

Space-Time Editing of Elastic Motion through Material Optimization and Reduction

Siwang Li¹ Jin Huang^{1*} Fernando de Goes² Xiaogang Jin¹ Hujun Bao^{1*} Mathieu Desbrun²
¹State Key Lab of CAD&CG, Zhejiang University ²Caltech



Figure 1: From an input sequence (bottom) of a standing *T. rex* simulated via a Saint Venant-Kirchhoff deformation model, we edit the input motion with 27 space-time constraints to make the dinosaur squat & jump while trying to catch the small plane flying around it (top).

Abstract

We present a novel method for elastic animation editing with space-time constraints. In a sharp departure from previous approaches, we not only optimize control forces added to a linearized dynamic model, but also optimize material properties to better match user constraints and provide plausible and consistent motion. Our approach achieves efficiency and scalability by performing all computations in a reduced rotation-strain (RS) space constructed with both cubature and geometric reduction, leading to two orders of magnitude improvement over the original RS method. We demonstrate the utility and versatility of our method in various applications, including motion editing, pose interpolation, and estimation of material parameters from existing animation sequences.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: motion editing; elastic animation; space and time constraints; model reduction.

Links: [DL](#) [PDF](#)

1 Introduction

Providing efficient control of elastic simulation has received considerable attention in recent years: be it for interpolation between poses or editing of existing animation sequences through position constraints, one always seeks a plausible motion requiring the least

amount of non-physical control forces to match the input poses and/or spatial constraints in time. However, most current methods assume a fixed elastic material (with given mass, stiffness, and damping matrices) during the optimization of control forces, which may not be appropriate for, or compatible with, the type of position constraints that the user imposes. Large edits may create unduly large control forces, and thus unrealistic motion at and around the constraints. Moreover, unavoidably large control forces may also affect the dynamics long before or after the edit in the animation sequence. In this paper, we improve upon motion editing approaches by incorporating material optimization within the formulation of a constrained optimal control problem.

The need for efficiency and scalability in motion editing dictates the use of reduced dynamics, while robustness to large deformation forces requires non-linear coordinates. Our approach achieves both efficiency and robustness by expressing shape deformation in a reduced rotation-strain space, obtained via geometric reduction and a cubature scheme. The dynamics in this representation can be encoded through a small number of non-linear modes and pose-independent material parameters, which are optimized to reduce the amount of control forces needed to match user-defined constraints. Typical artifacts of linear reduction are thus removed, while shape reconstruction is performed over two orders of magnitude faster than in previous non-linear methods.

1.1 Related Work

Physically-based simulation can generate exquisitely complex animation sequences, but fine-tuning the resulting motion requires a trial-and-error process involving time-consuming parameter tweaking. Early efforts in animation control mainly focused on character animation [Gleicher 1997], rigid body simulation [Popovic et al. 2000; Twigg and James 2007], and fluid simulation [Treuille et al. 2003; Kim et al. 2006; Nielsen and Bridson 2011]. Recently, a number of methods has been proposed to control elastic object animation as it represents a challenge in both geometry processing and animation. While pose (or keyframe) interpolation has been shown a useful tool [Barbič et al. 2009; Huang et al. 2011; Hildebrandt et al. 2012], the framework of space-time constraints [Witkin and Kass 1988; Fang and Pollard 2003; Safonova et al. 2004; Jeon and Choi 2007; Barbič et al. 2012; Li et al. 2013] offers a more general approach to intuitively edit an input sequence: editing is cast as a constrained optimal control problem, in which one solves for the

*Corresponding authors.

¹e-mail: lisiwang@zjucadcg.cn, {hj.jin, bao}@cad.zju.edu.cn

²e-mail: {fdgoes, mathieu}@caltech.edu

least amount of non-physical forces needed to satisfy user-defined constraints in space and time.

Linearization and reduction of full-blown elasticity models are necessary tools to enforce tractability. In particular, modal analysis [Pentland and Williams 1989; Hauser et al. 2003] was proposed to reduce the dynamics to only a few modes, computed through eigen analysis of the (constant) mass and stiffness matrices of the elastic object. However, such linear approximations are notoriously inadequate for large deformation. This issue was circumvented in the context of pose interpolation by either applying linearization around the keyframes [Hildebrandt et al. 2012] or introducing a Rotation-Strain (RS) space [Huang et al. 2011]; alas, these solutions do not address the general case of space-time and partial positional constraints. Still, some alternatives were proposed: Barbič et al. [2012] used [Huang et al. 2011] in a post-warping step, at the cost of violating position constraints; linearizing the equation of motion around the input trajectory [Barbič and Popović 2008; Li et al. 2013] often achieves tighter position constraints, but artifacts still appear when large motion editing is performed.

In addition to the limitations mentioned above, we point out that most previous work considers material parameters and eigenmodes fixed in their optimization of control forces. This choice can severely affect the overall dynamics of the edited sequence, particularly when significant changes to the input sequence are required. Stiffness and damping were successfully optimized in [Huang et al. 2011], but restricted to a small number of prescribed modes and key frame interpolation only. Recently, material optimization was applied to fabrication and 3D printing [Bickel et al. 2009; Bickel et al. 2010; Martin et al. 2011], but not with the goal of minimizing non-physical forces. The work of [Coros et al. 2012] also used material optimization in order to adapt rest shapes and then control the motion style of deformable characters, but without space-time constraints.

1.2 Contributions and Overview

In this paper, we present a new approach to space-time motion editing of elastic objects through material optimization. Our technique offers interactive manipulation of animation sequences that maintains the input dynamic behavior while ensuring physical plausibility of the edited sequence. We demonstrate that our formulation gives the user tight control over positions with much smaller forces compared to previous work, and no visual artifacts even for large deformation. Our approach can also be used to recover material properties from existing animation sequences.

At the core of our approach are three key ingredients. First, we adopt the modal Rotation-Strain (RS) space introduced in [Huang et al. 2011] to “diagonalize” the equation of motion and provide robustness to large deformation. Second, we construct a novel *reduced* RS space based on geometric reduction and cubature to significantly speed up the reconstruction of 3D coordinates from modal RS coordinates. And finally, we allow intuitive space-time editing of elastic animation by not only minimizing the magnitude of the control forces required to fit user-defined constraints, but also through an optimization of the modes, frequencies, and damping to use for the final animation—which amounts to an optimization of the object’s material coefficients.

2 Background

We begin by reviewing two crucial components that our method builds upon: modal analysis and rotation-strain coordinates.

2.1 Modal analysis

The linearized equation of motion for an elastic object with rest shape \bar{u} is

$$\tilde{M}\ddot{u} + \tilde{D}\dot{u} + \tilde{K}u = \tilde{f}_{ext}(t), \quad (1)$$

where $u \in \mathbb{R}^{3n}$ represents the displacements from \bar{u} of all n nodes of the discrete representation of the object, $\tilde{M}, \tilde{K}, \tilde{D} \in \mathbb{R}^{3n \times 3n}$ are the mass, stiffness, and damping matrices respectively, and $\tilde{f}_{ext}(t)$ is the external force at time t . The shape of the elastic object in time is reconstructed as $\bar{u} + u(t)$.

