

# Interactive Material Design Using Model Reduction

Hongyi Xu, Yijing Li, Yong Chen, Jernej Barbič  
University of Southern California

We demonstrate an interactive method to create heterogeneous continuous deformable materials on complex three-dimensional meshes. The user specifies displacements and internal elastic forces at a chosen set of mesh vertices. Our system then rapidly solves an optimization problem to compute a corresponding heterogeneous spatial distribution of material properties, using the Finite Element Method (FEM) analysis. We apply our method to linear and nonlinear isotropic deformable materials. We demonstrate that solving the problem interactively in the full-dimensional space of individual tetrahedron material values is not practical. Instead, we propose a new model reduction method that projects the material space to a low-dimensional space of material modes. Our model reduction accelerates optimization by two orders of magnitude, and makes the convergence much more robust, making it possible to interactively design material distributions on complex meshes. We apply our method to precise control of contact forces and control of pressure over large contact areas between rigid and deformable objects for ergonomics. Our tetrahedron-based dithering method can efficiently convert continuous material distributions into discrete ones and we demonstrate its precision via FEM simulation. We physically display our distributions using haptics, as well as demonstrate how haptics can aid in the material design. The produced heterogeneous material distributions can also be used in computer animation applications.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Physically based modeling*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*Virtual Reality*

General Terms: algorithms, design

Additional Key Words and Phrases: interactive, design, materials, model reduction, FEM

## 1. INTRODUCTION

Three-dimensional deformable objects are commonly present in our world and are highly relevant to many engineering and science disciplines, including computer graphics, animation, engineering, physics and medicine. The behavior of a deformable object is uniquely determined by the underlying material law linking the object's displacements (strains) to elastic forces (stresses). In this work, we propose a new method to design heterogeneous material distributions on complex three-dimensional meshes. Previous methods have largely focused on observation or capture of material properties, and a subsequent simulation or fabrication of the observed materials; or on non-interactive design. Once the materials are known, many methods can produce compelling three-dimensional animations, say, of buildings, botanical systems, muscles, skin or internal organs.

We propose an interactive inverse design method to create continuous heterogeneous material distributions on 3D volumetric meshes with arbitrary geometry, based on the Finite Element Method analysis. Our material distributions are designed to conform to prescribed displacements and internal elastic forces at the selected

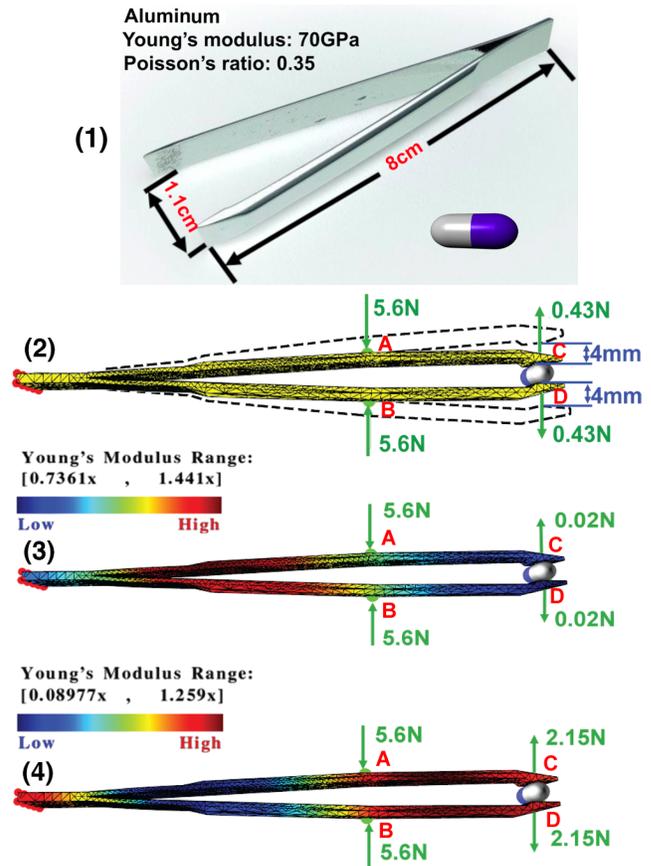


Fig. 1. **Interactive design of materials.** Our work makes it possible to interactively design spatially-varying material distributions that simultaneously conform to prescribed forces and displacements, based on the Finite Element Method analysis. (1) Medical tweezers with a default homogeneous material distribution and the pill. (2) Tweezers tetrahedral mesh (realistic size of 8cm) is loaded with forces at A,B, so that the tweezers deform and grasp the pill with a normal contact force of 0.43N at C,D. Undeformed tweezers are shown dashed. (3) The spatial material distribution of Young's moduli is optimized interactively in 27 seconds, using model reduction of materials (10 modes), so that the displacements at A,B,C,D remain the same, yet the normal contact force on the pill at C,D decreases 20 $\times$ , to 0.02N. The spatial distribution is indicated using the colors. Optimization running without reduction is 204 $\times$  slower. (4) Same as (3), except that the normal contact force is now made 5 $\times$  stronger (2.15N). Optimization using model reduction takes 13 seconds, whereas optimization running without reduction is 739 $\times$  slower, and converged to a suboptimal local minimum (produced negative Young's moduli). In the material legend, 1 $\times$  corresponds to aluminum.

vertices of the three-dimensional mesh (see Figure 1). Such a capability has numerous applications as many mechanical components,

structures and mechanisms have to produce predictable displacements under known forces or pressure distributions. With smoothly varying heterogeneous material distributions, multiple properties (functions) can be obtained in the same object, and high-stress regions at the material boundaries can be avoided. Applications include medical tools that exhibit prescribed forces to the environment when subjected to typical human hand forces (Figure 1), robotic links that bend by prescribed amounts under known contact or actuation forces, ergonomic shoes that exhibit prescribed pressure distributions against the human foot (Figure 8), or ergonomic chairs. We also demonstrate how to edit materials using force-feedback (haptics), and how to use our tool to create heterogeneous linear or nonlinear (supporting large deformations) material distributions for computer animation.

Our distributions are computed interactively based on sparse user input of displacements and forces at a set of mesh vertices. In contrast to shape editing where the input typically consists of positions of a set of handles, our input consists both of prescribed positions and prescribed elastic forces. This provides our system with the additional input to infer the spatial material distribution. The material distribution is computed simultaneously with the shape, combining both global object geometry and material distribution. We formulate and solve an optimization problem to produce smooth material distributions and shapes, consistent with the displacement-force input. We demonstrate that solving this optimization problem in the full space of materials where every element's material is an independent degree of freedom is not practical for interactive applications, due to very long optimization times (hours) and local minima. Instead, we demonstrate how to use model reduction to obtain a low-dimensional optimization problem that can be solved interactively. Whereas previous work applied model reduction to the object's shape, we apply model reduction to the object's *material* distribution and demonstrate that this greatly improves the force and displacement accuracy. We propose a quality low-dimensional space of materials, obtained from the eigenmodes of the Laplace operator, and provide an algorithm to rapidly evaluate the objective function and its gradient with respect to the reduced material parameters. The key technical benefit of our reduction is a substantial acceleration (two orders of magnitude) of the computation of the gradient of the objective function, as well as avoidance of local minima. Our model reduction also accelerates convergence and thus accelerates optimization by two orders of magnitude, rendering our system interactive and enabling intuitive editing of material distributions for complex meshes. To convey the resulting material distributions to the user, we either render them using volume rendering, or realize them in the physical world using a force-feedback haptic interface. We also propose a tetrahedron-based dithering method which can efficiently convert our optimized continuous material distributions into discrete ones. Using FEM simulation, we demonstrate that our method can approximate the force-displacement properties of the continuous materials very well using discrete materials. Our contributions include:

- interactive forward design of elastic materials using sparse inputs, tolerant to inconsistent user input,
- model reduction in the material space with two orders of magnitude speedup,
- user-friendly interface for both linear and nonlinear isotropic material design,
- tetrahedron-based dithering method for discretizing continuous material distributions.

## 2. RELATED WORK

**Material optimization for deformable objects:** Deformable object simulation is a well-studied problem in computer graphics, and the Finite Element Method is one of the most popular methods. For an overview of existing techniques, please refer to [Nealen et al. 2006; Sifakis and Barbič 2012]. In order to create visually-appealing simulations, one must select a proper material model and tune its parameters. To circumvent the complexity of parameter tuning, several authors obtain material parameters by measuring real objects. Methods have been proposed to estimate Young's modulus and/or Poisson's ratio [Schnur and Zabarar 1992; Becker and Teschner 2007; Lee and Lin 2012], cloth stretch and bending parameters [Wang et al. 2011], facial muscle activations [Sifakis et al. 2005], viscoelastic properties [Kauer et al. 2002; Gao et al. 2009] or plasticity parameters [Kajberg and Lindkvist 2004]. Bickel et al. [2009] acquired heterogeneous material distributions of real objects through force and displacement measurements, followed by high-dimensional material optimization of Young's moduli for all the tetrahedra. Bickel et al. [2010], in turn, designed objects that consist of parallel layers of homogeneous materials, and that match the measured forces and displacements, with the goal of replicating real objects. The materials for each layer are selected using a branch-and-bound discrete optimization from the exponential space of  $\#materials^{\#layers}$  designs. Unlike our paper, their work did not demonstrate interactive material editing, and reported running times on the scale of an hour in examples with five layers and nine base materials. We can accommodate general 3D geometry as opposed to a layered structure. Bickel's method [Bickel et al. 2010] produces meso-scale materials from micro-scale cells of basic materials. Our approach is a macro-scale method that optimizes the material distribution at a large scale, capable of rapidly modeling high-resolution continuous material distributions on meshes with thousands of tetrahedra. Our distributions are oriented arbitrarily in space and exhibit significant local detail (Figure 9). Although fabricating objects with continuous material distributions is still impractical nowadays, our tet-based dithering algorithm can convert continuous distributions into discrete distributions in seconds and still closely approximate the target mechanical properties. Furthermore, these data-driven approaches assume that the model to be designed or replicated is already physically available in real life, and that forces and displacements come from measuring a real object. In contrast, the goal of our work is to design new material distributions, of objects that may or may not be physically available. Because we design materials based on arbitrary user input, we have to tackle the situation where no reasonable material distribution can match input constraints, which is less likely to happen in data-driven approaches.

Instead of fitting material parameters, several methods fit global deformation response based on measurements on object boundaries [Pai et al. 2001; Lang et al. 2002; Schoner et al. 2004]. It is also possible to control the deformed shape under known force loads by optimizing the object's rest shape [Skouras et al. 2012]. Recently, example-based methods were proposed to avoid material fitting by guiding simulations using target example shapes [Martin et al. 2011; Schumacher et al. 2012]. Skouras et al. [2013] employed an *offline* optimization process to find actuation forces and discrete material distributions that match given input shapes. In our work, we assume that forces are prescribed by the user, but optimize *continuous* material distributions and optionally shapes. We demonstrate how to apply model reduction to the material distribution, and present an *interactive* design process where we are able to

rapidly iterate many design options and display the resulting shapes and forces using a haptic multi-modal interface. Material optimization for space-time motion editing has also recently been explored by [Li et al. 2014]. They optimized materials to minimize control forces under space-time position constraints, whereas in our method, forces are prescribed as inputs. More importantly, their method does not output world-space materials, but gives the output materials implicitly, by changing global linear modes and frequencies. Our method provides Young’s moduli and Poisson’s ratios for each tetrahedron.

