

Example-Based Damping Design

HONGYI XU, University of Southern California

JERNEJ BARBIČ, University of Southern California

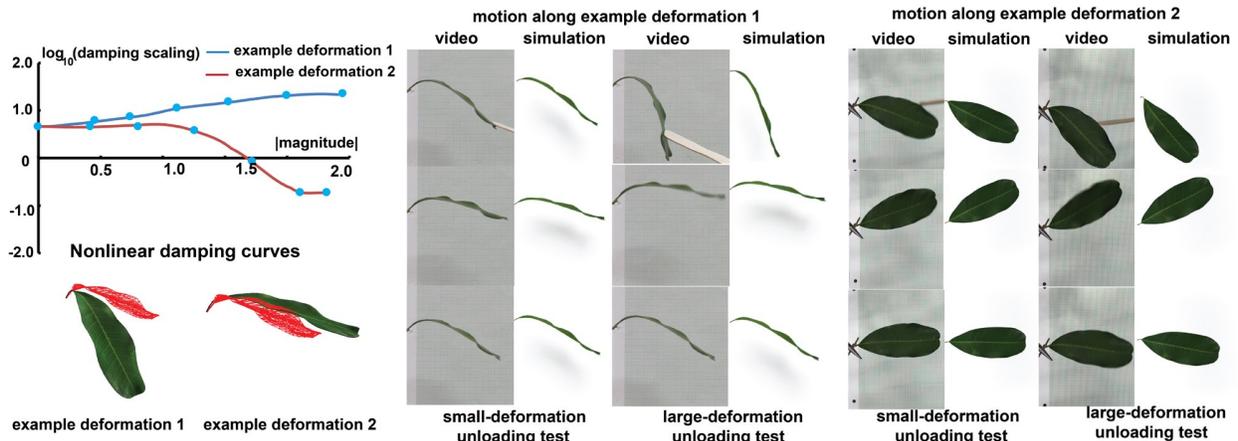


Fig. 1. **Nonlinear anisotropic damping that matches real-world leaf motion.** We experimentally estimated both the leaf elastic nonlinear material, and our nonlinear damping model, to match the real-world ground truth leaf motion. Our model accommodates anisotropic nonlinear damping, with different nonlinear damping properties for the leaf bending in the up-down (deformation 1) and left-right directions (deformation 2).

To date, material modeling in physically based computer animation has largely focused on mass and stiffness material properties. However, deformation dynamics is largely affected also by the damping properties. In this paper, we propose an interactive design method for nonlinear isotropic and anisotropic damping of complex three-dimensional solids simulated using the Finite Element Method (FEM). We first give a damping design method and interface whereby the user can set the damping properties so that motion aligned with each of a few chosen example deformations is damped by an independently prescribed amount, whereas the rest of the deformation space follows standard Rayleigh damping, or any viscous damping. Next, we demonstrate how to design nonlinear damping that depends on the magnitude of the deformation along each example deformation, by editing a single spline curve for each example deformation. Our user interface enables an art-directed and intuitive approach to controlling damping in solid simulations. We mathematically prove that our nonlinear anisotropic damping generalizes the frequency-dependent Caughey damping model, when starting from the Rayleigh damping. Finally, we give an inverse design method whereby the damping curve parameters can be inferred automatically from high-level user input, such as the amount of amplitude loss in one oscillation cycle along each of the chosen example deformations. To minimize numerical damping for implicit integration, we introduce an accurate and stable implicit integrator, which removes spurious high-frequency oscillations while only introducing a minimal amount of numerical damping. Our damping

can generate effects not possible with previous methods, such as controllable nonlinear decaying envelopes whereby large deformations are damped faster or slower than small deformations, and damping anisotropic effects. We also fit our damping to videos of real-world objects undergoing large deformations, capturing their nonlinear and anisotropic damping dynamics.

CCS Concepts: • **Computing methodologies** → **Physical simulation**;

Additional Key Words and Phrases: physically based simulation, damping, FEM, interactive, anisotropic, nonlinear, example-space

ACM Reference format:

Hongyi Xu and Jernej Barbic. 2017. Example-Based Damping Design. *ACM Trans. Graph.* 36, 4, Article 53 (July 2017), 14 pages.
DOI: <http://dx.doi.org/10.1145/3072959.3073631>

1 INTRODUCTION

Three-dimensional Finite Element Method (FEM) simulations are widely used in computer graphics, animation and related fields. The simulation behavior of a deformable object is uniquely determined by the mass properties, the underlying “stiffness” strain-stress material law, as well as the damping model that irreversibly dissipates the energy in a vibrating system and causes a decaying trend in the system response. Despite several previous papers to design and optimize the strain-stress relationships, we are not aware of any prior work on damping design for computer animation. Since the classic publication of “Theory of Sound” [Rayleigh 1896], characterization of the damping effects in a vibrating structure has long been an active research topic in the mechanical engineering field. However, there is still no single universally accepted damping model [Woodhouse 1998], due to the intricacies involved in understanding the

This research was sponsored in part by the National Science Foundation (CAREER-1055035, IIS-1422869), and USC Annenberg Graduate Fellowship to Hongyi Xu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 ACM. 0730-0301/2017/7-ART53 \$15.00
DOI: <http://dx.doi.org/10.1145/3072959.3073631>

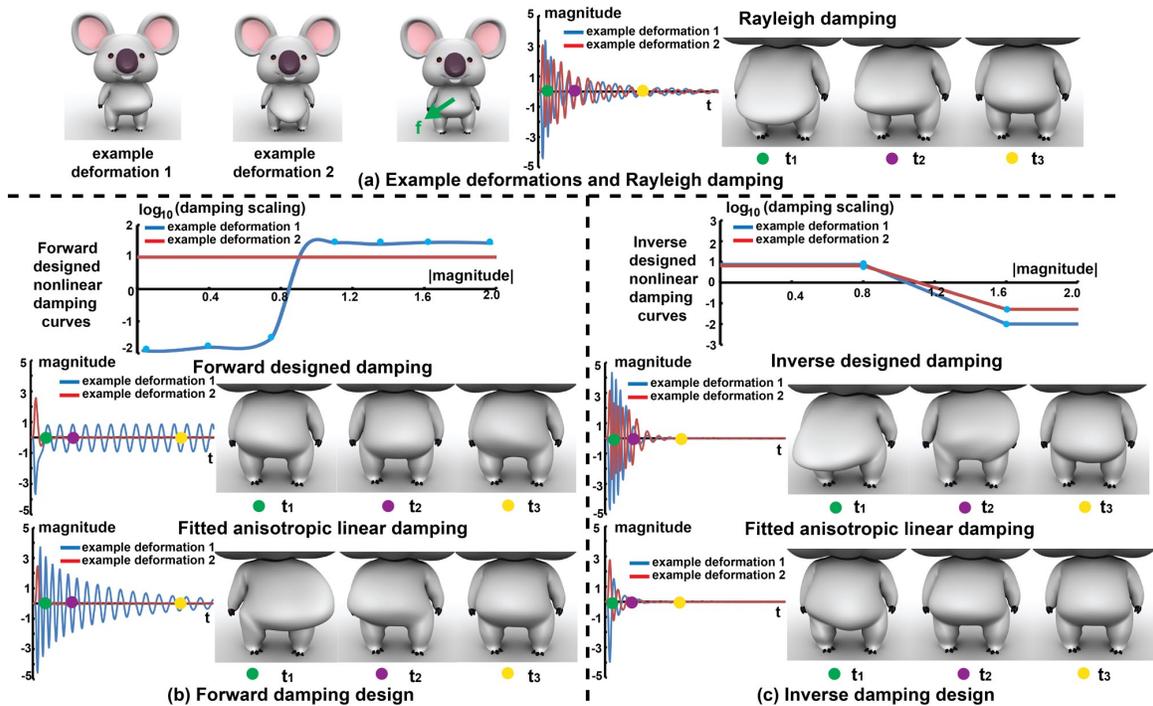


Fig. 2. **Our nonlinear damping produces decaying curves not achievable using linear damping.** (a) The two example belly deformations and the Rayleigh damping result, under the shown force. (b) Our method produces damping that is stronger under large deformations, but weakens under small deformations. Here, we use forward design, where the user designs nonlinear damping by manipulating spline control points. (c) Reverse of (b): weaker under large deformations and stronger under small deformations. Here, we use inverse design, where the user specifies the residual amplitude after one oscillation cycle of unloading the object from a prescribed initial amplitude; an optimizer finds the matching spline damping curve. In both (b) and (c), we show the best matching result under linear anisotropic damping. Such linear damping is not expressive enough to simulate the nonlinear effects.

state variables controlling the damping forces [Adhikari 2001]. Common practice in computer graphics is to use the linear viscous damping, in which the damping force linearly depends on the instantaneous velocities. Among them, Rayleigh damping [Rayleigh 1896], which assumes that the damping matrix is simply a linear combination of the mass and stiffness matrix, is probably the oldest but still most widely used form of viscous damping. In our design, the user starts with “default damping”, which is usually Rayleigh damping, but can be any viscous damping. We demonstrate how to then depart from it to design anisotropic and nonlinear damping effects, for computer animation purposes. In addition to design of artist-directed damping properties, our method can also be used to match the damping behavior of real systems, such as the leaf in Figure 1.

In Rayleigh damping, one controls the damping effects globally by tuning the mass and stiffness proportionality constants. However, different deformations typically present varying damping behaviors. A typical example are the frequency-dependent mode decay ratios observed in sound modal synthesis [Sterling and Lin 2016]. To create an animation, in artist endeavors, we typically envision desired dynamics on a few (low-frequency) primary vibrations. With global damping models, it can be cumbersome or impossible to tune the parameters to achieve the desired damping effects. Inspired by the example-based materials [Martin et al. 2011], our work

addresses this damping design challenge by introducing example-based anisotropic damping, enabling independent damping design for each example deformation. The user first prescribes a few example deformations to indicate the motions to design, and our method scales the default damping forces along the example deformations by the user-specified amount. We note that instead of designing element-wise damping properties, which often have unpredictable effects on global damping, our method is more artist-directed as it designs damping directly in the global shape space.

As a motivation for our nonlinear damping, Elliott et al. [2015] pointed out that the damping force function is “nonlinear in a number of mechanical systems and similar nonlinear damping behaviors is also seen in many electrical, biological and other dynamic systems.” In many cases, “it is the damping that is the dominant source of nonlinearity” [Elliott et al. 2015] and it is worthwhile to consider nonlinear damping models beyond the viscous linear damping. In our system, we approximate the nonlinear damping behaviors with damping scaling factors that depend nonlinearly on the magnitude of each example deformation. For each example deformation, the nonlinearity is modeled as a single 1-D spline curve and the artist can easily adjust the damping effects by editing the spline. We demonstrate that our nonlinear damping enables complex visual effects that cannot be achieved with standard linear viscous damping (Figure 2). In order to provide a high-level damping design interface, we also give a method for *inverse damping design*. In this

method, the user can tune the damping of each example deformation by prescribing an initial amplitude, and a residual amplitude after one oscillation cycle. Given a few such (initial, residual) amplitude pairs, we perform fast optimization to determine the 1-D nonlinear damping curve for this example deformation. Our method is inspired by “unloading tests” performed in mechanical engineering, and provides a direct user interface to specifying the damping.

Our damping matrix is non-negative definite and only dissipates energy, enabling stable integration under large timesteps. In our work, we choose to use implicit integration for numerical stability. Because our goal is to control damping, it is important to use a numerical integration scheme which does not introduce substantial numerical damping; hence, we do not use implicit backward Euler [Baraff and Witkin 1998]. Variational integrators [Marsden and West 2001] have good energy-preserving properties, and as we prefer implicit integration, we first considered perhaps the most widely used integrator in structural dynamics, the implicit Newmark (also called “Trapezoidal rule”) [Wriggers 2002], which has been shown to be variational in the Veselov sense [Kane et al. 2000]. However, implicit Newmark can suffer from stability problems, especially with low damping levels, large timesteps and nonlinear dynamics. To improve stability, we adopt a simple composite integration scheme that combines the lightly-dissipative trapezoidal rule with the strongly dissipative second-order implicit backward Euler process [Bathe 2007]. This method effectively maximizes high-frequency numerical dissipation to remove the spurious oscillations, without introducing excessive numerical damping in the important low-frequency region. To the best of our knowledge, this integration scheme has to date not been used for computer animation applications. It is second-order accurate and is significantly more stable than implicit Newmark under large deformations and timesteps, all the while introducing approximately equal amounts of artificial damping as implicit Newmark.

