e^2$, where $d_e = \frac{1}{3}(a_f + a_{f'})$ for the two faces f and f' adjacent to $e \in \mathcal{E}$ (a_f is the area of f) and $\mathbf{W}_{\text{mem}}[z, \tilde{z}] = \sum_{f \in \mathcal{F}} a_f \cdot W_{\text{mem}}(G[z, \tilde{z}]|_f)$, where $W_{\text{mem}}(A) := \frac{\mu}{2} \text{tr} A + \frac{\lambda}{4} \det A - \left(\mu + \frac{\lambda}{2}\right) \log \det A - \mu - \frac{\lambda}{4}$. Furthermore, μ and λ are positive material constants and $G[z, \tilde{z}]$ is the Cauchy–Green strain tensor of the deformation, which is a function of the edge lengths on each face. The logarithmic term in the energy density W_{mem} ensures that the triangle inequalities remain fulfilled for finite-energy deformations.

The path energy of a path $(z(t))_{t \in [0,1]}$ on \mathcal{M} is defined as

$$\mathbf{E} = \mathbf{E}[(z(t))_{t \in [0,1]}] = \int_0^1 g_z(\dot{z}, \dot{z}) dt$$

and, for fixed end points $z_0 = z(0)$ and $z(1)$, its minimizers are geodesics. Then, $\log_{z_0} z(1) = \dot{z}(0)$ defines the logarithm and $\exp_{z_0} v_0 = z(t)$ with $v_0 = \dot{z}(0)$ the exponential map (a solution of the ODE $\nabla_{\dot{z}(t)} \dot{z}(t) = 0$ for the connection ∇ associated with the metric).

For the Riemannian distance, we have that $\text{dist}^2(z(0), z(1)) = \mathbf{E}$ and straightforwardly $\mathbf{E} = K \sum_{k=1}^K \text{dist}^2(z_{k-1}, z_k)$ with $z_k = z(\frac{k}{K})$. In our case of the discrete shells, $\mathbf{W}[z, \tilde{z}] = \text{dist}^2(z, \tilde{z}) + O(\text{dist}^3(z, \tilde{z}))$. Thus, we replace the squared Riemannian distance by the computationally much cheaper elastic deformation energy and obtain the discrete path energy

$$\mathbf{E}^K[z_0, \dots, z_K] = K \sum_{k=1}^K \mathbf{W}[z_{k-1}, z_k]$$

for discrete paths $(z_0, \dots, z_K) \in (\mathbb{R}^{2|\mathcal{E}|})^{K+1}$. Then, discrete K -geodesics are defined as minimizers of the discrete path energy for given end points z_0 and z_K and the associated system of Euler-Lagrange equations is given by

$$\partial_2 \mathbf{W}[z_{k-1}, z_k] + \partial_1 \mathbf{W}[z_k, z_{k+1}] = 0 \quad \text{for } k = 1, \dots, K-1.$$

4. Principal Geodesic Analysis

In this section, we will discuss the generalization of Principal Component Analysis on linear spaces to Riemannian manifolds. In this context, the Riemannian logarithm and exponential map enable to transform nonlinear, large variations on the manifold to infinitesimal variations given as tangent vectors in a particular tangent space and vice versa. Given the already introduced discrete geodesic calculus, we use discrete counterparts of the continuous logarithm and the continuous exponential map. We define the discrete logarithm at some discrete shell z_0 as

$$\text{Log}_{z_0}(z_K) = v_0$$

where $v_0 = K \mathbf{P}_{T_{z_0} \mathcal{M}}(z_1 - z_0)$ is a discrete initial velocity of the discrete geodesic (z_0, z_1, \dots, z_K) based on time step size $\tau = \frac{1}{K}$. Here, $\mathbf{P}_{T_{z_0} \mathcal{M}}$ is the orthogonal projection onto the tangent space $T_{z_0} \mathcal{M}$ with respect to the metric g_{z_0} , which is extended to $\mathbb{R}^{2|\mathcal{E}|}$ per its definition. The corresponding exponential map $\text{Exp}_{z_0}(v_0)$ for $v_0 \in T_{z_0} \mathcal{M}$, as the inverse mapping, computes a discrete geodesic (z_0, z_1, \dots, z_K) for given discrete, initial velocity v_0 . In

fact, we first compute z_1 for given z_0 and v_0 via orthogonal projection of $z_0 + \frac{v_0}{K}$ onto \mathcal{M} with respect to g_{z_0} , i.e. $z_1 := \mathbf{P}_{\mathcal{M}}[z_0 + \frac{v_0}{K}]$. The projection $\mathbf{P}_{\mathcal{M}}$ can be computed using a Gauß–Newton scheme [FB11]. Then we recursively compute z_{k+1} solving $\partial_2 \mathbf{W}[z_{k-1}, z_k] + \partial_1 \mathbf{W}[z_k, z_{k+1}] = 0$ in a least squares sense, i.e.

$$z_{k+1} = \arg \min_{z \in \mathcal{M}} \|\partial_2 \mathbf{W}[z_{k-1}, z_k] + \partial_1 \mathbf{W}[z_k, z]\|_2^2.$$

After $K - 1$ steps this yields the requested discrete geodesic and the discrete exponential, i.e.

$$\text{Exp}_{z_0}(v_0) := z_K.$$

For given points $z^1, \dots, z^N \in \mathcal{M}$, the continuous Fréchet mean \bar{z} is defined as the $\arg \min_{z \in \mathcal{M}} \sum_{n=1, \dots, N} \text{dist}^2(z^n, z)$. In what follows, we replace continuous geodesics used in the definition of $\text{dist}(\cdot, \cdot)$ by discrete K -geodesics to define a discrete (Fréchet) K -mean \bar{z} . Of course, these definitions depend on the choice of K , however, we always used $K = 8$ for Exp and Log and did not observe any benefit by increasing it in our experiments, see also Section 9. Concerning the mean, we obtained already very good results for the 1-mean \bar{z} , which can be considered as the elastic mean [RW09].

Now, we are in the position to introduce the discrete Principal Geodesic Analysis (PGA) [FLPJ04] for \mathcal{M} . Again, let $z^1, \dots, z^N \in \mathcal{M}$ be a given data base of discrete shells.

Discrete PGA algorithm

1. compute the mean \bar{z} of the input shapes z^1, \dots, z^N .
2. compute $v^n = \text{Log}_{\bar{z}}(z^n)$ for $n = 1, \dots, N$.
3. perform a PCA on the tangent space $T_{\bar{z}}\mathcal{M}$ with $g_{\bar{z}}$ as the underlying scalar product to compute the dominant J principal modes of variation $u_1, \dots, u_J \in T_{\bar{z}}\mathcal{M}$.

The actual PCA computation (3.) can be formulated as

$$\begin{aligned} & \underset{\substack{U \in \mathbb{R}^{2|\mathcal{E}| \times J} \\ W \in \mathbb{R}^{J \times N}}}{\text{minimize}} \quad \|V - UW\|_g^2 \\ & \text{subject to} \quad g_z(u_i, u_j) = \delta_{ij} \text{ for } i, j \in \{1, \dots, J\}, \end{aligned} \quad (3)$$

where $V \in \mathbb{R}^{2|\mathcal{E}| \times N}$ is the matrix containing the v^j as columns and the columns of U are the modes u_1, \dots, u_J . The approximation error is measured in a Frobenius norm $\|\cdot\|_g$ weighted according to $g_{\bar{z}}$. Taking into account the minimizing property and the fact that $v^n \in T_{\bar{z}}\mathcal{M}$, one easily checks that all u_j are in $T_{\bar{z}}\mathcal{M}$.

For each u_j , the curve $t \rightarrow \text{Exp}_{\bar{z}}(t u_j)$ can be interpreted as the j th nonlinear mode of shape variation of the input data. Moreover, considering the whole subspace $\mathcal{U} = \text{span}\{u_1, \dots, u_J\}$ of the tangent space $T_{\bar{z}}\mathcal{M}$ the PGA allows us to define a submanifold

$$\mathcal{M}[\mathcal{U}] := \text{Exp}_{\bar{z}}(\mathcal{U})$$

of the NRIC manifold \mathcal{M} which approximates the input data.