**Model reduction** is a well-known technique in engineering. In computer graphics, model reduction has been used for fast simulation of deformable solids [James and Pai 2002; Hauser et al. 2003; Barbič and James 2005; An et al. 2008; Kim and James 2009; Hildebrandt et al. 2011], fluids [Wicke et al. 2009], shape decomposition [Huang et al. 2009], skinning [Jacobson et al. 2012] and for fast interactive design of animations [Hildebrandt et al. 2012; Barbič et al. 2012]. These methods perform model reduction on the space of deformations of the object, whereas we perform reduction on the material distribution of the object. Also, these methods assumed that the material distribution is known, whereas our goal is to design the material distributions from sparse user input. Lee et al. [2012] estimates elasticity and boundary forces using an iterative optimization framework and accelerates the process with shape reduction. However, in contrast with material reduction which preserves accuracy, shape reduction introduces undesirable inaccuracy into the process. Also, instead of requiring the entire target shape to be specified, our method only requires a sparse user input of displacements and forces on a set of mesh vertices. Starting with an existing high-resolution material distribution, one can also reduce it to a coarser mesh [Kharevych et al. 2009; Nesme et al. 2009]. In addition to requiring known detailed materials, these methods do not try to match given forces and displacements.

**Shape optimization:** Many methods exist to compute high-quality deformation shapes of surfaces and solids, based on user input handles [Igarashi et al. 2005; Botsch et al. 2007; Botsch and Sorkine 2008; Jacobson et al. 2012]. Similarly to one of our examples, Mori and Igarashi [Mori and Igarashi 2007] used a sketching interface to manipulate 3D shapes for plush toys. Different from these methods, we design not just the shape, but augment the handles with force input to also design the material distribution (stiffness) of the object.

**Physical fabrication:** Layer-based additive manufacturing, also known as 3D printing, is a collection of techniques for manufacturing solid objects by sequential delivery of energy and/or material to specified locations in space. Some 3D printers are capable of printing two or more base materials or functionally graded materials into a single printed part (*multi-material printing*). Many multi-material 3D printing methods are available, such as jetting materials from micro-scale inkjet printing nozzles, stereolithography [Han et al. 2010; Zhou et al. 2013], fused decomposition modeling [Khalil et al. 2005], and selective laser sintering [Liew et al. 2002]. These fabrication capabilities enable exciting manufacturing capabilities that were previously impossible. Heterogeneous object fabrication with smooth and versatile material distribution is widely studied [Kou et al. 2006; Huang et al. 2013]. These methods assume that the input material distributions are known; they are typically designed manually by users. However, in order to achieve desired functionality, intelligent design of multi-material distributions is required for future products. While many research papers on 3D printing focused on the fabrication process, we present a method to

design continuous heterogeneous material distributions to conform to given FEM displacement and force inputs. We are unaware of any prior work, in graphics or engineering, that optimizes a continuous 3D solid material distribution across the volume of a geometrically complex three-dimensional mesh, to conform to prescribed FEM displacements and forces. We believe this general problem will find several applications in the future. Once the continuous materials are known, each 3D printer’s driver can then decide how to best realize them given the printer’s technical capabilities. This is similar to how in computer graphics and vision, we optimize pixel R,G,B intensities as continuous signals, even though they are later going to be only displayed at discrete [0-255] levels on a computer monitor. It would be cumbersome to optimize pixels using discrete optimization. Instead, continuous optimization often makes most sense, whereas the discretization task can be left to the specifics of each display device.

### 3. MATERIAL GRADIENTS

In this section, we briefly introduce deformable materials and their gradients with respect to the material parameters. We limit our discussion to small deformations, linear tetrahedral elements and linear isotropic materials parameterized by Young’s modulus and Poisson’s ratio [Shabana 1990]. We later extend our method to support large deformations (Section 7). For a more extensive introduction to solid deformable object modeling we refer the reader to [Sifakis and Barbič 2012]. Given a tetrahedral mesh  $\Omega$  with  $n$  vertices and  $m$  elements, the stiffness matrix  $K_e \in \mathbb{R}^{12 \times 12}$  of element  $e$  is

$$K_e = V_e B_e^T \mathcal{E}_e B_e, \quad (1)$$

where  $V_e$  is volume of element  $e$ ,  $B_e \in \mathbb{R}^{6 \times 12}$  is a constant matrix that only depends on the tetrahedron’s rest shape, and the linear tensor of elasticity  $\mathcal{E}_e \in \mathbb{R}^{6 \times 6}$  is [Shabana 1990]

$$\mathcal{E}_e = \frac{E_e}{(1 + \nu_e)(1 - 2\nu_e)} \begin{bmatrix} \mathcal{E}_e^1 & 0 \\ 0 & \mathcal{E}_e^2 \end{bmatrix}, \quad \text{for} \quad (2)$$

$$\mathcal{E}_e^1 = \begin{bmatrix} 1 - \nu_e & \nu_e & \nu_e \\ \nu_e & 1 - \nu_e & \nu_e \\ \nu_e & \nu_e & 1 - \nu_e \end{bmatrix}, \quad \mathcal{E}_e^2 = (1 - 2\nu_e) I_{3 \times 3}. \quad (3)$$

Here,  $E_e > 0$  is the Young’s modulus defining material elasticity, while  $\nu_e \in (-1, 1/2)$  is the Poisson’s ratio defining material compressibility. We can assemble all element stiffness matrices into the global stiffness matrix  $K \in \mathbb{R}^{3n \times 3n}$ . The internal elastic forces  $f \in \mathbb{R}^{3n}$  then equal  $Ku$ , where  $u \in \mathbb{R}^{3n}$  gives the displacements of vertices away from the rest configuration. Note that the element stiffness matrix  $K_e$  and global stiffness matrix  $K$  are linear functions of scalar  $E_e$ . We can assemble all  $E_e$  and  $\nu_e$  into vectors  $E \in \mathbb{R}^m$  and  $\nu \in \mathbb{R}^m$ , and can therefore write

$$K(E, \nu) = \sum_{e=1}^m \frac{\partial K}{\partial E_e} E_e, \quad (4)$$

where the matrices  $\partial K / \partial E_e$  depend only on  $\nu$  but not on  $E$ . They can be obtained by setting  $E_e$  to 1 in (2). We assume that the mesh is anchored (fixed) at a set of vertices  $\mathcal{F}$ . In our examples,  $\mathcal{F}$  contains sufficient vertices (three vertices that are not on the same line) so that the rigid degrees of freedom are removed. We, however, also demonstrate an unanchored result ( $|\mathcal{F}| = 0$ , Figure 15). The degrees of freedom from  $\mathcal{F}$  are simply removed from the system. For notational convenience, we relabel all quantities ( $n, u, f, K$ ) to refer only to the remaining degrees of freedom.

#### 4. MATERIAL DESIGN

We now describe how we optimize the material based on given forces and displacements. In our work, we only optimize Young's moduli. Poisson's ratio can be made heterogeneous using the Laplace-solve procedure outlined in Section 4.2, but are not optimized in our work. We assume that the distribution of Poisson's ratio is known, and state the design problem for  $E$  as follows.

**Design Problem:** Given user-specified forces  $f \in \mathbb{R}^{3n}$  and displacements  $\bar{u} \in \mathbb{R}^{3c}$  on vertices from some given set  $\Gamma$ ,  $|\Gamma| = c \geq 1$  (the *position handles*), find a smooth (heterogeneous) material distribution  $E \in \mathbb{R}^m$  such that when the mesh is statically loaded with  $f$ , the displacements on  $\Gamma$  are  $\bar{u}$ .

If the user does not specify forces  $f \in \mathbb{R}^{3n}$  on every vertex, we assume the non-specified forces to be zero, or they can be any given external force such as gravity. For any particular material distribution  $E$ , the mesh displacements  $u$  are unique and are obtained by solving the linear system  $K(E)u = f$ . Therefore, for a general  $E$ , the displacements on  $\Gamma$  do not match  $\bar{u}$  and we need to design a procedure to discover  $E$ . By re-ordering the degrees of freedom so that set 2 refers to  $\Gamma$  and set 1 to the rest, we have

$$\begin{bmatrix} K_{11}(E) & K_{12}(E) \\ K_{21}(E) & K_{22}(E) \end{bmatrix} \begin{bmatrix} \hat{u} \\ \bar{u} \end{bmatrix} = f = \begin{bmatrix} \hat{f} \\ \bar{f} \end{bmatrix}. \quad (5)$$

We can now use block-Gaussian elimination to arrive at

$$\tilde{K}(E)\bar{u} = \bar{f} - K_{21}(E)K_{11}^{-1}(E)\hat{f}, \quad \text{for} \quad (6)$$

$$\tilde{K}(E) = K_{22}(E) - K_{21}(E)K_{11}^{-1}(E)K_{12}(E), \quad \text{and} \quad (7)$$

$$\hat{u} = K_{11}^{-1}(E)(\hat{f} - K_{12}(E)\bar{u}). \quad (8)$$

Observe that  $\tilde{K}(E) \in \mathbb{R}^{3c \times 3c}$  is a (small) dense matrix that encodes the relationship between displacements of constrained vertices in  $\Gamma$  and the resulting internal elastic forces on them. We note that this procedure is commonly referred to as *static condensation* [Bro-Nielsen and Cotin 1996; Cotin et al. 1999; James and Pai 2001]. However, in prior work it has been employed in simulation applications where  $E$  is fixed and known, whereas we will optimize for  $E$ . For smoothness of  $E$ , we measure it as  $E^T L E$ , where  $L$  is the discrete mesh Laplacian for scalar fields that are defined on tetrahedra of a mesh (such as our  $E$ ) [Zhang 2004; Bickel et al. 2009]. The discrete mesh Laplacian is given as

$$(LE)_i = \sum_{j=1}^m \omega_{i,j}(E_i - E_j) \quad (9)$$

where  $\omega_{i,j} = 1$  if tetrahedra  $i$  and  $j$  share a vertex, and 0 otherwise. Matrix  $L$  is positive semi-definite and rank-deficient; its nullspace is 1-dimensional and consists of homogeneous distributions. It can be easily shown that

$$E^T L E = \sum_{1 \leq i < j \leq m} \omega_{i,j}(E_i - E_j)^2, \quad (10)$$

and therefore our smoothness energy penalizes Young's modulus differences in adjacent tets. We now re-cast the Design Problem as

$$\min_{E, \bar{u}} \frac{1}{2} E^T L E + \frac{\alpha}{2} \|\tilde{f}(E) - \bar{f}\|_W^2 + \frac{\alpha\beta}{2} \|\bar{u} - \bar{u}\|^2, \quad (11)$$

$$\text{where } \tilde{f}(E) = \tilde{K}(E)\bar{u} + K_{21}(E)K_{11}^{-1}(E)\hat{f} \in \mathbb{R}^{3c}, \quad (12)$$

and where  $\alpha, \beta > 0$  are weights that define the importance of matching forces and displacements on  $\Gamma$ , respectively. Note that

forces outside of  $\Gamma$  (forces  $\hat{f}$ ) are matched automatically, by design of  $\tilde{K}$ . In some applications, we omit the last term, i.e., we impose a hard constraint that the handle vertices must meet prescribed positions exactly,  $\bar{u} = \bar{u}$ . We also tried replacing the force term in (11) with a hard constraint, however this resulted in optimizations that were too severely constrained and not convergent. The norm on the force term,  $\|x\|_W^2 = x^T W x$ , is weighted with a diagonal matrix  $W \in \mathbb{R}^{3c \times 3c}$  where diagonal entries  $3i, 3i+1, 3i+2$  equal  $\bar{f}_{max}^2 / \|\bar{f}_i\|^2$  if  $\|\bar{f}_i\| \geq \epsilon \bar{f}_{max}$  and  $1/\epsilon^2$  otherwise, for  $i = 1, \dots, c$ . Here,  $\bar{f}_{max}$  is the maximum value of  $\|\bar{f}_i\|$  and we use  $\epsilon = 10^{-8}$ . Matrix  $W$  is useful when the prescribed forces on the various handles differ by a large amount. Suppose a target force on handle A is 0.5N, but 100N on another handle B. Under an identity  $W$ , the optimizer would assign equal importance to an error of 1N on either handle, causing a large *relative* force error at A. Matrix  $W$  eliminates this bias by assigning greater importance to force errors at the handles where the prescribed force is small.