2 RELATED WORK

Damping causes energy to dissipate during mechanical vibrations of an elastic body. In computer graphics, various damping models have been adopted for physically-based simulation, such as strain-rate damping for solids [Debunne et al. 2001; O’Brien et al. 2002; O’Brien and Hodgins 1999], air drag and spring damping for cloth [Baraff and Witkin 1998; Bhat et al. 2003] and internal friction for modeling hysteresis effects [Lahey 2002; Miguel et al. 2013; Williams 2010]. However, among them, Rayleigh damping [Rayleigh 1896] is the oldest but still the most popular damping model for simulating a dynamic response of a structure, due to its simplicity and ease of implementation. Although our method generalizes viscous damping models, we start from the Rayleigh damping model in our system because this model is already familiar to the artists. The existing damping models control damping via a few global parameters. We decouple the damping control in the space of the user-prescribed example deformations and can design damping effects independently for each example deformation. In order to create desired vibration behaviors, several authors obtained the damping parameters by measuring real deformations or recording sound. Measured cloth deformation behaviors are reproduced via optimization of parameters for the spring damping [Bhat et al. 2003] and internal friction [Miguel

et al. 2013]. With linear modal analysis, modal decay rates are also obtained from multiple sound recordings [Lloyd et al. 2011; Pai et al. 2001] or through the space-time deformation constraints [Li et al. 2014]. For sound synthesis, Ren and colleagues [2013] extracted Rayleigh damping parameters from a single audio sample. Recently, the work was extended by [Sterling and Lin 2016] to fit the parameters of the Caughey damping [Caughey 1960; Caughey and O’Kelly 1965], which gives frequency-dependent damping ratios. To capture real-world deformations, Wang and her colleagues [2015] used heterogeneous corotational linear materials and optimized Rayleigh damping parameters. These papers obtained damping parameters from experimental data and did not aim to provide a method for easy user design, adjustment, or modification of damping. In our paper, we give a damping model that can be easily and independently tuned by the user using a small number of parameters. Furthermore, previous work only addressed linear damping, whereas we give a richer nonlinear damping model. In Figure 2, we demonstrate that this model cannot be matched using linear models such as [Sterling and Lin 2016]. Our damping model can also be used to fit viscoelastic behavior of real objects (dragon and leaf examples).

2.1 Example-Based Methods

Example-based methods have been employed for graphics applications, such as mesh posing [Sumner and Popović 2004], texture synthesis [Wei et al. 2009], sound synthesis [Ren et al. 2013], and deformations [Jones et al. 2016; Martin et al. 2011; Wang et al. 2010]. Different from these previous methods in which examples are used for parameter fitting or interpolation, we employ example deformations as our damping design space. More similar to our approach are example-based elastic materials [Martin et al. 2011; Schumacher et al. 2012], which employ examples of desired deformations to construct an anisotropic nonlinear elastic or plastic potential. This potential controlled elasticity, and not damping. While we draw inspiration from [Martin et al. 2011], we present an anisotropic nonlinear example-based *damping* model.

2.2 Material Design

FEM simulations are greatly influenced by the specific strain-stress material relationship, and the damping model. Since the pioneering work of deformable object simulation [Terzopoulos et al. 1987], there has been a lot of research on extending the expressive range of materials [Bargteil et al. 2007; Irving et al. 2004; Müller and Gross 2004], forward design of materials [Li and Barbič 2014; Martin et al. 2011; Miguel et al. 2016; Schumacher et al. 2012; Xu et al. 2015b] or material parameter optimizations [Becker and Teschner 2007; Bickel et al. 2009; Xu et al. 2015a]. However, few publications focused on the design of damping for physically-based simulation. To the best of our knowledge, previous methods only modeled linear damping, and we are the first work to present a design method for anisotropic and/or nonlinear damping effects. Different from [Li and Barbič 2014; Xu et al. 2015b] which needed to specify element-wise orientations for material anisotropy, our design of anisotropic damping is given in the space of example deformations and is therefore more artist-directable. Xu and colleagues [Xu et al. 2015b] presented a 1-D spline system to control stress-strain hyperelastic material laws. Similar to their work, we also employ 1-D splines, but use them to model

nonlinear *damping* dependence on shape deformation or velocity magnitude.

3 BACKGROUND: VISCOUS DAMPING

In solid mechanics, three-dimensional deformable objects are governed by the nonlinear partial differential equations of elasticity. After applying FEM, one obtains the equations of motion,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{f}_d(\mathbf{u}, \dot{\mathbf{u}}) + \mathbf{f}_{int}(\mathbf{u}) = \mathbf{f}_{ext}, \quad (1)$$

where $\mathbf{u} \in \mathbb{R}^{3n}$ is the displacement vector for the n mesh vertices, $\mathbf{M} \in \mathbb{R}^{3n \times 3n}$ is the mass matrix, and $\mathbf{f}_d, \mathbf{f}_{int}, \mathbf{f}_{ext} \in \mathbb{R}^{3n}$ are the damping, elastic internal and external forces, respectively. The gradient of the elastic internal forces $\mathbf{f}_{int}(\mathbf{u})$ with respect to \mathbf{u} , $\mathbf{K}(\mathbf{u}) = \partial \mathbf{f}_{int}(\mathbf{u}) / \partial \mathbf{u}$, is called the tangent stiffness matrix and is determined by the specific material law. *Viscous damping* is defined as damping where the force depends only on the current displacement \mathbf{u} and velocity $\dot{\mathbf{u}}$ [Adhikari 2001]. In non-viscous damping, the damping force depends also on the history of \mathbf{u} and $\dot{\mathbf{u}}$. In this paper, we only model viscous damping.

Viscous damping is achieved by the incorporation of Rayleigh's Dissipation Function [Rayleigh 1896] given as

$$\mathcal{F} = -\frac{1}{2} \sum_{i=1}^{3n} \sum_{j=1}^{3n} \mathbf{C}_{ij} \dot{\mathbf{u}}_i \dot{\mathbf{u}}_j = -\frac{1}{2} \dot{\mathbf{u}}^T \mathbf{C} \dot{\mathbf{u}}, \quad (2)$$

where $\mathbf{C} \in \mathbb{R}^{3n \times 3n}$ is a non-negative definite symmetric matrix, known as the viscous damping matrix. The non-conservative damping forces \mathbf{f}_d can be obtained as

$$\mathbf{f}_d = -\frac{\partial \mathcal{F}}{\partial \dot{\mathbf{u}}} = \mathbf{C} \dot{\mathbf{u}}. \quad (3)$$

The damping matrix \mathbf{C} is not necessarily constant and can depend on the deformations and velocities. The most common approach to forming \mathbf{C} is Rayleigh damping, whereby one uses linear combinations of the mass matrix \mathbf{M} and stiffness matrix \mathbf{K} , as $\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K}(\mathbf{u})$. Scalar $\alpha \geq 0$ sets the level of mass damping which slows down global deformations, and $\beta \geq 0$ sets the stiffness damping, useful to remove high-frequency instabilities. In this paper, the term *non-negative definite matrix* refers to any square matrix \mathbf{A} with the property that $\mathbf{y}^T \mathbf{A} \mathbf{y} \geq 0$ for any vector \mathbf{y} . Note that \mathbf{A} need not be symmetric for this definition. Because $\mathbf{y}^T \mathbf{A} \mathbf{y} = \mathbf{y}^T (\mathbf{A} + \mathbf{A}^T) \mathbf{y} / 2$, \mathbf{A} is non-negative definite if and only if the symmetric matrix $(\mathbf{A} + \mathbf{A}^T) / 2$ is semi-positive definite in the usual sense.

4 DAMPING DESIGN

In this section, we will show how to edit the viscous damping matrix for anisotropic and nonlinear damping effects and demonstrate the properties of our model when starting from Rayleigh damping.

4.1 Example-Based Damping

With global damping models, the damping behaviors along various example deformations are tightly coupled by the globally-tuned damping parameters. The damping parameters constitute the degrees of freedom of the corresponding damping model space. We note that for an animator, the primary concern is often the behavior of a few (typically low-frequency) prominent deformations whereas

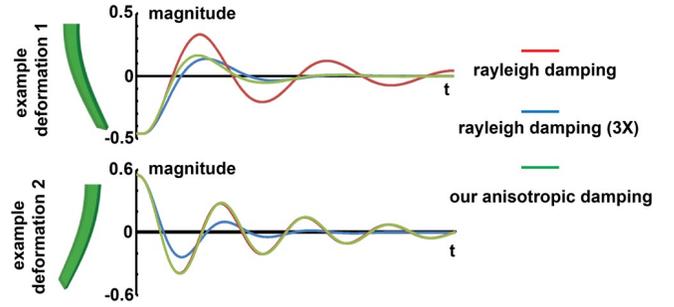


Fig. 3. **Illustration of example-based anisotropic damping.** Full simulation with St.Venant-Kirchhoff material. We start from Rayleigh damping (red) and design our anisotropic damping (green) where the motion along the first example deformation is 3× stronger, whereas the damping in other directions is left unmodified (the red and green curve are on top of each other for the second example deformation). Globally tuning the Rayleigh damping parameters to be 3× larger will affect motion along all the directions (blue).

for the rest of the deformation space, a natural but fast vibration decay is usually most desirable. However, with the existing global damping models, it is tedious and difficult to tune the global damping parameters to control the damping effects along each deformation shape of interest. In our work, we give artist the local control of damping effects with our anisotropic damping matrix.

Equipped with some default viscous damping force $\mathbf{C} \dot{\mathbf{u}}$, our system first lets the artist provide $k \geq 1$ example deformations $\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_k] \in \mathbb{R}^{3n \times k}$ to indicate the linear motions that are going to be designed with specific damping effects. The example deformations can be selected using any suitable method, such as via shape editing, modal analysis or quasi-static physical simulation. Then, we replace the component of the default damping force $\mathbf{f}_d^i(\dot{\mathbf{u}})$ that aligns with the example deformation \mathbf{x}_i with a scaled damping force $\hat{\mathbf{f}}_d^i(\dot{\mathbf{u}}) = \gamma_i \mathbf{f}_d^i(\dot{\mathbf{u}})$, and leave the orthogonal component unmodified. The scalar $\gamma_i \geq 0$ enables the artist to attenuate ($0 \leq \gamma_i < 1$), or strengthen ($\gamma_i > 1$) the default damping along the example deformation \mathbf{x}_i . Here, we first assume that all the example deformations are \mathbf{C} -orthonormalized (Appendix A) to each other, i.e., $\mathbf{x}_j^T \mathbf{C} \mathbf{x}_i = 0$, if $i \neq j$, and $\mathbf{x}_i^T \mathbf{C} \mathbf{x}_i = 1$. We will discuss the general case later. We achieve our designed damping effects by modifying the damping force to

$$\hat{\mathbf{f}}_d(\dot{\mathbf{u}}) = \mathbf{C} \dot{\mathbf{u}} + \sum_{i=1}^k (\hat{\mathbf{f}}_d^i(\dot{\mathbf{u}}) - \mathbf{f}_d^i(\dot{\mathbf{u}})), \quad (4)$$

where $\mathbf{f}_d^i(\dot{\mathbf{u}}) = \mathbf{C} \mathcal{P}_{\mathbf{x}_i}(\dot{\mathbf{u}})$, and $\mathcal{P}_{\mathbf{x}_i}(\dot{\mathbf{u}}) = (\mathbf{x}_i^T \mathbf{C} \dot{\mathbf{u}}) \mathbf{x}_i$ is the \mathbf{C} -weighted projection of the velocity $\dot{\mathbf{u}}$ to the example deformation \mathbf{x}_i . We note that when the velocity $\dot{\mathbf{u}}$ is \mathbf{C} -orthogonal to all the example deformations, $\mathcal{P}_{\mathbf{x}_i}(\dot{\mathbf{u}})$ are all $\mathbf{0}$, and therefore the damping force is unmodified. However, when $\dot{\mathbf{u}}$ is aligned with \mathbf{x}_i , the damping force along \mathbf{x}_i is scaled with γ_i , i.e.,

$$\hat{\mathbf{f}}_d(\mathbf{x}_i) = \gamma_i \mathbf{f}_d(\mathbf{x}_i) = \gamma_i \mathbf{C} \mathbf{x}_i. \quad (5)$$

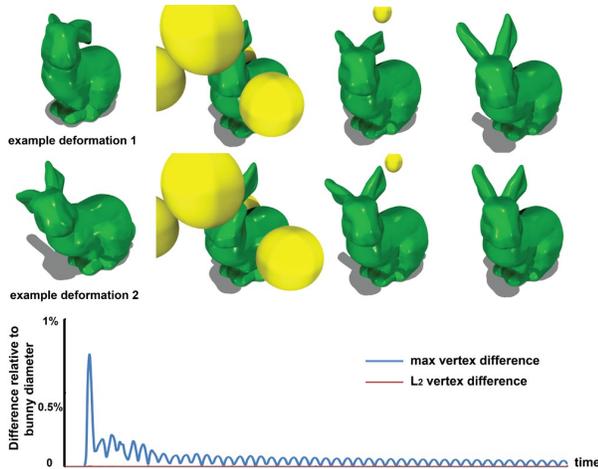


Fig. 4. **Anisotropic damping control in the presence of collisions.** The first and second row scale the damping $0.1\times$ compared to Rayleigh damping, along the forward-backward motion (example deformation 1) and left-right motion (example deformation 2), respectively. The rest of the deformation space uses unmodified Rayleigh damping. Under collision impact from the 4 balls, our anisotropic damping achieves much less damping along the example deformations. The bottom rows evaluates the visual difference between using non-negative definite Π (which provably dissipates energy) and $\bar{\Pi} = \mathbf{R}\mathbf{\Gamma}\mathbf{R}^{-1}$. It can be seen that the maximum vertex position difference is small: under 1% of the bunny’s diameter.

