Optimized Subspaces for Deformation-Based Modeling and Shape Interpolation

Philipp von Radziewsky[a], Elmar Eisemann[a], Hans-Peter Seidel[b], Klaus Hildebrandt[a]

[a] *Delft University of Technology, Netherlands*
[b] *Max-Planck-Institut für Informatik, Germany*

Abstract

We propose a novel construction of subspaces for real-time deformation-based modeling and shape interpolation. The scheme constructs a subspace that optimally approximates the manifold of deformations relevant for a specific modeling or interpolation problem. The idea is to automatically sample the deformation manifold and construct the subspace that best-approximates these snapshots. This is realized by writing the shape modeling and interpolation problems as parametrized optimization problems with few parameters. The snapshots are generated by sampling the parameter domain and computing the corresponding minimizers. Finally, the optimized subspaces are constructed using a mass-dependent principle component analysis. The optimality provided by this scheme contrasts it from alternative approaches, which aim at constructing spaces containing low-frequency deformations. The benefit of this construction is that compared to alternative approaches a similar approximation quality is achieved with subspaces of significantly smaller dimension. This is crucial because the run-times and memory requirements of the real-time shape modeling and interpolation schemes mainly depend on the dimensions of the subspaces.

Keywords: Shape Deformation, Shape Interpolation, Shape Modeling

1. Introduction

Creating digital geometric content is an important task for applications in various areas including digital manufacturing, computer animation, and virtual reality. Acquisition technologies, like 3D-scanning, allow for creating accurate digital copies of detailed real-world objects. Therefore, methods for modeling a single shape and for synthesizing new ones from a collection of shapes are essential for customizing digital content to the demands of users and applications. Here we consider two such methods: *deformation-based modeling* and *shape interpolation*. Deformation-based modeling tools provide a user with simple and intuitive interfaces for modifying a digital shape. For example, a user can translate and rotate parts of the object, so-called handles, and the rest of the shape follows automatically. Physical models of deformable objects are used to produce deformations that match the users intuition. For shape interpolation, we consider a set of example shapes, *e.g.*, different poses of one character. Shape interpolation allows for creating new "in-between shapes" and is a crucial module of schemes for tasks like morphing, deformation transfer, example-based shape editing, example-based materials for controlling simulations, and shape exaggeration.

A fundamental problem for both methods, modeling and interpolation, is that on the one hand, processing tools need to solve high-dimensional non-linear optimization problems to compute the deformed shapes, and, on the other hand, users expect fast or even interactive responses. Therefore, it is essential to design efficient approximation algorithms for these problems. Subspace methods proved to be very effective. The principle is to construct a low-dimensional approximation of the complex problem in a preprocess (offline phase) and to solve only the low-dimensional system in the interactive (online) phase. Different schemes for constructing subspaces for deformation-based modeling have been introduced based on space deformations, radial basis functions, bi-harmonic problems, low-frequency Laplace–Beltrami eigenfunctions or vibration modes. The common goal of these methods is to construct subspaces containing low-frequency deformations. An alternative approach is to learn the subspaces from observations. Methods following this idea, such as the method of snapshots, are prominent for the reduction of physical simulations.

In this work, we introduce constructions of subspaces that are optimized for deformation-based shape modeling and shape interpolation tasks. The constructions involve the following technical contributions. We formulate general frameworks for shape modeling and interpolation as parameterized optimization problems with low-dimensional compact parameter domains. Then, observations of the shape modeling or interpolation tasks can be obtained by samplings the solution space of the optimization problem. This in turn can be done by sampling the parameter domain of the optimization problem and computing the corresponding deformations. For shape interpolation, the parameters are the interpolation weights. Since the interpolation weights are positive and sum to one, the set of weights forms a simplex, whose dimension is one less than the number of example shapes to be interpolated. To generate the snapshots for the interpolation problem, we sample the simplex and compute the corresponding interpolating shapes. For deformation-based modeling, we consider deformation handles that can be translated and rotated in space. To obtain a compact parameter domain, we introduce a maximum translation for the handles, *e.g.*, the length of the objects bounding box diagonal, and parametrize the rotations using Euler angles. Then the parameter domain is a rectangular box (cuboid) of dimension $6(h - 1)$, where h is the number of handles. A sample point in this box specifies locations in $\mathbb{R}^3$ for all handles. To generate the snapshots for deformation-based shape modeling, we sample the box and for every sample point, we compute the deformation corresponding to the handle locations. Once the snapshots for a specific modeling or interpolation task have been generated, the subspaces that optimally approximate the snapshots are constructed. A mass-orthonormal basis of a such a subspace can be obtained by a mass-weighted principle component analysis (PCA) of the snapshots.

Our approach contrast from alternative subspace construction for deform-based modeling. Whereas our approach yields low-dimensional subspaces that are optimized for containing good approximations of the deformations that are relevant for a specific setting (*e.g.*, a set of handles defined on shape or a set of example shapes to be interpolated), alternative approaches aim at spaces containing low-frequency deformations. We analyze the quality of the resulting subspaces in experiments and comparisons to alternative approaches. In particular, we demonstrate that the proposed construction results in more efficient subspaces that achieve a comparable approximation quality with significantly smaller dimension. This is crucial because the computational cost for solving the reduced problems mainly depends on the dimension of the subspace.

2. Related Work

Deformation-based shape modeling and interpolation have received much attention in recent years. We can distinguish between linear and non-linear frameworks. For an in-depth discussion of linear schemes, which are not in the focus of this paper, we refer to the survey [1]. Examples of non-linear frameworks are co-rotated iterative Laplacian editing schemes [2, 3], PriMo [4] and As-Rigid-As-Possible [5, 6].

Dimensional reduction proved to be a powerful concept for designing computational schemes for (non-linear) deformation-based shape modeling that run at interactive or real-time rates. Such schemes construct a subspace of the search space of all possible deformations and restrict the optimization problems to these subspaces. Ideally, such a subspace is low-dimensional and contains good approximations of the desired deformations. Different approaches that aim at constructing subspaces of "low-frequency" deformations have been proposed. To construct low-frequency deformations in the signal theoretic sense, eigenfunctions of the Laplace–Beltrami operator have been used [7, 8, 9, 10]. An alternative is to use the low-frequency vibration modes of the deformable object to construct subspaces. To allow for larger deformations, these subspaces are augmented with modal derivatives [11, 12] or vectors obtained by transformations of the modes [13].

Subspaces can also be constructed by subsampling the surface and propagating displacements of the sample points to displacements of the vertices via a linear map [3, 14, 15, 16]. To construct this map radial basis functions with local support centered at the sample points are used. As a result the displacements of a single sample point only affect a local neighborhood of the surface. The resulting spaces contain only low-frequency deformations since localized deformations can only be represented up to the sampling density. Alternative to radial basis functions, bi-Laplacian systems on the volume enclosed by the surface are used to create the map from the samples to the mesh vertices. To ensure positive correlation of displacements of the samples and the vertices, the bounded biharmonic maps [17] impose a non-negativity constraint on the solutions of the bi-Laplace systems. In [18] the non-negativity constraints is dropped and the sampling-based approach is augmented with affine deformations of the handles. Though this approach considers deformations of the handles, the idea is orthogonal to the scheme we propose. While we consider the non-linear deformations of the shape induced by rigid deformations of handles to obtain snapshots, these consider affine deformations of the handle regions only and are directly added to the subspace basis. For the deformation of articulated characters, where the mesh describing the skin is attached to a skeleton, a dimensional reduction based on rotational regression was introduced in [19]. In [20], a scheme for context-aware skeletal shape deformation that uses example poses (provided by the user) was introduced.

The principle of space-deformation schemes is to deform the ambient space around the shape and thus implicitly the shape, as well. The deformations of the space are typically controlled by a cage [21, 22, 23, 24, 25]. The map from displacements of the cage to displacements of the mesh is linear. Hence, the cage implicitly defines a subspace. However, it is not efficient to explicitly construct the basis of the resulting subspace because the spaces are too high-dimensional. In [26], the set of shape deformations provided by the user is used to reduce the number of control points that influence the vertices, which helps localizing the influence of the control vertices.

The method we propose is the first subspace construction from snapshots for deformation-based modeling and shape interpolation. Still, constructing subspaces from snapshots an established approach and methods following this principle have been used, for example, for the reduction of simulations of deformable objects [11], character skinning [27] and for creating and animating human shape models [28]. Principle component analysis has also been applied for the construction of blend shape models, which are used for facial animation [29].