Observe that  $E$  only appears in the first two terms of (11), whereas  $\bar{u}$  only appears in the last two terms. Therefore, we can solve the optimization problem (11) by alternating the material optimization

$$\min_E \frac{1}{2} E^T L E + \frac{\alpha}{2} \|\tilde{f}(E) - \bar{f}\|_W^2 \quad (13)$$

with the handle position optimization

$$\min_{\bar{u}} \frac{1}{2} \|\tilde{f}(E) - \bar{f}\|_W^2 + \frac{\beta}{2} \|\bar{u} - \bar{u}\|^2. \quad (14)$$

Material optimization is performed under a fixed  $\bar{u}$  obtained during the last handle position optimization, and handle position optimization is performed under a fixed  $E$  obtained during the last material optimization. Note that this technique is commonly referred to as *block coordinate descent*, and each of these two optimizations decreases (or at least guarantees not to increase) the overall objective (11). Such a decomposition of the problem into two optimizations is intuitive because the user can control how often each of these two steps is performed. It also makes it possible to tune the weights  $\alpha$  and  $\beta$  in isolation.

Material optimization (14) is nonlinear in  $E$  because  $\tilde{K}(E)$  depends nonlinearly on  $E$ , due to inverse of  $K_{11}(E)$ . Therefore, we solve this unconstrained optimization problem using the conjugate gradient optimizer [Press et al. 2007]. The optimizer requires evaluations of objective functions (13) for arbitrary  $E$ , as well as its gradient with respect to  $E$ . The gradient can be obtained using the chain rule and by differentiating  $K_{11}(E)K_{11}^{-1}(E) = I$  with respect to  $E$ , and equals

$$LE + \alpha \left( \frac{d\tilde{f}(E)}{dE} \right)^T (\tilde{f}(E) - \bar{f}), \quad (15)$$

$$\text{where } \frac{d\tilde{f}(E)}{dE_e} = [K_{21}K_{11}^{-1} - I] \frac{dK}{dE_e} \begin{bmatrix} K_{11}^{-1}(K_{12}\bar{u} - \hat{f}) \\ -\bar{u} \end{bmatrix}, \quad (16)$$

for  $e = 1, \dots, m$ . The conjugate gradient optimizer works by forming a search direction obtained by evaluating the gradient and then properly subtracting previously explored directions. It then performs a line search along this direction. After each iteration, we update  $K(E)$  and  $\tilde{K}(E)$ , and visualize the current  $E$  to the user (Figure 1). Because the optimizer is iterative, the user can interrupt it any time, for example, when they are visually satisfied with the result. The ability to visualize and interrupt at any time is also very useful for parameter tuning, as it makes it possible to quickly abort a non-convergent optimization and re-tune parameters. We can also

stop the optimizer once it exceeds the maximal number of iterations or the material distribution converges.

The handle position optimization function in (14) is quadratic in  $\tilde{u}$  and therefore the optimization can be performed by solving a single dense linear system

$$\left(\tilde{K}(E)^T \tilde{K}(E) + \beta I\right) \tilde{u} = \tilde{K}(E)^T (\tilde{f} - K_{21}(E) K_{11}(E)^{-1} \hat{f}) + \beta \tilde{u}. \quad (17)$$

Throughout the paper, we report weighted relative force errors  $\epsilon_f$ , normal force errors  $\epsilon_f^n$  and handle position errors  $\epsilon_u$  as

$$\epsilon_f = \|\tilde{f}(E) - \tilde{f}\|_W / \|\tilde{f}\|_W, \quad (18)$$

$$\epsilon_f^n = \sqrt{\sum_{i=1}^c w_i \left(n_i^T (\tilde{f}_i(E) - \tilde{f}_i)\right)^2} / \sqrt{\sum_{i=1}^c w_i \left(n_i^T \tilde{f}_i\right)^2}, \quad (19)$$

$$\epsilon_u = \|\tilde{u} - \tilde{u}\| / \|\tilde{u}\|, \quad (20)$$

where  $w_i$  are the diagonal entries of matrix  $W$  (Equation 11). Displacements  $\tilde{u}$  depend on  $E$  and  $\tilde{u}$  (Equation 8) and are recomputed each time either of them changes.

We note that the Design Problem can be cast into a mathematical form in many ways, as there is a choice of hard vs soft constraints on forces or displacements. In our optimizations (11) and (13), we have chosen to use a soft penalty force term, i.e., we do not impose hard force constraints. This is because there are input forces and displacements for which there is no reasonable material distribution to match them. Under such inputs, the optimizer will either be unable to decrease the force term in (11) below a certain value, or, if the force weight  $\alpha$  is increased too much, will produce Young's moduli with negative values. Negative Young's moduli should obviously be avoided and we treat such results as failures. The main reason for such difficulties is that in isotropic materials, normal strains  $\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}$  only cause normal stresses  $\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$  (no shear stress), and therefore elastic forces cannot be arbitrarily rotated. Consider, for example, a beam loaded with a longitudinal force. It is not reasonable to command an internal elastic force at, for example, an angle of 90 degrees to the displacement (Figure 2, a). Because our optimization energy treats forces and displacements as soft constraints, it can still produce plausible outputs (Figure 2, b), whereas exact displacements (Figure 2, c) require a large error in force. We also implemented the Design Problem formulation where forces are met exactly but displacements are soft, presented in [Bickel et al. 2009; Bickel et al. 2010]. Such force-based methods work well when measuring real objects, where the forces and displacements are automatically consistent (modulo measurement error). For our forward design application where the user input may be arbitrary, soft penalty force and displacement terms improve robustness. With exact forces, the optimizer has to adjust materials so that under the fixed vertical force, the handle moves perpendicularly. No such solution exists and best that the optimizer can do is make the material very stiff, so that the handle does not move downwards and further distance itself from the goal (Figure 2, d).

#### 4.1 Choice of Laplacian discretization

The Laplace operator is used to enforce a smooth material distribution. We use the discrete mesh Laplacian ("umbrella operator") given by Equation 9, but our method is not specific to any specific Laplacian discretization. Prior work [Bickel et al. 2009] used a bi-Laplacian energy  $\|LE\|^2$ , which can be obtained by replacing  $L$  with  $L^T L$ . For meshes with non-uniform tetrahedron sizes, one

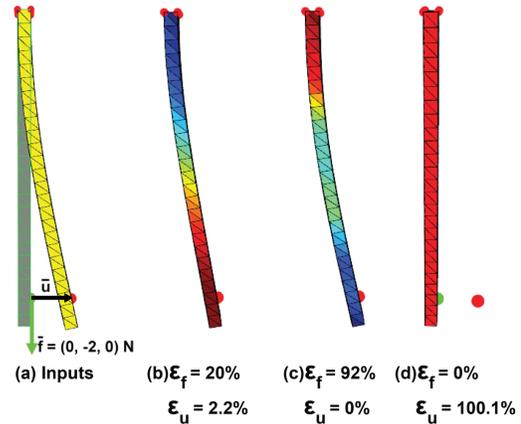


Fig. 2. **Relaxing both positions and forces is advantageous:** (a) Inconsistent input where force and displacement form a 90 degree angle. (b) Our method, with both material (13) and handle position optimization (14). (c) Our method, with material optimization only. (d) Exact force with a soft position term [Bickel et al. 2009; Bickel et al. 2010]. Red material color in (d) is not to scale with (b,c). The output in (d) is nearly homogeneous and approximately 200× stiffer than in (b,c). Full optimization. Green and red points are on top of each other in (b,c).

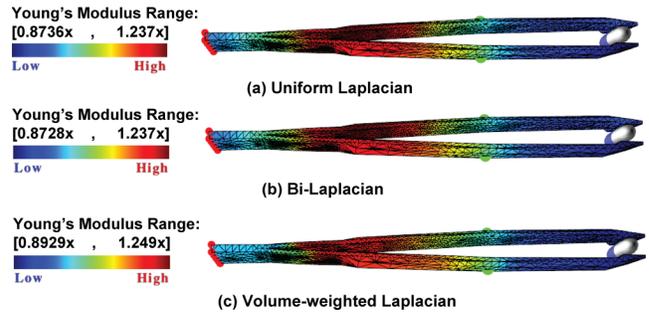


Fig. 3. **Comparison of different Laplacian discretizations.** Same force/displacement input. There are only minor differences in the optimization results.

could use a local volume-weighted Laplacian  $L_v$ , defined as

$$(L_v E)_i = \sum_{j=1}^m \omega_{i,j} (E_i - E_j), \quad (21)$$

where  $\omega_{i,j} = \frac{V_i + V_j}{2}$  if tetrahedra  $i$  and  $j$  share a vertex, and 0 otherwise. We compared all of these Laplacian discretizations, and the results were similar (Figure 3). If needed, one can simply replace  $L$  with  $L^T L$  or  $L_v$  everywhere in our method.

#### 4.2 Initial guess

We considered two initial guesses to our optimization. The first approach is to start with a homogeneous material with some known Young's moduli corresponding to a real material. The second approach is to prescribe Young's modulus at a few key vertices, and then smooth the Young's modulus across the entire mesh. Given the Young's modulus values at the handles, the distribution over

the mesh is computed by minimizing the optimization problem

$$\begin{aligned} \min_E \quad & \frac{1}{2} E^T L E \\ \text{subject to} \quad & S_i E = \bar{E}_i, \quad i = 1, \dots, c, \end{aligned} \quad (22)$$

where  $S_i$  is a selection matrix that selects the 1-ring of elements (denote their number by  $m_i$ ) around  $i$ -th handle vertex, and  $\bar{E}_i \in \mathbb{R}^{m_i}$  is the prescribed Young's modulus at handle  $i$ , with the same value replicated  $m_i$  times. We solve the linearly constrained quadratic optimization problem (22) using the standard Lagrange multiplier approach. Rank-deficiency of  $L$  is handled automatically by the presence of constraints. In our tweezers example, both approaches worked equally well. In the shoe example, there are many handles attached to each tet, so the first approach was preferable. The second approach accelerated convergence in our real-time demos (teddy bear, bunny). In our examples, we prescribe the target forces  $\bar{f}$  by setting force multipliers  $\lambda$  with respect to the default forces under default material. The prescribed Young's modulus values in these examples were simply the product of the default Young's modulus with the force multipliers set for that handle by the user.

## 5. MODEL REDUCTION

Optimization (13) is high-dimensional as every tetrahedron has an independent Young's modulus value. Therefore, optimization takes a long time (hours for models with a few thousands tetrahedra), and it is therefore not easily possible for the user to re-tune optimization parameters. This inhibits the design process, as only a few parameter values or inputs can be pursued. Furthermore, full optimization tends to progress more slowly per iteration (see Figure 5) and the optimizer often gets stuck in local minima, frequently producing negative Young's moduli (see Table I). We significantly accelerate optimization (13) by restricting the material distribution to a subspace of  $r$  material modes. Often, only a handful of material modes can offer a drastic improvement over a homogeneous distribution (which is equivalent to  $r = 1$ ) (Table I). Unlike prior model reduction methods [James and Pai 2002; Barbič and James 2005; An et al. 2008; Lee and Lin 2012], we do not perform shape reduction. In our method, both full and reduced optimization try to minimize the same energy (13). Therefore our reduced optimization can be seen as an improved optimizer to solve problem (13). Although we apply reduction to (13), other Design Problem formulations [Bickel et al. 2009; Bickel et al. 2010] would also benefit from it.