In comparison to linear PCA in $\mathbb{R}^{2|\mathcal{E}|}$, i.e. using the (projected) linear mean and linear differences, the nonlinearity allows us to better capture the articulation of the input shapes. This means for the same linear approximation quality, i.e. $\frac{\|V - UW\|_g^2}{\|V\|_g^2}$, the actual approximation of the input shapes z^n is more accurate using nonlinear PGA and the exponential map than using linear PCA and orthog-

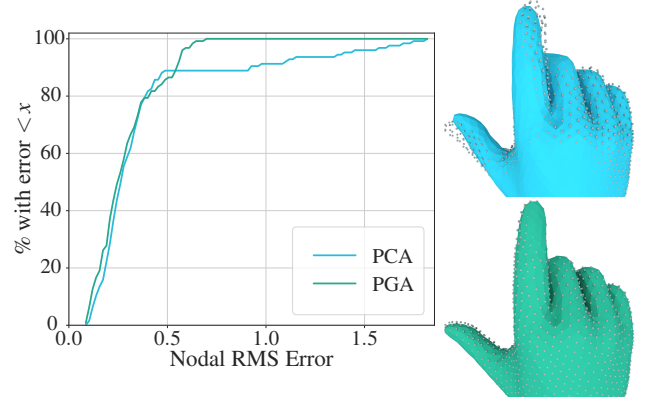


Figure 3: Comparison of nonlinear PGA (green) with linear PCA (blue) in NRIC. The histogram shows the distribution of errors in terms of mismatch of nodal position for reconstruction of the input shapes. Moreover, we compare for a selected input shape (vertices shown as gray point cloud) the approximation using PCA and PGA respectively. In both cases, we used $J = 8$ components and both models achieved a linear approximation quality of about 98%.

onal projection on \mathcal{M} . A comparison of this nonlinear PGA and a PCA in NRIC coordinates is given in Figure 3.

In general, $\text{Exp}_{\bar{z}}(t u_j + s u_i) \neq \text{Exp}_{\bar{z}}(t u_j) + \text{Exp}_{\bar{z}}(s u_i)$ by the nonlinearity of the exponential map. Nevertheless, after replacing standard PGA modes by sparse modes in NRIC coordinates in the next Section 5, the resulting manifold $\mathcal{M}[\mathcal{U}]$ itself will be approximated by linear superposition of very low-dimensional submanifolds in Section 7 – thus allowing for an efficient deformation synthesis in Section 8.

5. Sparse Principal Geodesic Analysis

By definition, conventional PGA outputs an orthonormal set of dominant deformation modes. These modes usually have large support as already the orthogonality implies largely overlapping supports of pairs of modes to ensure the annihilation in the $g_{\bar{z}}$ product. Motivated by the quest for more efficient parametrizations of data approximating submanifolds mentioned above and by the intuition that, in many applications, we are interested in sparse, spatially localized dominant modes with widely disjoint supports. For example, on human shapes, it seems natural to deal with dominant deformation modes which are supported only on a foot, the surrounding on a knee, or the area of a shoulder. The choice of NRIC, which represent local distortions via local coordinate changes, enables such a sparsity on articulated shapes in our approach.

To define sparse deformation modes, we follow the common approach of considering a sparsity-inducing regularization term. However, using an unweighted l_1 -norm on tangent vectors would be problematic as they consist of edge lengths and dihedral angles varying on vastly different scales. To this end, we introduce a family of weighted L^p -norms

$$\|u\|_p^p := \sum_{e \in \mathcal{E}} \left(\frac{1}{\bar{l}_e} |l_e| \right)^p + \sum_{e \in \mathcal{E}} \left(\frac{\bar{l}_e}{\sqrt{d_e}} |\theta_e| \right)^p, \quad u = (l, \theta) \in \mathbb{R}^{2|\mathcal{E}|},$$

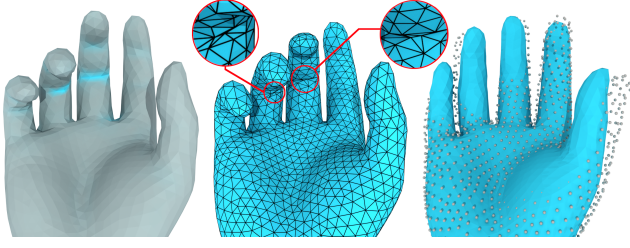


Figure 4: Comparison to sparse linear PCA with Euclidean metric in $\mathbb{R}^{2|\mathcal{E}|}$. Working without the tangent space constraint leads to modes with a mismatch between the length and the angle component. The dihedral angle support of a sparse linear PCA mode in $\mathbb{R}^{2|\mathcal{E}|}$ coordinates which has no length component at all is shown on the left. This leads to mesh artifacts already in case of short time extrapolation as seen in the middle picture. Furthermore, this is prohibitive also for the approximation of shapes, which can be seen in the last picture which shows the failure to approximate an input shape, which was easily handled even by the linear PCA in $\mathbb{R}^{2|\mathcal{E}|}$, even though the linear approximation quality is about 96%.

where \bar{d}_e is the area associated with e in the mean \bar{z} . The scalar product associated with the L^2 -norm has already been used in [FB11] to define a quadratic energy on lengths and angles. To simplify the notation in the following, we collect the weights in a vector $m \in \mathbb{R}^{2|\mathcal{E}|}$ such that $\|u\|_1 = |m \odot u|_1$, where \odot means entry-wise multiplication. With this L^1 -norm at hand, we generalize the PGA to a Sparse PGA (SPGA) as follows.

$$\begin{aligned} & \underset{\substack{U \in \mathbb{R}^{2|\mathcal{E}| \times J} \\ W \in \mathbb{R}^{J \times N}}}{\text{minimize}} & \|V - UW\|_g^2 + \lambda \sum_{j=1}^J \|u_j\|_1 \\ & \text{subject to} & u_j \in T_{\bar{z}}\mathcal{M} \text{ for } j \in \{1, \dots, J\} \\ & & \max_{i \in \mathbb{R}^N} |w_{ji}| \leq 1 \text{ for } j \in \{1, \dots, J\}. \end{aligned} \quad (4)$$

We dropped the orthonormality constraints as we do not aim for an orthonormal basis anymore but favor sparsity. To ensure that the magnitude of the coordinates of the u_j does not simply shrink to achieve a small L^1 -norm while the weights grow accordingly, we introduce a bound on the magnitude of the weights' entries. Different from standard PGA, we have to impose that the modes u_j are indeed tangent vectors in $T_{\bar{z}}\mathcal{M}$, which ensures that they yield admissible, infinitesimal deformations of the mean \bar{z} . Dropping this constraint $D\mathcal{Q}(z)u_j = 0$, one would leave the NRIC manifold \mathcal{M} and there is no guarantee that u_j represents a geometrically admissible variation of the underlying triangular mesh. To illustrate this, we added in Figure 4 just the L^1 -regularization to the linear PCA. As we can see, this led to components of edge length and dihedral angle variation with non-matching support. Their usability for approximation and the synthesis of new shapes is quite limited. In fact, it prohibits the use of extrapolation via the exponential map.

Finally, we can again by virtue of the exponential map define a submanifold

$$\mathcal{M}[U] = \{\text{Exp}_{\bar{z}}(v) \mid v \in U\}$$

of the NRIC manifold \mathcal{M} containing nonlinear deformations.

6. Quadratic Matrix Programming

We solve (4) by alternatingly solving for U while keeping W fixed and vice versa. When U is fixed, solving for W becomes a straightforward quadratic optimization problem. However, solving for U is more difficult due to the regularization term and the tangent space constraint.

To solve for the sparse modes efficiently, we rephrase (4) for fixed weights W as a *quadratic matrix programming* problem [Bec07]. To this end, we first reformulate the nonlinear matrix optimization problem, and then relax the L^1 -penalty term as in [Tib96; BH17] to obtain a quadratic problem.

Matrix optimization. Let us begin by reformulating the matrix optimization problem to prepare the relaxation. We denote by $G \in \mathbb{R}^{2|\mathcal{E}| \times 2|\mathcal{E}|}$ the matrix representation of the metric $g_{\bar{z}}$, i.e. $G = \mathbf{W}_{,22}[\bar{z}, \bar{z}]$. By the definition of the (weighted) Frobenius norm,

$$\begin{aligned} \|V - UW\|_g^2 &= \text{tr}((V - UW)^T G (V - UW)) \\ &= \text{tr}(W^T U^T G U W) - 2\text{tr}(V^T G U W) + \text{tr}(V^T G V) \\ &= \text{tr}(W^T U^T G U W) - 2\text{tr}(W V^T G U) + \text{tr}(V^T G V). \end{aligned}$$

To rephrase the regularization term, we denote by $M \in \mathbb{R}^{J \times 2|\mathcal{E}|}$ the matrix whose rows are all equal to the weight vector m . This allows the following reformulation

$$\sum_{j=1}^J \|u_j\|_1 = \text{tr}(M^T |U|), \quad |U| := (|u_{kj}|)_{\substack{k=1, \dots, 2|\mathcal{E}| \\ j=1, \dots, J}}$$

We express the tangent space constraint using the quaternion integrability operator \mathcal{Q} , i.e.

$$D_z \mathcal{Q}(\bar{z}) U = 0 \in \mathbb{R}^{3|V_0| \times J}$$

Together, this yields for given W the problem

$$\begin{aligned} & \underset{U \in \mathbb{R}^{2|\mathcal{E}| \times J}}{\text{minimize}} & \text{tr}(W^T U^T G U W) - 2\text{tr}((G V W^T)^T U) \\ & & + \lambda \text{tr}(M^T |U|) + \text{tr}(V^T G V) \\ & \text{subject to} & D_z \mathcal{Q}(\bar{z}) U = 0. \end{aligned} \quad (5)$$

While this problem already closely resembles a QMP problem, the choice of the L^1 -term still renders it nonlinear.