By employing $(b^T c)a = (a \otimes b)c$, Equation 4 is transformed into

$$\begin{aligned} \hat{\mathbf{f}}_d(\dot{\mathbf{u}}) &= \left(\mathbf{C} + \sum_{i=1}^k ((\gamma_i - 1)\mathbf{C}\mathbf{x}_i) \otimes (\mathbf{C}\mathbf{x}_i) \right) \dot{\mathbf{u}} = \\ &= \left(\mathbf{C} + \mathbf{C}\mathbf{X}(\mathbf{\Gamma} - \mathbf{I})(\mathbf{C}\mathbf{X})^T \right) \dot{\mathbf{u}}, \end{aligned} \quad (6)$$

where $\mathbf{\Gamma} = \text{diag}[\gamma_1, \gamma_2, \dots, \gamma_k]$. We can obtain the same formula as in Equation 4 by modifying the Rayleigh’s Dissipation Function as

$$\hat{\mathcal{F}} = -\frac{1}{2} \dot{\mathbf{u}}^T \mathbf{C} \dot{\mathbf{u}} - \frac{1}{2} \sum_{i=1}^k (\gamma_i - 1) \dot{\mathbf{u}}^T \mathbf{C} \mathcal{P}_{\mathbf{x}_i}(\dot{\mathbf{u}}), \quad (7)$$

which physically means scaling the energy dissipation along the example deformation \mathbf{x}_i by γ_i , while keeping the \mathbf{C} -orthogonal part unmodified. We note that we initially tried using mass-orthonormalized example deformations, where the modified damping matrix is $\mathbf{C} + \mathbf{M}\mathbf{X}(\mathbf{\Gamma} - \mathbf{I})(\mathbf{C}\mathbf{X})^T$. However, this damping matrix is not non-negative definite, which could lead to simulation instabilities with large or small damping factors γ . Such damping also tightly couples example deformations since the damping force along example deformation $\hat{\mathbf{f}}_d^i$ will also dissipate energy along the other example deformations. In contrast, our damping control is independent for each \mathbf{C} -orthogonal example deformation (Figure 3) since we have $\mathbf{x}_j^T \hat{\mathbf{f}}_d^i = 0$, if $i \neq j$.

Equation 6 assumes that the example deformations \mathbf{x}_i are already \mathbf{C} -orthogonal. Suppose the artist wants to design damping for non-orthogonal example deformations. We can express the artist’s intent via (yet unknown, to be determined) matrix $\hat{\mathbf{C}}$ such that the damping force for velocity \mathbf{x}_i equals $\hat{\mathbf{C}}\mathbf{x}_i = \gamma_i\mathbf{C}\mathbf{x}_i$, for $i = 1, \dots, k$, or, in matrix notation, $\hat{\mathbf{C}}\mathbf{X} = \mathbf{C}\mathbf{X}\mathbf{\Gamma}$. This property means that for velocities aligned with \mathbf{x}_i , the damping force is scaled by $\gamma_i \geq 0$, paralleling

Equation 5, except that \mathbf{x}_i are now non-orthogonal. We perform \mathbf{C} -weighted QR decomposition $\mathbf{X} = \mathbf{Q}\mathbf{R}$ (Appendix A), where $\mathbf{Q} = \{\mathbf{q}_1 | \mathbf{q}_2 | \dots | \mathbf{q}_k\} \in \mathbb{R}^{3n \times k}$ are \mathbf{C} -orthonormalized deformations, and $\mathbf{R} \in \mathbb{R}^{k \times k}$ is upper-triangular. Therefore, $\hat{\mathbf{C}}$ must satisfy

$$\hat{\mathbf{C}}\mathbf{Q} = \hat{\mathbf{C}}\mathbf{X}\mathbf{R}^{-1} = \mathbf{C}\mathbf{X}\mathbf{\Gamma}\mathbf{R}^{-1} = \mathbf{C}\mathbf{Q}\bar{\mathbf{\Pi}}, \quad \text{for } \bar{\mathbf{\Pi}} = \mathbf{R}\mathbf{\Gamma}\mathbf{R}^{-1}. \quad (8)$$

Note the similarity between Equations 5 and 8: both prescribe the damping on a set of \mathbf{C} -orthonormalized deformations, except that the damping in Equation 5 is independent for the different \mathbf{C} -orthonormalized example deformations, whereas in Equation 8, the coupling matrix $\bar{\mathbf{\Pi}} \in \mathbb{R}^{k \times k}$ is dense. In order to guarantee the dissipation of energy (as proven in the next paragraph), the damping coupling matrix must be non-negative definite. Because we have $\bar{\mathbf{\Pi}}\mathbf{R} = \mathbf{R}\mathbf{\Gamma}$, the eigenvalues of $\bar{\mathbf{\Pi}}$ equal γ_i , $i = 1, \dots, k$, and are therefore non-negative. However, as $\bar{\mathbf{\Pi}}$ is not symmetric, $\bar{\mathbf{\Pi}}$ could still fail to be non-negative definite, although we did not commonly encounter this case in practice. Intuitively, this case corresponds to the artists prescribing inconsistent damping scaling factors, such as prescribing very different scaling factors γ_i for two nearly identical non-orthogonal directions \mathbf{x}_i . In the spirit of positive-definite enforcement for elastic stiffness matrices [Teran et al. 2005], we can enforce non-negative definiteness by replacing $\bar{\mathbf{\Pi}}$ for a non-negative definite matrix $\mathbf{\Pi} = \bar{\mathbf{\Pi}} - \sum_{i=1}^p \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i$, where λ_i are the negative eigenvalues of $(\bar{\mathbf{\Pi}} + \bar{\mathbf{\Pi}}^T)/2$, and \mathbf{v}_i are their respective eigenvectors (proof is in Appendix B). We are now ready to modify the damping force. Similarly as in Equation 4, we replace the default damping force $\mathbf{f}_d^i(\dot{\mathbf{u}})$ with $\mathbf{\Pi}$ -weighted forces $\mathbf{f}_d^j(\dot{\mathbf{u}})$,

$$\begin{aligned} \hat{\mathbf{f}}_d(\dot{\mathbf{u}}) &= \mathbf{C}\dot{\mathbf{u}} + \sum_{i=1}^k \left(\sum_{j=1}^k (\mathbf{\Pi}_{ji} \mathbf{f}_d^j(\dot{\mathbf{u}})) - \mathbf{f}_d^i(\dot{\mathbf{u}}) \right) = \\ &= \mathbf{C}\dot{\mathbf{u}} + \left(\sum_{i=1}^k \sum_{j=1}^k (\mathbf{\Pi} - \mathbf{I})_{ji} \mathbf{C}\mathbf{q}_j \otimes \mathbf{C}\mathbf{q}_i \right) \dot{\mathbf{u}} = \\ &= \left(\mathbf{C} + \mathbf{C}\mathbf{Q}(\mathbf{\Pi} - \mathbf{I})(\mathbf{C}\mathbf{Q})^T \right) \dot{\mathbf{u}} = \underbrace{(\mathbf{C} + \mathbf{U}(\mathbf{\Pi} - \mathbf{I})\mathbf{U}^T)}_{\hat{\mathbf{C}}} \dot{\mathbf{u}}, \end{aligned} \quad (9)$$

where we define $\mathbf{U} = \mathbf{C}\mathbf{Q}$. For consistent artist input $\mathbf{\Pi} = \bar{\mathbf{\Pi}}$, matrix $\hat{\mathbf{C}}$ as defined in Equation 9 produces the desired damping on example deformations $\mathbf{X} = \mathbf{Q}\mathbf{R}$ (note that $\mathbf{Q}^T \mathbf{C}\mathbf{Q} = \mathbf{I}$):

$$\begin{aligned} \hat{\mathbf{C}}\mathbf{X} &= (\mathbf{C} + \mathbf{C}\mathbf{Q}(\mathbf{\Pi} - \mathbf{I})\mathbf{Q}^T \mathbf{C})\mathbf{X} = \mathbf{C}\mathbf{X} + \mathbf{C}\mathbf{Q}\mathbf{R}\mathbf{\Gamma}\mathbf{R}^{-1}\mathbf{Q}^T \mathbf{C}\mathbf{Q}\mathbf{R} - \\ &= \mathbf{C}\mathbf{Q}\mathbf{Q}^T \mathbf{C}\mathbf{Q}\mathbf{R} = \mathbf{C}\mathbf{X} + \mathbf{C}\mathbf{X}\mathbf{\Gamma} - \mathbf{C}\mathbf{X} = \mathbf{C}\mathbf{X}\mathbf{\Gamma}. \end{aligned} \quad (10)$$

Our example-based damping model can start with any viscous damping matrix \mathbf{C} , and preserves the default damping properties for dynamics orthogonal to the example deformations. If spatially-heterogeneous damping effects independent of the example deformations are desired, we can simply set heterogeneous damping parameters for the default damping matrix \mathbf{C} . Our method will preserve the spatial heterogeneity of damping. If one would like to design damping just for deformations in a local region, we can achieve that by selecting example deformations that have no displacement outside the local region. Note that the damping of any global deformations that are \mathbf{C} -orthogonal to the example local deformations won’t be affected by such design.

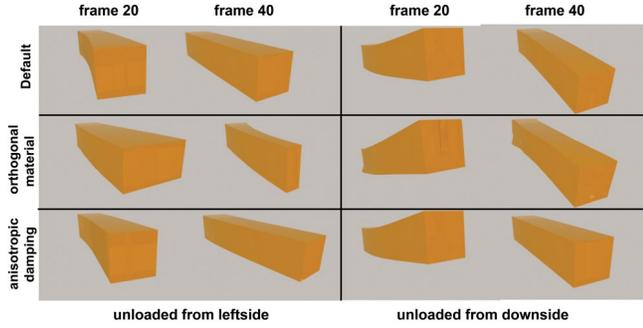


Fig. 5. **Anisotropic damping cannot be mimicked with anisotropic elastic forces.** The left and right parts of the image both show simulation frames computed under three methods (the rows): isotropic material and damping, anisotropic (orthotropic) elasticity [Li and Barbic 2014] but isotropic damping, and isotropic elasticity but anisotropic damping (our method). The user’s intention is to generate a damping effect where the left-right and up-down beam deflections are damped 0.1× and 5× more than the default Rayleigh damping, respectively. In order to produce the effect with orthotropic elasticity, the user made the material softer in the left-right direction, while making up-down stiffer. However, this drastically changes the object frequency, the resulting shapes and their dynamics.