Modal analysis builds a reduced space for this motion equation by making use of the solutions of the general eigenvalue problem

$$\tilde{K}\phi = \lambda\tilde{M}\phi. \quad (2)$$

By assembling the smallest r eigenvalues in a diagonal matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_r)$ and stacking their associated eigenvectors in a matrix $W = (\phi_1, \dots, \phi_r)$, one can rewrite Eq. (1) in this modal subspace as a set of r equations:

$$\ddot{z} + D\dot{z} + \Lambda z = f(t), \quad (3)$$

where $z \in \mathbb{R}^r$ stores the displacements in modal coordinates (representing the magnitudes in time of the eigenmodes), and $f(t) = W^T \tilde{f}_{ext}(t)$ represents the external forces in modal coordinates. One can then convert modal coordinates back into 3D coordinates by simply evaluating

$$u = Wz. \quad (4)$$

Observe that Eq. (1) assumes a displacement from a fixed rest shape \bar{u} . However, the same equation and modal analysis can be applied more generally for any animation sequence. When editing an existing sequence $\bar{u}(t)$, for instance, we seek an offset $u(t)$ that makes the final animation $\bar{u}(t) + u(t)$ satisfy user-specified space-time constraints through the least amount of control forces [Barbič et al. 2012; Li et al. 2013]. In the case of keyframe interpolation, we instead set \bar{u} as a rest shape and consider it constant in time.

2.2 Rotation-Strain coordinates

The linear reconstruction from modal coordinates to Euclidean coordinates in Eq. (4) is notoriously inadequate for large deformation. While the use of modal warping [Choi and Ko 2005] partially remedies this issue, the introduction of rotation-strain (RS) coordinates [Huang et al. 2011] as an *intermediary* representation was shown to be the most robust approach to reconstruct plausible shapes (see Fig. 2 for a side-by-side comparison).

RS coordinates are formed by a pair of 3×3 matrices $y_e \equiv (y_e^\theta, y_e^s)$ per element e in the set of tetrahedra \mathcal{T} of the input model, where y_e^θ is an antisymmetric matrix representing angular velocity and y_e^s is a symmetric matrix representing the Biot strain. As detailed in [Huang et al. 2011], the vector y of RS coordinates is constructed as the antisymmetric and symmetric parts of the deformation gradient G_u associated to the displacement u , where G is a discrete gradient operator defined on the rest shape. We can thus concisely express the map from u to y as a linear system $y = Qu$ with sparse matrix Q . Conversely, RS coordinates y are mapped back to displacements u in two steps: we first transform each RS coordinate y_e into a deformation gradient $g(y_e)$ via the exponential map:

$$g(y_e) = \exp(y_e^\theta)(I + y_e^s) - I, \quad (5)$$

and then we recover the displacement u by solving a Poisson reconstruction problem:

$$\min_u \sum_{e \in \mathcal{T}} \bar{V}_e \| (Gu)_e - g(y_e) \|_F^2, \quad (6)$$

where \bar{V}_e is the volume of tetrahedron e at rest, and $\|\cdot\|_F$ is the Frobenius norm. The solution of this quadratic minimization is found via a sparse linear system of the form:

$$Au = G^T \bar{V} g(y), \quad (7)$$

where $A = G^T \bar{V} G$ is a Laplacian-like, constant, and sparse matrix (thus prefactorizable) obtained from the second derivatives of the objective function, \bar{V} is a diagonal matrix with the volume \bar{V}_e of each tetrahedron e repeated 9 times, and $g \in \mathbb{R}^{9|\mathcal{T}|}$ is the row concatenation of the deformation gradient matrices $\{g(y_e)\}$. Observe that the recovery of 3D displacements u from the RS coordinates y via the exponential map matches their linear relationship ($y = Qu$) for small deformation [Huang et al. 2011]. Moreover, Eq. (7) needs at least one constrained node (or a regularization term) in order to remove the kernel of A . When several nodes are constrained, one can further simplify Eq. (7) by eliminating their corresponding rows and columns in A , G , and \bar{V} .

2.3 Modal Rotation-Strain coordinates

The work of [Huang et al. 2011] also proposed to combine modal analysis and RS coordinates by projecting the r bases ϕ_i in W into RS coordinates through the matrix Q . We can thus relate modal coordinates z to RS coordinates y via

$$y = QWz. \quad (8)$$

Note that, by sequentially applying Eqs. (8), (5) and (7), the displacement vector u is now reconstructed from the RS coordinates induced by modal coordinates z . This hybrid approach thus offers the benefits of both modal analysis and RS coordinates: while the former diagonalizes the equation of motion (Eq. (3)), the latter is robust to large deformation. We refer to z in Eq. (8) as the *modal RS coordinates* hereafter.

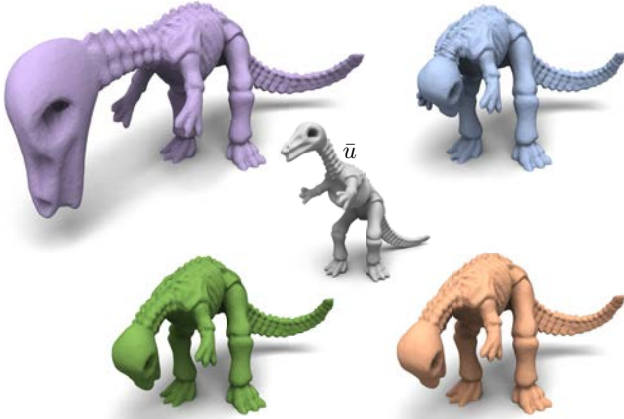


Figure 2: Shape reconstruction from a rest shape \bar{u} (center) and modal coordinates z using different methods. Top left: with linear modes. Top right: with modal warping. Bottom left: with the original RS method [Huang et al. 2011]. Bottom right: with our reduced RS method.

3 Reduced RS method

While modal RS coordinates are robust to large deformation, they require a global linear solve (namely, Eq. (7)) to reconstruct the 3D coordinates of a shape. This computational cost is an impediment

to the formulation of a constrained optimal control problem that relies on local evaluations of u and z . As our first contribution, we introduce a *reduced RS* method to dramatically improve the performance of the shape reconstruction from modal RS coordinates, and thus make our optimal control problem easier to solve for. We propose to approximate displacements u based on a combination of geometric reduction and a cubature scheme as we describe next.

3.1 Geometric reduction

The core idea of geometric reduction is to project the space of displacements u to a smaller set by exploring geometric correlations in the input shapes. We thus introduce an orthogonal matrix $B \in \mathbb{R}^{3n \times h}$ (i.e., $B^T B = I$), and approximate u in a subspace of dimension h (with $h \ll 3n$) via

$$u = Bq, \quad (9)$$

where $q \in \mathbb{R}^h$ represents the reduced geometric coordinates. Replacing Eq. (9) into Eq. (7) and multiplying both sides by B^T as suggested in [Stanton et al. 2013] yields

$$B^T A B q = B^T G^T \bar{V} g, \quad (10)$$

where A , G and \bar{V} remain unchanged from Section 2.3. Using the constant matrix $P = (B^T A B)^{-1} B^T G^T \bar{V}$, the reduced geometric coordinates are expressed as:

$$q = P g = \sum_{e \in \mathcal{T}} P_e g_e, \quad (11)$$

where $P_e \in \mathbb{R}^{h \times 9}$ is the submatrix in P for each tetrahedron e . Note that this geometric reduction now allows us to quickly and locally evaluate the 3D position of a node: picking rows $3k$, $(3k+1)$, and $(3k+2)$ of B and multiplying by q return the 3D coordinates u_k of the k^{th} node. It also bears pointing out that the matrices B and W play distinct roles in our reduced RS representation: while B is a purely geometric transformation to accelerate computations, the matrix W is used to diagonalize the equation of motion in RS space.