Solving the matrix optimization problem by relaxation. Next, we relax the L^1 -penalty (cf. [BH17; Tib96]) to obtain the quadratic problem with inequality constraints. To this end, we introduce non-negative variables $U^+, U^- \in \mathbb{R}^{2|\mathcal{E}| \times J}$ with

$$U = U^+ - U^-, \quad U^+, U^- \geq 0,$$

This way, using the componentwise triangle inequality $|U| \leq U^+ + U^-$ we obtain the estimate

$$\text{tr}(M^T |U|) \leq \text{tr}(M^T (U^+ + U^-))$$

for the L^1 -term. Analogously, adapting the other terms in (5) we finally arrive at the relaxed quadratic matrix programming problem

$$\begin{aligned}
 & \underset{\substack{U^+ \in \mathbb{R}^{2|\mathcal{E}| \times J} \\ U^- \in \mathbb{R}^{2|\mathcal{E}| \times J}}}{\text{minimize}} \quad \text{tr} \left(W^T \begin{pmatrix} U^+ \\ U^- \end{pmatrix}^T \begin{pmatrix} G & -G \\ -G & G \end{pmatrix} \begin{pmatrix} U^+ \\ U^- \end{pmatrix} W \right) \\
 & \quad + \text{tr} \left(\begin{pmatrix} \lambda M + 2GVW^T \\ \lambda M - 2GVW^T \end{pmatrix}^T \begin{pmatrix} U^+ \\ U^- \end{pmatrix} \right) \quad (6) \\
 & \quad + \text{tr}(V^T GV) \\
 & \text{subject to} \quad D_z \mathcal{Q}(\bar{z})U^+ - D_z \mathcal{Q}(\bar{z})U^- = 0, \\
 & \quad U^+ \geq 0, \quad U^- \geq 0.
 \end{aligned}$$

The above triangle inequality is an equality, i.e. $|U| = U^+ + U^-$ if $U^+ \odot U^- = 0$. Hence, the solutions U^+, U^- of (6) have disjoint support because, otherwise, we could move values from one matrix to the other without changing the value of the objective term while decreasing the regularization term. Thus, $U = U^+ - U^-$ is indeed the solution of the original problem. To solve the relaxed problem, one can first vectorize it (cf. [Bec07]) and then utilize off-the-shelf software for quadratic programming.

7. Effective Submanifold Approximation

In many applications, the sparse modes (e.g. see Figure 5) can at least partially be split into subsets with pairwise decoupled support. Recall, for human models one expects distinct, localized articulations for certain joints and muscle groups. Hence, the supports of corresponding groups of sparse modes representing these articulations are well separated. Furthermore, as pointed out in Figure 7, the support of these tangent vectors in $T_{\bar{z}}\mathcal{M} \subset \mathbb{R}^{2|\mathcal{E}|}$ is only very moderately extended under application of the exponential $\text{Exp}_{\bar{z}}$ even for large K . Given such a strong separation of supports of two subspaces \mathcal{U}_a and \mathcal{U}_b one observes that

$$\text{Exp}_{\bar{z}}(u_a + u_b) - \bar{z} \approx (\text{Exp}_{\bar{z}}(u_a) - \bar{z}) + (\text{Exp}_{\bar{z}}(u_b) - \bar{z})$$

for $u_a \in \mathcal{U}_a$ and $u_b \in \mathcal{U}_b$. This is certainly not the case for subspaces corresponding to modes with overlapping supports, which is illustrated in Figure 6 using the inset modes. In fact, the defect of a potentially non exact separation of supports after applying the exponential map can be measured using

$$\mathcal{D}^{\text{Exp}}(u_a, u_b) := \frac{(|\text{Exp}_{\bar{z}}(u_a) - \bar{z}|, |\text{Exp}_{\bar{z}}(u_b) - \bar{z}|)_2}{\|\text{Exp}_{\bar{z}}(u_a) - \bar{z}\|_2 \|\text{Exp}_{\bar{z}}(u_b) - \bar{z}\|_2},$$

where $(\cdot, \cdot)_2$ is the scalar product associated with the weighted L^2 -norm introduced in Section 5, see also Figure 7. Before, we have already argued that the support of the tangent vectors is only moderately extended under the exponential map. The same then holds for the coupling of pairs of tangent vectors, which allows us to measure the separation of modes and induced submanifolds $\text{Exp}_{\bar{z}}(\mathcal{U}_a)$

and $\text{Exp}_{\bar{z}}(\mathcal{U}_b)$ using

$$\mathcal{D}(u_a, u_b) := \frac{(|u_a|, |u_b|)_2}{\|u_a\|_2 \|u_b\|_2},$$

which can also be seen when comparing Figure 8 and Figure 7.

This observation motivates the following splitting approach. Let us suppose that the space $\mathcal{U} \subset T_{\bar{z}}\mathcal{M}$ spanned by the sparse principal modes u_1, \dots, u_J can be written as the direct sum of L subspaces \mathcal{U}_l for $l = 1, \dots, L$, i.e.

$$\mathcal{U} = \mathcal{U}_1 \oplus \mathcal{U}_2 \oplus \dots \oplus \mathcal{U}_L,$$

where each of these subspaces is spanned by a subset of the principal modes. In fact, as discussed above, we suppose that due to our sparse PGA approach the subspaces \mathcal{U}_j can be chosen to have at least approximately pairwise disjoint supports. For each vector $v \in \mathcal{U}$, we have a corresponding decomposition $v = v_1 + v_2 + \dots + v_L$ with $v_l \in \mathcal{U}_l$. Now, we consider a superposition of exponential maps via summation of NRIC coordinates

$$\mathbf{Z}[v] = \bar{z} + \sum_{l=1}^L (z_l - \bar{z}) \quad \text{with } z_l = \text{Exp}_{\bar{z}} v_l.$$

Finally, $\mathbf{Z}[v]$ is not necessarily on \mathcal{M} and we have to project back onto \mathcal{M} and obtain

$$z[v] = \mathbf{P}_{\mathcal{M}} \mathbf{Z}[v].$$

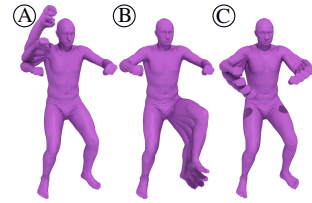


Figure 5: Three sparse modes A, B, C for the human model (used in Figure 6 and Figure 10).

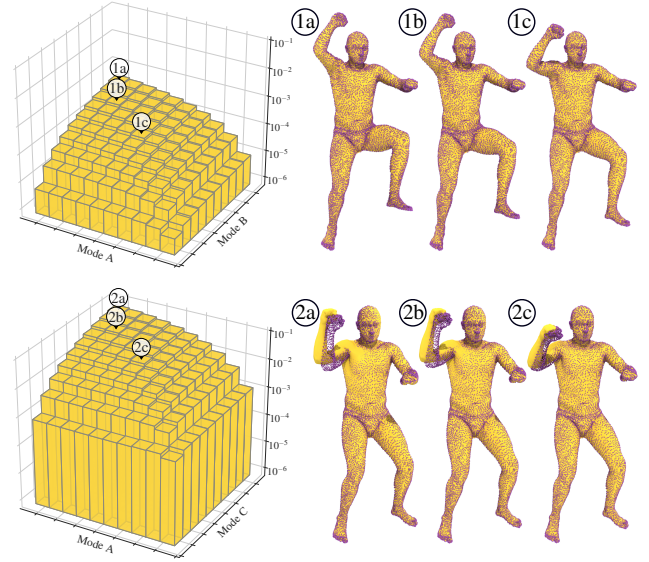


Figure 6: Comparison of splitting error for two separated modes (top row) and two overlapping modes (bottom) (cf. Figure 5). We consider a deformation synthesis $z[v]$, which treats the modes separately and show the relative approximation error in the energy, i.e. $\mathbf{W}[\text{Exp}_{\bar{z}}(v), z[v]] / \mathbf{W}[\bar{z}, \text{Exp}_{\bar{z}}(v)]$, on periodically distributed samples as bar plot along with three examples (approximation in yellow, correct extrapolation as purple point cloud).