4.1.1 Dissipation of energy. Our damping model ensures energy dissipation. For damping to dissipate energy, its power $\dot{\mathbf{u}}^T \hat{\mathbf{C}} \dot{\mathbf{u}}$ must be non-negative at all times. Note that the sign of power here adheres to the usual convention of placing the damping to the left-hand side of Equation 1. Therefore, energy dissipation is equivalent to non-negative definiteness of $\hat{\mathbf{C}}$. Our anisotropic damping matrix $\hat{\mathbf{C}}$ is not necessarily symmetric under non-orthogonal example deformations. However, much like the original damping matrix \mathbf{C} , it is non-negative definite, mathematically ensuring that our damping always dissipates energy. This can be seen as follows. Given that the deformations \mathbf{q}_i are \mathbf{C} -orthonormalized, we can split any nonzero vector $\mathbf{y} \in \mathbb{R}^{3n}$ into two \mathbf{C} -orthogonal parts, $\mathbf{z} = \mathbf{y} - \bar{\mathbf{z}}$, for $\bar{\mathbf{z}} = \sum_{i=1}^k \eta_i \mathbf{q}_i$, where $\mathbf{z}^T \mathbf{U} = 0$ and $\mathbf{z}^T \mathbf{C} \bar{\mathbf{z}} = 0$. Therefore,

$$\begin{aligned} \mathbf{y}^T \hat{\mathbf{C}} \mathbf{y} &= (\mathbf{z} + \bar{\mathbf{z}})^T (\mathbf{C} - \mathbf{U} \mathbf{U}^T + \mathbf{U} \mathbf{\Pi} \mathbf{U}^T) (\mathbf{z} + \bar{\mathbf{z}}) = \\ &= \mathbf{z}^T \mathbf{C} \mathbf{z} + (\bar{\mathbf{z}}^T \mathbf{U}) \mathbf{\Pi} (\mathbf{U}^T \bar{\mathbf{z}}) + \bar{\mathbf{z}}^T \mathbf{C} \bar{\mathbf{z}} - \bar{\mathbf{z}}^T \mathbf{U} \mathbf{U}^T \bar{\mathbf{z}} = \\ &= \mathbf{z}^T \mathbf{C} \mathbf{z} + (\bar{\mathbf{z}}^T \mathbf{U}) \mathbf{\Pi} (\mathbf{U}^T \bar{\mathbf{z}}), \quad \text{as } \bar{\mathbf{z}}^T \mathbf{C} \bar{\mathbf{z}} = \bar{\mathbf{z}}^T \mathbf{U} \mathbf{U}^T \bar{\mathbf{z}} = \sum_{i=1}^k \eta_i^2. \end{aligned} \quad (11)$$

Because both \mathbf{C} and $\mathbf{\Pi}$ are non-negative definite, it follows that $\mathbf{y}^T \hat{\mathbf{C}} \mathbf{y} \geq 0$, i.e., $\hat{\mathbf{C}}$ is non-negative definite. Because \mathbf{C} is symmetric and because we use the Woodbury formula to solve the linear system at each timestep (Section 6.1), the non-symmetry of $\hat{\mathbf{C}}$ does not affect simulation stability. However, if needed, one can achieve matrix symmetry by replacing $\mathbf{\Pi}$ by $(\mathbf{\Pi} + \mathbf{\Pi}^T)/2$. The non-negative definite property of our damping matrix enables stable integration even with large timesteps and with contact (Figure 4). We note that we performed many simulations also directly with matrix $\bar{\mathbf{\Pi}}$, as opposed to $\mathbf{\Pi}$. While $\bar{\mathbf{\Pi}}$ does not offer a theoretical guarantee of damping energy dissipation, the visual output difference between $\bar{\mathbf{\Pi}}$ and the theoretically stable $\mathbf{\Pi}$ was small in our examples (Figure 4, bottom).

4.1.2 Generalization of Caughey damping. The deformation \mathbf{u} can be represented as a linear combination of linear modes ϕ which can be obtained by linear modal analysis, $\mathbf{M} \phi = \lambda \mathbf{K} \phi$, where $\phi_i^T \mathbf{M} \phi_i = 1$, $\phi_i^T \mathbf{K} \phi_i = \omega_i^2$ and ω_i is the frequency of mode i . The damping ratio of a deformation is a measure of how much amplitude the oscillation loses in one period. The damping ratio ξ_i of linear mode ϕ_i under Rayleigh damping is

$$\xi_i = \frac{1}{2\omega_i} \phi_i^T \mathbf{C} \phi_i = \frac{1}{2\omega_i} (\alpha \phi_i^T \mathbf{M} \phi_i + \beta \phi_i^T \mathbf{K} \phi_i) = \frac{1}{2} \left(\frac{\alpha}{\omega_i} + \beta \omega_i \right). \quad (12)$$

An extended Rayleigh damping, called Caughey damping [Caughey 1960; Caughey and O’Kelly 1965], is defined as

$$\mathbf{C} = \mathbf{M} \sum_{j=n_1}^{n_2} \alpha_j (\mathbf{M}^{-1} \mathbf{K})^j, \quad (13)$$

where $0 \leq n_1 < n_2$; note that Rayleigh damping follows for $n_1 = 0$ and $n_2 = 1$. Coefficients α_j relate to the damping ratios ξ_i by

$$\xi_i = \frac{1}{2\omega_i} \sum_{j=n_1}^{n_2} \alpha_j \omega_i^{2j}. \quad (14)$$

This model has adequate degrees of freedom to control the damping ratio for each mode, although the coefficients α_j still couple all the modes together. We note that the Caughey damping model will produce a dense damping matrix (Equation 13) which is computationally expensive to obtain, and is therefore mostly just useful for reduced simulations with linear modes. We can design our damping to behave like Caughey damping in the space of example deformations, as follows. In our damping model, if we choose the Rayleigh damping as the default damping, and select linear modes as the example deformations, we then have $\phi_j^T \mathbf{C} \phi_i = 0$, if $i \neq j$. The damping ratio for each mode in our model is

$$\xi_i = \frac{1}{2\omega_i} \phi_i^T \hat{\mathbf{C}} \phi_i = \frac{\gamma_i}{2} \left(\frac{\alpha}{\omega_i} + \beta \omega_i \right). \quad (15)$$

Note that we are controlling the damping ratio for each example deformation independently with a single degree of freedom γ_i . Therefore, our model achieves the same damping along the example deformations as Caughey damping if we set $\gamma_i = \sum_{j=n_1}^{n_2} \alpha_j \omega_i^{2j} / (\alpha + \beta \omega_i^2)$. Our model converges to Caughey damping under more and more examples, and becomes identical to Caughey damping when the examples span the complete deformation space of the object. Note that Caughey damping is also a viscous damping model, and therefore it could also be used as the default damping directly.

4.1.3 Comparison to anisotropic materials. Changing material stiffness affects Rayleigh damping due to the stiffness-damping term, when $\beta \neq 0$. Therefore, one may try to mimic anisotropic damping with anisotropic elasticity. However, under modified elasticity, the deformation and frequency change significantly (Figure 5). Our method can produce the desired damping effect easily without any change to the underlying elasticity. In practice, this means that the user can keep the existing scene setup, gravity strength, collision parameters and timesteps when tuning damping. Our method also makes it much easier to design the damping, as it operates in the space of example deformations, which are of direct concern to the

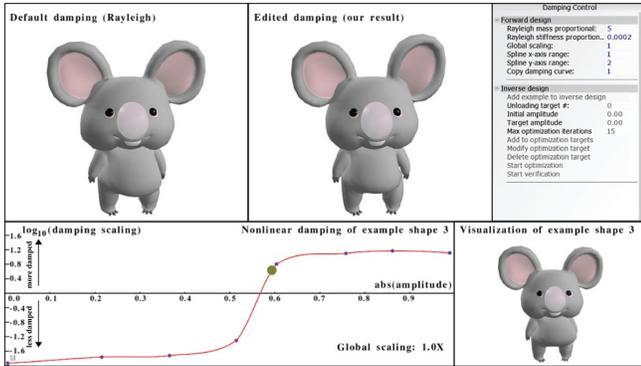


Fig. 6. **User interface for forward design of damping.** In the bottom-left corner, the user can edit the spline curve that controls the nonlinearity of damping for each example deformation. The three preview windows show the result under Rayleigh damping, our damping corresponding to the adjusted curve, and a visualization of the example deformation whose damping is currently being edited.

user. In contrast, the element-level anisotropic elasticity parameters are indirectly related to the damping effects and harder to tune.

4.2 Nonlinear Damping

Although widely used, viscous damping models that linearly depend on velocity only constitute a small subset of all viscous damping models. Under many circumstances, the damping matrix \mathbf{C} can nonlinearly depend on the deformation or velocity, such as in fluid damping, air drag or in structural Coulomb deformation damping [De Silva 2007]. In this section, we extend our example-based damping model to easily and intuitively model nonlinear damping along the example deformations, using a simple spline system.

Given the current deformation \mathbf{u} , we can obtain the magnitudes (excitations) $\chi = [\chi_1, \dots, \chi_k] \in \mathbb{R}^k$ of the examples by a least-square minimization of $\|\mathbf{X}\chi - \mathbf{u}\|_{\mathbf{M}}^2$, which has the solution $\chi = (\mathbf{X}^T \mathbf{M} \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{M} \mathbf{u})$. The inverse of $\mathbf{X}^T \mathbf{M} \mathbf{X} \in \mathbb{R}^{k \times k}$ can be pre-computed. Then, instead of setting a constant damping control parameter γ_i , we vary it based on some user-prescribed function of the deformation magnitude χ_i . Note that the index i here refers to the deformation example index. To locally control the nonlinear damping, we model the function $\gamma_i(\chi_i)$ as a 1D spline (see Figure 6). In an interactive process, for each nonlinear curve, the user provides $m_i \geq 1$ samples $(|\chi_i^j|, \log_{10}(\gamma_i^j)), j = 1, 2, \dots, m_i$, over a selected range $[0, |\chi_i|^{\max}]$. Here, we choose the absolute value of χ_i (as opposed to χ_i directly) as the x -axis, so that the resulting damping is symmetric with respect to direction of deformation. We model the splines in the logarithmic space so that the damping design is symmetric under strengthening and attenuating. Although our spline system is not limited to a specific spline type, we use the Bézier spline to interpolate and extrapolate the sample points, to accommodate the entire range of deformations. The user can freely move the sample points to edit the curve and the damping control parameters γ_i are ensured to be positive by our logarithmic space design. We can similarly extend our method to model the nonlinear damping effects dependent on velocity magnitude with a spline-based nonlinear function $\gamma_i(\dot{\chi}_i)$, where $\dot{\chi} = (\mathbf{X}^T \mathbf{M} \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{M} \dot{\mathbf{u}})$

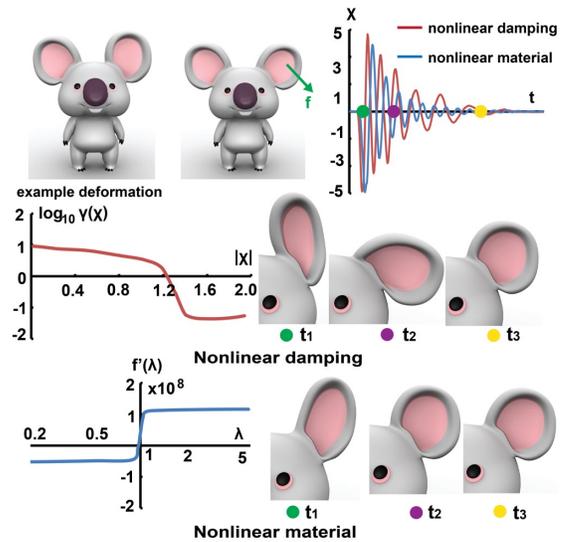


Fig. 7. **Our nonlinear damping effects cannot be obtained by adjusting nonlinear elastic properties.** Middle row: our nonlinear damping curve and frames of the resulting animation. Bottom row: our best attempt to obtain a similar result simply by using linear Rayleigh damping, but setting nonlinear elastic properties. Even with excessive tuning, result is still different and fundamentally changes the object's dynamics (right column).

(Figure 1). In order to control the nonlinear damping isotropically and independently of example deformations, our spline-based nonlinear damping scaling can also be applied to the complete damping matrix, using a factor $\gamma(\|\mathbf{u}\|_2)$, in which case the spline x -axis is the deformation magnitude $\|\mathbf{u}\|_2$. We summarize our damping force as

$$\hat{\mathbf{f}}_d(\mathbf{u}, \dot{\mathbf{u}}) = \tilde{\mathbf{C}} \dot{\mathbf{u}}, \quad \text{for} \quad (16)$$

$$\tilde{\mathbf{C}} = \gamma(\|\mathbf{u}\|_2) \left(\mathbf{C} + \mathbf{U} (\mathbf{\Pi}(\chi_1, \chi_2, \dots, \chi_k) - \mathbf{I}) \mathbf{U}^T \right), \quad (17)$$

where $\mathbf{\Pi}(\chi_1, \chi_2, \dots, \chi_k)$ is the non-negative definite matrix obtained from $\mathbf{R} \text{diag}[\gamma_1(\chi_1), \gamma_2(\chi_2), \dots, \gamma_k(\chi_k)] \mathbf{R}^{-1}$, as explained in Section 4.1. We note that our model essentially approximates nonlinear damping with an equivalent velocity-linear damping, whose matrix $\tilde{\mathbf{C}}$ depends on excitations χ . Different from quasi-linear elasticity which makes the system nonconservative, our quasi-linear damping model does not introduce visual artifacts due to the energy dissipation property of the damping force [Elliott et al. 2015].