To get interactive or real-time rates for larger meshes, in addition to the dimensional reduction, a scheme for efficient evaluation of the reduced deformation energy and its gradient are needed. Different schemes have been proposed, some use the specific structure of a particular deformation energy, others are more general. For iterative Laplacian editing, the reduced Laplace matrix can be precomputed and prefactored. Then the linear systems, which must be solved in every iteration, can directly be solved in reduced coordinates. For fast evaluation of the As-Rigid-As-Possible energy and its gradient, the precomputation of the reduced Laplace matrix is combined with a clustering scheme for fast estimation the rotational parts of the deformation gradients [30]. Finite element discretizations of linear materials (St. Venant–Kirchhoff materials) yield a deformation energy that is a fourth-order polynomial. The coefficients of these polynomials can be precomputed [11], which yields exact evaluation of the reduced energy and forces at costs depending only on the subspace dimension. The optimized cubature [31, 13, 32] can be applied for approximating the reduced energy and forces of more general materials. An alternative is the mesh coarsening approach proposed in [12].

Linearized deformation models have also been used for shape interpolation. An example is Poisson shape interpolation [33]. Due to the linearization this scheme develops artifacts when interpolating between large deformations. Examples and a discussion of this effect can be found in [34]. Interpolation schemes based on non-linear deformation models have been proposed in [35] for implicitly defined shapes in 2D and 3D images, in [36] for planar shapes, and in [34, 37, 38, 39, 40, 41, 42] for meshes. To compute one interpolating shape, a non-linear optimization problem has to be solved. Even with multi-grid solvers the computation of an interpolated shape for meshes with 20-50k vertices requires several seconds on a regular desktop computer. In recent work efficient approximation algorithms have been proposed. In [37], a combination of mesh coarsening and deformation-transfer has been used to speed-up the computation of interpolated shapes. In [41] an efficient block-coordinate descent scheme has been introduced to speed-up the as-isometric-as-possible shape interpolation scheme. These schemes can achieve interactive rates (one or two interpolated shape per second for meshes with 20-50k vertices on a regular desktop computer). In [38], the mesh is embedded in a coarser mesh and example shapes of the coarse mesh are constructed. To speed-up the shape interpolation, only the coarse meshes are interpolated and the deformations of the coarse mesh are propagated to the finer mesh. A subspace approach that yields real-time computation rates even for larger meshes has been proposed in [42]. A major difference to the proposed scheme is that in [42] the subspace are constructed solely using forces and Hessians of the example shapes and the computation of samples is avoided. This lowers the computational effort for the subspace construction, but does not provide the optimality the proposed scheme offers.

Shape spaces encode individual shapes as points. The spaces of meshes can be equipped with Riemannian metrics [43, 44]. The shortest curve connecting two shapes is a geodesic. Traversing such a geodesic, provides a way of interpolating shapes. In [45], shape interpolation is used for smoothing curves in shape space. Based on shape interpolation, they introduce a curve shortening flow that evolves curves towards geodesics in shape space.

3. Parametrized Optimization Problems in Shape Deformation

Computing optimal deformations of shapes is important for many applications in geometry processing. Here, we consider optimization problems that are controlled by few parameters. This means, we are looking at problems of the form

$$\underset{x\in\mathbb{R}^n}{\arg\min}\, F(x,\omega), \tag{1}$$

where $x \in \mathbb{R}^n$ is the search space, $\omega \in \Omega \subset \mathbb{R}^m$ is a vector listing the parameters, and we assume that n is large and m is small. We specifically address two applications, namely deformation-based shape modeling and interpolation. Before discussing these two problems, we first briefly discuss deformation energies.

3.1. Deformation energies

We are considering a flexible discrete shape that has n degrees of freedom. For example, a triangle or tetrahedral mesh with $n/3$ vertices. In this case, the coordinates of the vertices are the degrees of freedom. One ingredient to variational shape deformation problems in geometry processing is a deformation energy. This is a functional $E : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ on the space of configurations of the shape which vanishes for the neutral configuration $\bar{x}$. In terms of continuum mechanics, the shape represents a deformable object consisting of a hyperelastic material. A material is elastic if the forces acting on the object depend only on the actual configuration x and is independent of the deformation path and speed. This means that 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 deformation energy (or potential) $E(x)$ is the function (determined up to a constant) whose negative gradient equals the force. It measures the energy stored in the object when it is deformed to the configuration x.

There are various deformation energies that have been used in geometry processing. Examples are PriMo [4], As-Rigid-As-Possible [5, 6], and Discrete Shells [46] for surface meshes and finite elements discretizations of elastic solids with different hyperelastic materials for tetrahedral meshes [47]. The approach we are presenting can be applied to various deformation energies and discrete shape representations including triangle and tetrahedral meshes as well as spline type representations or point clouds. For our experiments, we used tetrahedral meshes and a finite elements discretization of St. Venant–Kirchhoff materials for elastic solids.

3.2. Deformation-based shape modeling

The general idea of deformation-based shape modeling is to consider the shape as a deformable object and to allow the user to define constraints on the mesh to model it. Different user-interfaces for specifying these have been introduced. One effective approach is the handle-based interface: the user defines regions of the object to be *handles*. Then in the modeling phase, rigid transformations of the handles are specified, which define prescribed positions for the handles' constituent vertices. Explicitly, we include this interface to the objective by adding a quadratic penalization term:

$$\mu \textstyle\sum_{H_i} \sum_{v_j \in H_i} \bar{m}_j \left\| v_j - R_i \bar{v}_j + t_i \right\|^2, \tag{2}$$

where H_i denotes the handles, $v_j, \bar{v}_j \in \mathbb{R}^3$ the vertices in the deformed and neutral configuration, and R_i and t_i the rotation and translation of the ith handle, *i.e.*, the user parameters. The contribution of each vertex is weighted with the mass $\bar{m}_i$ of $\bar{v}_i$, (a fourth of the combined volume of all tetrahedra adjacent

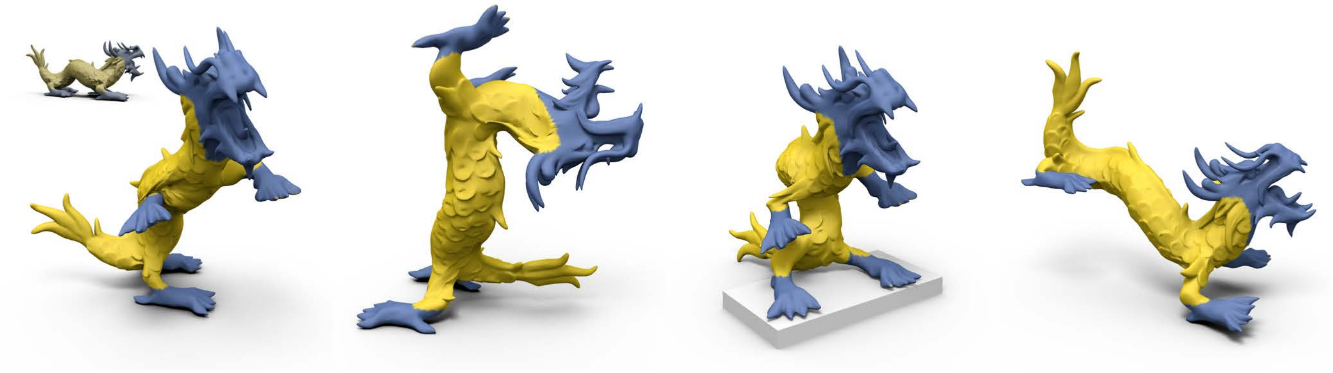


Figure 1: Large deformations of the dragon model with 2 handles (hind legs and forelegs plus head) and 6 non-rigid degrees of freedom computed in a 40 dimensional subspace.

to $\bar{v}_i$) to make the energy robust against remeshing and mesh coarsening. The parameter μ is a global multiplier weighting the penalization term against the elastic potential.

This constraint term can be written as a parameter-dependent function on $\mathbb{R}^n$. A rigid transformation can be specified by six parameters, three for the translations and three for the rotations (*e.g.*, using Euler angles). Let ω be a vector listing the six parameters and x the vector listing the vertex coordinates of all vertices, then the potential (2) is a function $E^H(x, \omega)$ depending on x and ω.

The deformations we want to compute are the minimizers of the sum of the two energies. This means, for modeling the shape, we need to solve optimization problems of the form (1), where the objective functional is

$$F(x, \omega) = E(x) + E^H(x, \omega). \tag{3}$$

Since the deformation energy E is invariant under rigid motion, applying the same rigid transformation to all handles means that the minimizers are transformed with the same rigid transformation. Therefore, $m + 1$ handles effectively lead to $6\,m$ parameters.