In choosing the material modes, we pursue *a priori* model reduction, i.e., we select material modes without needing any material data or specializing to any concrete user input. Such a construction is general, versatile, and easy to use as the user does not need to prepare data or tweak the subspace. For the material modes, we essentially want smooth scalar functions over the entire mesh which can progressively build more and more localized material distributions. A natural choice is to use the eigenvectors of the Laplacian operator  $L$  [Lévy and Zhang 2010; Zhang et al. 2010]. Note that the continuous Laplacian in 1D is the operator that computes the second derivative of the function. Its eigenvectors are the constant function, and then sin and cos functions with progressively larger frequencies. Similar behavior is observed in the mesh Laplacian: its first eigenvector is the constant material distribution (eigenvalue is zero), and then higher modes permit progressively more locality (see Figure 4). To obtain the material modes, we solve the general-

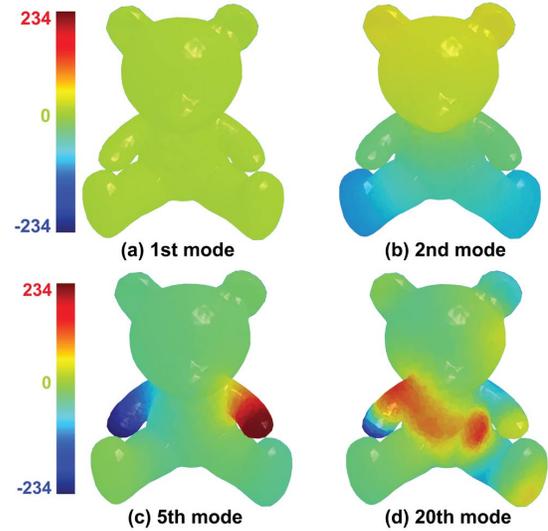


Fig. 4. **Volume-rendered material modes.** Teddy bear example. The first mode is constant across the mesh (homogeneous material), the second mode varies nearly linearly across the mesh, and higher modes provide increasing local detail.

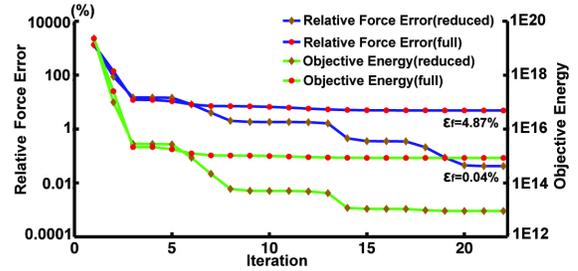


Fig. 5. **Reduced optimization converges faster than full optimization.** In addition to faster progress per iteration, reduced optimization is also 66× faster computationally per iteration. Tweezers example, 10 material modes. Log scale. The conjugate gradient optimizer sometimes makes slow progress (plateaus), followed by a period of rapid progress.

ized eigenproblem

$$L y_j = \mu_j V y_j, \quad (23)$$

where  $V \in \mathbb{R}^{m \times m}$  is the diagonal weighting matrix with per-tetrahedron volumes  $V_e$  on the diagonal. We assemble the first  $r$  eigenvectors ( $y_1, \dots, y_r$ ) as columns of the *material basis matrix*  $\Phi \in \mathbb{R}^{m \times r}$ , where each column is one mode. The spatial material distribution is expressed as  $E = \Phi z$ , where  $z \in \mathbb{R}^r$  is the *reduced material vector*. We note that the entries of  $\Phi$  may be negative. This is not a problem because the first mode's (homogeneous material) reduced coordinate  $z_1$  is initialized to a large value and in practice remains sufficiently large to prevent negative materials. Although we use eigenvectors of the Laplace operator for the modes, arbitrary scalar mesh functions could be used instead. They could be painted by an artist, or obtained from existing material data. Layered structures [Bickel et al. 2010] or other homogeneous regions are a special case where each mode is 1 on a region and zero elsewhere, and where  $L$  is modified so that changes in the material across the region boundaries are not penalized in (11).

Applying  $E = \Phi z$  to optimization (13), the reduced optimization problem becomes

$$\min_{z, \tilde{u}} \frac{1}{2} z^T Q z + \frac{\alpha}{2} \|\tilde{f}(z) - \bar{f}\|^2 + \frac{\alpha\beta}{2} \|\tilde{u} - \bar{u}\|^2, \quad (24)$$

$$\text{for } \tilde{f}(z) = \tilde{K}(z)\tilde{u} + K_{21}(z)K_{11}^{-1}(z)\hat{f} \in \mathbb{R}^{3c}, \quad (25)$$

where  $Q = \Phi^T L \Phi \in \mathbb{R}^{r \times r}$  is the reduced Laplacian matrix. Matrix  $Q$  is diagonal and its entries consist of the eigenvalues  $\mu_j$ . Matrices that depend on  $z$  are evaluated by forming full-space material distributions. For example,  $\tilde{K}(z)$  is computed by forming  $E = \Phi z$  and evaluating  $\tilde{K}(E)$ . The gradient of the objective function (24) with respect to  $z$  is

$$Qz + \alpha \left( \frac{d\tilde{f}(z)}{dz} \right)^T (\tilde{f}(z) - \bar{f}), \quad \text{where} \quad (26)$$

$$\begin{aligned} \frac{d\tilde{f}(z)}{dz} &= \sum_{e=1}^m \frac{d\tilde{f}}{dE_e} \Phi^e \stackrel{(16)}{=} H \sum_{e=1}^m \left( \frac{dK}{dE_e} \begin{bmatrix} K_{11}^{-1}(K_{12}\tilde{u} - \hat{f}) \\ -\tilde{u} \end{bmatrix} \right) \Phi^e = \\ &= HG\Phi, \quad \text{for } H = \begin{bmatrix} K_{21}K_{11}^{-1} & -I \end{bmatrix}. \end{aligned} \quad (27)$$

Here,  $\Phi^e$  is  $e$ -th row of  $\Phi$ , and column  $e$  of matrix  $G \in \mathbb{R}^{3n \times m}$  equals

$$\frac{dK}{dE_e} \begin{bmatrix} K_{11}^{-1}(K_{12}\tilde{u} - \hat{f}) \\ -\tilde{u} \end{bmatrix}. \quad (28)$$

Note that all but 12 entries in  $dK/dE_e$  are zero, and therefore  $G$  can be computed efficiently by forming localized matrices  $dK/dE_e \in \mathbb{R}^{12 \times 12}$ , which can be precomputed. In order to compute the gradient, we need to perform a multiplication with  $H$ , i.e., solve a linear system with  $K_{11}(E)$ . We do so by computing Cholesky decomposition of  $K_{11}(E)$ , and then solve  $r$  linear systems for the right hands in  $G\Phi$ . The systems can be solved in parallel. Reduction provides the benefit of only needing to solve  $r$  linear systems, as opposed to  $m \gg r$  systems without reduction (Equation 16), which makes reduced iterations significantly faster, e.g., 70× speedup with 10 modes compared to unreduced iterations in the tweezers example (Table I). The multiplication  $E = \Phi z$  is performed efficiently once per iteration using Intel MKL library and represents negligible computational overhead, as the computation time is dominated by solving linear systems given by  $K_{11}(E)$ .

## 6. MATERIAL DISCRETIZATION

In this section, we demonstrate how our continuous distributions could be converted to discrete material distributions, and analyze the accuracy of this process using FEM simulation. Such *material dithering* is similar to how continuous pixel color in image processing can be dithered on displays with a limited number of colors [Floyd and Steinberg 1976]. The resulting material distributions could be useful with a future 3D manufacturing process. In contrast to the existing dithering methods [Vidimčič et al. 2013; Liu et al. 2004; Cho et al. 2003] for multi-material printing which apply dithering to slices or voxels, we give a method to dither continuously-defined materials on a tetrahedral mesh. This capability is important because simulations in computer graphics often use tetrahedral meshes. Furthermore, existing commercially available multi-material printers (the Objet Connex family) require each material region (potentially irregularly shaped or disconnected) to be specified by its triangle mesh boundary. Such a triangulated boundary could be computed easily from a tet mesh. We demonstrate how to dither a single, continuous quantity (Young’s modulus); the same

algorithm could also be applied, say, to dither Poisson’s ratios. We assume that a library of  $M \geq 2$  base materials is available. The base materials must respect the minimum and maximum values of the target Young’s modulus continuous distribution  $E$ , as it is impossible to generate a continuous material outside of the convex combination of the available base materials.

Our algorithm spreads the residual quantization error of a tet onto its neighboring tets, to be processed later. Each tet is assigned a target continuous Young’s modulus value. At the beginning of the algorithm, this value is the input continuous Young’s modulus value, obtained, say, using the optimization methods described in this paper. During the algorithm, these values change as the tets spread their quantization error to their neighbors. We maintain a list of the tets that have already been dithered, and process the tets using a priority queue. The queue ensures that the tets are visited in a breadth-first pattern. Initially, we push a boundary tet into the queue; we pick the tet whose centroid has the largest  $y$ -coordinate. Then we process the tets one by one, always popping the first tet (call it  $i$ ) from the queue. We assign it a discrete material  $\bar{E}_i$  with the smallest absolute value of the residual  $\varepsilon_i = E_i - \bar{E}_i$ , where  $E_i$  is the current target continuous value, and  $\bar{E}_i$  is the closest discrete material. We then mark  $i$  as dithered and spread the residual  $\varepsilon_i$  to the neighboring (1-ring) tets that have not been dithered yet, as follows. For each tet  $j$  in the 1-ring set of vertex-neighboring, edge-neighboring and face-neighboring tets of tet  $i$  (call them  $\mathcal{V}(i), \mathcal{E}(i), \mathcal{F}(i)$ , respectively), we add  $\varepsilon_i \frac{w_v V_j}{\chi}, \varepsilon_i \frac{w_e V_j}{\chi}, \varepsilon_i \frac{w_f V_j}{\chi}$  to the current target Young’s moduli values, respectively, where

$$\chi = w_v \sum_{k \in \mathcal{V}(i)} V_k + w_e \sum_{k \in \mathcal{E}(i)} V_k + w_f \sum_{k \in \mathcal{F}(i)} V_k. \quad (29)$$

Here,  $V_j$  is the volume of element  $j$ , and  $\chi$  was chosen so that the sum of the distributions is  $\varepsilon_i$ . Parameters  $w_v, w_e, w_f$  are the weights for error distribution to the three types of neighboring tets. In our examples, we set  $w_v = 1, w_e = 2$  and  $w_f = 3$  to bias distributing the residual error based on the number of tet shared vertices (vertex-neighbor=1, edge-neighbor=2, face-neighbor=3). Notice that the residual error distribution is also weighted by element volume  $V_j$  to accommodate non-uniformly meshed objects. We push all the undithered neighboring tets onto the queue, in the order of  $\mathcal{F}(i), \mathcal{E}(i), \mathcal{V}(i)$ . The process is terminated when all the tets have been assigned a discrete material, i.e, no more tets are left in the queue.

Figure 6 shows dithering using two discrete materials. When applying the same handle displacements on the two material distributions, we only observed 2.5% and 1% force differences (defined in (18)). The dithering process takes 8 and 7 seconds. We also compare our method to the discrete optimization method of [Bickel et al. 2010], where we treat each tet as one layer. The layers follow the same sequence as our tet-based dithering method. Without applying branch-and-bound and/or clustering, the design space of [Bickel et al. 2010] is exponential with  $M^m$  choices ( $M = 2, m = 10,046$  for the teddy bear and  $M = 2, m = 9577$  for the tweezers in Figure 6), which is intractable. The branch-and-bound algorithm with clustering [Bickel et al. 2010] reduces the complexity to  $KM^2m$ , where  $K$  is the number of clusters (we use  $K = 20$ ). However, the running time is significantly (5,000×) longer, and the error is 10× larger than in our method (Figure 6, c, f). We also observe that for the Young’s modulus differences to the optimal continuous distribution  $\|\sum_e (E_e - E_e^c)\|$ , our dithering output is 300× and 296× smaller than the discrete optimization result.