4.2.1 Comparison to nonlinear materials. With nonzero stiffness damping, one may try to achieve nonlinear damping by controlling the nonlinearity of the material. In Figure 7, we select a koala waving ear as an example deformation, and design a nonlinear damping curve such that the damping decreases with the deformation. Therefore when pulled with a force, the ear initially behaves very lively when it is mostly in the large deformation range. The motion is damped much faster as soon as the amplitude decays to small deformations. In order to achieve similar nonlinear damping effects, we attempt to tune the spline-based material [Xu et al. 2015b] to make it stiff around the rest shape, but soft for large deformations. From the decaying amplitude curves, we can see that although the initial deformation was similar, the spline-based nonlinear material

significantly alters the vibration frequency (Figure 7). Tuning the material nonlinearity also globally affects the dynamic behavior for all the deformations, regardless of the deformation direction. Furthermore, spline-based materials work in the space of principal stretches of each individual element, whereas our example-space damping model is more direct and intuitive to control.

5 INVERSE DAMPING DESIGN

Although our presented damping design is easy to use, it may require a few manual design and simulation cycles to obtain the expected damping effects. In this section, we demonstrate how to infer the nonlinear damping control curves $\gamma_i(\chi_i)$, $i = 1, 2, \dots, k$, from a high-level user-provided damping specification.

We first need to define a high-level specification for the user to describe the desired nonlinear damping. In mechanical engineering, the linear damping parameters are typically measured from a few “unloading” tests, where the object is unloaded from some displacements and the amount of energy dissipated per cycle is a measure of the structure’s damping level. Similarly, we let the user control damping at a high level, by prescribing the amplitude of an example deformation after one oscillation cycle, for various initial amplitudes. Therefore, for each example deformation i , we unload the object with some user-prescribed initial displacement $a_i \mathbf{x}_i$, $a_i > 0$, and let the user prescribe the amplitude $0 \leq b_i \leq a_i$ that remains after one oscillation cycle. The end of an oscillation cycle is detected as the timestep when the deformation magnitude switches from increasing to decreasing. We also enforce that the signal has to pass through the origin twice in each oscillation cycle in case of local peaks due to high-frequency vibrations. We note that the residual amplitude after one oscillation cycle monotonically decreases as the damping function γ_i increases (see Figure 9). It approaches the minimum value of 0 when the system is strongly overdamped (γ_i is very large). The maximum value of b_i is reached when only artificial damping exists in the system ($\gamma_i = 0$). With only one prescribed (a_i, b_i) pair (an “unloading target”), linear damping alone is already enough. To inversely design nonlinear damping, the user needs to specify multiple unloading targets (a_i^j, b_i^j) , $j = 1, 2, \dots, m_i$. We model the nonlinear damping curve $\gamma_i(\chi_i)$ using piecewise-linear splines (see Figure 8, bottom-left), for reasons explained later. We use a_i^j as the x -axis coordinates of the control points (the “knees” of the piecewise-linear splines) to cover the entire range of deformations in the unloading tests. Each unloading test j of example deformation i will add one spline control point $(a_i^j, \log_{10}(\gamma_i^j))$. We optimize γ_i^j , for all examples i and all unloading tests j , to minimize the difference to the user-prescribed residual amplitudes. We express our optimization problem as

$$\min_{\gamma_1^1, \dots, \gamma_1^{m_1}, \dots, \gamma_k^1, \dots, \gamma_k^{m_k}} \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^{m_i} \left(\frac{f(\gamma_i, a_i^j)}{b_i^j} - 1 \right)^2, \quad (18)$$

where $f(\gamma_i, a_i^j)$ is a black-box function which returns the amplitude of the example deformation i after one oscillation cycle, under the initial amplitude a_i^j , and based on the damping curve γ_i modeled

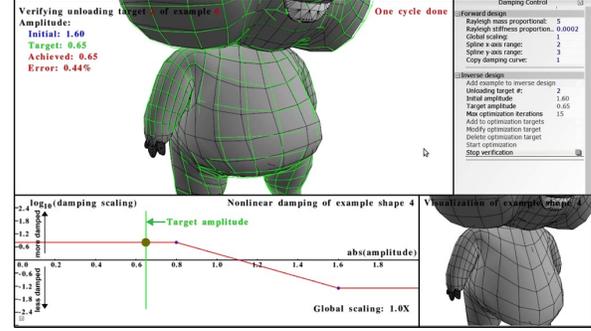


Fig. 8. User interface for inverse damping design. For each example, the users enter a few (initial amplitude, amplitude after one cycle) targets. Our optimizer then produces the nonlinear damping curve (bottom-left).

with the current values of γ_i^k , $i = 1, \dots, k$, $j = 1, \dots, m_i$. We evaluate the black-box function f using simulation. We use relative errors so that errors at all unloading tests are given equal importance.

The optimization problem 18 is highly nonlinear and one could potentially obtain the gradient using the adjoint method [Barbič et al. 2009; McNamara et al. 2004]. However, exploiting the fact that b_i is a monotonic function of γ_i , we can simply optimize our problem using bisection (Figure 9). We first discuss the case when there is only one example deformation i with multiple unloading targets (a_i^j, b_i^j) , $j = 1, 2, \dots, m_i$. We note that for an unloading task with an initial amplitude a_i^j , only the control points in the range $[0, a_i^j]$ affect the simulation. Therefore, we sort the initial amplitudes a_i^j in ascending order, and proceed to optimize the control points $(a_i^j, \log_{10}(\gamma_i^j))$, $j = 1, 2, \dots, m_i$, one at a time. We run several one-cycle-oscillation simulations from each amplitude a_i^j , searching for optimal $\log_{10}(\gamma_i^j)$ using bisection. The previously optimized parameters are all in the deformation range $[0, a_i^j]$, and we keep them fixed. If we were to use a Bézier spline to represent the damping curve, the position of the control point $(a_i^j, \log_{10}(\gamma_i^j))$ will affect the curve in the previously optimized deformation range; hence, we use piecewise linear segments (Figure 8, bottom-left). We start with the current damping parameters and compare the amplitude at the end of one oscillation cycle to b_i^j . If larger, we increase γ_i^j to 1000 (our upper limit), in order to properly bracket the true γ_i^j value. If the residual amplitude still remains larger, we terminate with $\gamma_i^j = 1000$. Similarly, if the residual amplitude is smaller than b_i^j , we decrease γ_i^j to 0.001. Otherwise, we proceed via bisection, setting the next $\log(\gamma_i^j)$ using linear interpolation of the two previous values that came closest to the target. We terminate the optimization when the error is smaller than some threshold (we use 1%), or when reaching an iteration limit (we use 15). Our employed constants are examples that worked well for us, and can be adjusted by the user.

We use the same method to optimize all the damping curves when there are multiple example deformations. However, the unloading simulations of one example deformation may be coupled with the damping curves of the other examples. Therefore, for cases with multiple example deformations, we use block-coordinate descent

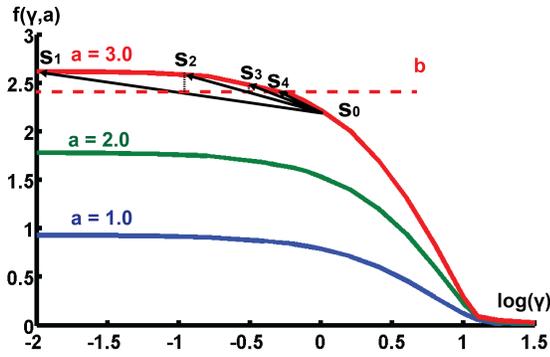


Fig. 9. **Illustration of inverse design.** X-axis is the damping strength γ (the unknown) for an example deformation of interest (Koala demo). Single example deformation, single unloading test; hence there is a single control point and γ is independent of the deformation magnitude in this illustration. Y-axis gives the amplitude of the example deformation after one oscillation cycle, for three separate initial amplitudes 1.0, 2.0, 3.0. For each of these amplitudes, note that the fall of amplitude at $\gamma = 0$ is purely due to artificial damping. The user specified that for the initial amplitude of 3.0, the amplitude after one oscillation cycle should be $b = 2.45$. We show the iterations S_i of the automatic search algorithm for γ .

optimization where we fix the all but one damping curve, optimize that damping curve, and proceed round-robin until convergence. In practice, we observe that the coupling is often weak, and we typically reach convergence in just two iterations.

Our optimization method is applicable both to full and reduced simulation. For interactive optimization performance, we choose to use reduced simulation. Given the reduced simulation basis $\Phi \in \mathbb{R}^{3n \times r}$, we project all the example deformations \mathbf{X} into reduced space as $\tilde{\mathbf{X}} = \Phi^T \mathbf{X} \in \mathbb{R}^{r \times k}$. Then, our reduced damping matrix is

$$\tilde{\mathbf{C}} = \gamma(\|\tilde{\mathbf{u}}\|_2) \left(\tilde{\mathbf{C}} + \tilde{\mathbf{U}} \left(\Pi(\chi_1, \chi_2, \dots, \chi_k) - \mathbf{I} \right) \tilde{\mathbf{U}}^T \right) \dot{\tilde{\mathbf{u}}}. \quad (19)$$

Matrix $\tilde{\mathbf{C}} = \Phi^T \mathbf{C} \Phi \in \mathbb{R}^{r \times r}$ gives reduced default damping. $\tilde{\mathbf{X}} = \tilde{\mathbf{Q}}\mathbf{R}$, $\tilde{\mathbf{U}} = \tilde{\mathbf{C}}\tilde{\mathbf{Q}}$, and $\tilde{\mathbf{u}}, \dot{\tilde{\mathbf{u}}} \in \mathbb{R}^r$ are the reduced deformation and velocity.

6 COMPOSITE IMPLICIT INTEGRATION

In computer animation, we often choose implicit backward Euler [Baraff and Witkin 1998] or implicit Newmark (trapezoidal rule) [Wriggers 2002] for stable integration of equations of motion. The trapezoidal rule is very popular, mainly because it dissipates the energy lightly and achieves second-order timestep accuracy. However, in nonlinear dynamic analyses, it can become unstable under large deformations. Implicit backward Euler offers much better simulation stability, but introduces excessive numerical damping. A natural idea for stable and accurate integration is to combine a lightly dissipative and a strongly dissipative integration scheme.

Following the idea of [Bathe 2007], we calculate the unknown displacements, velocities and accelerations by considering the timestep Δt to consist of two equal substeps of size $\Delta t/2$. In the first half sub-step, we employ the trapezoidal rule

$$\dot{\mathbf{u}}_{t+\Delta t/2} = \dot{\mathbf{u}}_t + \frac{\Delta t}{4} (\ddot{\mathbf{u}}_t + \ddot{\mathbf{u}}_{t+\Delta t/2}) \quad (20)$$

$$\mathbf{u}_{t+\Delta t/2} = \mathbf{u}_t + \frac{\Delta t}{4} (\dot{\mathbf{u}}_t + \dot{\mathbf{u}}_{t+\Delta t/2}), \quad (21)$$

to advance Equation 1 to time $t + \Delta t/2$. Applying Newton-Raphson iterations $i = 1, 2, 3, \dots$, we obtain

$$\begin{aligned} & \left(\frac{16}{\Delta t^2} \mathbf{M} + \frac{4}{\Delta t} \mathbf{C}_{t+\Delta t/2}^{(i-1)} + \mathbf{K}_{t+\Delta t/2}^{(i-1)} \right) \Delta \mathbf{u}^{(i)} = \\ & = \mathbf{f}_{ext} - \mathbf{f}_{int}(\mathbf{u}_{t+\Delta t/2}^{(i-1)}) - \mathbf{M} \left(\frac{16}{\Delta t^2} (\mathbf{u}_{t+\Delta t/2}^{(i-1)} - \mathbf{u}_t) - \frac{8}{\Delta t} \dot{\mathbf{u}}_t - \ddot{\mathbf{u}}_t \right) - \\ & \quad - \mathbf{C}_{t+\Delta t/2}^{(i-1)} \left(\frac{4}{\Delta t} (\mathbf{u}_{t+\Delta t/2}^{(i-1)} - \mathbf{u}_t) - \dot{\mathbf{u}} \right), \end{aligned} \quad (22)$$

where $\mathbf{u}_{t+\Delta t/2}^{(i)} = \mathbf{u}_{t+\Delta t/2}^{(i-1)} + \Delta \mathbf{u}^{(i)}$. Velocity $\dot{\mathbf{u}}_{t+\Delta t/2}$ and acceleration $\ddot{\mathbf{u}}_{t+\Delta t/2}$ can be obtained via Equations 20 and 21, once the Newton-Raphson method converges or reaches an iteration limit.