3.3. Shape interpolation

For shape interpolation, we consider a set of $m + 1$ meshes with the same connectivity but different vertex positions $\bar{x}_k \in \mathbb{R}^n$. This setting naturally arises when the shapes are $m+1$ poses of a character, $m + 1$ deformations of an object, or the shapes have the same texture. For a general set of shapes, the requirement that all shapes have the same connectivity means that there is a consistent set of correspondences between the shapes given and the interpolation will respect the correspondences. For an overview of methods for establishing correspondences between meshes, we refer to the recent survey [48].

Our goal is to compute shapes that interpolate between the $m + 1$ example shapes. Analogous to barycentric interpolation, a positive weight ω_i is assigned to every shape such that the weights sum to one. For every vector of $m + 1$ weights, an interpolated shape can be computed. If one of the weights takes the value one, the interpolated shape equals the corresponding example shape. The *mean shape* is the interpolated shape we obtain if all weights are $1/(m + 1)$. For any choice of weights, the corresponding interpolated shape is computed by solving an optimization problem. To model this problem, we consider $m + 1$ deformation energies E_k. The kth energy has the shape $\bar{x}_k$ as its neutral configuration. Since all $m + 1$ example shape have the same connectivity, one vector $x \in \mathbb{R}^n$ (listing coordinates of all vertices) specifies deformed configurations for all $m + 1$ example shapes. Then, the $E_k(x)$ measure the energy stored in the material if it is deformed from its neutral configuration $\bar{x}_k$ to x. The interpolating shape corresponding to a weight vector $\omega = (\omega_0, \omega_1, ...\omega_m)$ is the configuration x that minimizes the weighted sum of the deformation energies E_k. This means to compute an interpolating shape, an optimization problem of the form (1) must be solved, where the objective functional is

$$F(x, \omega) = \sum_k \omega_k E_k(x).$$

For a continuous formulation and properties of the shape interpolation, we refer to [35, 42]. Since the weights sum to one, the optimization problem has m independent parameters.

4. Sampling the Space of Solutions

Before we consider the construction of optimized subspaces in the next section, we look at strategies for sampling the solution space of the optimization problems. A reasonable approach is to sample the parameter domain following a particular distribution and compute minimizers of the samples. In this section, we first discuss the sampling of the parameter domains for modeling and interpolation, then the efficient computation of minimizers for lifting the sampling from the parameter domain to the solution space, and, finally, rigid registration of the samples for shape interpolation.

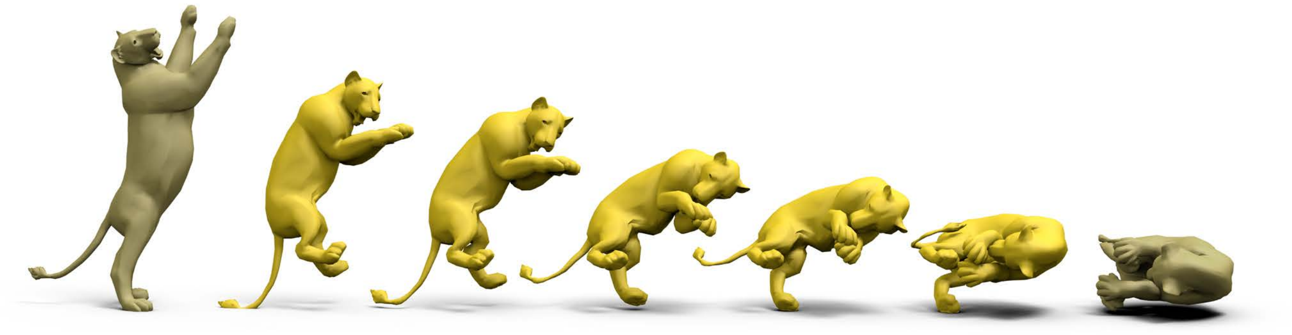


Figure 2: Shapes that interpolate between two poses (left and right) of a lion computed in a subspace of only 6 dimensions.

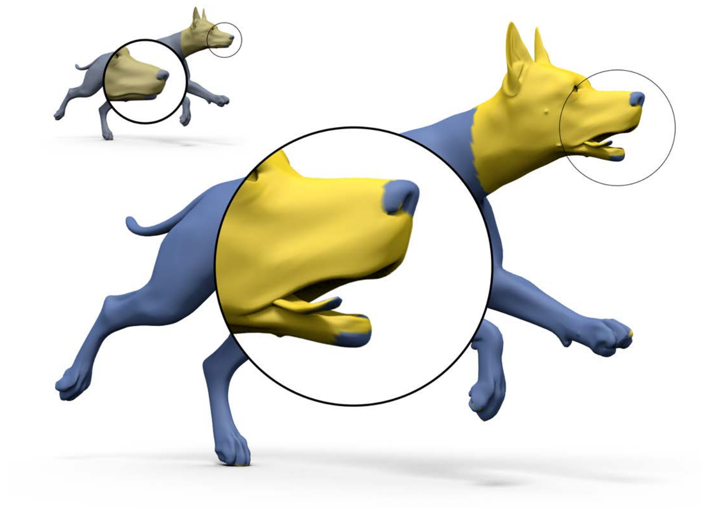


Figure 3: Example showing local deformation computed using our subspace approach.

4.1. Parameter sampling

The parameter domain of the shape interpolation problem is compact by construction due to the constraints that the weights are positive and sum to one. Explicitly, the domain forms an m-dimensional simplex. It is a natural idea to draw samples uniformly distributed on this domain.

The parameter space for deformation-based editing consists of rigid transformations of the handles. Explicitly, we parametrize the rotational part by extrinsic Euler rotations about a coordinate system fixed at the handle's center, which allows to describe any orientation in space and the translation part as some offset to the neutral configuration. To obtain a compact parameter domain, we specify a maximum for each of the Euler angles and translations in the coordinate directions. The resulting set is rectangular box in the parameter domain that we sample following a uniform distribution. It is noteworthy that this approach gives a significantly different training set compared to a sampling with a normal distribution around the neutral configuration. While neither choice has a natural justification, a normal distribution would favor small deviations resulting in an undesired emphasis towards small displacements. Furthermore, we want to remark that instead of Euler angles, other parametrizations of the rotation group could be used. This would potentially alter the resulting sampling because we are not compensating for the distortion introduced by the parameterization.

Based on these distributions, we tested different ways to generate the sample set, which are sampling on a uniform grid, random sampling, and *Sobol sequences* [49], a quasi-Monte Carlo (QMC) sampling method. However, we did not notice a significant difference in the performance of the whole subspace construction when altering the construction of the sample set in our experiments. Therefore, we used the simplest strategy, the random sampling, for producing the results shown in the paper.

4.2. Numerical solver for generating the samples

The first step for constructing an optimized subspace is to compute a number of snapshots of the unreduced optimization problem. For this we use Newton's method. Since the deformation energies are not convex, the Hessian is, in general, not positive definite. This can be rectified by applying some scheme to force positive-definiteness by either explicitly or implicitly modifying the eigenvalues of the Hessian [50]. Explicit modification of the Hessian is generally costly as it requires eigendecomposition of the large matrix. A method for Hessian modification that is specifically tailored for our problem are the so-called *invertible finite elements*, which were introduced in [51]. The technique uses the fact that the Hessian of the deformation energy is assembled from contributions of the tetrahedral elements. Instead of decomposing the whole Hessian, it modifies the elemental contributions of the individual tetrahedra by computing a modified singular value decomposition (SVD) $U\Sigma V$ of the deformation gradient. The standard SVD is altered in the sense that U, V are chosen such that $\det(U) = \det(V) = 1$. Hence, negative singular values are allowed and a negative value indicates that the tetrahedron is in an inverted state. For the computation of an element's contribution, these values are clamped at some threshold and the contribution is computed for this clamped state. This provides a tool

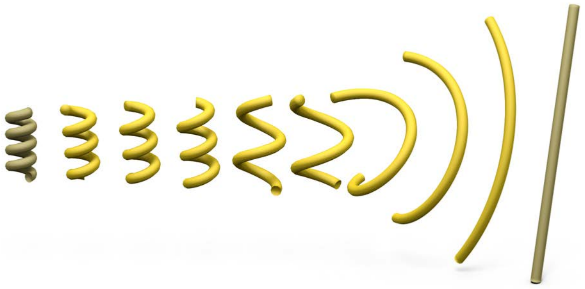


Figure 4: Shapes that interpolate between a cylinder and a helix computed in a 20 dimensional subspace.

to modify the Hessian. The choice of the clamping threshold is heuristic and depends on the configuration of the mesh. Setting all singular values to one means to always use the Hessian at the reference shape, which is a minimizer of the elastic energy and, hence, guarantees positive definiteness of the contribution to the Hessian. To generate the samples for our experiments, the threshold was chosen between 0.5 and 0.8.