The construction of the matrix B is performed only once in a pre-computation stage, and it can be based on any geometry processing approach that effectively captures the input shape and its possible deformation. In our implementation, we settled on modal derivatives [Barbič and James 2005] using the leading 30 modes, uniform density, Young's modulus, and Poisson's ratio. When an input sequence is available (e.g., for animation editing or material recovery), the matrix B can be enriched through Principal Component Analysis. The method in [von Tycowicz et al. 2013] can also be used to speed up the initialization of B .

3.2 Cubature

Geometric reduction significantly speeds up the Poisson reconstruction needed to convert RS coordinates y to 3D coordinates u . However, the evaluation of the RS coordinates for all the tetrahedron elements is still required. We propose to further reduce computations through a cubature scheme in RS space. Given a set \mathcal{T}_{cub} of cubature sample tetrahedra (with $|\mathcal{T}_{\text{cub}}| \ll |\mathcal{T}|$) and their associated cubature weights $\{\omega_e\}_{e \in \mathcal{T}_{\text{cub}}}$, we approximate q via:

$$q \approx \sum_{e \in \mathcal{T}_{\text{cub}}} \omega_e P_e g_e \triangleq \omega^c P^c g^c, \quad (12)$$

where ω^c , P^c , g^c are matrices assembled from $\{\omega_e, P_e, g_e\}_{e \in \mathcal{T}_{\text{cub}}}$, respectively. With this cubature approximation, only the RS coordinates y^c of the tetrahedra in the cubature set \mathcal{T}_{cub} are required:

$$y^c = (QW)^c z, \quad (13)$$

where $(QW)^c$ is formed by rows of QW in Eq. (8) associated to the elements in \mathcal{T}_{cub} . This approach is, in practice, more than three hundred times faster than solving u through the global linear system in Eq. (7), with little to no visual impact as illustrated in Fig. 2.

We precompute the cubature elements and their weights based on a training data of the form $\{(z_s, u_s)\}_{s=1..N_t}$. Following the method in [An et al. 2008], we use a greedy approach to select cubature samples and optimize weights via a non-negative least squares minimization of the fitting error:

$$\sum_{s=1}^{N_t} \left\| \frac{1}{\|u_s\|} (u_s - B \omega^c P^c g((QW)^c z_s)) \right\|^2. \quad (14)$$

Notice that this minimization can be further simplified by employing the geometric reduction presented in Section 3.1. To this end, we approximate $u_s \approx Bq_s$ by setting q_s as the minimizer of $\|Bq_s - u_s\|^2$, which leads to $q_s = B^T u_s$ since $B^T B = I$, so that Eq. (14) reduces to:

$$\sum_{s=1}^{N_t} \left\| \frac{1}{\|q_s\|} (q_s - \omega^c P^c g((QW)^c z_s)) \right\|^2. \quad (15)$$

If an input animation is not available, we generate training data via simulation. Poses u_s from the simulation sequence are first transformed into RS coordinates y_s , then into the modal RS coordinates z_s through $\|QW z_s - y_s\|^2$. As shown in the inset, the relative error decreases quickly as the number of cubature samples increases. For 80 cubature samples (green tetrahedra), the relative reconstruction error for the flower model is below 1.45%. In our implementation, we greedily add cubature elements until it guarantees less than 3% relative error, leading to a range of 60 to 80 cubature elements.

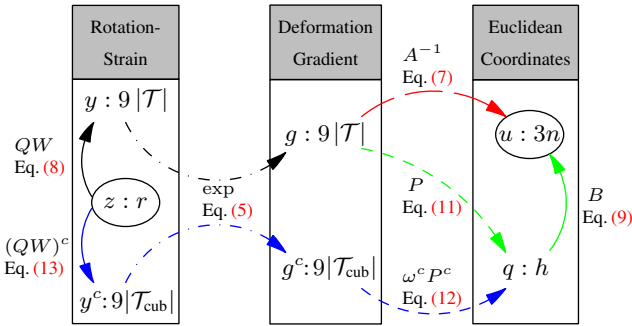
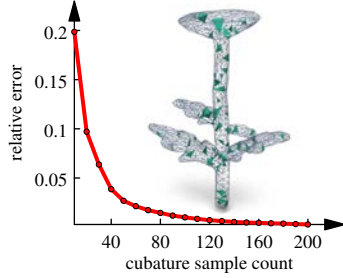


Figure 3: Diagram illustrating different approaches for the reconstruction of 3D coordinates. The expression after each colon indicates the dimension of its associated element listed before the colon.

3.3 Summary

We summarize in Fig. 3 all the components involved in our reduced RS method that transforms modal RS coordinates to Euclidean coordinates. Matrices and equations (along with their dimensions) used to map between RS, deformation gradient, and Euclidean coordinates are labeled for clarity. While the original RS method re-

quires solving a large sparse system (red arrow) to go from deformation gradients to Euclidean space, we bypass this step via geometric reduction using only matrix multiplications of size h (green arrows). Our cubature scheme further accelerates this procedure by only evaluating RS coordinates and deformation gradients on $|\mathcal{T}_{\text{cub}}|$ cubature tetrahedra (blue arrows). All computations in our reduced RS method are thus performed in low dimensional spaces (of dimensions r , $|\mathcal{T}_{\text{cub}}|$, and h), except for the final reconstruction that requires a multiplication by B to convert to 3D coordinates.

4 Material optimization

Equipped with our reduced RS space, we now introduce our approach to motion editing through optimal control of both modal RS coordinates and material properties.