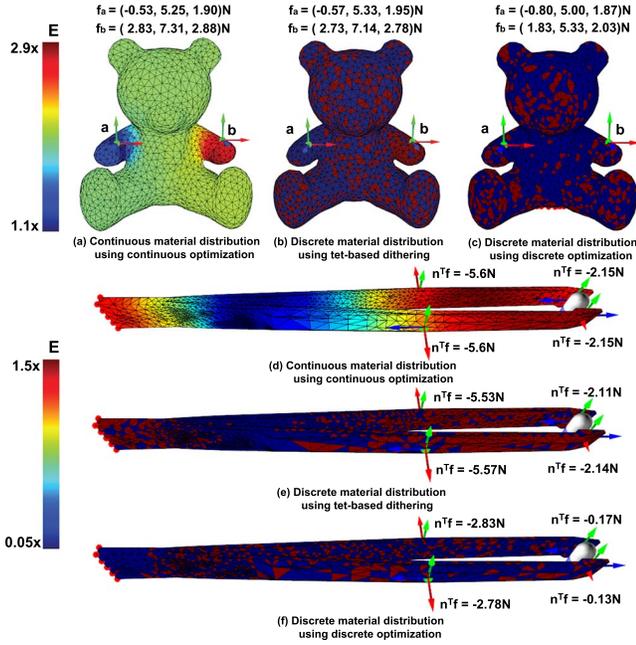


Fig. 6. **Material dithering and discrete optimization.** (a) and (d) Continuous material distribution optimized using our method. (b) and (e) Discrete material distribution using tet-based dithering using two base materials. For the teddy bear example,  $E$  is  $1.3\times$  and  $2.9\times$  larger than the default Young’s modulus, respectively. For the tweezers example,  $E$  is  $0.05\times$  and  $1.5\times$  larger than the default Young’s modulus, respectively. Under the same input handle displacements to (a) vs (b), (d) vs (e), the force error is only 2.5% and 1.0%. The dithering process takes 8 and 7 seconds, respectively. (c) and (f) Discrete material distribution using discrete optimization [Bickel et al. 2010]. Same base materials as in (b). Discrete optimization times are 13 and 12 hours, and the force error is 23.9% and 57.3%, respectively.

## 7. LARGE DEFORMATIONS

In previous sections, we focused on the linear isotropic material, which is limited to small deformations. However, we notice that many of the common nonlinear isotropic material models, including Saint Venant Kirchhoff material [Bonet and Wood 2008], corotational materials [Müller and Gross 2004] or neo-Hookean materials [Bonet and Wood 2008], have the property that the stiffness matrix  $K(E, u)$  depends linearly on  $E$ . This makes it possible to extend our optimization framework to accommodate nonlinear materials under large deformations, and accelerate the process with material reduction. The objective is exactly the same as in Equations (11),(24). However, unlike with the linear material model, we have to solve for the deformation shape  $u$  iteratively using Newton’s method,

$$\begin{bmatrix} K(E, u^{(i)}) & \bar{S}^T \\ \bar{S} & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \bar{\lambda} \end{bmatrix} = - \begin{bmatrix} f_{\text{int}}(E, u^{(i)}) + \bar{S}^T \bar{\lambda}^{(i)} - f_{\text{ext}} \\ \bar{S} u^{(i)} - \bar{u} \end{bmatrix}, \quad (30)$$

$$u^{(i+1)} = u^{(i)} + \Delta u, \quad \bar{\lambda}^{(i+1)} = \bar{\lambda}^{(i)} + \Delta \bar{\lambda}, \quad (31)$$

where  $\bar{S} \in \mathbb{R}^{c \times 3n}$  selects the handle DOFs, and  $i$  denotes the iteration index. We initialize the iteration with  $\bar{\lambda}^{(0)} = 0$ , and by setting  $u^{(0)}$  to the previous handle deformation. In order to obtain the gra-

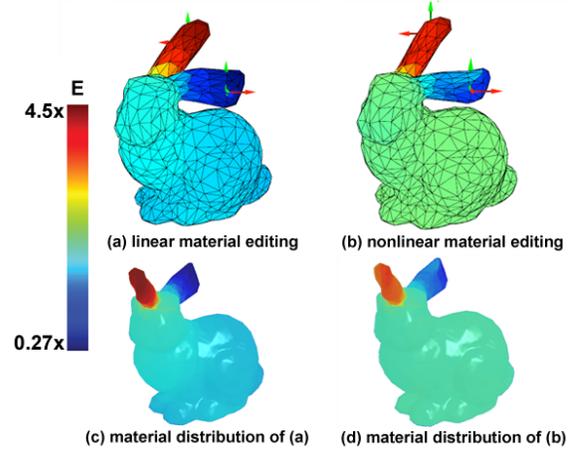


Fig. 7. **Large deformation material optimization.** Editing the deformations under linear isotropic (a,c) and nonlinear St. Venant-Kirchhoff (b,d) materials, respectively. In both cases, the two ears are made  $4.0\times$  and  $1.0\times$  stiffer. Optimizations took 18 and 24 seconds to converge in (a) and (b), respectively. When we apply the linearly-optimized material (c) onto nonlinear StVK simulation, the force error is 15.1%. Force error of nonlinearly-optimized material (d) is significantly smaller, 3.2%.

dient  $\partial \tilde{f} / \partial E$ , we first need to define a function  $G \in \mathbb{R}^{3n}$ ,

$$G(E, \hat{f}, \tilde{f}, \hat{u}, \tilde{u}) = f_{\text{int}} \left( \begin{bmatrix} \hat{u} \\ \tilde{u} \end{bmatrix}, E \right) - \begin{bmatrix} \hat{f} \\ \tilde{f} \end{bmatrix} = 0. \quad (32)$$

By differentiating with respect to  $E_e$  on both sides, we obtain a linear system

$$\begin{bmatrix} \frac{\partial G}{\partial \hat{u}} & \frac{\partial G}{\partial \tilde{f}} \end{bmatrix} \begin{bmatrix} \frac{\partial \hat{u}}{\partial E_e} \\ \frac{\partial \tilde{f}}{\partial E_e} \end{bmatrix} = - \frac{\partial G}{\partial E_e}, \quad (33)$$

$$\frac{\partial G}{\partial \hat{u}} = \begin{bmatrix} K_{11}(E, u) \\ K_{21}(E, u) \end{bmatrix}, \quad \frac{\partial G}{\partial \tilde{f}} = \begin{bmatrix} 0 \\ -I \end{bmatrix}, \quad (34)$$

$$\frac{\partial G}{\partial E_e} = \frac{1}{E_e} f_{\text{int}}([0 \dots E_e \dots 0]^T, u), \quad (35)$$

where the terms  $\frac{\partial G}{\partial \hat{u}}, \frac{\partial G}{\partial \tilde{f}}, \frac{\partial G}{\partial E_e}, \frac{\partial G}{\partial \tilde{f}}$  are 0 and excluded from the system. Note that we need to solve  $m$  linear systems while we only need to do the Cholesky decomposition once. We apply material reduction by multiplying  $\partial G / \partial E$  with  $\Phi$ . Thus, we now need to solve  $r$  linear systems, where the right hand sides are columns of  $\frac{\partial G}{\partial E} \Phi \in \mathbb{R}^{3n \times r}$ . Figure 7 analyzes the performance and accuracy of our nonlinear material optimization.

## 8. RESULTS

We demonstrate four application areas of our method: precise control of contact forces (tweezers example), control of contact pressure distributions under large contact areas for ergonomics (foot and shoe), haptic design of heterogeneous soft objects (teddy bear) and design of material distributions for animation (bunny).

Table I. Comparison to unreduced optimization

$r$	$\lambda = 0.05$					$\lambda = 0.1$			$\lambda = 0.2$					$\lambda = 5$				
	time	error	#iter	spT	spI	time	error	#iter	time	error	#iter	spT	spI	time	error	#iter	spT	spI
1	0.6s	67.1%	1	4415	119	0.6s	63%	1	0.6s	55.6%	1	6643	117	0.6s	55%	1	16625	124
2	5.5s	67%	5	482	65	4.4s	62%	4	3.2s	55.3%	3	1246	66	2.5s	54%	3	3990	89
3	6.5s	0.4%	7	408	77	4.9s	0.4%	5	4.89s	0.29%	5	815	72	4.7s	48%	4	2122	63
5	18s	0.49%	28	147	114	12s	0.49%	12	4.84s	0.44%	5	824	72	13.3s	0.37%	16	750	90
10	13s	0.48%	12	204	66	22s	0.5%	20	7.8s	0.5%	8	511	72	13.5s	0.4%	16	739	88
50	27s	0.3%	23	98	61	21.4s	0.4%	19	6.9s	0.4%	6	578	61	12.9s	0.35%	15	773	87
full	2649s	0.47%	37	1	1	1630s	0.47%	22	3986s	0.5%	57	1	1	9975s	1.33%	134(F)	1	1

Tweezers example (Figure 1), optimized under varying contact forces and numbers of material modes  $r$ . Time gives the optimization time, #iter is number of conjugate gradient iterations of Equation 13. spT=total speedup, spI=speedup per iteration,  $\lambda$  is the multiplicative factor of changing the normal contact force on the pill. Error is the achieved relative normal force error (defined in (18)), weighted with matrix  $W$  (13) to ensure the small target forces are actually met and not masked by large forces. For brevity, we omitted speedups for  $\lambda = 0.1$ ; they can be computed from the provided data.

## 8.1 Precise control of contact forces

In our first example, we optimize the Young’s modulus distribution of a pair of realistically-sized (8cm) *tweezers* made of aluminum (3,024 vertices, 9,577 tetrahedra, Figure 1). Our design goal is that when the human hand applies realistic prescribed normal forces to the center of each side of the tweezers, both the center and the end deform by prescribed amounts, and the tweezers pick up a pill with a prescribed normal contact force. Normal forces can be incorporated into (11) by replacing forces with the dot product of forces with a normal  $n_j$  at each handle vertex  $j$ , i.e.,

$$\sum_{j=1}^c \left( n_j^T (\bar{f}_j(E) - \bar{f}_j) \right)^2. \quad (36)$$

To set up our experiment, we first determined that when we apply normal forces of 4.9N to the center of each side of the tweezers (in opposite directions on each side; vertices A, B in Figure 1), the tweezers deform, under a homogeneous aluminium distribution, just enough to touch the pill with zero contact force. This gives us 3D displacements  $\bar{u}_C, \bar{u}_D$  (approximately 4mm in the normal direction) on two vertices (C, D) on each side of the end of the tweezers. We then increase the applied forces on A, B to 5.6N while keeping C, D fixed, essentially grasping the pill and causing a contact normal force of 0.43N to C and D. We record the 3D displacements  $\bar{u}_A, \bar{u}_B$  of A, B (approximately 2.2mm in the normal direction). We then use our method to design a material distribution that meets both the prescribed displacements  $\bar{u}$  and prescribed normal forces on A, B, C, D:

$$n_A^T \bar{f}_A = n_B^T \bar{f}_B = 5.6\text{N}, \quad n_C^T \bar{f}_C = n_D^T \bar{f}_D = 0.43\lambda\text{N}. \quad (37)$$

We repeat the experiment for several values of  $\lambda$  and numbers of material modes  $r$  (see Table I). We only use material optimization (Equation 13,  $\alpha = 10^5$ ; no handle position optimization). Initial guess is the initial homogeneous material (aluminium,  $E = 70\text{GPa}$ ,  $\nu = 0.35$ ). We stopped the optimization when the relative force error dropped below 0.5%, or, when this did not happen, at the first iteration that was followed by a long plateau where the optimizer was unable to reduce the force error further. It can be seen that our method is able to reach the prescribed forces very closely; the relative force error  $\epsilon_f^n$  is under 0.5%. It can be also seen that the number of conjugate gradient iterations generally increases with  $r$ . In this example, approximately five modes are sufficient to solve the problem reasonably well. For  $\lambda = 5$ , full optimization converged to a suboptimal minimum (marked with “F” in Table I) where some tets were assigned negative materials. Our model reduction method offers substantial benefits over full optimization. Because the re-

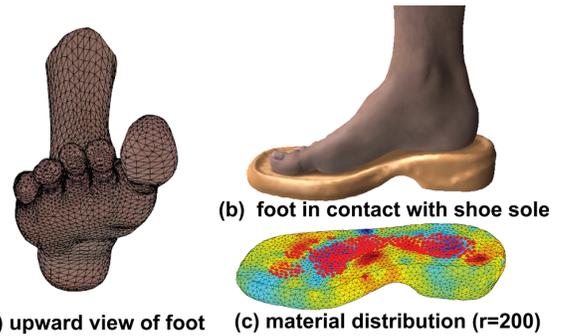


Fig. 8. **Rigid foot in contact with a deformable shoe.** Part (c) shows the 384 contact vertices (in red) and the optimized material distribution that yields constant normal pressure on the foot.

sults are obtained two orders of magnitude faster, it is easily possible for the user to iterate parameters. Reduced optimization are also less prone to producing negative materials. Furthermore, our results suggest that a search in the full-dimensional material space is often unnecessary, as the target material distributions are relatively smooth and low-dimensional and can be resolved well using material model reduction with a small number of modes.