In the second half sub-step, the second-order implicit Euler backward method [Hairer et al. 1993] is applied for stability as

$$\dot{\mathbf{u}}_{t+\Delta t} = \frac{1}{\Delta t} \mathbf{u}_t - \frac{4}{\Delta t} \mathbf{u}_{t+\Delta t/2} + \frac{3}{\Delta t} \mathbf{u}_{t+\Delta t} \quad (23)$$

$$\ddot{\mathbf{u}}_{t+\Delta t} = \frac{1}{\Delta t} \dot{\mathbf{u}}_t - \frac{4}{\Delta t} \dot{\mathbf{u}}_{t+\Delta t/2} + \frac{3}{\Delta t} \dot{\mathbf{u}}_{t+\Delta t}. \quad (24)$$

Each Newton-Raphson iteration of advancing Eq. 1 to $t + \Delta t$ is

$$\begin{aligned} & \left(\frac{9}{\Delta t^2} \mathbf{M} + \frac{3}{\Delta t} \mathbf{C}_{t+\Delta t}^{(i-1)} + \mathbf{K}_{t+\Delta t}^{(i-1)} \right) \Delta \mathbf{u}^{(i)} = \mathbf{f}_{ext} - \mathbf{f}_{int}(\mathbf{u}_{t+\Delta t}^{(i-1)}) - \\ & - \mathbf{M} \left(\frac{9}{\Delta t^2} \mathbf{u}_{t+\Delta t}^{(i-1)} - \frac{12}{\Delta t^2} \mathbf{u}_{t+\Delta t/2} + \frac{3}{\Delta t^2} \mathbf{u}_t - \frac{4}{\Delta t} \dot{\mathbf{u}}_{t+\Delta t/2} + \frac{1}{\Delta t} \dot{\mathbf{u}}_t \right) - \\ & \quad - \mathbf{C}_{t+\Delta t}^{(i-1)} \left(\frac{3}{\Delta t} \mathbf{u}_{t+\Delta t}^{(i-1)} - \frac{4}{\Delta t} \mathbf{u}_{t+\Delta t/2} + \frac{1}{\Delta t} \mathbf{u}_t \right), \end{aligned} \quad (25)$$

where $\mathbf{u}_{t+\Delta t}^{(i)} = \mathbf{u}_{t+\Delta t}^{(i-1)} + \Delta \mathbf{u}^{(i)}$, and we get the final velocity and acceleration from Equations 23 and 24.

Although this composite integration scheme requires about twice as much computational effort as the one-step integrators, it achieves second-order accuracy. In particular, the method remains stable even when the trapezoidal rule becomes unstable. Figure 10 shows that even under a $2\times$ larger timestep (and hence equal invested computational effort), the implicit Bathe integrator still achieves the same accuracy as the trapezoidal rule (Figure 10 (a)). Under nonlinear dynamics, the trapezoidal rule can become unstable due to the non-conserved energy and momentum. The Newmark method with $\beta > 0.25$ (also called the *generalized- α method*) can improve stability, but introduces numerical dissipation. Implicit Bathe integrator effectively maximizes the dissipation of spurious high-frequency vibrations without adding excessive numerical damping to low-frequency deformations (Figure 10 (b)). Compared to the generalized- α method, implicit Bathe produces much less artificial damping (Figure 10 (a)).

6.1 Linear System Solve for Implicit Integration

In our system, we modify the damping matrix \mathbf{C} to our anisotropic nonlinear damping matrix $\hat{\mathbf{C}}$, which unfortunately is a dense matrix, and therefore we cannot directly use sparse linear solvers to perform implicit integration during a timestep. However, from Equation 6, we observe that the modified damping matrix $\hat{\mathbf{C}}$ is actually obtained by perturbing the default \mathbf{C} with a low-rank (rank = k) dense matrix $\mathbf{U}(\Gamma - \mathbf{I})\mathbf{U}^T$. Therefore, we can employ the Woodbury formula [Woodbury 1950] to solve the linear system as

$$(\mathbf{A} + \mathbf{wUV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{U} (\mathbf{I}/\mathbf{w} + \mathbf{V} \mathbf{A}^{-1} \mathbf{U})^{-1} \mathbf{V} \mathbf{A}^{-1}, \quad (26)$$

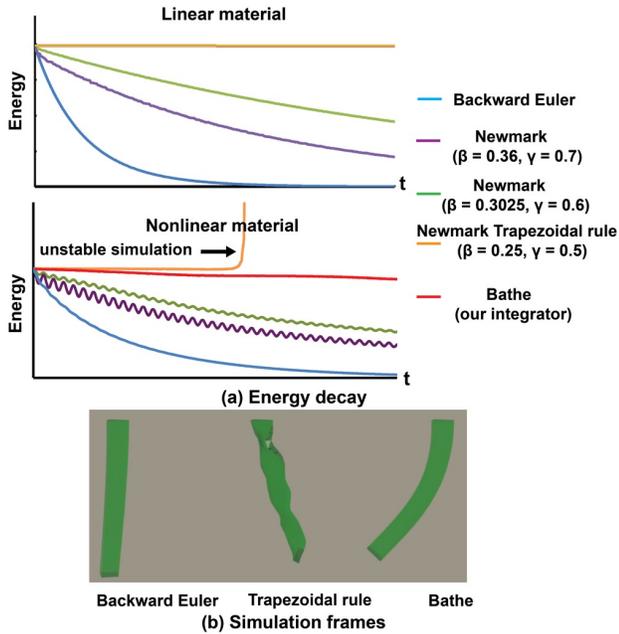


Fig. 10. **Comparison of integration accuracy and stability.** A beam is unloaded from an initial displacement without any external damping. We show the energy decaying curves under linear and nonlinear material in (a), and the simulation frames in (b). We use $2\times$ larger timestep for implicit Bathe than the other integrators, to normalize the computational effort in each timestep. Trapezoidal rule does not remove spurious high-frequency vibrations and fails to remain stable under large timesteps for nonlinear material. Our method remains stable and preserves energy without introducing excessive damping, unlike the implicit backward Euler or Newmark with $\beta > 0.25$. The curves for implicit Bathe and trapezoidal methods are on top of each other for linear material simulation.

where $\mathbf{V} = (\mathbf{\Gamma} - \mathbf{I})\mathbf{U}^T$ and \mathbf{A} is the system matrix (i.e., the left-hand-side matrix) under default damping in Equations 22 ($w = 4/\Delta t$) and 25 ($w = 3/\Delta t$). In our work, we use the Pardiso linear system solver. The main overhead for our system solve is that we need to solve $k + 1$ linear systems, instead of just one linear system as in the standard damping case. However, we can reduce the performance overhead significantly by prefactoring the matrix \mathbf{A} with Cholesky decomposition, and solving the $k + 1$ linear systems in parallel (we use 16 threads, Table 1). We note that this construction is only required for anisotropic damping. For isotropic nonlinear damping, one can continue to solve a single linear system per timestep, as with standard damping. For large-scale simulations, one could replace the direct solver for an iterative solver. Because the system matrices in Equations 22 and 25 are perturbed with low-rank matrices, it is straightforward to explicitly multiply any vector with these system matrices, as needed for iterative solvers. Preconditioning schemes such as Jacobi or Gauss-Seidel are also easily possible, as one can quickly construct the diagonal of the system matrix.

7 RESULTS

We demonstrate our method with several examples (Table 2). All examples used an Intel Xeon E5-2690 PC (1st gen., 2 processors @

k	0	1	2	4	8	10
system solve (sec)	0.09	0.09	0.14	0.16	0.22	0.24
one timestep (sec)	0.14	0.15	0.19	0.23	0.31	0.32

Table 1. Performance under a different number of example deformations k . Full implicit Bathe simulation on Koala. If k is smaller than the number of physical cores (which is often the case in practice), the computation time increases sublinearly with k .

	#tets	#modes	#examples	method	fps
Elephant	10,416	-	4	AF	3
Koala (forward)	11,080	74	4	NAR	110
Koala (inverse)	11,080	74	4	NAR	110
Dragon	8,252	-	1	NAF	5
Leaf	6,210	-	2	NAF	10

Table 2. **Summary of results.** Method column: N=nonlinear damping, A=anisotropic damping, R=reduced simulation, F=full simulation.

2.90 GHz, each 8 cores), with 32GB of RAM. We first demonstrate our example-based anisotropic damping on a walking elephant (St.Venant-Kirchhoff material). The artist rigged and keyframed the walking motion in Maya. We add secondary physically based motion to the trunk, ears and belly, using a constraint-driven FEM simulation, similar to [Li et al. 2016]. We select the left-right and forward-backward deformations of the trunk and the forward-backward bending ears motion as the four example deformations (see Figure 11 (a)). For evaluation, we design three sets of anisotropic damping models. In the first model, we make the damping of the trunk along the left-right example deformation $10\times$ weaker, and the other direction $3\times$ stronger. In the output motion, we can clearly see that compared to default Rayleigh damping, our damping makes the left-right trunk motion much more lively, while the forward-backward motion is damped out much faster (Figure 11 (b)). Our second damping setting reverses the damping scales of the two trunk motions. Therefore, the trunk dynamics is “biased” into the direction of forward-backward motion (Figure 11 (c)). In the third damping setting, we leave the trunk at its default damping levels, but make the forward-backward motion of the ears $3\times$ more damped (Figure 11 (c)). We observed that the initial acceleration of the forward-walking elephant brings the ears to about the same large deformations, under both the default and our damping. However, our damping causes the ear to more slowly recover to the rest shape. In contrast, with default damping, the ears are vibrating around the rest shape. Such example-based anisotropic effects are difficult to achieve with element-level strain-rate damping [O’Brien and Hodgins 1999] because the example deformations overlap in space in arbitrary ways.

In our second example, we forward-design the nonlinear damping curve to match the real-world damping observed on a 3D-printed dragon (Figure 12). We recorded a video of the recovery of the dragon mouth from an initial large deformation. The TangoBlackPlus material is very viscous, devoid of oscillations; its visual appearance can be characterized as “overdamped”. We observe that, as the mouth is released from a large deformation, it first quickly recovers by a certain amount, and then slowly continues to recover back to the rest shape. In order to match the dynamic behavior, we tuned the mass, stiffness and damping properties. We first set the mass density (1008 kg/m^3) by dividing the weighted mass with the volume of the tetrahedral simulation mesh. Next, we tune the elastic material, by statically loading the real dragon mouth with 5 sets of loading

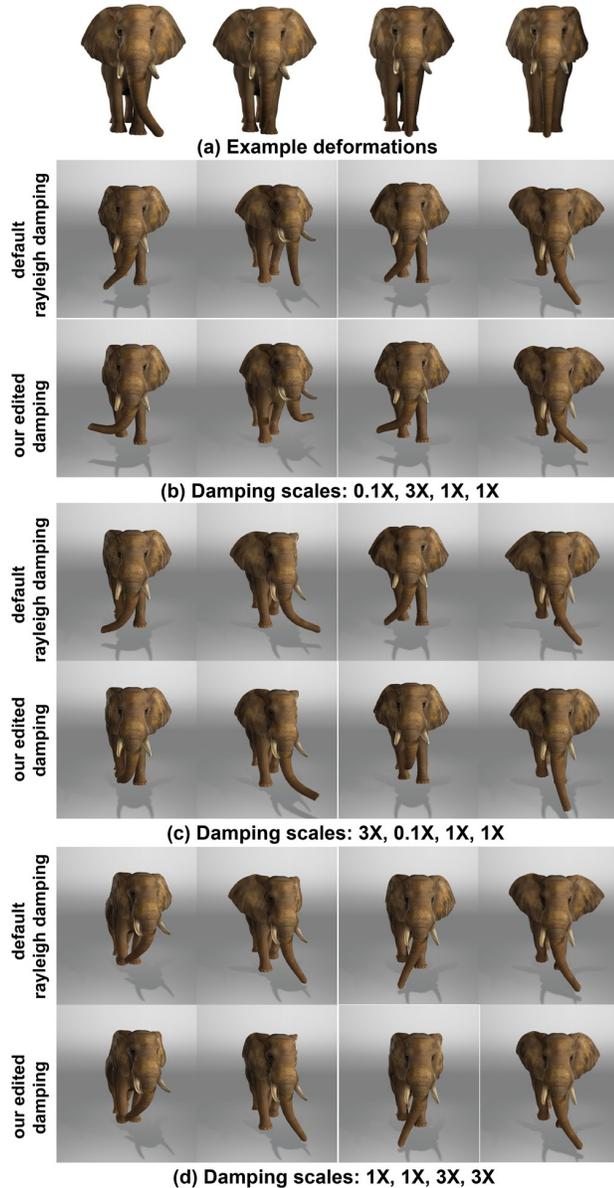


Fig. 11. **Anisotropic damping design for the walking elephant.** Part (a) shows the four chosen example deformations: left-right deflection of the trunk, forward-backward deflection of trunk, left ear bending, right ear bending. In parts (b,c,d), we specify a damping scale for each of the four example deformations, and show four selected simulation frames. Our method clearly produces the prescribed anisotropic damping effects.