The number of steps the Newton solver requires to converge depends on the choice of the initial estimate. Since many solutions are computed to generate the samples, their locality can be exploited. To achieve a good initial guess, we choose from the computed samples the closest point (in parameter space) to the next sample.

4.3. Rigid registration of the samples

For shape interpolation, an additional step is required to obtain a meaningful training set. The deformation energies are invariant to rigid transformation of both the neutral and the deformed configuration. As a consequence, the interpolated shapes are only defined up to rigid motion. However, the subspace construction is not invariant to rigid motion of the samples. Therefore, we apply a rigid registration to the samples and the example shapes before the subspace construction. For each of the shapes, we determine the rigid motion that minimizes the squared distance to one selected reference shape (*e.g.*, the mean shape). Determining the optimal rotation amounts to computing an SVD, see [52] for details. For measuring the distance between two shapes, we use the L^2-norm (or mass weighted norm), which we introduce in equation (5) of the following section.

In constrast to shape inerpolation, the objective function used for deformation-based shape modeling is not invariant under rigid motion as the handle positions are fixed. Therefore no registration is needed. The fact that applying the same rigid motion to all the handles only rigidly transforms the minimizer is considered during the sampling of the parameter domain by fixing the location and orientation of one handle.

5. Subspace Construction

In this section, we discuss the dimensional reduction of the optimization problem (1). The motivation is that the optimization problem is controlled by only a few parameters (rigid motions of handles for deformation and weights for the interpolation), which means that in the solution space the n degrees of freedom of the shape (*e.g.*, the vertex positions) are correlated. In other words, the solution space of the optimization problem is a low-dimensional object in $\mathbb{R}^n$. The idea is to construct a low-dimensional (affine) subspace of $\mathbb{R}^n$ that approximates the solution space and restrict the optimization to the subspace. Since the solution space is non-linear, the dimension of the affine space will be higher than the set of parameters of the optimization problem. The fidelity of the reduction of the problem depends on how well the subspace can approximate the solution space.

In this section, we introduce a technique for constructing subspaces that are optimized for providing good approximations. Since checking how well a subspace approximates the whole solution space is infeasible, we first sample the solution space (as described in Section 4) and then optimize for the approximation of the sampling.

An affine subspace S of $\mathbb{R}^n$ can be represented as a linear subspace of $\mathbb{R}^n$ that is attached to a reference point $\tau \in \mathbb{R}^n$. Let $U \in \mathbb{R}^{n \times d}$ be a matrix whose columns are the basis vectors of the linear space. Then the affine space is

$$S = \{Uq + \tau \mid q \in \mathbb{R}^d\}. \tag{4}$$

For any point $x = Uq+\tau$ in S, we call q the reduced coordinates (w.r.t. U and τ).

For the construction of the subspace for deformation-based modeling, we want the neutral pose $\bar{x}$ to be an element of the subspace. Therefore, we set τ to be the neutral pose and construct only the linear subspace that we attach to this point. For the interpolation problem, we proceed similarly and set τ to be the mean shape of the example shapes.

5.1. Optimal approximation of the samples

Let $\{\tilde{y}_j\}_{i\in\{1,2,\ldots,n_s\}} \subset \mathbb{R}^n$ be the samples of the solution space of the optimization problem (1). For the computation, it is convenient to describe them as displacements y_j of the shape τ, *i.e.*, $y_j = \tilde{y}_i - \tau$. To measure the distance of a sample from a subspace of $\mathbb{R}^n$, we use the scalar product on $\mathbb{R}^n$ induced by the *mass matrix* M of the shape τ (hence the neutral shape for deformation and the mean shape for interpolation), which is a numerical discretization of the L^2-scalar product on the displacement fields on the shape. For the explicit formulae for the entries of the mass matrix for tetrahedral meshes, we refer to [53, 54]. Using the mass matrix allows to measure the distance of a sample to a subspace in a manner that is robust against remeshing, coarsening or refining of the meshes. We denote the scalar product and norm by

$$\langle x, y\rangle_M = x^T My \qquad \text{and} \qquad \|x\|_M = \sqrt{\langle x, x\rangle_M}. \tag{5}$$

and create the optimized subspace by PCA. Specifically, we construct the basis matrix U that minimizes the approximation of the samples as

$$\begin{aligned} &\underset{\{u_i\}_{i\in\{1,\dots,d\}}}{\arg\min} \sum_{j=1}^{n_s} \left\| y_j - \sum_{i=1}^{d} \left\langle y_j, u_i \right\rangle_M u_i \right\|_M^2 \qquad (6)\\ &\text{subject to } \left\langle u_i, u_j \right\rangle_M = \delta_{ij}. \end{aligned}$$

with (mass)-orthonormal columns $\{u_i\}_{i\in\{1,2,..,d\}}$ of U.

5.2. *Method of snapshots*

There are different ways for solving (6). In our case, where the number n_s of samples is small compared to the dimension n of the search space, the following technique, which is often called the *method of snapshots*, is efficient. For a proof that the resulting basis solves the problem (6), we refer to [55].

The first step is to construct the symmetric positive-definite matrix $Y^T MY$, where the vectors y_i form the columns of Y. This is a dense matrix of size n_s. Then, the first d eigenvectors ϕ_i (in order of descending eigenvalue) solving the problem

$$Y^T MY\phi_i = \lambda_i\phi_i$$

are computed. The PCA basis u_i is obtained by multiplication of the eigenvectors with the matrix Y and normalization (w.r.t. $\|\cdot\|_M$) of the result. This can be done by setting

$$u_i = \frac{1}{\sqrt{\lambda_i}} Y\phi_i.$$

The approximation error achieved by the PCA basis with d elements can be expressed in terms of the eigenvalues λ_i

$$\textstyle\sum_{i=d+1}^{n_s} \lambda_i.$$

Instead of computing the basis with a given number of elements, one can compute a basis that satisfies a prescribed approximation goal. Such a goal should be chosen such that

$$\frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{n_s} \lambda_i}$$

is close to one. However, this only guarantees the approximation of the set of samples. The approximation of the solution manifold additionally depends on how well the sampling approximates the solution space.

6. The Reduced Optimization Problem

After the subspace has been constructed, we restrict the optimization problem to this space. This means, we consider the reduced objective function

$$\hat{F}(q, \omega) = F(Uq + \tau, \omega)$$

and instead of (1) and solve the reduced problem

$$\underset{q\in\mathbb{R}^d}{\arg\min}\, \hat{F}(q, \omega).$$

Our goal is to construct the reduced optimization problem such that the cost for solving it depends only on the subspace dimension and the number of parameters of the optimization problem. In particular, it should be independent of the resolution of the shapes to be edited or interpolated. To achieve this goal, we need a strategy for evaluating the reduced objective functional (and its derivatives) at costs independent of the resolution of the shapes.

6.1. *Evaluation of the reduced functional and gradient*

St. Venant–Kirchhoff materials have the property that the corresponding deformation energy is a quartic polynomial in the coordinates of a particular displacement. By substituting x in (1) for $Uq + \tau$ and grouping the terms by the entries of q one obtains a quartic polynomial in $\mathbb{R}^d$, for which the coefficients can be precomputed. This yields a scheme for exact evaluation of the reduced deformation energy at a computational costs depending only on d. For details, we refer to [11].

For deformation-based modeling, the objective functional additionally comprises the energy E^H. This energy depends on the vertex positions of the deformed mesh, *i.e.*, the unknowns of the optimization process, and the rigid transformations of the handles, *i.e.*, the user constraints, which change regularly (as opposed to the rest pose vertices $\bar{v}_i$ in (2)). To turn this energy into a quadratic form, which we can easily reduce, we extend the domain and allow for arbitrary affine, instead of rigid, transformations of the handles. It is easy to see that the extended energy depends quadratically on both, the vertex positions and the 3×4 transformation matrices A_i specifying the affine transformations of the handles H_i. By substituting x for $Uq + \tau$ in E^H and grouping for the entries of q and the A_i's one obtains a quadratic polynomial in $d + 12 \cdot h$ unknowns, where h is the number of handles. Thus, one obtains a scheme for which the computational effort depends on the subspace dimension and the number of handles, but is independent of both the number of vertices of the mesh and the number of vertices in each handle. We want to emphasize that this yields an exact evaluation of the energy E^H. Though the reduced energy could be queried with arbitrary affine transformations, we consider only rigid transformations of the handles both in the offline phase (random rigid transformations of the handles are generated for sampling) and in the online phase (user specifies rigid transformations using the mouse).