4.1 Rationale

Finding the “most physical motion” satisfying a series of space-time constraints can be abstracted into an optimal control formulation, where one looks for the modal RS coordinates z that minimizes the amount of external forces needed to match user-defined position constraints. This assumes an *a-priori* stiffness matrix \tilde{K} and a mass matrix \tilde{M} , which define the elastic object’s behavior. However, pre-assigned material properties may degrade the resulting motion when the user imposes large deformation that are incompatible with this choice of material parameters. Instead, we propose to optimize not only the modal RS coordinates z , but also the modal bases W , frequencies Λ , and damping D used for the animation sequence. Assuming that $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_r)$ and $D = \text{diag}(d_1, \dots, d_r)$ are both diagonal matrices, we formulate our optimal control problem as follows:

$$\begin{aligned} \arg \min_{z, \Lambda, D, W} E_f(z, \Lambda, D) + \gamma E_c(W, z) \\ \text{subject to } \lambda_k, d_k \geq 0 \quad \forall k \in [1, r], \end{aligned} \quad (16)$$

where E_f measures the amount of external forces required to generate the motion, and E_c measures how closely the user-specified space-time constraints should be met in \mathbb{R}^3 , i.e.,

$$\begin{cases} E_f(z, \Lambda, D) = \frac{1}{2} \sum_{i=2}^{N-1} \|\ddot{z}_i + D\dot{z}_i + \Lambda z_i\|_2^2, & (17a) \\ E_c(W, z) = \frac{1}{2} \sum_{(i,j) \in \mathbb{C}} \|u_j^i(W, z_i) - \hat{u}_j^i\|_2^2. & (17b) \end{cases}$$

In these energies, finite differences in time are used to express $\ddot{z}_i = z_{i+1} - 2z_i + z_{i-1}$ and $\dot{z}_i = z_{i+1} - z_i$, \mathcal{N} is the total number of frames, and \hat{u}_j^i denotes the specified target displacements (with respect to the input sequence \bar{u}) for node j at time i , while \mathbb{C} denotes the set of user-defined constraints (i, j) . The parameter γ can be used to offer loose or strong enforcement of constraints; we typically used $\gamma = 10^{-6}$ for motion editing via position constraints, and $\gamma = 1.0$ for key pose interpolation and material parameter recovery. Note that enforcing positive coefficients $\{\lambda_k\}$ and $\{d_k\}$ is necessary to ensure proper physical behavior, as the λ_k ’s represent squared natural frequencies, while the damping factors d_k ’s should never introduce energy to the system.

4.2 Material bases sampling

An obvious hurdle in the above formulation is that it now requires an optimization with respect to a dense $3n \times r$ matrix W , which is very costly. We reduce complexity by, instead, picking W to be an arbitrary (but small) subspace of modes: while the traditional

W for modal analysis is composed of the first r eigenvectors of Eq. (2) (corresponding to the smallest eigenvalues λ_k), we choose W to be a r -dimensional subspace formed by a linear combination of a larger set of $R > r$ modes. More concretely, we precompute a matrix \widehat{W} whose columns are the first R eigenvectors of the generalized eigenvalue problem in Eq. (2) (corresponding to the R smallest eigenvalues λ_k), and restrict W to be of the form:

$$W = \widehat{W}S, \quad (18)$$

where the ‘‘sampling’’ matrix S of size $R \times r$ adds new degrees of freedom in our solver. This parameterization of W renders the optimal control problem much more tractable, while still only discarding the modes corresponding to the highest frequencies of the original elastic model. The stiffness and damping matrices (respectively, Λ and D) are still assumed to be diagonal, with their diagonal elements now being unknowns to be optimized.

4.3 Material bases optimization

Given our choice of material bases, our optimization problem is finally formulated as a function of z , Λ , D , and S through

$$\begin{aligned} & \arg \min_{z, \Lambda, D, S} E_f(z, \Lambda, D) + \gamma E_c(\widehat{W}S, z) + \mu E_s(S) \\ & \text{subject to } \lambda_k, d_k \geq 0 \quad \forall k \in [1, r], \end{aligned} \quad (19)$$

where the regularization term

$$E_s(S) = \|\text{diag}(S^T S) - \mathbf{1}_{r \times 1}\|_2^2, \quad (20)$$

is used to eliminate the trivial solution $z \rightarrow 0$, $\|S\|_2 \rightarrow +\infty$. We set $\mu = 10^{-8}$ in all our experiments, but results are similar for a large range of regularization parameters as demonstrated in Fig. 4.

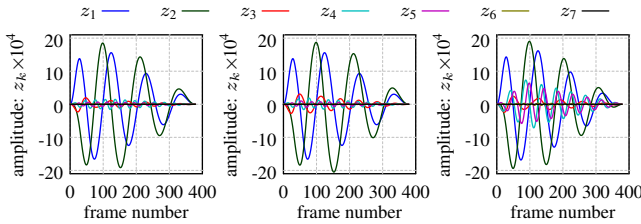


Figure 4: From left to right: the results of the flower example with modal RS coordinates $z_k(t)$ ($k = 1 \dots 7$) and $\mu = 10^{-10}, 10^{-8}, 10^{-6}$, respectively, are visually indistinguishable as the low frequencies remain very similar across changes of μ spanning several orders of magnitude.

5 Optimal control solver

We now delve into the algorithmic details to solve for our constrained optimal control formulation in Eq. (19).

5.1 Approach

We adopt an iterative scheme to efficiently compute an optimal motion. We optimize the variables one by one, keeping the others fixed to leverage the quadratic nature of most terms in Eq. (19):

- Fix $\Lambda = \Lambda^{(n)}$, $D = D^{(n)}$, $S = S^{(n)}$, optimize $z^{(n+1)}$;
- Fix $z = z^{(n+1)}$, $S = S^{(n)}$, optimize $\Lambda^{(n+1)}$, $D^{(n+1)}$;
- Fix $z = z^{(n+1)}$, $\Lambda = \Lambda^{(n+1)}$, $D = D^{(n+1)}$, optimize $S^{(n+1)}$.

5.2 Initialization

We initialize modal RS coordinates with $z^{(0)} = 0$, the material basis matrix with $S^{(0)} = (I_{r \times r}, 0)^T$ (corresponding to the usual choice of W using the r smallest eigenvalues), and a damping matrix $D^{(0)} = 0$. To provide a reasonable initial value for the stiffness matrix $\Lambda^{(0)}$, we exploit the property that the natural frequency of a motion is linearly proportional to $\sqrt{\lambda}$ for low damping motion. We thus uniformly scale the eigenvalues from the general eigenproblem to fix time scale such that the first non-zero eigenvalue is at $(2\pi/L)^2$, where L is the user-defined period (in number of frames) of the most significant mode present in the motion. In order to favor smoother deformation, we also recommend to scale each column of the modal basis matrix \widehat{W} by the inverse square root of each of their associated eigenvalue λ from Eq. (2). Other adjustments of \widehat{W} could also be used based on user-specific requirements.

5.3 Optimization of z

We use a Quasi-Newton method to optimize z based on an approximated Hessian evaluated as

$$H_z = H_{f,z} + \gamma J_z^T J_z, \quad (21)$$

where

$$H_{f,z} = \frac{\partial^2 E_f}{\partial z^2} \quad \text{and} \quad J_z = \frac{\partial u^c}{\partial z}. \quad (22)$$

with u^c as the vector containing all node displacements $u_j^i(\widehat{W}S, z)$ in Eq. (17b) related to the user-defined constraints—see Appendix A for closed-form expressions of J_z . Note that the computational cost involved in evaluating J_z depends on the number of constrained nodes, but $H_{f,z}$ does not. We used CHOLMOD [Chen et al. 2008] to solve for these Newton’s steps in our implementation.

5.4 Optimization of Λ, D

Since we keep z and S fixed, $E_f(\Lambda, D)$ is a quadratic function of stiffness Λ and damping D . We thus update stiffness coefficients λ_k (resp., damping coefficients d_k) by directly solving a diagonal $r \times r$ linear system involving the modal RS coordinates z (resp., their time derivatives \dot{z}). If some of the resulting stiffness or damping values become negative, we simply set them to zero to enforce the conditions $\lambda_k \geq 0$ and $d_k \geq 0$.