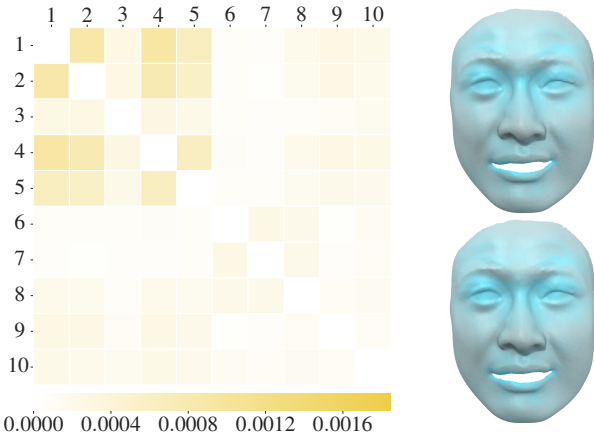


Figure 7: $\mathcal{D}^{\text{Exp}}(u_i, u_j)$ is plotted as heatmap on the left for $i, j \in \{1, \dots, J\}$. On the right, we show the extrapolation $\text{Exp}_{\bar{z}}(u)$ of a randomly sampled mode $u \in \mathcal{U}$, once colored according to the absolute tangential mode $|u|$ (top) and once colored according to the absolute difference $|\text{Exp}_{\bar{z}}(u) - \bar{z}|$ (bottom) both on the same color scale underlining the close similarity of the support of $u \in T_{\bar{z}}\mathcal{M} \subset \mathbb{R}^{2|\mathcal{E}|}$ and $\text{Exp}_{\bar{z}}(u) - \bar{z} \in \mathbb{R}^{2|\mathcal{E}|}$.

This defines the submanifold

$$\mathcal{M}[\mathcal{U}_1, \dots, \mathcal{U}_L] = \{z[v] \mid v \in \mathcal{U}\} \subset \mathcal{M}$$

which approximates the original SPGA submanifold $\mathcal{M}[\mathcal{U}]$.

8. Efficient Deformation Synthesis

So far, we introduced an approximation of the submanifold by splitting the application of the exponential map into applications on smaller subspaces. It remains to show how these exponential maps and the needed projection can be approximated efficiently to enable the fast synthesis of nonlinearly deformed shapes on the submanifold. We propose an approach based on sampling in the preprocessing phase and multilinear interpolation in the online phase.

To this end, we first consider a single subspace \mathcal{U}_l of dimension d with basis u_1^l, \dots, u_d^l . In this subspace, we take a lattice $\{\sum_{n=1}^d \alpha_n \tau u_n^l \mid \alpha \in \mathbb{Z}^d\}$ with mesh size $\tau \in \mathbb{R}$. For $\alpha \in \mathbb{Z}^d$, we denote by $v^\alpha = \sum_{n=1}^d \alpha_n \tau u_n^l$ the corresponding lattice point. Now, an arbitrary $v = \sum_{n=1}^d w_n(v) u_n^l \in \mathcal{U}_l$ lies in a cell of the lattice with nodes $v^{\alpha(v)+\beta}$ for $\beta \in \{0, 1\}^d$, where $\alpha(v) = \lfloor \frac{w(v)}{\tau} \rfloor \in \mathbb{Z}^d$ element-wise. In two dimensions, this would mean that $\alpha(v)$ identifies the lattice point to the lower-left of v . To approximate $\text{Exp}_{\bar{z}}(v)$, we consider a piecewise multilinear interpolation $\mathbf{Z}_{\tau, l}$ of the values at lattice points, *i.e.*

$$\mathbf{Z}_{\tau, l}[v] := \sum_{\beta \in \{0, 1\}^d} w_\beta(v) \text{Exp}_{\bar{z}}(v^{\alpha(v)+\beta})$$

where the $w_\beta(v)$ are the multilinear interpolation weights (*cf.* [WZ88]) such that $v = \sum_{\beta \in \{0, 1\}^d} w_\beta(v) v^{\alpha(v)+\beta}$. In the offline phase, exponentials of lattice points are computed for a suitable finite subset of \mathbb{Z}^d and stored in lookup tables, which can be used

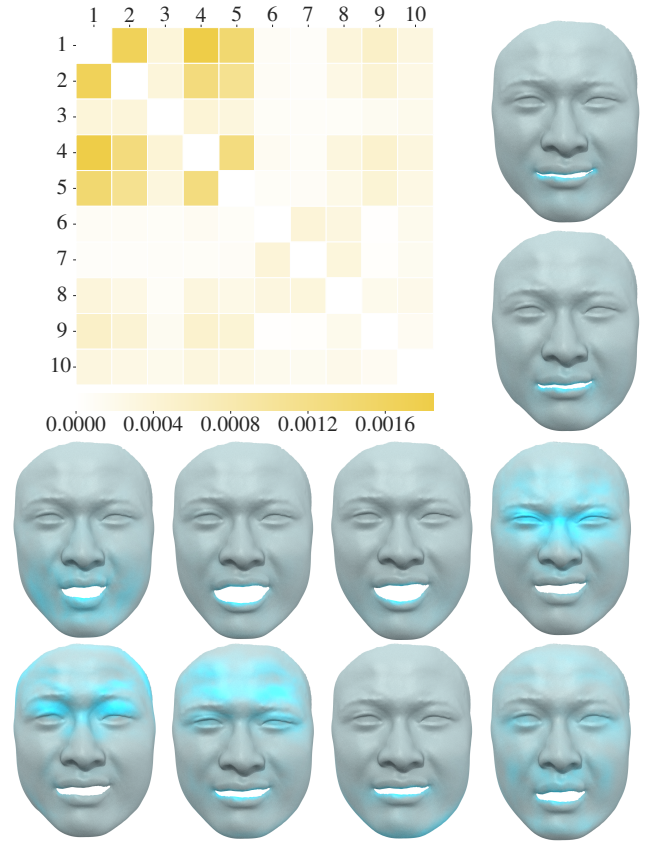


Figure 8: Coupling of tangential modes on a face data set [ZSCS04] exhibiting non-isometric deformations. The heatmap shows the coupling between the different tangential modes, *i.e.* $\mathcal{D}(u_i, u_j)$ for $i, j \in \{1, \dots, J\}$. It is accompanied by the extrapolated modes colored according to the absolute tangential modes $|u_i|$ (weighted as in the L^1 -norm), each on a different scale to improve the visual appearance of the modes.

to cheaply evaluate the piecewise multilinear interpolation in the online phase.

Considering the whole subspace $\mathcal{U} = \mathcal{U}_1 \oplus \dots \oplus \mathcal{U}_L$, we use for general $v \in \mathcal{U}$ the splitting $v = v_1 + \dots + v_L$ and repeat the interpolation above for each subspace. Then, we evaluate the superposition of these interpolations

$$\mathbf{Z}_\tau[v] = \bar{z} + \sum_{l=1}^L (\mathbf{Z}_{\tau, l}[v_l] - \bar{z})$$

for the actual synthesis of nonlinear deformations. As above for $\mathbf{Z}[v]$ the superposition $\mathbf{Z}_\tau[v]$ is in general not in \mathcal{M} and we define the submanifold $\mathcal{M}^\tau[\mathcal{U}_1, \dots, \mathcal{U}_L] = \{z_\tau[u] \mid u \in \mathcal{U}\}$ for

$$z_\tau[u] = \mathbf{P}_\mathcal{M} \mathbf{Z}_\tau[u]$$

which again is expected to approximate the submanifold $\mathcal{M}[\mathcal{U}]$. The necessity of the projection is demonstrated in Figure 9.

Surely, when dealing with the submanifolds $\text{Exp}_{\bar{z}}(\mathcal{U}_l)$ in our

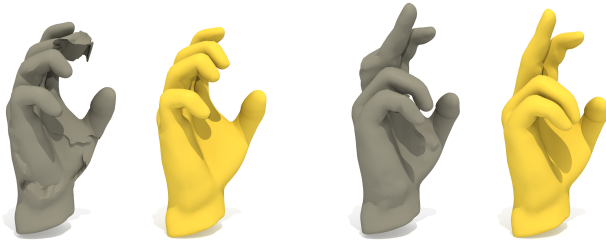


Figure 9: Two comparisons of projected shapes $z_\tau[v]$ (yellow) with adaptive frame-based reconstruction [SHHR20] of $Z_\tau[v]$ without projecting it on the manifold (gray). This shows that the projection is required.

computational setup, we are interested only in compact subsets of the subspace \mathcal{U}_l representing plausible deformations. Furthermore, in the applications, it turned out to be sufficient to restrict to subspaces \mathcal{U}_l of low dimensionality between one and four. Hence, the multilinear interpolation of precomputed samples on a rectangular grid becomes a feasible option. A comparison of the exact submanifold $\mathcal{M}[\mathcal{U}_l] = \text{Exp}_{\bar{z}}(\mathcal{U}_l)$ and the approximation $\mathcal{M}^\tau[\mathcal{U}_l]$ for a single two dimensional subspace \mathcal{U}_l is shown in Figure 10. The efficiency of this approach based on multilinear interpolation of precomputed samples is evaluated in the next section and in particular in Table 4.