Alternatives to the exact evaluation are the optimized cubature [31, 13], mesh coarsening [12], and rotation clustering [30] (the latter is only applicable for the as-rigid-as-possible deformation energy). All these schemes approximate the reduced functional and gradient. We want to mention that compared to the approximation schemes, the exact evaluation comes at additional computational costs in the offline and the online phase. We still used the exact evaluation in our experiments because this focuses the evaluation on the subspace quality (since the quality of the force approximation does not influence the results).

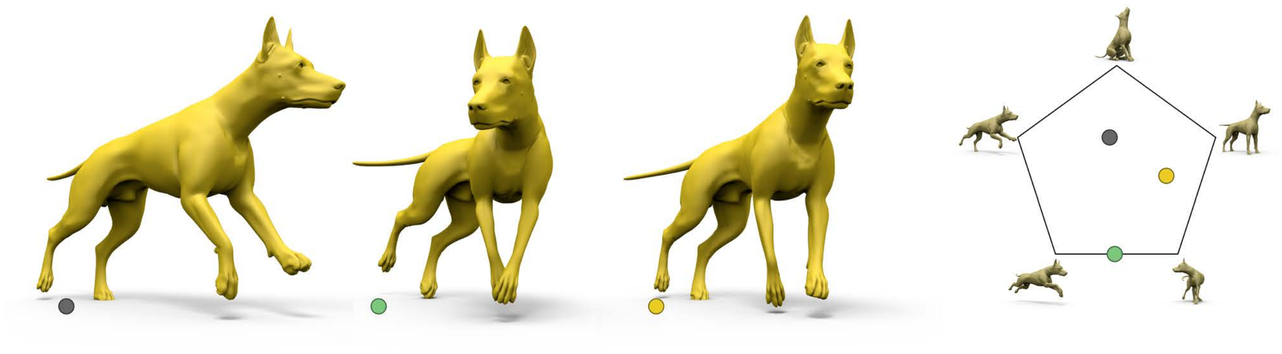


Figure 5: Shape interpolation of multiple reference shapes of the dog in a 10 dimensional subspace. The interpolated shape ● was recorded with a weight of 0 for the sitting reference pose. Hence, the bending of the hind legs is not present.

Model			Preprocess					Online	
Name	V	T	Sample Count	Sample Generation	Iterations	Subspace Size	Coefficients	BFGS	Iterations
Shape Interpolation									
Lion	13k	65k	4	26 s	28	6	1.18 s	0.057 ms	7
Helix	1k	2.5k	30	27 s	13	20	5.88 s	0.6 ms	3
Dog	32k	122k	20	477 s	13	10	25 s	0.6 ms	3
Centaur	16k	54k	40	561 s	17	24	726 s	1.16 ms	6
Deformation-based Shape Editing									
Dinosaur	27k	107k	50	2099s	12	40	2124 s	1 ms	10
Dragon	32k	121k	300	8121s	19	40	5162 s	1.1 ms	9
Elephant	24k	82k	200	3608s	12	45	5663 s	2.63 ms	8
Dog	32k	122k	200	4862s	25	17	380 s	0.25 ms	8

Table 1: Performance statistics measured on an quad-core Intel Core i7-4820K running at 3.7GHz. From left to right: the number of vertices, number of tetrahedra, recorded samples, total time to generate the samples, average number of Newton iterations per sample point, subspace dimension, the construction time for the basis including the polynomial coefficient computation, average time for one BFGS iteration in the reduced model, and average number of iterations required. The online results have been recorded in a live-modeling session.

6.2. Solving the reduced optimization problem

After the reduction, we have to solve a low-dimensional nonlinear optimization problem for computing a deformed or interpolated shape. Since this is done during the online phase the performance of the solver is of particular interest.

Using Newton's method yields the same problem as discussed for the unreduced case. Namely, that the Hessian may not be positive definite. In contrast to the unreduced case, the invertible finite elements are not applicable here since we do not construct the Hessian by computing the contributions of the individual tets, but evaluate the precomputed polynomials. In our experiments, we found it preferable to use a quasi-Newton method instead. The scheme constructs and updates a symmetric positive-definite approximation of the Hessian during the optimization. In our experiments, we found the BFGS method to be effective. This scheme maintains an approximation of the inverse of the Hessian. Hence in every iteration not only the time for setting up the Hessian is saved, but also the time required for computing the descent direction is reduced. For a in-depth discussion of the BFGS scheme including a formula of the update rule, we refer to [50]. The default initialization of the inverse Hessian approximation is the identity matrix. However, the performance can be improved by initializing with a better approximation of the true inverse Hessian. We found using the inverse of the Hessian of the neutral and mean shape for shape modeling and interpolation effective in our experiments.

The termination criterion we use follows the framework discussed in [56]. It decides whether to stop based on the change of the objective value F, the convergence of the sequence x_k and the necessary condition $\| \nabla E_k \| = 0$ and is given by

$$
\begin{aligned}
E_{k-1} - E_k &< \epsilon(1 + |E_k|) \\
\|x_{k-1} - x_k\| &< \sqrt{\epsilon}(1 + \|x_k\|) \\
\| \nabla E_k \| &\leq \sqrt[3]{\epsilon}(1 + |E_k|)
\end{aligned}
$$

where $\epsilon \geq 0$ is the convergence tolerance. At optimization time, a value between 10^{-3} and 10^{-7} is picked.

7. Results and Discussion

The supplementary video shows examples of real-time shape modeling and interpolation performed with our implementation of the proposed framework. In all the shown examples, we use very low-dimensional subspaces (6-60 dim.) and demonstrate

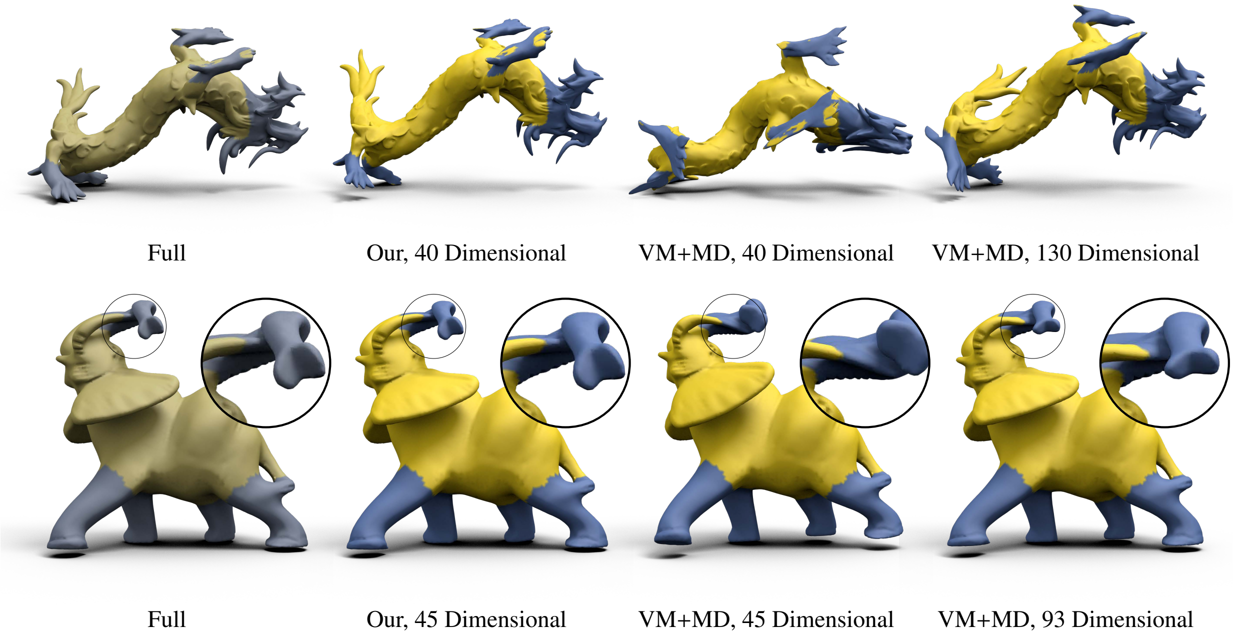


Figure 6: Comparison of results produced in different subspaces. From left to right: Unreduced reference solution, our construction, spaces of the same dimension constructed using vibration modes and modal derivatives, same construction with higher dimension. In a subspace created by our method, the handle constraints specified by the user (blue regions) match the positions of the full optimization much closer. This allows a more fine-grained positioning of local parts while retaining a plausible elastic behavior of the whole object.