5.5 Optimization of S

Given that the Hessian of E_c with respect to S is dense, we choose a Hessian-free optimization for this step, using the implementation of [Bochkanov] of the L-BFGS method [Byrd et al. 1995]. We find that this approach performs systematically better than Newton’s method. In all our experiments, we start with the current sampling matrix $S^{(n)}$, from which a maximum of 50 iterations is sufficient to get satisfactory results.

6 Results

We demonstrate next how one can use our approach to handle a variety of applications related to motion editing.

6.1 Space-time constrained animation editing

A first application of our method is to edit an existing animation sequence by allowing the user to interactively select position constraints in order to change the motion. From the input sequence given as positions $\bar{u}(t)$ of nodes in time, one could find an optimal

offset $u(t)$ to the input sequence that satisfies user-specified constraints as in [Barbič et al. 2012; Li et al. 2013] with least control force, and reconstruct the shape through $\bar{u}(t)+u(t)$. However, such a linearized scheme leads to significant artifacts if large offsets are needed, and even the use of post-warping [Barbič et al. 2012] to mitigate these effects often incurs violation of position constraints. Instead, RS coordinates offer dramatic improvement to the editing process even for constraints involving large offsets. We first convert the initial sequence \bar{u} into its RS vector \bar{y} containing the RS coordinates \bar{y}_e of each cubature tetrahedron $e \in \mathcal{T}_{\text{cub}}$ of the model through the sequence. We then optimize our optimal control formulation in Eq. (19) in reduced RS space, set the final RS coordinates to $y_e = \bar{y}_e + (Q(\widehat{W}S))_e z$, and finally reconstruct the displacement u as described in Section 3.

Fig. 5 shows a comparison of our methods versus [Barbič et al. 2012]. We use as input a steady animation in which each frame is the rest shape of a flower, with properly scaled uniform material parameters. The resulting sequence is constrained to have one of its nodes following a prescribed trajectory, and to match the rest pose at the beginning and the end of the sequence. As expected, our resulting animation (even without material optimization) displays a circular motion that matches the imposed spatial constraints (right column). For constraints that are quite near to the rest shape (top row), the method of [Barbič et al. 2012] with 200 modes is able to provide low distortion shapes through post-warping. Instead, our method with 30 modes matches even extreme constraints (bottom row) and induces a plausible and smooth motion.

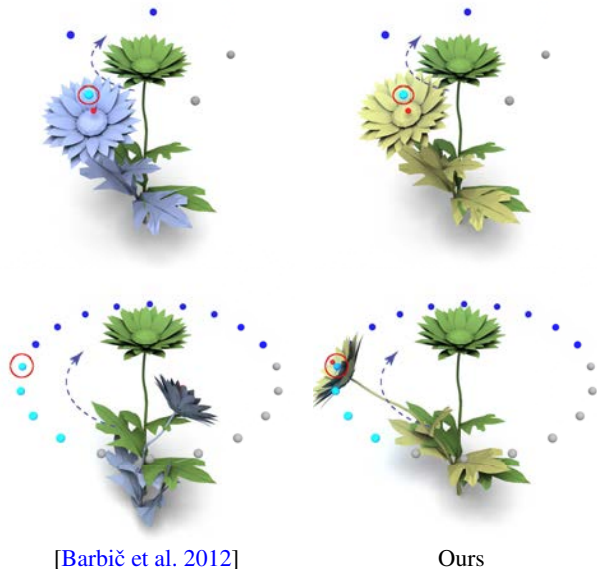


Figure 5: Animation editing for the animation sequence of a flower. The spatially-constrained node is in red (at the top of the flower), while the other markers indicate user-prescribed space-time constraints applied to this node. The rest position of the flower is shown, as well as its shape at the time corresponding to the spatial constraint associated with the circled light-blue marker. Top row: 5 constraints (uniformly distributed from frame 20 to 100) are applied to a 200-frame sequence. Bottom row: 20 constraints are applied from frame 30 to 125 in a 380-frame sequence.

For an input animation containing more complex motions, [Li et al. 2013] improves upon [Barbič et al. 2012] by linearizing the equation of motion around the input sequence. However, large amounts of distortion are visible if significant editing is performed, as in Fig. 6. In this example, we use an input sequence generated via

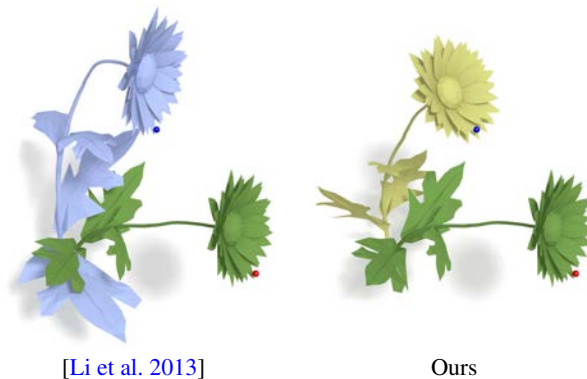
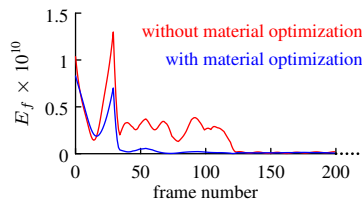


Figure 6: Comparison of our method to [Li et al. 2013]. A simulation sequence (in green) is edited by constraining the node in red to the blue marker. Our results preserve the shape better.

a full simulation of a flower using a Saint Venant-Kirchhoff model with external forces and uniform material assignment. We do not apply the material optimization in this result to demonstrate the advantage of solely using our reduced RS method. For this example, our method took 0.017s, while the approach of [Li et al. 2013] (using 30 modes too) took 0.19s to solve.

We provide an additional comparison in Fig. 7 for the flower model to highlight the importance of our material optimization approach. If material parameters induced from modal analysis are used, the control forces required to match the constraints are large (red curve in the inset). As a consequence, the flower abruptly changes behavior before and after the period of time containing position constraints. Material optimization is able to significantly reduce the magnitude of control forces (blue curve in the inset), and makes the flower keep its circular motion for a while before restoring the rest shape, which is expected due to angular momentum.



Finally, we present two complex examples of animation editing (see resulting animations in the supplemental video). In Fig. 1, a T. rex model is simulated by lifting its feet up and down to create a base sequence. This sequence motion is then altered via 27 space-time constraints in order to make the dinosaur jump trying to catch a plane circling around it. In Fig. 14, we produce a skateboard animation for the dinosaur by applying both partial and keyframe space-time constraints.