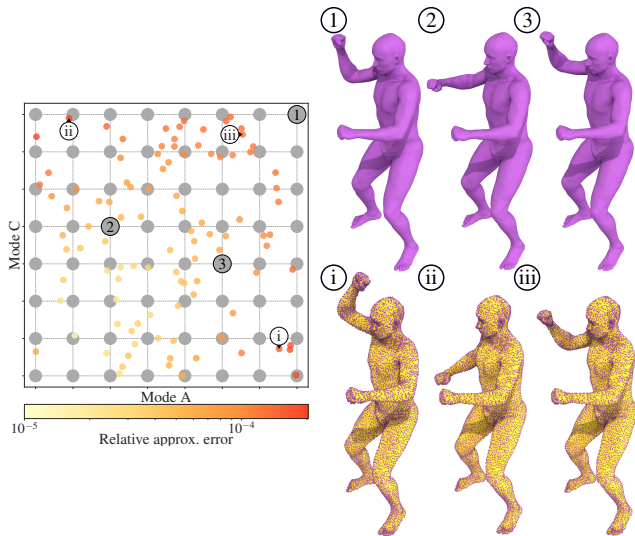


Figure 10: Approximation of the exponential map by multilinear interpolation of lattice points for two strongly coupled modes (cf. Figure 5). On the left, we show the lattice (grey) in the two-dimensional subspace along with randomly sampled points colored according to the logarithm of their relative approximation error in the energy, i.e. $\mathbf{W}[\text{Exp}_{\bar{z}}(v), z_\tau[v]] / \mathbf{W}[\bar{z}, \text{Exp}_{\bar{z}}(v)]$. On the right, selected extrapolated lattice points are shown in purple and we highlight the quality of the approximation of shapes (purple point clouds) by our deformation synthesis approach (yellow) for samples with relatively high approximation error.

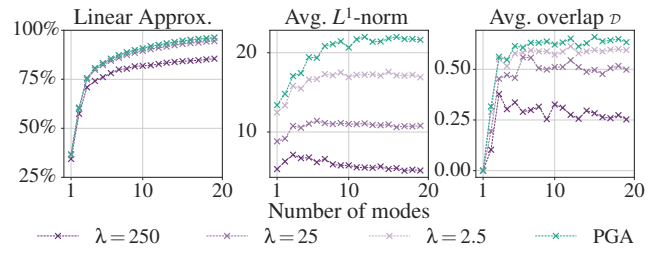


Figure 11: Comparison of multiple choices of J and λ on the Face dataset. The linear approximation quality is measured as $\|V - UW\|^2 / \|V\|^2$ and weighted L^1 -norm and overlap are computed as described before. Noticeably larger values for λ led to vanishing modes. In this case, we chose $J = 10$ and $\lambda = 250$.

9. Experimental Results

We have implemented our method in C++ with the Geometric Optimization And Simulation Toolbox (GOAST) [HS*20], where we use the Eigen library [GJ*10] for numerical linear algebra and CHOLMOD [CDHR08] and UMFPACK [Dav04] from the SuiteSparse collection as direct linear solvers. For the quadratic problem (6), we use MOSEK [MOS20] as efficient off-the-shelf solver.

We use a multi-resolution approach to enable fast computations while retaining the possibility to produce high-quality deformations. In this approach, the input shapes are coarsened by one of two methods: For the human model and the horse input data, we used an iterative edge collapse approach based on minimizing the quadric error metric [GH97] computed in groups [MG03] to preserve the dense correspondence between input shapes. For the hand and face input data, we remeshed a reference input shape using OpenFlipper [MK12] and computed coarse representations of the other input shapes with the same approach as in the prolongation described next. In both cases, the coarse results were prolonged to the fine level using a representation of the fine mesh vertices in terms of intrinsic positions and normal displacement with respect to the coarse mesh and computed on a reference shape similar to [KMP07]. See Table 4 for a comparison of original and coarse resolutions of the different input data.

Selection of number of modes and sparsity. In the computation of modes, we have two parameters which are currently chosen manually based on heuristics: The number of modes J and the sparsity weight λ in (4). This is linked to a trade-off between the approximation quality of the model, the sparsity and overlap of support of the modes, and the size of the model. In Figure 11, we compare these quantitatively for a number of choices and also to PGA. For the approximation, not only the average overlap is important but also the resulting subspace dimensions. Thus, a higher number of modes is not necessarily beneficial.

Selection of coupling spaces. The selection of the subspace decomposition $\mathcal{U} = \mathcal{U}_1 \oplus \dots \oplus \mathcal{U}_L$, i.e. the grouping of the sparse tangential modes based on their coupling, is an important step to obtaining high-quality results with our proposed method as exemplarily pointed out in Figure 6. In our implementation, we employ

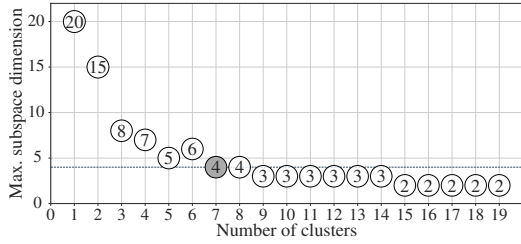


Figure 12: Comparison of maximal subspace dimension for different numbers of clusters on the Horse dataset. The selected number highlighted in gray.

spectral clustering [SM00] of a set of modes using the sparse tangential coupling $\mathcal{D}(u_i, u_j)$ as the underlying similarity measure. To this end, we use the coupling matrix of the sparse modes, cf. Figure 8, as affinity or similarity matrix. The number of clusters is then determined such that the resulting dimensionality of the subspaces \mathcal{U}_j is in the range $1, \dots, 4$, see also Figure 12, and thus small enough for the lattice generation described in Section 8 to be computationally feasible. The decomposition into subspaces of different dimensions used in our examples is listed in Table 1. The impact of the coupling of sparse tangential modes on the decoupling of the associated nonlinear submanifolds is depicted in Figure 7. Examples of groups of modes extracted from the SCAPE data set are shown in Figure 1. The examples illustrate that modes that move the same part of the body, for example the left leg in the leftmost groups, can be in different groups. The reason is that the distortions induced by the modes are located in different areas of the body, in the hip and knee regions for the leftmost groups.

Example	J	1D	2D	3D	4D
SCAPE	40	0	5	6	3
Hands	12	0	1	2	1
Faces	10	4	1	0	1
Horse	20	0	2	4	1

Table 1: Distribution of the dimensionality of subspaces \mathcal{U}_j used in the different examples.

Deformation synthesis. A central aim of our method is the fast synthesis of high-quality and large nonlinear deformations in articulated motion, *i.e.* the evaluation of our discrete approximation z_τ of the exponential map for varying subspace coordinates $\alpha \in \mathbb{R}^J$. We demonstrate this by considering curves $C: \mathbb{R} \rightarrow \mathcal{U}$ in the subspace and their counterpart $c: \mathbb{R} \rightarrow \mathcal{M}^\tau$ on the manifold \mathcal{M} obtained via our deformation synthesis, *i.e.* $c(t) := z_\tau[C(t)]$. Such curves can, for example, be used to obtain smooth deformations interpolating given key poses on the manifold. In Figure 13, we show such an interpolating curve based on a Catmull–Rom spline $t \mapsto C(t)$ [CR74], for a set of hand poses. To compute the projection onto the manifold, we follow [FB11] and formulate it as nonlinear least squares problem in nodal positions, *i.e.* $\min_{X \in \mathbb{R}^{3|\mathcal{V}|}} \|(l, \theta)(X) - \mathbf{Z}_\tau[u]\|_g^2$. This directly gives us the nodal positions needed for visualizations and can be solved efficiently using the Gauß–Newton method. Then, evaluating the curve $t \mapsto$

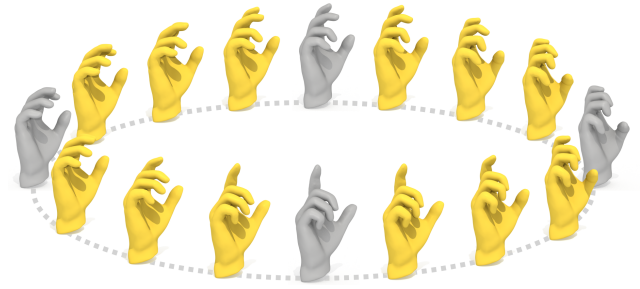


Figure 13: An interpolating deformation path computed using our method. The interpolation is obtained by fitting a periodic Catmull–Rom spline through the subspace coordinates of the input shapes (shown in grey) and evaluating our discrete parametrization for evenly spaced points on the spline (yellow).

$z_\tau[C(t)]$ along consecutive points can be done in (near) real-time in these examples as each step only needs very few Gauß–Newton iterations to compute the projection $\mathbf{P}_\mathcal{M}$ due to the good initialization from the previous step. See also the paragraph *Timings* below.