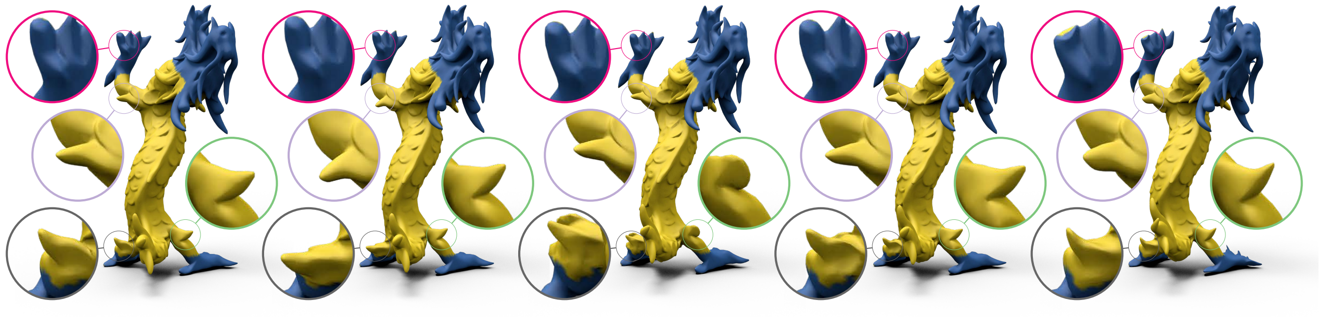
that large deformations including twists can be represented. One example, is the interpolation between a cylindrical shape and a twisted helix, shown in Figure 4. The example nicely illustrates the high-quality of the shape interpolation resulting from the physically-based problem modeling. The fact that the whole sequence of shapes lies in a 20-dimensional subspace, which has been computed using the proposed scheme, illustrates effectiveness of the subspace construction. Further examples of shape modeling and interpolation that include large deformations are the dragon, the dogs, and the lion shown in Figs. 1, 5, and 2. Figure 3 shows an example in which the subspace method is used to perform local modeling operations (modeling of the tongue of the dog).

Comparison with Biharmonic Coordinates and Radial Basis Functions. We compare our subspace construction method to Biharmonic Coordinates (BC) Wang et al. [18] and radial basis functions (RBF). For the comparison, we use the Chinese Dragon model with a St. Venant–Kirchhoff material (both Lamé parameters equal 1) and a weight of $\mu = 1000$ for the handle constraints and the Armadillo model with the Lamé parameters set to 1 and $\mu = 1$. For computing the deformation in all cases, we initialize the Newton scheme with the mass-orthogonal projection of the result derived from a full optimization to the respective subspace and iterate until convergence with a tolerance of $\epsilon = 10^{-7}$.

Model		Subspace		L^2-Distance
Name	V	Construction	d	(Std. Dev.)
Dragon	32k	Our	40	0.042 (0.013)
		VM+MD	40	0.151 (0.058)
			130	0.045 (0.022)
Elephant	24k	Our	45	0.009 (0.008)
		VM+MD	45	0.101 (0.040)
			93	0.056 (0.045)
Helix	1k	Our	20	0.002 (0.002)
		VM+MD	20	0.263 (0.074)
			100	0.083 (0.015)
			260	0.004 (0.001)

Table 2: Comparison of the relative L^2 error of our subspace construction methods to a space obtained from vibration modes and modal derivatives.

Both, BC and RBF require a point sampling of the shape, for which we use farthest point sampling on the mesh vertices. Starting with this sampling, we construct the BC subspace by generating the weights with `libigl`'s `biharmonic_coordinates` implementation, then, constructing three degrees of freedom (x,y,z movement) from each scalar



Full Optimization

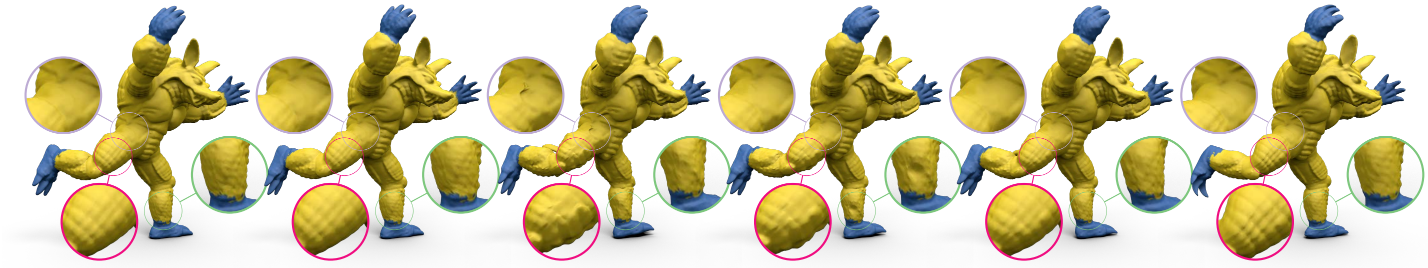
Ours,
40 Dimensional

Biharmonic Coordinates,
384 Dimensional

Biharmonic Coordinates,
744 Dimensional

Radial Basis Functions,
744 Dimensional

Figure 7: Visual comparison of a subspace created from sampling of the solution space (ours) vs. the subspace construction from [18] (Biharmonic Coordinates), and a spatial sampling followed by a reconstruction of the displacement field using radial basis functions. The degrees of freedom can be roughly an order of magnitude less, when sampling the solution space.



Full Optimization

Ours,
60 Dimensional

Biharmonic Coordinates,
228 Dimensional

Biharmonic Coordinates,
408 Dimensional

Biharmonic Coordinates,
768 Dimensional

Radial Basis Functions,
768 Dimensional

Figure 8: Visual comparison of a subspace created from sampling of the solution space (ours) vs. the subspace construction from [18] (Biharmonic Coordinates), and a spatial sampling followed by a reconstruction of the displacement field using radial basis functions.

weight. For radial basis functions, a number of choices exist, which include isotropic Gaussians [57, 58] and various kinds of piecewise defined polynomials [15, 59, 60]. We opt for isotropic Gaussians and compute the weights by using the distances in the embedding space.

Table 3 lists objective value and gradient magnitude (measured in the unreduced space) for the respective shapes for a different subspace dimensions. Note that these additionally depend on the total volume, which is 19.1 for the dragon and 234741 for the armadillo. The subspace dimension d depicts all independent degrees of freedom; for a direct comparison with the number h reported in [18], it must be divided by three (*e.g.*, $d = 384$ corresponds to 2 handles and 120 additional points). Fig. 7 and 8 depict visual comparisons. The table shows that, from a deformation energy perspective, our subspace yields states comparable to RBF and BC subspaces with up to 20× the number of DoFs. This is a crucial advantage since the computation time required for solving the reduced optimization problems mainly depends on the dimension of the reduced space. The order of the dependency varies with the method used for energy and gradient approximation. The exact evaluation of the reduced energy gradients by polynomial reduction, we used here, yields a force evaluation cost of $O(d^4)$. Faster schemes that approximate the energy gradients, like the optimized cubature, yield a cost of $O(d^2)$.