6.2 Keyframe interpolation

Interpolation between poses of a mesh can also be achieved efficiently with our method. While one could simply constrain all vertices of the mesh at their keyframe positions to achieve such a special case of space-time constraints, this would incur a non-negligible computational cost. One could instead only constrain cubature sample tetrahedra to achieve pose interpolation at much reduced cost. We propose an even faster approach by directly transforming, via \widehat{W} , the key poses $\{\bar{u}_i\}$ into their modal RS coordinates $\{\bar{z}_i\}$: given a keyframe \bar{u}_i , we first evaluate the RS coordinates \bar{y}_i^c of all the cubature tetrahedra, then compute $\bar{z}_i = ((Q\widehat{W})^c)^\dagger \bar{y}_i^c$ where $(\cdot)^\dagger$ denotes the pseudo-inverse operator. Our optimal control approach can now use a RS version E_c^{RS} of the penalty energy

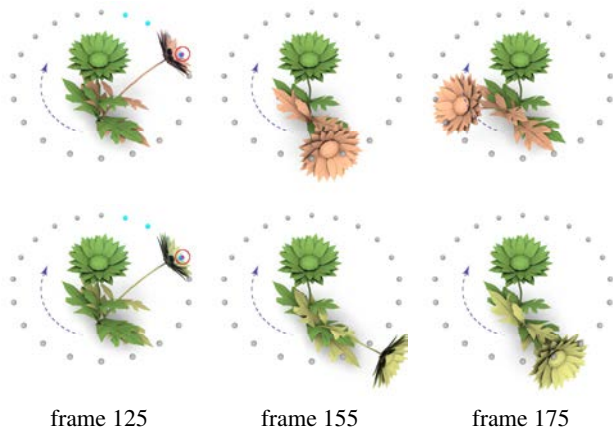


Figure 7: Path control through position constraints, with (top) and without (bottom) material optimization. The constrained node is in red, frame 125 has a constraint indicated by the circled blue dot, while the other two frames are free of constraints.

in Eq. (17b) to reduce complexity, with

$$E_c^{\text{RS}}(S, z) = \frac{1}{2} \sum_i \|S z_i - \bar{z}_i\|_2^2. \quad (23)$$

This interpolation in RS space is similar to [Huang et al. 2011], with the important difference that bases are now optimized via S .

We demonstrate the efficiency of this method in Fig. 8 for a bird mesh. By setting three poses as constraints (at the 10th, 20th and 30th frames) and fixing the first two and last two frames to be the bird’s rest shape (to enforce zero velocity at each end of the sequence), a smooth interpolation through these keyframes is created. Notice that our material optimization results in a symmetric flapping of the wings as expected. In contrast, no material optimization leads to spurious, asymmetric artifacts due to the fact that the volumetric embedding mesh is not exactly symmetric.

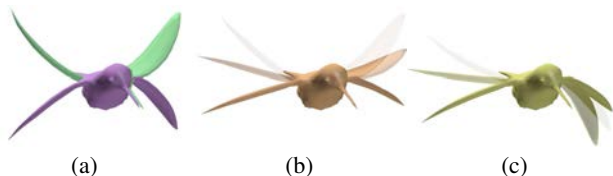


Figure 8: (a) Key poses are selected at frames 10 and 30 (same purple pose) and 20 (green pose). (b) Our approach with material optimization results in the expected flapping of the wings. (c) Instead, the same method without material optimization develops asymmetric behavior.

6.3 Recovering material parameters

For a given sequence of shape deformation, our material optimization method is also able to recover approximated elastic material parameters. We demonstrate this material parameter estimation in two sequences generated by the simulation of Saint Venant-Kirchhoff models with non-uniform material (Fig. 9). For the beam model, we fix one of its ends and apply external forces at the first 30 frames to make it vibrate up and down. For the dinosaur example, we fix the feet and start the simulations with a deformed shape, zero initial velocity, and no external forces. Note that our material parameter recovery can also be applied to captured data.

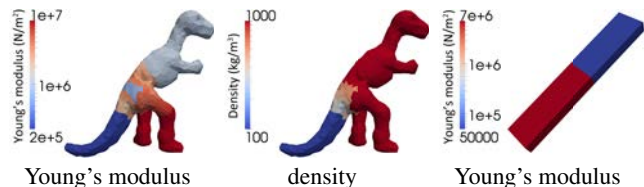


Figure 9: Pseudocolors indicating non-uniform material parameters used in our experiments. The Poisson’s ratio is constant for both models ($\nu = 0.45$), and the density of the beam model is uniform ($\rho = 100 \text{ kg/m}^3$).

In both examples, we use the first $\mathcal{N} = 150$ frames (out of 300) in our material optimization. We first perform modal RS analysis as in Section 2.3 to map the input sequence $\{\bar{u}(i)\}_{i=1 \dots \mathcal{N}}$ into initial modal RS coordinates $\{\bar{z}_k(i)\}$ for mode k at frame i with a generic material of constant Young’s modulus $E = 1 \text{ N/m}^2$, Poisson’s ratio $\nu = 0.45$, and mass $\rho = 1 \text{ kg/m}^3$. We then use frequency analysis to initialize our material optimization: we convert the sequence of modal RS coordinates $\bar{z}_k(i)$ for each mode k into its frequency spectrum $\tilde{z}_k(p)$ via FFT, where p is the frequency index, evaluate the significance of mode k by computing the energy of its spectrum

$$\eta_k = \sum_{p=1}^{\mathcal{N}} \tilde{z}_k(p)^2,$$

and then compute the most significant natural frequency from the maximum of the spectrum

$$\xi_k = 2\pi (\text{argmax}_p \{\tilde{z}_k(p)\} - 1) / \mathcal{N}.$$

We select the most significant dynamic mode in the input sequence $m = \text{argmax}_k \{\eta_k\}$, and finally scale the eigenvalues of the initial matrix Λ to enforce $\lambda_m = \xi_m^2$ before using them as the initial value $\Lambda^{(0)}$ of our material optimization.

To evaluate the results of our material parameter recovery, we use the 150th frame of the input sequence as initial condition, and simulate the animation with our reduced RS method and compare it with the rest of the input simulation. As shown in Fig. 10, using the material properties recovered from our method, the simulation result matches the remainder of the initial sequence better than using the initial matrices $\Lambda = \Lambda^{(0)}$, $D = D^{(0)}$, $S = S^{(0)}$.



Figure 10: Our material optimization can extrapolate the dynamics of an input sequence: from a 300-frame physically-based simulation, the last frame (left) of the sequence is well approximated (center) if material optimization is performed using only the first half of the original sequence and the resulting material bases in reduced RS space are used to simulate the second half of the animation. Without material optimization, the non-uniformity of Young’s modulus in this model leads to erroneous dynamics as evidenced by the last frame of an extrapolated sequence (right).

For the beam model with non-uniform material properties, the modal RS coordinates show rapid changes of frequency and damping if no material optimization is performed, since the control forces steer the object towards matching the constraints in space and time (Fig. 11). Simply optimizing Λ and D is unsatisfactory as well, especially when a small number of the modes are used (top row in Fig. 12). However, material optimization dramatically improves the results, demonstrating that the added degrees of freedom offered by S are successfully leveraged by our solver.