Mesh editing. The deformation synthesis restricted to the approximating submanifold $\mathcal{M}^\tau[\mathcal{U}_1, \dots, \mathcal{U}_L]$ for a sparse mode decoupling $\mathcal{U} = \mathcal{U}_1 \oplus \dots \oplus \mathcal{U}_L$ can also be used as deformation prior for mesh editing. In this case, we assume the editing information is given as M sparse handle positions $x_1, \dots, x_M \in \mathbb{R}^3$ corresponding to the positions of vertices $v_1, \dots, v_M \in \mathcal{V}$. Then we perform the editing via a quadratic penalty method, *i.e.* we consider a series of optimization problems of the form

$$\min_{X \in \mathbb{R}^{3|\mathcal{V}|}, u \in \mathcal{U}} \|(l, \theta)(X) - \mathbf{Z}_\tau[u]\|_g^2 + \gamma \sum_{m=1}^M \|X_{v_m} - x_m\|_{\mathbb{R}^3}^2 \quad (7)$$

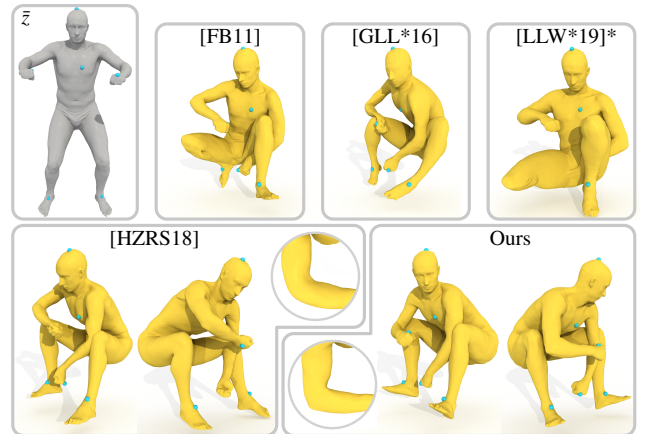


Figure 14: Handle editing comparison on SCAPE. Top row: Rest pose, results using the methods in [FB11], [GLL*16], and an adapted version of [LLW*19] (see main text for more information); Bottom row: results from [HZRS18] and ours along with close ups of the right elbows.

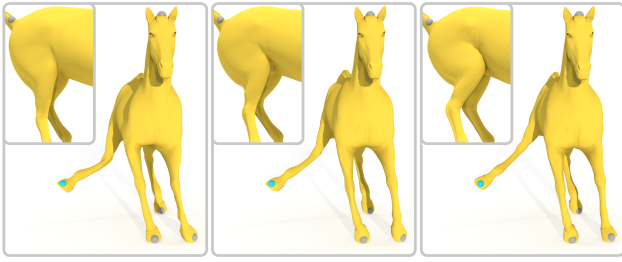


Figure 15: Challenging handle editing example on the horse as suggested in [LLW*19]. Left to right: editing with linear interpolation in $\mathbb{R}^{2|\mathcal{E}|}$ and hard constraints; editing with nonlinear interpolation and hard constraints; editing with nonlinear interpolation and soft constraints, i.e. via the quadratic penalty method (7).

for increasing γ , each again solved with Gauß–Newton. We choose the starting value for γ and increase it by a fixed factor until the mesh fulfills the handle positions up to a pre-defined tolerance.

In Figure 14, we compare this mesh editing approach to other state-of-the-art, data-driven approaches. To compare our approach to [LLW*19], we replicated their approach without localization, i.e. using a spatially constant l_1 -term in the mode computation, and with a fixed penalty parameter (see the right-most example in the top row). The other two methods, [FB11] and [GLL*16], are applied as described in the original publications. Our method delivers plausible deformations on-par with [HZRS18], while being nearly two orders of magnitude faster. Some details are even superior, for example, the bending of the right arm, shown in the close-ups, appears more natural in our results with the sharp elbow contour being better preserved.

Using the superposition of nonlinear subspaces represented by z_τ instead of the linear combinations of modes in $\mathbb{R}^{2|\mathcal{E}|}$ allows our method to generalize better for edits requiring deformations far from the input data set. For example, in Figure 15, we consider the challenging example of dragging the right hind leg outwards of a horse shape, while the input data only consists of a galloping sequence. Here, besides the penalty approach used in Figure 14 we used hard constraints for the vertex positions at the handles. We compare the editing via a linear basis and hard constraints in $\mathbb{R}^{2|\mathcal{E}|}$ (left), which leads to unnatural bending similar to the one reported

Example	PGA		SPGA	
	Off-	Online	Off-	Online
Fig. 13 (Interpolation)	3min	3s	90min	60ms
Fig. 14 (Fitting)	2min	919s*	660min	10s (100ms)

Table 2: Comparison of runtimes for PGA and SPGA. In the interpolation case, we report the average time to evaluate the spline in Figure 13 at 120 evenly spaced points. In the fitting case, the total amount of time to produce the results in Figure 14 is reported and for SPGA, in brackets, also the time for incremental editing steps.

* as reported in [HZRS18]

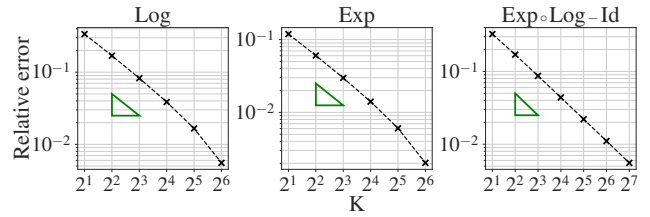


Figure 16: Convergence of time-discrete exponential and logarithm for $K \rightarrow \infty$ on the SCAPE dataset. For the logarithm and exponential, we show the relative error in the weighted L^2 -norm using $K_{max} = 128$ as pseudo ground truth. Furthermore, due to the tangent space projection the discrete exponential does not coincide with the inverse of the discrete logarithm. But, we observe that it is an approximation with decreasing relative error in the weighted L^2 -norm for $K \rightarrow \infty$.

in [LLW*19, Figure 17], with editing based on the large deformation interpolation $z_\tau[\cdot]$, both for hard constraints (middle), and using the quadratic penalty method (7) (right), where the latter two yield far more natural results.

Comparison PGA to SPGA. In comparison to PGA [HZRS18], our SPGA computes sparse modes at the expense of approximation accurateness. Hence, to achieve the same approximation quality with SPGA as with PGA one will typically need more modes (cf. Figure 11). The sparsity, however, is crucial for the submanifold approximation in Section 7, as otherwise the resulting error would be large (cf. Figure 6). When working with PGA, we cannot use our efficient deformation synthesis and instead have to directly use the nonlinear exponential map. In editing applications, when additional handle constraints come into play, this implies that we have to evaluate derivatives of the discrete exponential and hence use methods from PDE-constrained optimization, see [HZRS18]. This results in online runtimes which are far from interactive rates (Table 2). In contrast, our approximation built on SPGA allows for interactive rates but requires more runtime in the offline phase for the quadratic optimization and sampling of the exponential map.

Comparison to ADMM. Another method commonly used to compute sparse deformation components is the alternating direction method of multipliers (ADMM), cf. [NVW*13; HYZ*14; WLZH17]. We have also investigated this approach to solve for the sparse modes. To deal with the tangent space constraint in (4) one needs to add a third term to the commonly used ADMM problem representing this constraint as a convex indicator function. Then,

Solver	SCAPE		Hands		Faces	
	Obj.	Time	Obj.	Time	Obj.	Time
QMP	2618	1553s	575	205s	28765	127s
ADMM	2755	2092s	629	382s	31114	408s

Table 3: Objective values after ten outer iterations and runtime per outer iteration for QMP and ADMM. For ADMM, we choose the penalty parameters as $\eta = 10$ after comparing multiple options.

Dataset	Data & Parameters				Preprocessing				Online		
	N	$ \mathcal{V} $	$ \mathcal{V}_{\text{coarse}} $	J	Mean	Log	QMP	Exp	\mathbf{Z}_τ -eval.	GN iter (proj)	GN iter (edit)
SCAPE [ASK*05]	71	12.5k	1.3k	40	105s	30s	1553s	3s	0.5ms	18ms	100ms
Hands [YLSL11]	126*	6.1k	1.9k	12	230s	90s	205s	6s	0.3ms	25ms	–
Faces [ZSCS04]	384	24k	2.1k	10	420s	70s	127s	6s	0.3ms	35ms	–
Horse [SP04]	50	8.5k	1.3k	20	100s	50s	375s	4s	0.3ms	16ms	60ms

Table 4: Overview of used datasets with corresponding preprocessing and online timings listing from left to right: the data set, the number of input shapes N , the number of vertices of the full mesh $|\mathcal{V}|$ and the coarse mesh $|\mathcal{V}_{\text{coarse}}|$, the number of modes J , the timings in the preprocessing phase to compute the mean \bar{z} , a discrete logarithm $\text{Log}_{\bar{z}}$, one solution of the quadratic matrix programming problem, a discrete exponential map $\text{Exp}_{\bar{z}}$, and the timings in the online phase to compute the multilinear interpolations \mathbf{Z}_τ , a Gauß–Newton iteration used for the projection on the manifold \mathcal{M} , and a Gauß–Newton iteration required in the mesh editing context.