Method	Chinese Dragon (Fig. 7)			Armadillo (Fig. 8)		
	d	Objective	Gradient	d	Objective	Gradient
Full Optimization	–	$4.37 \cdot 10^{-1}$	$8.51 \cdot 10^{-3}$	–	$1.42 \cdot 10^{3}$	$3.48 \cdot 10^{1}$
Ours	40	$1.03 \cdot 10^{0}$	$1.24 \cdot 10^{0}$	60	$9.12 \cdot 10^{3}$	$9.03 \cdot 10^{3}$
Wang et al. [18]	69	$1.43 \cdot 10^{2}$	$5.87 \cdot 10^{2}$	138	$2.33 \cdot 10^{6}$	$5.02 \cdot 10^{5}$
	114	$6.15 \cdot 10^{1}$	$1.31 \cdot 10^{2}$	228	$4.44 \cdot 10^{4}$	$8.24 \cdot 10^{4}$
	204	$5.99 \cdot 10^{1}$	$7.85 \cdot 10^{1}$	408	$1.09 \cdot 10^{4}$	$3.27 \cdot 10^{3}$
	384	$6.12 \cdot 10^{0}$	$1.06 \cdot 10^{1}$	768	$8.54 \cdot 10^{3}$	$7.59 \cdot 10^{2}$
	744	$1.33 \cdot 10^{0}$	$4.40 \cdot 10^{0}$			
RBF	69	$2.46 \cdot 10^{3}$	$8.15 \cdot 10^{3}$	138	$1.29 \cdot 10^{6}$	$9.42 \cdot 10^{5}$
	114	$7.49 \cdot 10^{2}$	$3.73 \cdot 10^{3}$	228	$3.75 \cdot 10^{5}$	$1.58 \cdot 10^{5}$
	204	$1.94 \cdot 10^{2}$	$9.18 \cdot 10^{2}$	408	$5.11 \cdot 10^{4}$	$4.40 \cdot 10^{4}$
	384	$1.83 \cdot 10^{1}$	$2.43 \cdot 10^{2}$	768	$9.89 \cdot 10^{3}$	$8.00 \cdot 10^{3}$
	744	$2.22 \cdot 10^{0}$	$3.72 \cdot 10^{1}$			

Table 3: Quantititive comparison of our construction to BC and RBF subspaces. We list objective values and gradients of a full optimization evaluated at the respective results of a reduced optimization.

Comparison with vibration modes and modal derivatives. We compare the subspaces computed with our scheme to the ones spanned by vibration modes and modal derivatives (VB & MD), which have been used for deformation-based modeling in [12]. This approach constructs the subspace from eigenvectors of the deformation energy's Hessian at the model's rest pose, *i.e.*, derivatives of E in (3). Fig. 6 shows a visual comparison of solutions that include large deformations (rotations of the handles). Like our method, the use of VB & MD exhibits smooth and plausible, elastic deformations of the model; it favors deformations of low change of elastic potential to be included in the degrees of freedom of the subspace. However, this construction scheme does not account for the constraints, which limits how precise the local parts of the model can be placed by the user. In order to compensate, new degrees of freedom need to be included in the subspace. Our scheme can represent very vivid deformations with roughly half as many dimensions (see also the dinosaur and dragon model in the accompanying video). For interpolation between two shapes, we compare with subspaces spanned by vibration modes and modal derivatives of both example shapes. We compare the results for the interpolation between a cylinder and a helix—a very large deformation. Approximation errors compared to the unreduced solutions are given in Table 2. Fig. 3 demonstrates that the adaption of our scheme to the constraints can easily be exploited for local deformations. In this example the influence region of the deformation is limited to the head of the dog model. The interpolation of the helix and the cylinder shows that for very large deformations, our approach performs much better than the construction of subspaces only from derivatives of the example shapes. Table 2 shows that in order to obtain a comparable approximation quality to our 20-dimensional subspaces, the construction using modes and modal derivatives of the example shapes needs a 260-dimensional subspace.

Computation times. The computational effort for generating the samples is driven by three factors. The first is the number of samples that is to be computed. The second one is the number of iterations needed to find a sample solution. Thanks to better initial estimates that are sourced from the previous solutions, the average of this number usually decreases the more samples are to be generated. The third factor is the time spent in one iteration of the iterative solver. This mostly depends on the number of tetrahedra and, for shape interpolation, it also depends on the number of example shapes. Table 1 lists details and timing for these operations. Since we use the exact evaluation of the reduced forces, the preprocess also includes the precomputation of the polynomial coefficients of the reduced deformation energy. In particular, for the shape interpolation (where polynomials coefficients for all example shapes must be precomputed), this computation is costly. Switching to alternative approaches, like optimized cubature [31] or rotation clustering for the ARAP energy [30], would speed-up this computation, and, additionally, the evaluation costs at runtime. We still used the exact force evaluation to focus on the experimental evaluation on the subspaces (and eliminate the influence of force approximation).

Sample Size and Convergence. We tested the robustness of the subspace construction in the shape interpolation setting. Generally speaking, the generation of a linear subspace from a sampling of the shape space, *i.e.*, the solution space of (1), is prone to overfitting. We evaluated how well the subspaces generalize for different sample sizes computed for 6 poses of the Centaur model (see also accompanying video). Fig. 9 shows the results of leave-one-out cross-validation on different sample sizes and fixed dimensions; the measured quantity (summarized by mean and standard deviation) is the relative L_2 error when approximating one interpolated shape in a subspace constructed from the remaining samples.

The results show that the choice of sampling points has a relatively low influence and even small sample sizes give usable degrees of freedom for finding solutions in the reduced problem. The observation that a small number of DOFs suffices agrees with [42], who used 64 DOFs – roughly 3 times as many as we – to interpolate 5 shapes of the Centaur. Note that the numbers are not directly comparable, as [42] used triangle meshes and discrete shell energies.

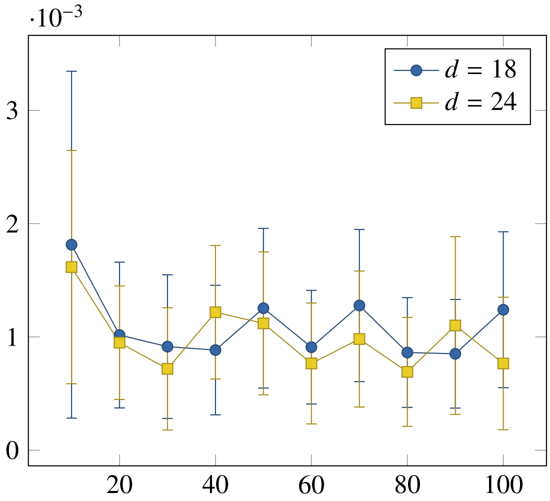


Figure 9: Mean L_2 error and variance on subspaces of fixed size r created from various sample sizes (horizontal axis). One nicely sees that the number of samples needed is roughly the same as the desired dimension of the subspace. The 18 dimensional subspace is sufficient to smoothly interpolate between 6 poses of the Centaur model, which is presented in the accompaniyng video.

8. Conclusion

We present a novel subspace construction for deformation-based shape modeling and interpolation. In contrast to previous approaches, our construction produces subspaces that are explicitly optimized for approximation of the deformations reachable with a specific user interface. With this method very low-dimensional subspace for modeling and interpolation can be produced. Comparisons to alternative approaches illustrate the benefits of subspace optimization. These spaces are well-suited for real-time shape modeling and interpolation.

8.1. Limitations and challenges

The price to pay for the optimization is that samples have to be generated. This process is automatic but requires computational effort. Since all samples could be computed independently, computation times could be greatly reduced by parallelization. A consequence of the optimization to a specific setting (*e.g.*, choice of handles) is that when the handle set is modified the spaces loose their optimality. Strictly speaking, new samples have to generated and a new space must be set up.

We are planning to integrate the subspace construction to a real-time template-based capture system. We are convinced that the combination of non-linear and physically-based deformation and real-time responses has potential for this area. Furthermore, it would be interesting to integrate shape analysis into the subspace construction process. For example, it may be possible to use symmetries of a shape to further reduce the size of the subspaces.

Acknowledgements

We would like to thank Christopher Brandt, Christian Schulz and Christoph von Tycowicz for inspiring discussions and for sharing code and data, Yu Wang, Alec Jacobson, Jernej Barbič and Ladislav Kavan for making their *biharmonic coordinates* code available as part of `libigl`, and the anonymous reviewers for helpful comments and suggestions.

9. References

[1] Botsch M, Sorkine O. On linear variational surface deformation methods. IEEE Trans Vis Comput Graphics 2008;14(1):213–230.

[2] Au OKC, Tai CL, Liu L, Fu H. Dual Laplacian editing for meshes. IEEE Trans Vis Comput Graphics 2006;12(3):386–395.

[3] Huang J, Shi X, Liu X, Zhou K, Wei LY, Teng SH, et al. Subspace gradient domain mesh deformation. ACM Trans on Graph 2006;25(3).

[4] Botsch M, Pauly M, Gross M, Kobbelt L. PriMo: Coupled prisms for intuitive surface modeling. In: Symposium on Geometry Processing. 2006, p. 11–20.

[5] Sorkine O, Alexa M. As-rigid-as-possible surface modeling. In: Symposium on Geometry Processing. 2007, p. 109–16.

[6] Chao I, Pinkall U, Sanan P, Schröder P. A simple geometric model for elastic deformations. ACM Trans Graph 2010;29:38:1–38:6.

[7] Rong G, Cao Y, Guo X. Spectral mesh deformation. Visual Computer 2008;24(7-9):787–796.