forces (see Figure 12 (a)), and recording the static deformations with a millimeter paper. We then manually fit the nonlinear spline-based material curve [Xu et al. 2015b] such that the simulation matches the real quasi-static tests, using trial and error. We then proceeded to fitting the damping. First, we observed that the Rayleigh damping model is insufficient to match the recorded dynamics (Figure 12 (c)). We therefore model the dragon using our damping model, as follows. We use one simulation frame as an example deformation,

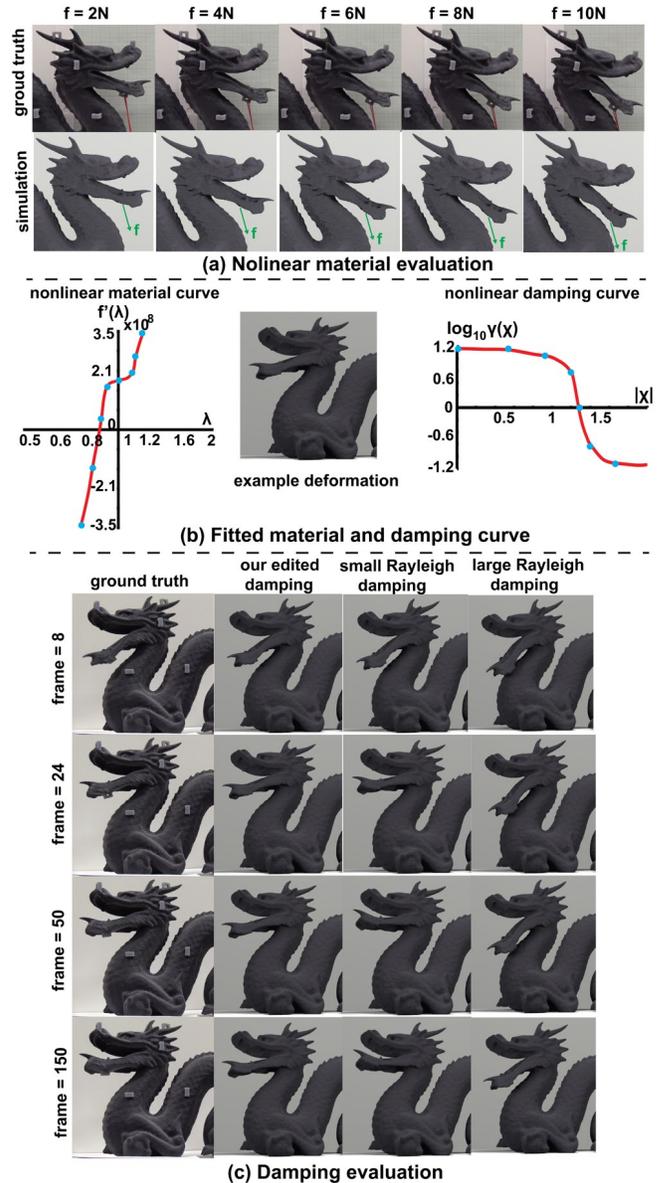


Fig. 12. **Fitting the elastic material and damping based on real-world observations.** We subjected this 3d-printed dragon (Stratasys TangoBlack-Plus material) to several static force loads and adjusted the spline elastic material curve [Xu et al. 2015b] to best match the resulting deformation. We then adjusted the damping of the “open-mouth” example deformation, to match the real-world reference dragon video (“ground truth”). This particular material is more damped under small deformations than under large deformations. Such an effect is impossible to match using linear damping. One can either try to match the ground truth damping under large deformations, which causes small deformation damping to be visibly too weak, or match small deformation damping, which causes the simulation to be severely overdamped under large deformations. Our nonlinear damping can be seen to match the ground truth well.

and then edit our nonlinear damping curve to model the observed motion (Figure 12 (b)). Starting from default Rayleigh damping, we

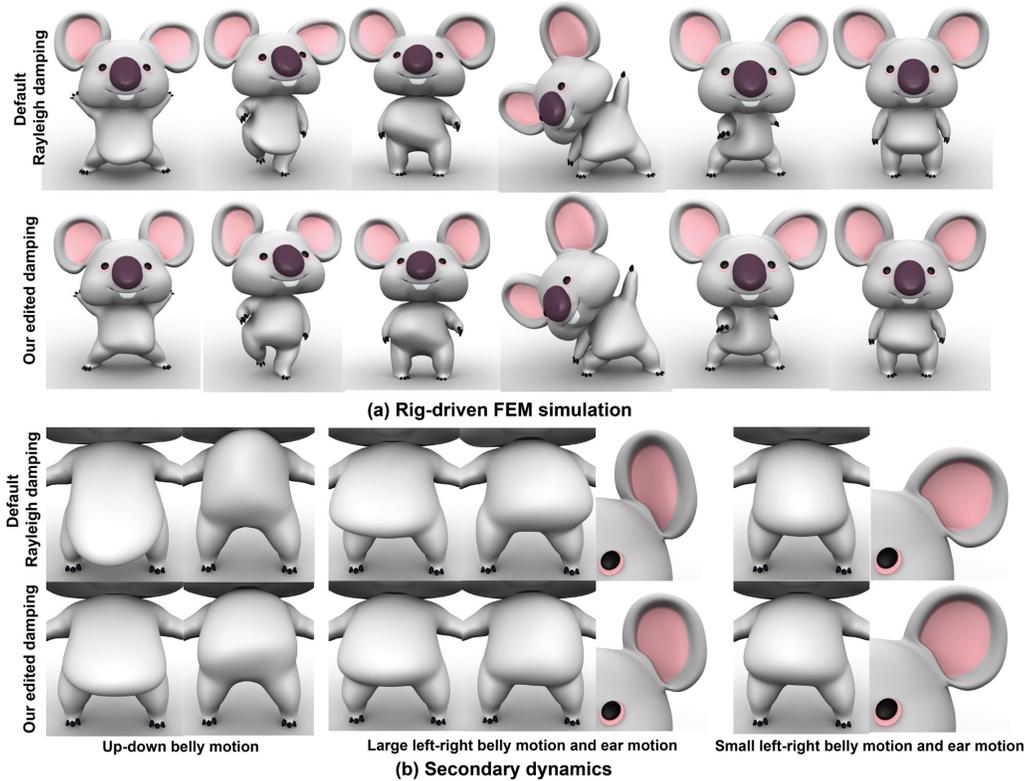


Fig. 13. **Our damping applied to a rig-driven physically-based simulation** [Xu and Barbic 2016]. The damping parameters can be adjusted once (done in Figure 2), and then applied to an arbitrary number of physically based character simulations in real time. Part (a) shows the animation frames, whereas part (b) removes the rig from display, showing only the secondary dynamics which shows our intended anisotropic and nonlinear damping effects.

edit the damping spline curve such that at large deformations, the material is lightly damped. The magnitude of damping is made to increase to a large value around the rest shape (Figure 12 (b)). With our tuned mass, stiffness and damping properties, we were able to visually fit the observed dynamic behaviors closely.

We use our system to interactively perform forward and inverse damping design for model-reduced simulations of the koala. We select the left-right and up-down belly motion (Figure 2), and the bending motion of each ear (Figure 7) as four example deformations. In the forward design process, we edit the nonlinear spline damping curve such that the ear and left-right belly motions are much less damped than the default Rayleigh damping under small deformations, whereas for large deformations, they are made more damped (Figure 2). We set the up-down direction of belly motion to be $10\times$ more damped, producing anisotropic damping effects on the belly. The artist can obtain interactive design feedback by simulating the koala (pulling with the mouse). The damping needs to be designed only once, after which it can be applied to an arbitrary number of simulation inputs, such as for example, arbitrary skeleton animation encountered, say, in a computer game. We demonstrate this by applying the designed damping to a rig-driven FEM simulation [Xu and Barbic 2016] (Figure 13), for several input skeleton motions that were not used during the damping design. Under the designed damping, the up-down motion of the belly is quickly damped during the high koala jumps. Large left-right belly deformations quickly

dissipate to a small value, after which the belly continues to vibrate in the small deformation range for a long time, due to the designed nonlinear damping curve. Similar dynamic behavior is observed in the ears. We choose to illustrate inverse damping design by producing a different kind of a belly damping curve where large deformations are less damped, but small deformations are subjected to large damping (Figure 2). For the left-right belly motion, we set the after-one-cycle residual amplitudes to 0.2, 0.75, when unloading it from amplitudes of 0.8, 1.6, respectively. For the up-down belly motion, the residual amplitudes are set to 0.25, 0.65, for the same initial amplitudes as in the left-right motion. Our optimization is able to reach these targets within 1% for all of the four unloading tests in 5 – 10 iterations. In simulation, the belly energy then decays slowly under large deformations, but once the motion is mostly in the small deformation region, it dissipates the energy much faster.

Our nonlinear damping effects are impossible to achieve with linear damping. In Figure 2, we tried to approximate the nonlinear damping effects using linear anisotropic damping. For the forward-designed koala curve, we averaged the nonlinear damping curves, essentially flattening the nonlinear curve into a constant-valued line. We obtained the damping scales $\gamma = 0.6, 10$ for the two example deformations. For the inverse-designed curve, we optimized a single damping scale parameter for each example deformation based on our 4 unloading tests. With optimized damping scales $\gamma = 6.4, 5.5$ for the two example deformations, the linear damping

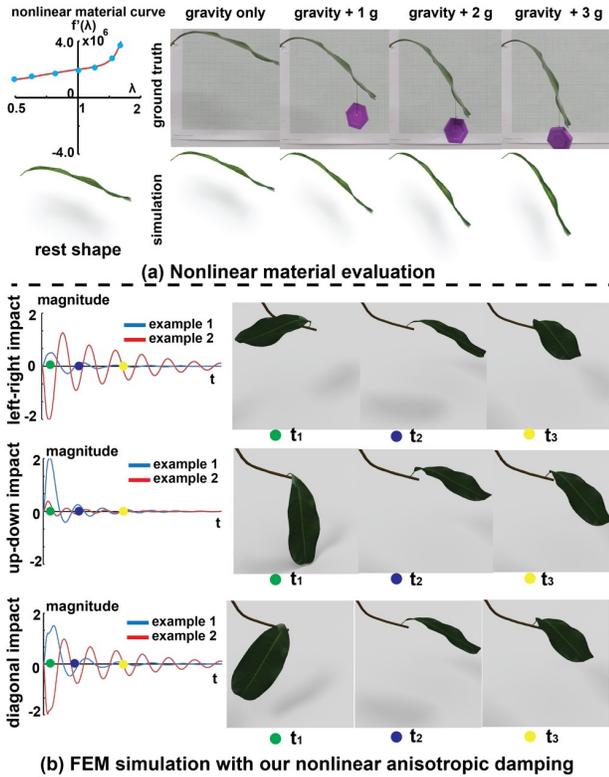


Fig. 14. Leaf FEM simulation with nonlinear elasticity and damping. (a) We fit the nonlinear material elasticity and the rest shape based on the static equilibrium under gravity and external loads. (b) We impact the leaf with external forces in three directions. From the decaying amplitude curves and simulation frames, we can see that the dynamics along example deformation 1 (up-down, blue) is more damped than in the other direction (sideways, red). The amplitudes decay nonlinearly. The nine images show our simulation results.

model can achieve one-cycle oscillation amplitudes 0.22 (12% error), 0.47 (37% error), 0.27 (7% error), 0.53 (19% error) for the 4 unloading targets, respectively. The amplitudes in linear damping must decay exponentially with constant rates, and therefore damping envelopes with non-constant rates cannot be reproduced (Figure 2).

In our last example (Figure 1), we fit the anisotropic nonlinear damping curves to model the leaf vibration of the Carrotwood tree (*Cupaniopsis anacardioides*). We observed that when the leaf vibrates, the sideways (i.e., left-right) motion is much more lively than the up-down motion. Nonlinear damping effects can also be seen, especially in the up-down direction, where the motion first quickly gets damped and then vibrates for a few cycles with small amplitudes. To reproduce this nonlinear anisotropic damping effects, we fit both the nonlinear material and damping curves. Similar to our dragon example, we first measure the mass density ($135\text{kg}/\text{m}^3$) and then manually tune the spline-based material curves using trial and error, such that the FEM simulation matches the camera-recorded leaf deformations under gravity and external loads (0, 1, 2, 3 gram, Figure 14 (a)). We recover the rest shape of the leaf after tuning the material, by running reverse gravity simulation. We iterate to

tweak the spline material curve and the rest shape until convergence. Then, we select the first two linear vibration modes as example deformations; first mode vibrates up-down, and second mode vibrates left-right. We then fit the two damping curves based on unloading tests (Figure 1). Each direction has a small-deformation and a large-deformation unloading test. We note that the leaf has low mass and therefore the air drag in the up-down direction is prominent. Therefore, we model our damping to be nonlinearly dependent on the *velocity* magnitude. Based on our fitted damping curves, we can clearly see that the damping in the up-down direction is larger than in the left-right direction, which explains the anisotropic damping effects. Consistently with the air drag force, we obtain an increasing nonlinear damping curve for the up-down direction, i.e., damping scales superlinearly with velocity. However, for the left-right motion, which is subjected both to structural damping and a smaller amount of air drag, our experiment produced a damping curve that decreases with velocity. We then use our fitted material and damping in a FEM simulation, producing nonlinear and anisotropic vibration effects close to real-world leaf motion (Figure 14 (b)).