* obtained from six shapes by computing weighted elastic averages

as usual for ADMM one alternately evaluates the proximal mappings for each term. A more detailed explanation of this can, for example, be found in [BPC*11]. However, we observed a slower convergence, *i.e.* compared to our approach outlined in Section 5 the objective value was larger after the same number of outer iterations while still requiring more time per outer iteration. We provide objective values and runtimes for the two methods in Table 3.

Time discretization. To use the Riemannian exponential map and logarithm as tools in our method, we needed to discretize them in time, *cf.* Section 3. The discretization is controlled by the number of steps K in a discrete geodesic and converges for $K \rightarrow \infty$ to the corresponding continuous notions, which is shown in [RW15]. We demonstrate this convergence empirically for the SCAPE dataset in Figure 16 and also saw that for $K = 8$ we already achieve an acceptable error. Thus, we chose it for all experiments reported above.

Timings. In Table 4, we list detailed timings of all components of our method. For the quadratic problem, we report runtimes with enabled parallelization, while it was disabled for all other timings. The offline phase was performed on a workstation with two AMD EPYC 7601 CPUs, the high number of cores allows to efficiently parallelize the offline phase by computing many logarithms or exponential maps in parallel. All other results were computed on a laptop with an Intel Core i7-9750H CPU.

In the preprocessing phase, we used 25 outer iterations, *i.e.* solving once for the weights and modes, in each of the examples. Precomputing the exponential for lattice points (*cf.* Section 8) required about 3000 (horse), 4000 (hands), 20000 (human), and 30000 (face) evaluations for the respective examples.



Figure 17: Incremental mesh editing example with low-frequency deformations in the torso. Also shown in Figure 1.

In the online phase, evaluating our deformation synthesis z_τ for a random $v \in \mathcal{U}$ using the mean shape as initialization needed in all examples at most ten Gauß–Newton iterations. For larger, instant changes, such as in Figures 14 and 15, we need multiple increase of the penalty parameter in (7) and thus a larger number of iterations. Concretely, the computation for the result in Figure 14 took about ten seconds while for Figure 15 it took about five seconds. In applications with incremental changes, we only need a very small number of Gauß–Newton iterations. For example, each step of a spline for the shapes in Figure 13 evaluated at 120 evenly spaced points required two to three iterations, while for the incremental handle editing (shown in Figures 1 and 17) even one was sufficient. This makes the method applicable for interactive applications.

10. Conclusions

We introduced a framework for data-driven surface processing that allows synthesizing large, nonlinear deformations based on a Sparse Principal Geodesic Analysis on the space of discrete shells and enables close to real-time articulation of surface models and interactive mesh editing. Thereby, we use the Nonlinear Rotation-Invariant Coordinates to deal with the space of discrete shells as a Riemannian submanifold of $\mathbb{R}^{2|\mathcal{E}|}$.

The approach consists of an offline and an online phase.

– In the offline phase, we first use a discrete geodesic calculus to compute a mean shape and the discrete Riemannian logarithms of the input surfaces. From these logarithms, we identify sparse, dominant modes of shape variability. Second, the modes are grouped with respect to their overlap and by superposition of the discrete Riemannian exponential map acting on the span of each group a submanifold of the shape manifold, which fits the input data, is defined. Third, we compute a lookup table of sample shapes extrapolated this way.

– In the online phase, the synthesis of large nonlinear deformations is performed by summing of multilinear interpolations of samples from the lookup tables and finally projecting these close approximations by very few steps of a Gauß–Newton scheme onto the shape manifold.

We tested our framework on different shape data sets, evaluated the performance quantitatively and compared it to different state of the art methods.

Limitations and challenges. The current implementation can only

handle simply connected surfaces and input data which is in dense correspondence. A generalization to higher-genus surfaces would require the handling of additional, nonlocal integrability conditions. Computing the geodesics requires sufficient mesh quality of the input shapes to prevent local minima. We witnessed on the SCAPE dataset, that our method sometimes got stuck in such local minima. This, however, did not degrade the performance of the method due to its rareness. Still, extending the geodesic calculus to less quality meshes would increase the method's usability to a wider variety of datasets. The method still requires a projection onto the shape manifold after the superposition of groupwise extrapolations, constructing an approximation omitting this would improve performance and enable more flexible applications. Furthermore, a multiscale approach tailored to NRIC could help to further increase the method's performance. We have reported error measurements for the superposition of extrapolated groups of modes. Here, an a priori control of the error would enable a theoretically rigorous grouping of modes. Moreover, it would be interesting to combine the variational sparsity approach based on L^1 -norms with a variational mode untangling as in [SBBG11]. Finally, recent hierarchical tensor product approximation tools could allow generalizing the currently single scale grouping of modes and support an improved and more efficient manifold approximation.

Acknowledgments

The authors are grateful to Behrend Heeren for proofreading the manuscript. J. Sassen and M. Rumpf gratefully acknowledge the support provided by the Austrian Science Fund (FWF) through project NFN S117 and by the Deutsche Forschungsgemeinschaft (DFG) through the Hausdorff Center for Mathematics (GZ 204711, Project ID 390685813). Open access funding enabled and organized by Projekt DEAL. [Correction added on 19 November 2020, after first online publication: Projekt Deal funding statement has been added.]