[8] Rustamov RM. On mesh editing, manifold learning, and diffusion wavelets. IMA International Conference on Mathematics of Surfaces XIII 2009;:307–321.

[9] Dey T, Ranjan P, Wang Y. Eigen deformation of 3d models. Visual Computer 2012;28(6-8):585–595.

[10] Wu Y, Au OKC, Taic CL, Lu T. Hirm: A handle-independent reduced model for incremental mesh editing. Geometric Modeling and Processing 2015;.

[11] Barbič J, James DL. Real-time subspace integration for St. Venant-Kirchhoff deformable models. ACM Trans Graph 2005;24(3):982–990.

[12] Hildebrandt K, Schulz C, von Tycowicz C, Polthier K. Interactive surface modeling using modal analysis. ACM Trans Graph 2011;30(5):119:1–119:11.

[13] von Tycowicz C, Schulz C, Seidel HP, Hildebrandt K. An efficient construction of reduced deformable objects. ACM Trans Graph 2013;32(6):213:1–10.

[14] Sumner RW, Schmid J, Pauly M. Embedded deformation for shape manipulation. ACM Trans Graph 2007;26(3).

[15] Adams B, Wicke M, Ovsjanikov M, Wand M, Seidel H, Guibas LJ. Meshless shape and motion design for multiple deformable objects. Comput Graph Forum 2010;29(1):43–59.

[16] Wu X, Wand M, Hildebrandt K, Kohli P, Seidel HP. Real-time symmetry-preserving deformation. Computer Graphics Forum 2014;33(7):229–238.

[17] Jacobson A, Baran I, Popović J, Sorkine O. Bounded biharmonic weights for real-time deformation. ACM Transactions on Graphics 2011;30(4):78:1–78:8.

[18] Wang Y, Jacobson A, Barbič J, Kavan L. Linear subspace design for real-time shape deformation. ACM Trans Graph 2015;34(4):57:1–57:11.

[19] Wang RY, Pulli K, Popović J. Real-time enveloping with rotational regression. ACM Trans Graph 2007;26(3).

[20] Weber O, Sorkine O, Lipman Y, Gotsman C. Context-Aware Skeletal Shape Deformation. Computer Graphics Forum 2007;.

[21] Botsch M, Pauly M, Wicke M, Gross M. Adaptive space deformations based on rigid cells. Computer Graphics Forum 2007;26(3):339–347.

[22] Lipman Y, Levin D, Cohen-Or D. Green coordinates. ACM Trans Graph 2008;27(3):1–10.

[23] Ben-Chen M, Weber O, Gotsman C. Variational harmonic maps for space deformation. ACM Trans Graph 2009;28(3).

[24] García FG, Paradinas T, Coll N, Patow G. *cages: : A multilevel, multi-cage-based system for mesh deformation. ACM Trans Graph 2013;32(3):24.

[25] Zhang J, Deng B, Liu Z, Patanè G, Bouaziz S, Hormann K, et al. Local barycentric coordinates. ACM Trans Graph 2014;33(6).

[26] Landreneau E, Schaefer S. Poisson-based weight reduction of animated meshes. Computer Graphics Forum 2010;29(6):1945–1954.

[27] Kry PG, James DL, Pai DK. Eigenskin: Real time large deformation character skinning in hardware. In: Symposium on Computer Animation. 2002, p. 153–9.

[28] Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J, Davis J. Scape: Shape completion and animation of people. ACM Trans Graph 2005;24(3):408–416.

[29] Bouaziz S, Wang Y, Pauly M. Online modeling for realtime facial animation. ACM Trans Graph 2013;32(4):40:1–40:10.

[30] Jacobson A, Baran I, Kavan L, Popović J, Sorkine O. Fast automatic skinning transformations. ACM Trans Graph 2012;31(4):77:1–77:10.

[31] An SS, Kim T, James DL. Optimizing cubature for efficient integration of subspace deformations. ACM Trans Graph 2008;27(5):1–10.

[32] Yang Y, Li D, Xu W, Tian Y, Zheng C. Expediting precomputation for reduced deformable simulation. ACM Trans Graph 2015;34(6):243:1–13.

[33] Xu D, Zhang H, Wang Q, Bao H. Poisson shape interpolation. In: Symp. on Solid and Phys. Mod. 2005, p. 267–74.

[34] Winkler T, Drieseberg J, Alexa M, Hormann K. Multi-scale geometry interpolation. Computer Graphics Forum 2010;29(2):309–318.

[35] Rumpf M, Wirth B. A nonlinear elastic shape averaging approach. SIAM J Img Sci 2009;2(3):800–833.

[36] Chen R, Weber O, Keren D, Ben-Chen M. Planar shape interpolation with bounded distortion. ACM Trans Graph 2013;32(4):108:1–108:12.

[37] Fröhlich S, Botsch M. Example-driven deformations based on discrete shells. Comp Graph Forum 2011;30(8):2246–2257.

[38] Martin S, Thomaszewski B, Grinspun E, Gross M. Example-based elastic materials. ACM Trans Graph 2011;30(4):72:1–72:8.

[39] Marras S, Cashman TJ, Hormann K. Efficient interpolation of articulated shapes using mixed shape spaces. Computer Graphics Forum 2013;32(8):258–270.

[40] Levi Z, Gotsman C. Smooth rotation enhanced as-rigid-as-possible mesh animation. IEEE Trans Vis Comput Graph 2015;21(2):264–277.

[41] Zhang Z, Li G, Lu H, Ouyang Y, Yin M, Xian C. Fast as-isometric-as-possible shape interpolation. Computers & Graphics 2015;46:244 – 256. Shape Modeling International 2014.

[42] von Tycowicz C, Schulz C, Seidel HP, Hildebrandt K. Real-time nonlinear shape interpolation. ACM Trans Graph 2015;34(3).

[43] Kilian M, Mitra NJ, Pottmann H. Geometric modeling in shape space. ACM Transactions on Graphics 2007;26(3):64:1–64:8.

[44] Heeren B, Rumpf M, Wardetzky M, Wirth B. Time-discrete geodesics in the space of shells. Computer Graphics Forum 2012;31(5):1755–1764.

[45] Brandt C, von Tycowicz C, Hildebrandt K. Geometric flows of curves in shape space for processing motion of deformable objects. Computer Graphics Forum 2016;35(2).

[46] Grinspun E, Hirani AN, Desbrun M, Schröder P. Discrete Shells. Symposium on Computer Animation 2003;:62–67.

[47] Bonet J, Wood RD. Nonlinear Continuum Mechanics for Finite Element Analysis. Cambridge Press; 2008.

[48] van Kaick O, Zhang H, Hamarneh G, Cohen-Or D. A survey on shape correspondence. Comp Graph Forum 2011;30(6):1681–1707.

[49] Sobol I. On the distribution of points in a cube and the approximate evaluation of integrals. Computational Mathematics and Mathematical Physics 1967;7(4):86 – 112.

[50] Nocedal J, Wright SJ. Numerical Optimization. Springer; 2006.

[51] Irving G, Teran J, Fedkiw R. Invertible finite elements for robust simulation of large deformation. In: Symposium on Computer Animation. 2004, p. 131–40.

[52] Eggert DW, Lorusso A, Fisher RB. Estimating 3-D rigid body transformations: a comparison of four major algorithms. Mach Vis Appl 1997;9(5-6):272–290.

[53] Wardetzky M, Bergou M, Harmon D, Zorin D, Grinspun E. Discrete quadratic curvature energies. Computer Aided Geometric Design 2007;24(8-9):499–518.

[54] Cook RD, Malkus DS, Plesha ME, Witt RJ. Concepts and Applications of Finite Element Analysis. Wiley; 2001.

[55] Hinze M, Volkwein S. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems. In: Dimension Reduction of Large-Scale Systems. Springer; 2005, p. 261–306.

[56] Gill PE, Murray W, Wright MH. Practical optimization. Academic Press; 1981.

[57] Lewis JP, Cordner M, Fong N. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In: SIGGRAPH. 2000, p. 165–72.

[58] Wu L, Tiso P. Modal derivatives based reduction method for finite deflections in floating frame. In: World Congress on Computational Mechanics. 2014, p. 10:3125–3135.

[59] Müller M, Keiser R, Nealen A, Pauly M, Gross M, Alexa M. Point based animation of elastic, plastic and melting objects. In: Symposium on Computer Animation. 2004, p. 141–51.

[60] Müller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications. In: Symposium on Computer Animation. 2003, p. 154–9.