We also optimized the tweezers using three-dimensional force constraints as opposed to normal force constraints. Such constraints are more difficult to enforce than scalar constraints, causing the convergent  $\lambda$  range to be somewhat narrower. The relative force errors (10 modes) were slightly larger for large force modifications: 3.8% with  $\lambda = 5$ , but otherwise similar: 0.2% with  $\lambda = 0.3$ .

## 8.2 Pressure control over large contact areas

In our second example, we apply our method to controlling the contact pressure distribution between a rigid foot and a deformable shoe. This example also serves to demonstrate that our method scales to a large number of vertices with prescribed displacements and forces ( $c = 384$ ). Our rigid object is a human foot in contact with the deformable sole of a shoe (Figure 8). The shoe has realistic geometry (length is 25cm). We optimize the distribution of Young’s moduli of the shoe so that various pressure criteria are met. For example, we can make the pressure constant (for ergonomics) across the entire contact area, or, to demonstrate the ability to control pressures, impose three-fold higher pressure in the front part of the foot versus the heel, and vice-versa.

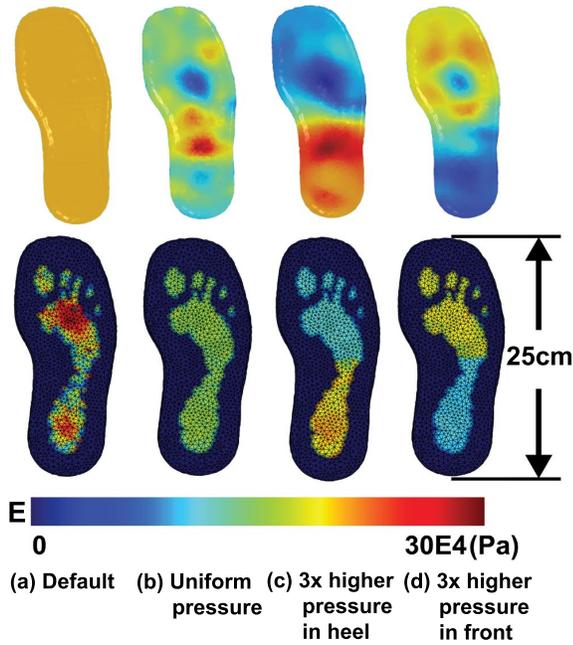


Fig. 9. **Controlled shoe contact pressures:** Top row: volume rendering of the material distribution  $E$ . Bottom row: Contact pressures. Left column is the input obtained under a homogeneous material distribution. Second, third and fourth column give results under a target constant pressure distribution, 3x higher distribution in front and heel, respectively. We used  $r = 100$  modes. Computation times to steady convergence (approximately 150 iterations) were 4.0 min, 3.9 min, 3.5 min, for (b),(c),(d), respectively. Relative force and displacement errors were (7.6%,7.3%), (8.0%,7.6%), (7.5%,8.1%), for (b),(c),(d), respectively.

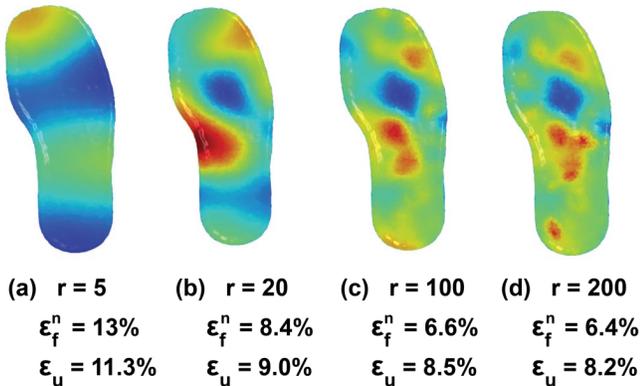


Fig. 10. **Results under an increasing number of material modes (constant pressure experiment).** It can be seen that progressive material local detail is added as  $r$  increases. Full optimization diverged. Relative pressure and position errors are indicated. It can be seen that the errors decrease as  $r$  is increased. Computation times were 3.5 min, 3.5 min, 3.7 min, 4.5min, for 125 iterations, for (a), (b), (c), (d), respectively.

We start our experiment by positioning the rigid foot triangle mesh (1,635 vertices, 3,266 faces) in contact with the tetrahedral mesh (5,095 vertices, 20,954 tetrahedra) of the shoe. In this paragraph,

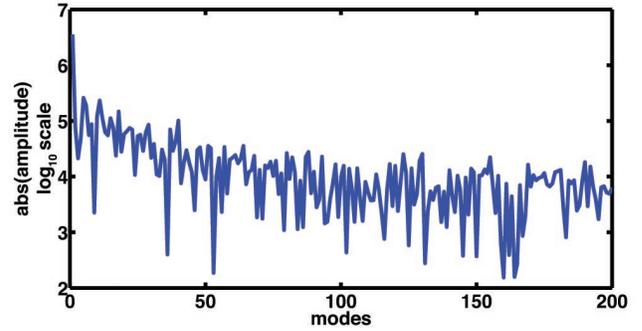


Fig. 11. **Participation of modes:** Shoe example,  $r = 200$ , constant pressure target. Y-axis values are the absolute-valued components of the reduced material vector,  $|z_i|$ . Log scale. The first mode (constant material) has the strongest participation as it determines the overall strength of the material.

we describe how we determine a set of handles on the shoe mesh, their displaced positions, and the initial pressure distribution. We consider these quantities as input to our method and note that they could be obtained using other means; for example, by using a contact pressure sensor [Yin and Pai 2003], followed by loading a homogeneous shoe mesh with those pressures. The foot triangle mesh is a closed manifold mesh. For each vertex of the shoe tetrahedral mesh, we determine if it is inside or outside of the foot mesh. If inside, we then determine the closest location on the foot mesh. For each vertex of the shoe mesh inside the foot mesh, we declare a contact, place a displacement/force handle, and displace the vertex to the closest point position on the foot. There were 384 such vertices in our example. We then automatically load these vertices and their displacements  $\bar{u}$  into our application program as displacement-only handles (force is not prescribed yet). Next, we find a homogeneous Young's modulus so that the normal contact forces support a human with a weight of 80kg (400N on each foot). This is done by selecting any positive Young's modulus value, computing and summing the elastic forces on all handles, and then scaling it so that the total elastic force is 400N. This distribution forms the initial guess to our experiments and we plot its normal pressures in Figure 9 (a). Normal pressure on a vertex (normal traction) is computed as the ratio of the normal force divided by the surface area belonging to a vertex (1/3 of the surface area of the 1-ring of triangles).

We now proceed to finding material distributions that (1) make the pressure on all the 384 contact points equal, (2) give two constant pressure regions where the pressure is  $3\times$  higher in the front part of the foot than in the heel, and (3) reverse of the previous. In each case, we compute the target normal contact forces on each handle so that their sum supports a person with a weight of 80kg. In (1), we first compute the constant pressure by dividing 400N with the total foot contact area. We then multiply the pressure with the surface area of each handle. In (2) and (3), we compute pressures in the two regions by solving  $A_1 p_1 + A_2 p_2 = 400N$ , for  $p_2 = 3p_1$ , where  $A_i$  are the areas of the two regions. We then run optimization (11), optimizing both materials and handle displacements to match  $\bar{u}$  and  $\bar{f}$  on the 384 handles. We only perform handle position optimization (14) once, at the end of the material optimization. It can be seen that our optimizer succeeded in making the pressure distribution uniform across the shoe, or controlling it spatially (Figure 9).

In Figure 10, we tested our method under a progressively increasing number of modes  $r$ . It can be seen that the pressure distribution ex-

hibits increasing local detail as  $r$  is increased. Full optimization was  $270\times$  slower per iteration than reduced simulation with  $r = 200$ , and diverged to negative materials after 3.7 hours (27 iterations). Figure 11 demonstrates the participation of individual modes in our resulting pressure distribution.

In the shoe example, we found it useful to perform material optimization under an automated *weight update* mechanism, where the  $\alpha$  weight in (13) is made to increase slowly after every  $N_{up}$  conjugate gradient iterations (we used  $N_{up} = 5$ ), by a user-adjustable factor  $\gamma$  (we used  $\gamma = 1.2$ ), for a total number of  $N_{iter}$  iterations (we used  $N_{iter} = 125$ ), after which the weight is kept constant. Although not strictly necessary, we found that this mechanism is more stable (less prone to produce negative materials) and more capable to decrease the force error than if a constant  $\alpha$  was used throughout the optimization. We observed that under a constant  $\alpha$ , the optimizer often makes good initial progress, but then gets stuck at some force error level. In order to decrease the force error, one must make smoothness less important, i.e., increase  $\alpha$ . However, one cannot increase  $\alpha$  too aggressively, as otherwise, the distribution “escapes” to a local minimum with negative materials. Negative materials could be prevented by forcing  $E$  to remain above some threshold  $E \geq E_{min}$ , e.g., using a general solver such as SNOPT [Gill et al. 1997]. We pursued this approach, but did not find it convenient because it requires adding an inequality to the optimization problem. More importantly, we found that the negativity of  $E$  is much like a run-away reaction. Once it starts (by too high  $\alpha$ ), Young’s moduli continually decrease below all limits, regardless of their initial values. Therefore, blocking the decrease with a constraint  $E \geq E_{min}$  merely gives some partially diverged distribution. The best remedy is to instead decrease  $\alpha$ . As a curiosity, we note that negative values do not always imply unstable results, as the entire shape can still be statically stable if there are only a few elements or regions with negative stiffness; but will lead to strange dynamics.

### 8.3 Haptic design of heterogeneous materials

In this application, we exploit the speed of our optimization for interactive editing of heterogeneous material distributions using haptics (Figure 12). Under any distribution  $E$ , and any set of handles, we form the matrix  $\tilde{K}$  (Equation 7). Given the handle displacements  $\tilde{u}$ , we then compute the handle reaction forces  $\tilde{f}$  using Equation 6, as introduced by [James and Pai 2001]. Note that this approach is very popular in haptic rendering, where it is commonly used to display deformable objects. Previous papers, however, did not interactively change or optimize the material distribution, but displayed a fixed distribution to the user. Our work provides the capability to design material distributions and feel them with haptics.