References

- [AM00] ALEXA, MARC and MÜLLER, WOLFGANG. "Representing Animations by Principal Components". *Comput. Graph. Forum* 19.3 (2000), 411–418 3.
- [ASK*05] ANGUELOV, DRAGOMIR, SRINIVASAN, PRAVEEN, KOLLER, DAPHNE, et al. "SCAPE: shape completion and animation of people". *ACM Trans. Graph.* Vol. 24. 3. 2005, 408–416 12.
- [Bec07] BECK, AMIR. "Quadratic Matrix Programming". *SIAM Journal on Optimization* 17.4 (2007), 1224–1238 6, 7.
- [BH17] BRANDT, CHRISTOPHER and HILDEBRANDT, KLAUS. "Compressed vibration modes of elastic bodies". *Computer Aided Geometric Design* 52–53 (2017), 297–312 6.
- [BPC*11] BOYD, STEPHEN, PARIKH, NEAL, CHU, ERIC, et al. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". *Foundations and Trends in Machine Learning* 3 (2011), 1–122 12.
- [BVG09] BARAN, ILYA, VLASIC, DANIEL, GRINSPUN, EITAN, and POPOVIC, JOVAN. "Semantic deformation transfer". *ACM Trans. Graph.* 28.3 (2009), 36 3.
- [CDHR08] CHEN, YANQING, DAVIS, TIMOTHY A., HAGER, WILLIAM W., and RAJAMANICKAM, SIVASANKARAN. "Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate". *ACM Trans. Math. Softw.* 35.3 (2008) 9.
- [CR74] CATMULL, EDWIN and ROM, RAPHAEL. "A class of local interpolating splines". *Computer Aided Geometric Design*. Academic Press, 1974, 317–326 10.
- [Dav04] DAVIS, TIMOTHY A. "Algorithm 832: UMFPACK V4.3—An Unsymmetric-Pattern Multifrontal Method". *ACM Trans. Math. Softw.* 30.2 (June 2004), 196–199. ISSN: 0098-3500 9.
- [FB11] FRÖHLICH, STEFAN and BOTSCH, MARIO. "Example-Driven Deformations Based on Discrete Shells". *Comput. Graph. Forum* 30.8 (2011), 2246–2257 3, 5, 6, 10, 11.
- [FLPJ04] FLETCHER, P. THOMAS, LU, CONGLIN, PIZER, STEPHEN M., and JOSHI, SARANG. "Principal geodesic analysis for the study of nonlinear statistics of shape". *IEEE Trans. Med. Imaging* 23.8 (2004), 995–1005 3, 5.
- [GH97] GARLAND, MICHAEL and HECKBERT, PAUL. "Surface simplification using quadric error metrics". *ACM SIGGRAPH*. 1997, 209–216 9.
- [GHDS03] GRINSPUN, EITAN, HIRANI, ANIL N., DESBRUN, MATHIEU, and SCHRÖDER, PETER. "Discrete Shells". *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2003, 62–67 3.
- [GJ*10] GUENNEBAUD, GAËL, JACOB, BENOÎT, et al. *Eigen v3*. <http://eigen.tuxfamily.org>. 2010 9.
- [GLL*16] GAO, LIN, LAI, YU-KUN, LIANG, DUN, et al. "Efficient and Flexible Deformation Representation for Data-Driven Surface Modeling". *ACM Trans. Graph.* 35.5 (2016), 158:1–158:17 3, 10, 11.
- [GLY*20] GAO, L., LAI, Y., YANG, J., et al. "Sparse Data Driven Mesh Deformation". *IEEE Trans. Vis. Comput. Graph.* 26 (2020) 3.
- [HRS*14] HEEREN, BEHREND, RUMPF, MARTIN, SCHRÖDER, PETER, et al. "Exploring the Geometry of the Space of Shells". *Comput. Graph. Forum* 33.5 (2014), 247–256 3.
- [HRWW12] HEEREN, BEHREND, RUMPF, MARTIN, WARDETZKY, MAX, and WIRTH, BENEDIKT. "Time-Discrete Geodesics in the Space of Shells". *Comput. Graph. Forum* 31.5 (2012), 1755–1764 3, 4.
- [HS*20] HEEREN, BEHREND, SASSEN, JOSUA, et al. *The Geometric Optimization And Simulation Toolbox*. 2020. URL: <https://gitlab.com/numod/goast> 9.
- [HSS*09] HASLER, NILS, STOLL, CARSTEN, SUNKEL, MARTIN, et al. "A Statistical Model of Human Pose and Body Shape". *Comput. Graph. Forum* 28.2 (2009), 337–346 3.
- [HSvTP11] HILDEBRANDT, KLAUS, SCHULZ, CHRISTIAN, von TYCOWICZ, CHRISTOPH, and POLTHIER, KONRAD. "Interactive Surface Modeling Using Modal Analysis". *ACM Trans. Graph.* 30.5 (2011), 119:1–119:11 3.
- [HYZ*14] HUANG, ZHICHAO, YAO, JUNFENG, ZHONG, ZICHUN, et al. "Sparse Localized Decomposition of Deformation Gradients". *Comput. Graph. Forum* 33.7 (2014), 239–248 3, 11.
- [HZRS18] HEEREN, BEHREND, ZHANG, CHAO, RUMPF, MARTIN, and SMITH, WILLIAM. "Principal Geodesic Analysis in the Space of Discrete Shells". *Comput. Graph. Forum* 37.5 (2018) 2, 3, 10, 11.
- [JBK*12] JACOBSON, ALEC, BARAN, ILYA, KAVAN, LADISLAV, et al. "Fast automatic skinning transformations". *ACM Trans. Graph.* 31.4 (2012), 77:1–77:10 3.
- [Kar77] KARCHER, HERRMANN. "Riemannian center of mass and mollifier smoothing". *Communications on Pure and Applied Mathematics* 30.5 (1977), 509–541 3.
- [KG08] KIRCHER, SCOTT and GARLAND, MICHAEL. "Free-form Motion Processing". *ACM Trans. Graph.* 27.2 (2008), 12:1–12:13 3.
- [KMP07] KILIAN, MARTIN, MITRA, NILOY J., and POTTMANN, HELMUT. "Geometric Modeling in Shape Space". *ACM Trans. Graph.* 26.64 (2007), 1–8 3, 9.
- [LLW*19] LIU, YUXIN, LI, GUIQING, WANG, YUPAN, et al. "Discrete shell deformation driven by adaptive sparse localized components". *Computers & Graphics* 78 (2019), 76–86 3, 10, 11.

- [LSC*04] LIPMAN, YARON, SORKINE, OLGA, COHEN-OR, DANIEL, et al. “Differential coordinates for interactive mesh editing”. *Proceedings Shape Modeling International*. 2004, 181–190 2.
- [LSLC05] LIPMAN, YARON, SORKINE, OLGA, LEVIN, DAVID, and COHEN-OR, DANIEL. “Linear Rotation-invariant Coordinates for Meshes”. *ACM Trans. Graph.* 24 (2005), 479–487 2.
- [MA07] MEYER, MARK and ANDERSON, JOHN. “Key Point Subspace Acceleration and soft caching”. *ACM Trans. Graph.* 26.3 (2007), 74 3.
- [MG03] MOHR, ALEX and GLEICHER, MICHAEL. *Deformation Sensitive Decimation*. Tech. rep. University of Wisconsin, 2003 9.
- [MK12] MÖBIUS, JAN and KOBBELT, LEIF. “OpenFlipper: An Open Source Geometry Processing and Rendering Framework”. *Curves and Surfaces*. Vol. 6920. Lecture Notes in Computer Science. Springer, 2012, 488–500 9.
- [MOS20] MOSEK APS. *The MOSEK Optimizer API for C manual. Version 9.2*. 2020. URL: <https://docs.mosek.com/9.2/capi/index.html> 9.
- [NVW*13] NEUMANN, THOMAS, VARANASI, KIRAN, WENGER, STEPHAN, et al. “Sparse localized deformation components”. *ACM Trans. Graph.* 32.6 (2013), 1–10 2, 3, 11.
- [RW09] RUMPF, MARTIN and WIRTH, BENEDIKT. “A nonlinear elastic shape averaging approach”. *SIAM J. Imaging Sci.* 2.3 (2009), 800–833 5.
- [RW15] RUMPF, MARTIN and WIRTH, BENEDIKT. “Variational time discretization of geodesic calculus”. *IMA J. Numer. Anal.* 35.3 (2015), 1011–1046 12.
- [SA07] SORKINE, OLGA and ALEXA, MARC. “As-Rigid-As-Possible Surface Modeling”. *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. 2007, 109–116 3.
- [SBBG11] SOLOMON, JUSTIN, BEN-CHEN, MIRELA, BUTSCHER, ADRIAN, and GUIBAS, LEONIDAS. “Discovery of intrinsic primitives on triangle meshes”. *Computer Graphics Forum* 30.2 (2011), 365–374 13.
- [SCL*04] SORKINE, OLGA, COHEN-OR, DANIEL, LIPMAN, YARON, et al. “Laplacian surface editing”. *Symposium on Geometry Processing*. 2004, 175–184 2.
- [SHHR20] SASSEN, JOSUA, HEEREN, BEHREND, HILDEBRANDT, KLAUS, and RUMPF, MARTIN. “Geometric optimization using nonlinear rotation-invariant coordinates”. *Computer Aided Geometric Design* 77 (2020) 3, 4, 9.
- [SM00] SHI, JIANBO and MALIK, JITENDRA. “Normalized cuts and image segmentation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), 888–905 10.
- [SP04] SUMNER, ROBERT W. and POPOVIĆ, JOVAN. “Deformation transfer for triangle meshes”. *ACM Trans. Graph.* 23.3 (2004), 399–405 2, 12.
- [Tib96] TIBSHIRANI, ROBERT. “Regression Shrinkage and Selection via the Lasso”. *Journal of the Royal Statistical Society* 58.1 (1996), 267–288 6.
- [WDAH10] WINKLER, TIM, DRIESEBERG, JENS, ALEXA, MARC, and HORMANN, KAI. “Multi-Scale Geometry Interpolation”. *Comp. Graph. Forum* 29.2 (2010), 309–318 3.
- [WLT12] WANG, YUANZHEN, LIU, BEIBEI, and TONG, YIYING. “Linear Surface Reconstruction from Discrete Fundamental Forms on Triangle Meshes”. *Comput. Graph. Forum* 31.8 (2012), 2277–2287 3, 4.
- [WLZH17] WANG, YUPAN, LI, GUIQING, ZENG, ZHICHAO, and HE, HUAYUN. “Articulated-Motion-Aware Sparse Localized Decomposition”. *Comput. Graph. Forum* 36.8 (2017), 247–259 3, 11.
- [WZ88] WEISER, ALAN and ZARANTONELLO, SERGIO E. “A note on piecewise linear and multilinear table interpolation in many dimensions”. *Mathematics of Computation* 50.181 (1988), 189–196 8.
- [XZWB05] XU, DONG, ZHANG, HONGXIN, WANG, QING, and BAO, HUIJUN. “Poisson shape interpolation”. *Symp. on Solid and Phys. Mod.* 2005, 267–274 2.
- [YLSL11] YEH, I-CHENG, LIN, CHAO-HUNG, SORKINE, OLGA, and LEE, TONG-YEE. “Template-Based 3D Model Fitting Using Dual-Domain Relaxation”. *IEEE Trans. Vis. Comput. Graph.* 17.8 (2011), 1178–1190 12.
- [You10] YOUNES, LAURENT. *Shapes and Diffeomorphisms*. Springer, 2010 3.
- [ZSCS04] ZHANG, LI, SNAVELY, NOAH, CURLESS, BRIAN, and SEITZ, STEVEN M. “Spacetime Faces: High Resolution Capture for Modeling and Animation”. 23.3 (2004) 8, 12.