8 CONCLUSION

We gave a method to design example-based linear or nonlinear damping properties for solid three-dimensional deformable models simulated using the Finite Element Method. We start with the familiar Rayleigh damping, and then permit the user to modify the damping strength along each of a few chosen example deformations. We proved that our damping is dissipative, and generalizes Caughey damping. We also gave an intuitive system to model nonlinear damping properties using easily tunable one-dimensional splines. We demonstrated a high-level inverse interface whereby the user controls the damping simply by prescribing the initial and residual amplitudes of one-cycle unloading simulations. We compared our method to Rayleigh damping and demonstrated visual effects impossible with Rayleigh damping, but possible with our method. We also introduced an implicit integrator with good stability and damping properties, suitable for removing high-frequency oscillations while only causing a minimal amount of numerical damping.

Our example-based constructions requires solving several linear systems at each timestep; however, this is mitigated by the fact that Cholesky decomposition during each timestep needs to be performed only once, and the multiple right-hand sides can then be solved in parallel. Our example magnitude coefficients χ_i are linear in the deformation u , and therefore nonlinear damping may not be modeled precisely under large rotations. We do not model non-viscous properties which may require modeling viscoelastic functions that depend on states at several previous timesteps. Although we can model damping effects that depend nonlinearly on u and \dot{u} , our system models only a subset of all possible damping relationships; further exploration of this space is left for future work.

ACKNOWLEDGEMENTS

This research was sponsored in part by the National Science Foundation (CAREER-1055035, IIS-1422869), and USC Annenberg Graduate Fellowship to Hongyi Xu. We would like to thank Yijing Li for helping with constraint-based dynamics [Li et al. 2016], Carlos Joy for rigging and keyframing, and Nadja Skale for the video voice-over.

A C-ORTHOGONALITY

For the symmetric non-negative definite matrix \mathbf{C} , two vectors \mathbf{x}, \mathbf{y} are C-orthogonal (by definition) if $\mathbf{x}^T \mathbf{C} \mathbf{y} = 0$. A vector is C-normalized if $\mathbf{x}^T \mathbf{C} \mathbf{x} = 1$. Mathematically, this concept is identical to the more familiar mass-orthogonality commonly used in computer simulation (see, e.g., [Barbic and James 2005]), except that mass-orthogonality uses the mass matrix. In a C-weighted QR decomposition $\mathbf{X} = \mathbf{Q}\mathbf{R}$, the matrix \mathbf{Q} is C-orthogonal. We calculate it using the standard QR algorithm (Gramm-Schmidt), except that all inner products $\mathbf{x}^T \mathbf{y}$ are replaced with C-weighted products $\mathbf{x}^T \mathbf{C} \mathbf{y}$.

B NON-NEGATIVE DEFINITENESS OF Π

Let the eigenvalues and their orthogonal eigenvectors of the symmetric matrix $(\bar{\Pi} + \bar{\Pi}^T)/2$ be λ_i and \mathbf{v}_i , for $i = 1, \dots, k$; assume that the first p are negative, for $0 \leq p \leq k$. Let $\mathbf{y} \in \mathbb{R}^k$ be an arbitrary vector; expand it as $\mathbf{y} = \sum_{i=1}^k \alpha_i \mathbf{v}_i$. Then,

$$\begin{aligned} \mathbf{y}^T (\bar{\Pi} - \sum_{i=1}^p \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i) \mathbf{y} &= \mathbf{y}^T \left(\frac{\bar{\Pi} + \bar{\Pi}^T}{2} - \sum_{i=1}^p \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i \right) \mathbf{y} = \\ &= \mathbf{y}^T \left(\sum_{i=p+1}^k \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i \right) \mathbf{y} = \sum_{i=p+1}^k \lambda_i \alpha_i^2 \geq 0. \end{aligned} \quad (27)$$

REFERENCES

- Sondipon Adhikari. 2001. *Damping models for structural vibration*. Ph.D. Dissertation. University of Cambridge.
- David Baraff and Andrew P. Witkin. 1998. Large Steps in Cloth Simulation. In *Proc. of ACM SIGGRAPH 98*, 43–54.
- Jernej Barbic, Marco da Silva, and Jovan Popovic. 2009. Deformable Object Animation Using Reduced Optimal Control. *ACM Trans. on Graphics (SIGGRAPH 2009)* 28, 3 (2009).
- Jernej Barbic and Doug L. James. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. on Graphics (SIGGRAPH 2005)* 24, 3 (2005), 982–990.
- Adam W Bargteil, Chris Wojtan, Jessica K Hodgins, and Greg Turk. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans. on Graphics (SIGGRAPH 2007)* 26, 3 (2007), 16.
- Klaus-Jürgen Bathe. 2007. Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Computers & structures* 85, 7 (2007), 437–445.
- Markus Becker and Matthias Teschner. 2007. Robust and efficient estimation of elasticity parameters using the linear Finite Element Method. In *Simulation und Visualisierung Conf. (SimVis)*, 15–28.
- Kiran S Bhat, Christopher D Twigg, Jessica K Hodgins, Pradeep K Khosla, Zoran Popovic, and Steven M Seitz. 2003. Estimating cloth simulation parameters from video. In *Symp. on Computer Animation (SCA)*, 37–51.
- B. Bickel, M. Baecher, M. Otaduy, W. Matusik, H. Pfister, and M. Gross. 2009. Capture and Modeling of Non-Linear Heterogeneous Soft Tissue. *ACM Trans. on Graphics (SIGGRAPH 2009)* 28, 3 (2009), 89:1–89:9.
- TK Caughey. 1960. Classical normal modes in damped linear dynamic systems. *Journal of Applied Mechanics* 27, 2 (1960), 269–271.
- TK Caughey and MEJ O’Kelly. 1965. Classical normal modes in damped linear dynamic systems. *Journal of Applied Mechanics* 32, 3 (1965), 583–588.
- Clarence W De Silva. 2007. *Vibration damping, control, and design*. CRC Press.
- Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. 2001. Dynamic Real-Time Deformations Using Space & Time Adaptive Sampling. In *Proc. of ACM SIGGRAPH 2001*, 31–36.
- SJ Elliott, M Ghandchi Tehrani, and RS Langley. 2015. Nonlinear damping and quasi-linear modelling. *Phil. Trans. R. Soc. A* 373, 2051 (2015), 20140402.
- E Hairer, SP Nørsett, and G Wanner. 1993. Solving ordinary differential equations I: nonstiff problems, vol. 8. *Springer* (1993).
- G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible Finite Elements for Robust Simulation of Large Deformation. In *Symp. on Computer Animation (SCA)*, 131–140.
- Ben Jones, Nils Thuerey, Tamar Shinar, and Adam W Bargteil. 2016. Example-based plastic deformation of rigid bodies. *ACM Trans. on Graphics (TOG)* 35, 4 (2016), 34.

- C. Kane, J. E. Marsden, M. Ortiz, and M. West. 2000. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *Internat. J. Numer. Methods Engrg.* 49, 10 (2000), 1295–1325.
- Timothy Lahey. 2002. Modelling hysteresis in the bending of fabrics. (2002).
- Siwang Li, Jin Huang, Fernando de Goes, Xiaogang Jin, Hujun Bao, and Mathieu Desbrun. 2014. Space-time editing of elastic motion through material optimization and reduction. *ACM Trans. on Graphics (SIGGRAPH 2014)* 33, 4 (2014), 108:1–108:10.
- Yijing Li and Jernej Barbic. 2014. Stable Orthotropic Materials. In *Symp. on Computer Animation (SCA)*, 41–46.
- Yijing Li, Hongyi Xu, and Jernej Barbic. 2016. Enriching Triangle Mesh Animations With Physically Based Simulation. *IEEE Transactions on Visualization and Computer Graphics* (2016).
- D Brandon Lloyd, Nikunj Raghuvanshi, and Naga K Govindaraju. 2011. Sound synthesis for impact sounds in video games. In *Symp. on Interactive 3D Graphics and Games*.
- J. E. Marsden and M. West. 2001. Discrete Mechanics and Variational Integrators. *Acta Numerica* 10 (2001), 357–514.
- Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based elastic materials. *ACM Trans. on Graphics (SIGGRAPH 2011)* 30, 4 (2011), 72.
- Antoine McNamara, Adrien Treuille, Zoran Popovic, and Jos Stam. 2004. Fluid control using the adjoint method. *ACM Trans. on Graphics (SIGGRAPH 2004)* 23, 3 (2004), 449–456.
- Eder Miguel, David Miraut, and Miguel A. Otaduy. 2016. Modeling and Estimation of Energy-Based Hyperelastic Objects. *Computer Graphics Forum* 35, 2 (2016), 385–396.
- Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C Schwartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, and Miguel A Otaduy. 2013. Modeling and estimation of internal friction in cloth. *ACM Transactions on Graphics (SIGGRAPH Asia 2013)* 32, 6 (2013), 212.
- M. Müller and M. Gross. 2004. Interactive Virtual Materials. In *Proc. of Graphics Interface 2004*, 239–246.
- James O’Brien, Adam Bargteil, and Jessica Hodgins. 2002. Graphical modeling and animation of ductile fracture. *ACM Trans. on Graphics (SIGGRAPH 2002)* 21, 3 (2002), 291–294.
- J.F. O’Brien and J.K. Hodgins. 1999. Graphical Modeling and Animation of Brittle Fracture. In *Proc. of ACM SIGGRAPH 1999*, 111–120.
- Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. 2001. Scanning Physical Interaction Behavior of 3D Objects. In *Proc. of ACM SIGGRAPH 2001*, 87–96.
- John William Strutt Baron Rayleigh. 1896. *The theory of sound*. Vol. 2. Macmillan.
- Zhimin Ren, Hengchin Yeh, and Ming C Lin. 2013. Example-guided physically based modal sound synthesis. *ACM Transactions on Graphics (TOG)* 32, 1 (2013), 1.
- Christian Schumacher, Bernhard Thomaszewski, Stelian Coros, Sebastian Martin, Robert Sumner, and Markus Gross. 2012. Efficient simulation of example-based materials. In *Symp. on Computer Animation (SCA)*, 1–8.
- Auston Sterling and Ming C Lin. 2016. Interactive modal sound synthesis using generalized proportional damping. In *Symp. on Interactive 3D Graphics and Games*.
- Robert W Sumner and Jovan Popovic. 2004. Deformation transfer for triangle meshes. *ACM Trans. on Graphics (SIGGRAPH 2004)* 23, 3 (2004), 399–405.
- Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation. In *Symp. on Computer Animation (SCA)*, 181–190.
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. *Computer Graphics (Proc. of ACM SIGGRAPH 87)* 21(4) (1987), 205–214.
- Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation capture and modeling of soft objects. *ACM Trans. on Graphics (SIGGRAPH 2015)* 34, 4 (2015), 94.
- Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James F O’Brien. 2010. Example-based wrinkle synthesis for clothing animation. *ACM Trans. on Graphics (SIGGRAPH 2010)* 29, 4 (2010), 107.
- Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. 2009. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, 93–117.
- Robert W Williams. 2010. *Measuring and modeling the anisotropic, nonlinear and hysteretic behavior of woven fabrics*. Ph.D. Dissertation. University of Iowa.
- Max Woodbury. 1950. Inverting modified matrices. *Memorandum report* 42 (1950), 106.
- J Woodhouse. 1998. Linear damping models for structural vibration. *Journal of Sound and Vibration* 215, 3 (1998), 547–569.
- Peter Wriggers. 2002. *Computational Contact Mechanics*. John Wiley & Sons, Ltd.
- Hongyi Xu and Jernej Barbic. 2016. Pose-space subspace dynamics. *ACM Trans. on Graphics (SIGGRAPH 2016)* 35, 4 (2016), 35.
- Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbic. 2015a. Interactive material design using model reduction. *ACM Trans. on Graphics (TOG)* 34, 2 (2015), 18.
- Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbic. 2015b. Nonlinear material design using principal stretches. *ACM Trans. on Graphics (SIGGRAPH 2015)* 34, 4 (2015), 75.