We use the Sensable Omni haptic device to display the reaction forces  $\tilde{f}$  to the user. Because  $\tilde{K}$  is constant and small ( $3c \times 3c$ ),  $\tilde{f}$  can be computed very rapidly, at haptic rates. During handle vertex manipulation, the user can also change the reaction force on that vertex, effectively imposing a  $\tilde{f}$  constraint. In our user interface, the user can change the reaction force  $-S_i K(E_{default})u$  on handle  $i$  by scaling it with a scalar  $\lambda_i > 0$ ,

$$\tilde{f}_i = -\lambda_i S_i K(E_{default})u, \quad (38)$$

where  $S_i$  selects the  $i$ -th handle degrees of freedoms, and  $u = [\hat{u} \tilde{u}]^T$  is the object’s deformation. The material optimizer (13) is then launched in a separate thread. After each conjugate gradient iteration, a new  $\tilde{K}$  is computed and communicated to the haptic servo loop via a mutex. Because the iterations run at interactive rates

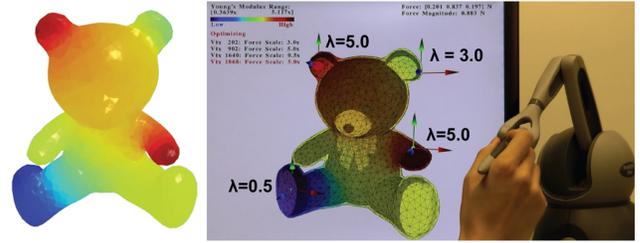


Fig. 12. **Editing of material distributions using haptics.** The output distribution is shown on the left.

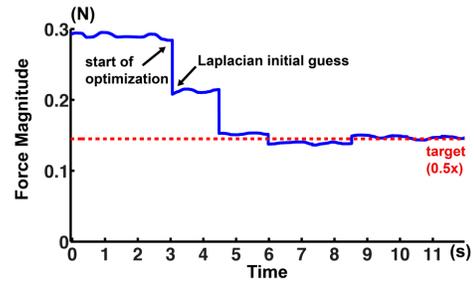


Fig. 13. **Haptic force signal** after the user decreased (at  $t = 3s$ ) the target force on the manipulated handle by  $0.5\times$ . Two handles were used ( $c = 2$ ). Teddy bear example. Plot shows magnitude of force. The force decreased to the desired level in a few optimizer iterations. Optimizer iterations are visible as constant plateaus of the force. Initial guess is computed using the Laplacian solver (Section 4.2). Signal variation is due to the shaking of user’s hand.

( $\sim 1.4$  second per iteration), the user can feel the force grow/shrink as requested (see Figure 13), while the current material distribution is visualized on the screen (Figure 12). It can be seen in Figure 13 that the Laplacian initial guess (Section 4.2) is reasonable, but it alone does not minimize our problem, and must be refined through optimization. In a manner of seconds, the new material distribution converges to a minimum of Equation (24). The user can interrupt the optimization at any time. Throughout the process, the user can continue the editing process by adding, removing or re-adjusting the forces and handles. Such a tool enables the user to rapidly create interesting material distributions and feel them using haptics. For example, we made the ear region and one arm of the teddy bear (2,690 vertices, 10,046 tetrahedra)  $5\times$  and  $0.5\times$  stiffer than the rest of the model.

For large requested changes of reaction forces (typically  $\lambda < 1/2$  or  $\lambda_i > 2$ ), we found that we can obtain improved results if we properly rotate the input forces at the handles. This is because a change in the materials causes the entire object’s shape  $u$  to change at each iteration of material optimization (13). This, in turn, rotates the elastic force at every handle, causing the original forces  $\tilde{f}_i$  to be misaligned in direction. Thus, after  $N'_{up}$  conjugate gradient iterations of (13) (we use  $N'_{up} = 3$ ), and after each update of  $\tilde{u}$  via (14), we update  $\tilde{f}_i$  according to (38), using the *current* object shape  $u$ . Such a modification effectively asks for a shape  $u$  and material distribution  $E$  with the property that the internal elastic forces at the handles are  $\lambda_i\times$  the elastic forces in shape  $u$  under the default homogeneous distribution.

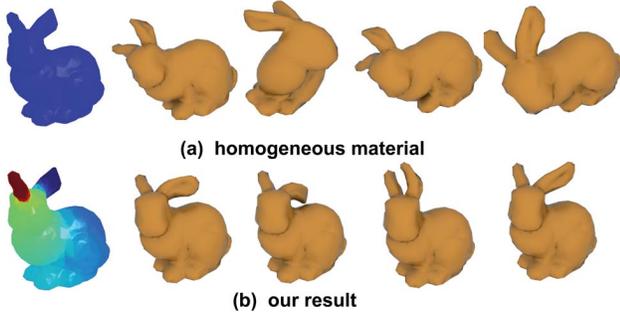


Fig. 14. **Interactively edited material distributions for animation.** Top row: animation frames under the homogeneous material. Bottom: our result. Material distribution is shown on the left. The stiffness of the bunny body was increased  $2.0\times$ , and the ears were made  $6.0\times$  and  $0.5\times$  stiffer. In the resulting animation frames, note how the left ear is stiffer and the right ear is softer.

#### 8.4 Material design for animation

In the bunny example (777 vertices, 2,605 tetrahedra), we demonstrate interactive material editing with the goal of creating general-purpose material distributions, say for subsequent physically-based simulation. The user can specify the forces using our  $\lambda$ -based interface. Although our distributions are computed under small-displacement assumptions, this example demonstrates that they can be easily applied to any large-deformation simulation that is parameterized by Young’s modulus and Poisson’s ratio, e.g., the Saint Venant Kirchhoff material [Bonet and Wood 2008], co-rotational materials [Müller and Gross 2004] or neo-Hookean materials [Bonet and Wood 2008] (Figure 14).

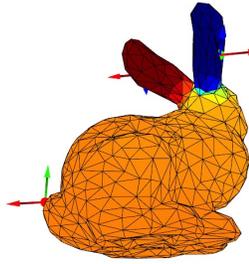


Fig. 15. Editing unanchored objects.

Alternatively, we can also directly optimize these nonlinear materials under large deformations (Section 7). In this example, we also demonstrate editing of unanchored objects (Figure 15). Unanchored objects only require solving the  $K_{11}(E)$  linear systems with a solver that can handle singular matrices, such as conjugate gradients that remove the nullspace component at every iteration (note that the nullspace of  $K_{11}$  is known). Matrix  $K_{11}(E)$  becomes non-singular when at least 3 handles are selected by the user, at which point the special handling is no longer required.

#### 8.5 Comparison to shape reduction

An alternative approach to accelerate the material optimization is to employ model reduction to the object’s deformations (but not materials) [Lee and Lin 2012]. The reduced basis of deformations can be obtained by solving the generalized eigenproblem

$$K(E_{\text{default}})x_j = \lambda_j M x_j. \quad (39)$$

We truncate the eigenbasis, retaining the first  $r'$  vibration modes as columns of  $U = (x_1 \dots x_{r'}) \in \mathbb{R}^{3n \times r'}$ . The deformation can then be approximated as  $u = Uq$ , where  $q \in \mathbb{R}^{r'}$  is the reduced displacement

vector. It can be obtained by solving the optimization problem,

$$\begin{aligned} \min_q \quad & \frac{1}{2}(Uq)^T K(Uq) \\ \text{subject to} \quad & \bar{S}Uq = \bar{u}, \end{aligned} \quad (40)$$

which is equivalent to solving a linear system

$$\begin{bmatrix} U^T K(E)U & (\bar{S}U)^T \\ \bar{S}U & 0 \end{bmatrix} \begin{bmatrix} q \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{u} \end{bmatrix}. \quad (41)$$

For computational reasons, it is only practical to compute the modes once (with the default material). Therefore, when the optimizer adjusts  $E$  during its iterations,  $U^T K(E)U$  is not necessarily a diagonal matrix. However, unlike with the method of Section 4 that does not employ shape reduction, the resulting force  $f(E) = K(E)Uq$  is not necessarily zero outside of the handles ( $\hat{f} \neq 0$ ). Therefore, we extend our objective function (13) to also penalize  $\hat{f}$ ,

$$\min_E \quad \frac{1}{2}E^T L E + \frac{\alpha}{2} \|f(E) - [0 \ \bar{f}]^T\|_W^2. \quad (42)$$

The gradient of (42) is

$$LE + \alpha \left( \frac{df(E)}{dE} \right)^T (f(E) - [0 \ \bar{f}]^T), \quad \text{where} \quad (43)$$

$$\frac{df(E)}{dE_e} = \frac{dK(E)}{dE_e} Uq + K(E)U \frac{dq}{dE_e}, \quad (44)$$

for  $e = 1 \dots m$ . To obtain  $dq/dE_e$ , we differentiate Equation 41 with respect to  $E_e$ . This yields a  $r' \times r'$  linear system for  $dq/dE_e$ ,

$$\begin{bmatrix} U^T K(E)U & (\bar{S}U)^T \\ \bar{S}U & 0 \end{bmatrix} \begin{bmatrix} \frac{dq}{dE_e} \\ \frac{d\lambda}{dE_e} \end{bmatrix} = \begin{bmatrix} -U^T \frac{dK(E)}{dE_e} U \\ 0 \end{bmatrix}. \quad (45)$$

Shape reduction accelerates the material optimization by reducing the dimension of linear systems from  $3n \times 3n$  to  $r' \times r'$ , although the number of linear systems still equals the number of mesh elements ( $m$ ). However, unlike performing reduction in the material space, shape reduction trades force accuracy for performance. Figure 16 demonstrates that shape reduction requires a substantial number of modes to reach close to the optimal solution. In this experiment, we placed one vertex handle on a beam (600 DOFs) and set  $\lambda = 2$  as in Section 8.3. The optimal solution is a homogeneous material distribution where  $E$  has been globally scaled by  $2\times$ . Shape reduction produces a heterogeneous material distribution that only becomes homogeneous for a large number of modes. Material reduction, in turn, produces a homogeneous material and matches the target force exactly, all the while accelerating the optimization significantly. We also conducted a comparison for a heterogeneous material distribution. We specified this distribution by employing two handles and setting target force multipliers  $\lambda$  as  $1\times$  and  $1.5\times$ , respectively. Our material reduction optimization output is as good as full modes optimization, while shape reduction gives material distributions with at least  $200\times$  larger force error  $\|f(E) - [0 \ \bar{f}]^T\|_W^2$ .

## 9. CONCLUSION

We presented an interactive method to design material properties of three-dimensional elastic objects. We demonstrated scalability to complex examples using material model reduction, where our

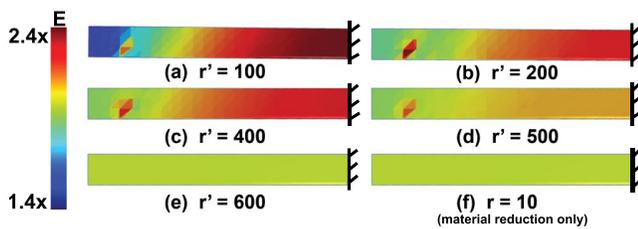


Fig. 16. Comparison of shape reduction and material reduction on a beam model. (a)-(e): Shape reduction with full material modes ( $r = 450$ ). (f) Material reduction with full deformation modes ( $r' = 600$ ). The singular spot is the place of the handle.

key insight is that reduction makes it possible to compute the reduced gradient two orders of magnitude faster than the full gradient, which in turn makes the optimization interactive. We used our method to control pressure distributions of rigid objects in contact with deformable objects. Our method can be combined with haptics, to give the user both intuitive interactive control and feedback over the heterogeneous stiffness of the object. Our interactive design tool is also useful in computer animation. We also present a tetrahedron-based dithering algorithm that can discretize continuous material distributions into discrete ones.

In this paper, we only optimize Young's moduli, but Poisson's ratios could also be optimized, for example, in the 2-dimensional space of Lamé parameters. We do not edit dynamic properties of objects such as damping, or design materials where the displacement input is prescribed to vary in time (animation input). We only optimize material distributions, but do not optimize the geometry of the object, which is a complementary approach to achieve a desired functionality. We use Laplacian eigenmodes for our basis, but the material reduced space could be refined further to better optimize the objective. Although we demonstrate the effectiveness of our dithering method using simulation, our dithering algorithm is greedy. Much like with Floyd-Steinberg dithering for pixels, a theoretical optimality condition is difficult to give, even though results are good in practice. In the future, we would also like to 3D-print real objects based on physically available materials.