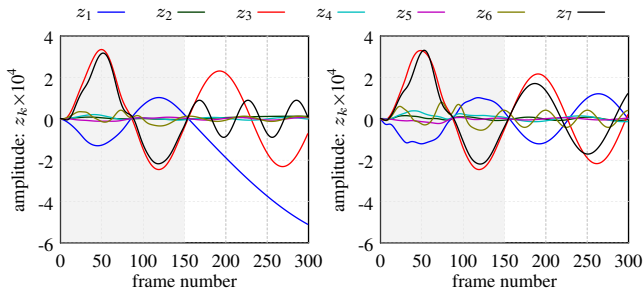


Figure 11: Without material optimization (left), the behavior of the beam shown in Fig. 12 using reduced coordinates $z_k(t)$ ($k = 1 \dots 7$ in this plot) changes drastically as soon as we remove all constraints (frame 150 and beyond). Results with material optimization (right) produce frequency and damping quite similar to the real simulation.

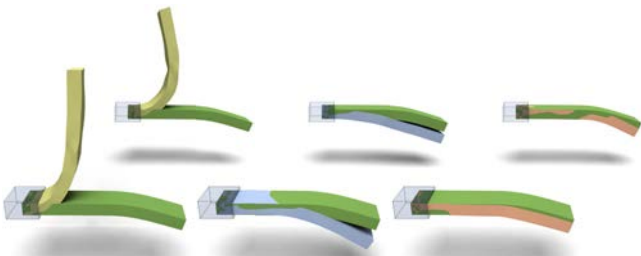


Figure 12: From left to right: material parameter recovery by optimizing only (z), or (z, Λ, D), or the complete set (z, S, Λ, D). The top row uses only 2 modes, while the bottom row uses 10 modes. The green bar represents the ground truth shape.

Finally, Fig. 13 provides another insight on how crucial our optimization of S to select bases from \tilde{W} is to minimizing control forces and improving position constraint enforcement. Note that if a larger number of modes is allowed, the benefit of our sampling matrix S becomes less pronounced, because modes may now become redundant as they can represent similar stiffness and damping coefficients (Fig. 11).

6.4 Performance

We demonstrate the efficiency of our approach by providing the convergence plots of our optimization for the flower and dinosaur examples in Fig. 15. In these plots, we count as one inner iteration each step of Newton’s method (for z), each solve of a quadratic optimization for Λ and D , and each iteration of L-BFGS for S . An outer iteration is the sequence of inner iterations corresponding to one round of optimization for z, Λ, D and S . The total energy in our optimization converges quickly and reliably in both examples as we perform our iterative method described in Section 5. These two examples are representative of all the examples we tried.

Performance statistics for editing with material optimization can

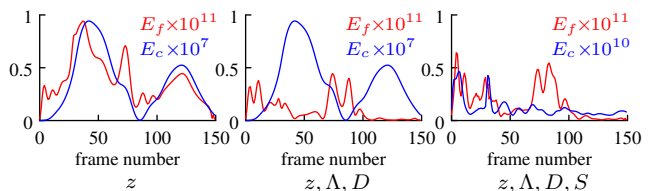


Figure 13: Optimizing the material in the bar experiment in Fig. 12 dramatically reduces the energies measuring the strength of control forces (E_f , in red) and the error in position constraints (E_c , in blue). Values are scaled as indicated by the graph legend to improve plot legibility. With the optimization of S , the error in constraint enforcement drops by three orders of magnitude when using only two modes.

also be found in Table 1. The most costly part of our method comes from optimizing z and S , and it heavily depends on the number of constraints. In the current CPU implementation, we can solve the optimization in a matter of seconds, even for relatively complex constraints. When we perform interactive editing, we use the current configuration as initial values for the next round of optimization, achieving approximately 5 frames per second.

7 Conclusion

In this paper, we introduced a new approach for material optimization that leads to reliable animation editing with space-time constraints. We also presented a novel method to reconstruct 3D shapes from RS coordinates based on geometric reduction and cubature, which improves performance compared to previous work by over two orders of magnitude. We demonstrated the robustness of our tools in various applications, such as pose interpolation, animation editing, and material parameter recovery.

The main limitation of our approach comes from the non-linear nature of its formulation. While our material optimization improves considerably the range of possible editing compared to previous methods, it may lead to local minima, especially when constraints are significantly different from the initial motion. Similar to [Barbič et al. 2012], our implementation does not support collision detection and response. As future work, we are currently investigating extensions to our material optimization scheme for the case of elasto-plasticity, in which damping and stiffness change throughout the sequence. Further improvements to our numerical solver and our material recovery are also of interest. Finally, it would be interesting to use our resulting optimized material bases to recover actual local material parameters (Poisson’s ratio, Young’s modulus, and density) of real objects based on 3D reconstructed sequences.

Acknowledgments

The authors wish to thank Katherine Breeden for her comments and corrections. Jin Huang and Hujun Bao were partially supported by NSFC (No.61170139, No.61210007). Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant nos. 61272298, 61328204). Fernando de Goes and Mathieu Desbrun were partially supported through NSF grant CCF-1011944 and a PhD Google Fellowship.

References

AN, S. S., KIM, T., AND JAMES, D. L. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM Transactions on Graphics* 27, 5, 165:1–165:10.



Figure 14: A skateboarding dinosaur sequence generated through material optimization, using both partial constraints and keyframe constraints.

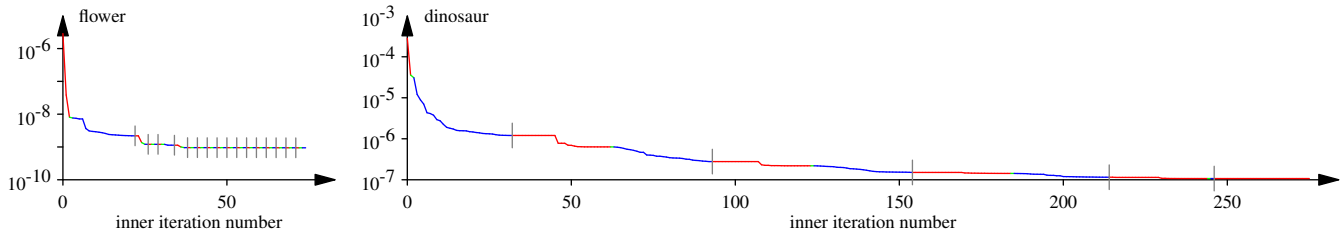


Figure 15: Convergence plots for the optimal control energy for both the flower example in the top row of Fig. 7 (left, using very few position constraints) and the dinosaur example in Fig. 10 (right, with a large number of pose constraints). The red parts of the curves represent the decrease of energy during the optimization of z , green during the optimization of Λ and D , and blue during the optimization of S . The gray ticks represent outer iterations. These examples are representative of all the other examples we tested.

BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics* 24, 3, 982–990.

BARBIČ, J., AND POPOVIĆ, J. 2008. Real-time control of physically based simulations using gentle forces. *ACM Transactions on Graphics* 27, 5, 163:1–163:10.

BARBIČ, J., DA SILVA, M., AND POPOVIĆ, J. 2009. Deformable object animation using reduced optimal control. *ACM Transactions on Graphics* 28, 3, 53:1–53:9.

BARBIČ, J., SIN, F., AND GRINSPUN, E. 2012. Interactive editing of deformable simulations. *ACM Transactions on Graphics* 31, 4, 70:1–70:8.

BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Transactions on Graphics* 28, 3, 89:1–89:9.

BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics* 29, 4, 63:1–63:10.

BOCHKANOV S. Alglib (www.alglib.net).

BYRD, R. H., LU, P., AND NOCEDAL, J. 1995. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* 16, 5, 1190–1208.

CHEN, Y., DAVIS, T. A., HAGER, W. W., AND RAJAMANICKAM, S. 2008. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.* 35, 3, 22:1–22:14.

CHOI, M. G., AND KO, H.-S. 2005. Modal warping: Real-time simulation of large rotational deformation and manipula-

tion. *IEEE Transactions on Visualization and Computer Graphics* 11, 1, 91–101.

COROS, S., MARTIN, S., THOMASZEWSKI, B., SCHUMACHER, C., SUMNER, R., AND GROSS, M. 2012. Deformable objects alive! *ACM Transactions on Graphics* 31, 4 (July), 69:1–69:9.

FANG, A. C., AND POLLARD, N. S. 2003. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics* 22, 3, 417–426.

GLEICHER, M. 1997. Motion editing with spacetime constraints. In *Symposium on Interactive 3D graphics*, 139–149.

GRASSIA, F. S. 1998. Practical parameterization of rotations using the exponential map. *J. Graph. Tools* 3, 3 (Mar.), 29–48.

HAUSER, K. K., SHEN, C., AND O'BRIEN, J. F. 2003. Interactive deformation using modal analysis with constraints. In *Proceedings of Graphics Interface*, vol. 3, 16–17.

HILDEBRANDT, K., SCHULZ, C., VON TYCOWICZ, C., AND POLTHIER, K. 2012. Interactive spacetime control of deformable objects. *ACM Transactions on Graphics* 31, 4, 71:1–71:8.

HUANG, J., TONG, Y., ZHOU, K., BAO, H., AND DESBRUN, M. 2011. Interactive shape interpolation through controllable dynamic deformation. *IEEE Transactions on Visualization and Computer Graphics* 17, 7, 983–992.

JEON, H., AND CHOI, M.-H. 2007. Interactive motion control of deformable objects using localized optimal control. In *Proceedings of ICRA*, 2582–2587.

KIM, Y., MACHIRAJU, R., AND THOMPSON, D. 2006. Path-based control of smoke simulations. In *Symposium on Computer Animation*, 33–42.

scene		#tet	#node	r	t_B	$t_{cub.}$	Train.	\mathcal{T}_{cub}	Cub. error	$ \mathcal{C} $	\mathcal{N}	#outer	#inner	time
Tyrannosaurus	Fig. 1	6658	2398	30	21.8	19.8	250	74	2.90%	27	350	3	490	3.040
flower	Fig. 7	5078	1828	30	6.7	16.1	300	80	1.45%	20	380	17	76	1.189
bird	Fig. 8	3113	1111	30	4.2	14.4	200	77	1.80%	3	150	12	797	1.052
beam 2 modes	Fig. 12	6144	1377	2	15.1	17.1	500	69	0.46%	150	150	23	796	0.379
beam 10 modes	Fig. 12	6144	1377	10	15.1	17.1	500	69	0.46%	150	150	9	425	0.796
dinosaur	Fig. 10	5249	1493	30	16.7	18.6	500	79	1.00%	150	150	6	277	3.340
dinosaur	Fig. 14	5249	1493	20	16.9	17.4	420	80	1.35%	112	620	7	123	1.731

Table 1: Performance (in seconds) measured on a PC with Intel CPU E7500, 2.93GHz, with 8 GB of memory. The number of columns of basis B ($= h$) and \tilde{W} ($= R$) are all 80. From left to right, number of tetrahedrons (#tet), nodes (#node), and modes (r), time to compute basis B using modal derivatives (t_B), time to find a cubature scheme ($t_{cub.}$), number of frames in training set (Train.), number of cubature samples (\mathcal{T}_{cub}), relative cubature error (Cub. error), number of constraints or keyframes ($|\mathcal{C}|$), number of frames in the optimization problem (\mathcal{N}), number of outer iterations (#outer), inner iterations (#inner) and total computational time for the optimization.

LI, S., HUANG, J., DESBRUN, M., AND JIN, X. 2013. Interactive elastic motion editing through spacetime position constraints. *Computer Animation and Virtual Worlds* 24, 3-4, 409–417.

MARTIN, S., THOMASZEWSKI, B., GRINSPUN, E., AND GROSS, M. 2011. Example-based elastic materials. *ACM Transactions on Graphics* 30, 4, 72:1–72:8.

NIELSEN, M. B., AND BRIDSON, R. 2011. Guide shapes for high resolution naturalistic liquid simulation. *ACM Transactions on Graphics* 30, 4, 83:1–83:8.

PENTLAND, A., AND WILLIAMS, J. 1989. Good vibrations: modal dynamics for graphics and animation. *Proceedings of ACM SIGGRAPH* 23, 3, 207–214.

POPOVIC, J., SEITZ, S., ERDMANN, M., POPOVIC, Z., AND WITKIN, A. 2000. Interactive manipulation of rigid body simulations. In *Proceedings of ACM SIGGRAPH*, 209 – 218.

SAFONOVA, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *Proceedings of ACM SIGGRAPH*, 514–521.

STANTON, M., SHENG, Y., WICKE, M., PERAZZI, F., YUEN, A., NARASIMHAN, S., AND TREUILLE, A. 2013. Non-polynomial galerkin projection on deforming meshes. *ACM Transactions on Graphics* 32, 4, 86:1–86:14.

TREUILLE, A., MCNAMARA, A., POPOVIĆ, Z., AND STAM, J. 2003. Keyframe control of smoke simulations. In *Proceedings of ACM SIGGRAPH*, 716–723.

TWIGG, C. D., AND JAMES, D. L. 2007. Many-worlds browsing for control of multibody dynamics. *ACM Transactions on Graphics* 26, 3, 14.

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

WITKIN, A., AND KASS, M. 1988. Spacetime constraints. In *Proceedings of ACM SIGGRAPH*, 159–168.

A Closed-form expression for Jacobian J_z

Suppose $u_i^j \in \mathbb{R}^3$ is the displacement of the constrained node j at frame i , and let B^j indicate the submatrix in B associated to j .

Based on the equations in Section 3, we can evaluate u_i^j as:

$$\begin{aligned} u_i^j &= B^j w^c P^c g^c(y_i^c) \\ &= \sum_{e \in \mathcal{T}_{cub}} \left(B^j w^c P^c \right)_e g_e^c((QW)^c z_i). \end{aligned}$$

The entry for the Jacobian matrix J_z corresponding to the derivative of u_i^j with respect to z_i is then computed as:

$$\frac{\partial u_i^j}{\partial z_i} = \sum_{e \in \mathcal{T}_{cub}} \left(B^j w^c P^c \right)_e \left(\frac{\partial g_e^c(y_i^c)}{\partial y_e} \right) (QW)^c,$$

where $\partial g_e / \partial y_e$ involves the derivative of the exponential map in Eq. (5), computed as proposed in [Grassia 1998].