#### ACKNOWLEDGMENTS

This research was sponsored in part by the National Science Foundation (CAREER-1055035, IIS-1422869), Sloan Foundation, USC Annenberg Graduate Fellowship to Hongyi Xu and Yijing Li, and a donation of two workstations by the Intel Corporation. We would also like to thank Malancha Gupta for her suggestions.

#### REFERENCES

- AN, S. S., KIM, T., AND JAMES, D. L. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM Trans. on Graphics (SIGGRAPH Asia 2008)* 27, 5, 165:1–165:10.
- BARBIČ, J. AND JAMES, D. L. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. on Graphics (SIGGRAPH 2005)* 24, 3, 982–990.
- BARBIČ, J., SIN, F., AND GRINSPUN, E. 2012. Interactive editing of deformable simulations. *ACM Trans. on Graphics (SIGGRAPH 2012)* 31, 4, 70:1–70:8.

- BECKER, M. AND TESCHNER, M. 2007. Robust and efficient estimation of elasticity parameters using the linear finite element method. In *SimVis*. 15–28.
- 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 Trans. on Graphics (SIGGRAPH 2010)* 29, 4, 63:1–63:10.
- BICKEL, B., BAECHER, M., OTADUY, M., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. on Graphics (SIGGRAPH 2009)* 28, 3, 89:1–89:9.
- BONET, J. AND WOOD, R. D. 2008. *Nonlinear Continuum Mechanics for Finite Element Analysis, 2nd Ed.* Cambridge University Press.
- BOTSCH, M., PAULY, M., WICKE, M., AND GROSS, M. 2007. Adaptive space deformations based on rigid cells. *Comp. Graphics Forum* 26, 3, 339–347.
- BOTSCH, M. AND SORKINE, O. 2008. On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on* 14, 1, 213–230.
- BRO-NIELSEN, M. AND COTIN, S. 1996. Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. *Comp. Graphics Forum* 15, 3, 57–66.
- CHO, W., SACHS, E. M., PATRIKALAKIS, N. M., AND TROXEL, D. E. 2003. A dithering algorithm for local composition control with three-dimensional printing. *Computer-aided design* 35, 9, 851–867.
- COTIN, S., DELINGETTE, H., AND AYACHE, N. 1999. Realtime Elastic Deformations of Soft Tissues for Surgery Simulation. *IEEE Trans. on Vis. and Comp. Graphics* 5, 1, 62–73.
- FLOYD, R. AND STEINBERG, L. 1976. An adaptive algorithm for spatial gray scale. *Proc. Society of Information Display* 17, 2, 75–77.
- GAO, Z., KIM, T., JAMES, D. L., AND DESAI, J. P. 2009. Semi-automated soft-tissue acquisition and modeling for surgical simulation. In *Proc. of the Fifth Annual IEEE Int. Conf. on Automation Science and Engineering*. 268–273.
- GILL, P. E., MURRAY, W., MICHAEL, AND SAUNDERS, M. A. 1997. Snoop: An sqp algorithm for large-scale constrained optimization.
- HAN, L.-H., SURI, S., SCHMIDT, C. E., AND CHEN, S. 2010. Fabrication of three-dimensional scaffolds for heterogeneous tissue engineering. *Biomedical microdevices* 12, 4, 721–725.
- HAUSER, K. K., SHEN, C., AND O'BRIEN, J. F. 2003. Interactive deformation using modal analysis with constraints. In *Proc. of Graphics Interface*. 247–256.
- HILDEBRANDT, K., SCHULZ, C., TYCOWICZ, C. V., AND POLTHIER, K. 2011. Interactive surface modeling using modal analysis. *ACM Transactions on Graphics (TOG)* 30, 5, 119.
- HILDEBRANDT, K., SCHULZ, C., VON TYCOWICZ, C., AND POLTHIER, K. 2012. Interactive spacetime control of deformable objects. *ACM Trans. on Graphics (SIGGRAPH 2012)* 31, 4, 71:1–71:8.
- HUANG, P., DENG, D., AND CHEN, Y. 2013. Modeling and fabrication of heterogeneous three-dimensional objects based on additive manufacturing. In *ASME 2013 Int. Mechanical Engineering Congress and Exposition*. V02AT02A056–V02AT02A056.
- HUANG, Q., WICKE, M., ADAMS, B., AND GUIBAS, L. 2009. Shape decomposition using modal analysis. *Computer Graphics Forum* 28, 2, 407–416.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics (SIGGRAPH 2005)* 24, 3, 1134–1141.
- JACOBSON, A., BARAN, I., KAVAN, L., POPOVIĆ, J., AND SORKINE, O. 2012. Fast automatic skinning transformations. *ACM Trans. on Graphics (SIGGRAPH 2012)* 31, 4, 77:1–77:10.

- JACOBSON, A., WEINKAUF, T., AND SORKINE, O. 2012. Smooth shape-aware functions with controlled extrema. *Computer Graphics Forum (Proc. of Symposium on Geometry Processing)* 31, 5, 1577–1586.
- JAMES, D. L. AND PAI, D. K. 2001. A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics. *Haptics-e, The Electronic J. of Haptics Research* 2, 1.
- JAMES, D. L. AND PAI, D. K. 2002. DyRT: Dynamic Response Textures for Real Time Deformation Simulation With Graphics Hardware. *ACM Trans. on Graphics (SIGGRAPH 2002)* 21, 3, 582–585.
- KAJBERG, J. AND LINDKVIST, G. 2004. Characterisation of materials subjected to large strains by inverse modelling based on in-plane displacement fields. *International Journal of Solids and Structures* 41, 13, 3439 – 3459.
- KAUER, M., VUSKOVIC, V., DUAL, J., SZÉKELY, G., AND BAJKA, M. 2002. Inverse finite element characterization of soft tissues. *Medical Image Analysis* 6, 3, 275–287.
- KHALIL, S., NAM, J., AND SUN, W. 2005. Multi-nozzle deposition for construction of 3d biopolymer tissue scaffolds. *Rapid Prototyping Journal* 11, 1, 9–17.
- KHAREVYCH, L., MULLEN, P., OWHADI, H., AND DESBRUN, M. 2009. Numerical coarsening of inhomogeneous elastic materials. *ACM Trans. on Graphics* 28, 3, 51:1–51:8.
- KIM, T. AND JAMES, D. 2009. Skipping steps in deformable simulation with online model reduction. *ACM Trans. on Graphics (SIGGRAPH Asia 2009)* 28, 5, 123:1–123:9.
- KOU, X., TAN, S., AND SZE, W. 2006. Modeling complex heterogeneous objects with non-manifold heterogeneous cells. *Computer-Aided Design* 38, 5, 457–474.
- LANG, J., PAI, D. K., AND WOODHAM, R. J. 2002. Acquisition of elastic models for interactive simulation. *The International Journal of Robotics Research* 21, 8, 713–733.
- LEE, H.-P. AND LIN, M. 2012. Fast optimization-based elasticity parameter estimation using reduced models. *The Visual Computer* 28, 6-8, 553–562.
- LÉVY, B. AND ZHANG, H. R. 2010. Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses*. SIGGRAPH '10. ACM, New York, NY, USA, 8:1–8:312.
- LI, S., HUANG, J., DE GOES, F., JIN, X., BAO, H., AND DESBRUN, M. 2014. Space-time editing of elastic motion through material optimization and reduction. *ACM Trans. Graph.* 33, 4, 108:1–108:10.
- LIEW, C. L., LEONG, K. F., CHUA, C. K., AND DU, Z. 2002. Dual material rapid prototyping techniques for the development of biomedical devices. part 2: Secondary powder deposition. *Int. J. of Advanced Manufacturing Technology* 19, 9, 679–687.
- LIU, H., MAEKAWA, T., PATRIKALAKIS, N., SACHS, E., AND CHO, W. 2004. Methods for feature-based design of heterogeneous solids. *Computer-Aided Design* 36, 12, 1141–1159.
- MARTIN, S., THOMASZEWSKI, B., GRINSPUN, E., AND GROSS, M. 2011. Example-based elastic materials. *ACM Trans. on Graphics (SIGGRAPH 2011)* 30, 4, 72.
- MORI, Y. AND IGARASHI, T. 2007. Plushie: an interactive design system for plush toys. *ACM Trans. on Graphics (SIGGRAPH 2007)* 26, 3, 45.
- MÜLLER, M. AND GROSS, M. 2004. Interactive Virtual Materials. In *Proc. of Graphics Interface 2004*. 239–246.
- NEALEN, A., MÜLLER, M., KEISER, R., BOXERMAN, E., AND CARLSON, M. 2006. Physically based deformable models in computer graphics. *Computer Graphics Forum* 25, 4, 809–836.
- NESME, M., KRY, P. G., JEŘÁBKOVÁ, L., AND FAURE, F. 2009. Preserving topology and elasticity for embedded deformable models. *ACM Trans. on Graphics (SIGGRAPH 2009)* 28, 3, 52:1–52:9.
- PAI, D. K., DOEL, K. V. D., JAMES, D. L., LANG, J., LLOYD, J. E., RICHMOND, J. L., AND YAU, S. H. 2001. Scanning physical interaction behavior of 3d objects. In *Proc. of ACM SIGGRAPH 2001*. 87–96.
- PRESS, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. 2007. *Numerical recipes: The art of scientific computing*, third ed. Cambridge University Press, Cambridge, UK.
- SCHNUR, D. AND ZABARAS, N. 1992. An inverse method for determining elastic material properties and a material interface. *Int. J. for Numerical Methods in Engineering* 33, 10, 2039–2057.
- SCHONER, J. L., LANG, J., AND SEIDEL, H.-P. 2004. Measurement-based interactive simulation of viscoelastic solids. In *Computer Graphics Forum*. Vol. 23. 547–556.
- SCHUMACHER, C., THOMASZEWSKI, B., COROS, S., MARTIN, S., SUMNER, R., AND GROSS, M. 2012. Efficient simulation of example-based materials. In *Symp. on Computer Animation (SCA)*. 1–8.
- SHABANA, A. A. 1990. *Theory of Vibration, Volume II: Discrete and Continuous Systems*. Springer-Verlag, New York, NY.
- SIFAKIS, E. AND BARBIČ, J. 2012. FEM simulation of 3D deformable solids: A practitioner’s guide to theory, discretization and model reduction. In *SIGGRAPH 2012 Course Notes*.
- SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. on Graphics (SIGGRAPH 2005)* 24, 3 (Aug.), 417–425.
- SKOURAS, M., THOMASZEWSKI, B., BICKEL, B., AND GROSS, M. 2012. Computational design of rubber balloons. *Computer Graphics Forum (Eurographics 2012)* 31, 2pt4, 835–844.
- SKOURAS, M., THOMASZEWSKI, B., COROS, S., BICKEL, B., AND GROSS, M. 2013. Computational design of actuated deformable characters. *ACM Trans. on Graphics (SIGGRAPH 2013)* 32, 4, 82:1–82:10.
- VIDIMČE, K., WANG, S.-P., RAGAN-KELLEY, J., AND MATUSIK, W. 2013. Openfab: A programmable pipeline for multi-material fabrication. *ACM Trans. Graph.* 32, 4, 136:1–136:12.
- WANG, H., O’BRIEN, J. F., AND RAMAMOORTHI, R. 2011. Data-driven elastic models for cloth: modeling and measurement. *ACM Trans. on Graphics (SIGGRAPH 2011)* 30, 4, 71.
- WICKE, M., STANTON, M., AND TREUILLE, A. 2009. Modular bases for fluid dynamics. *ACM Trans. on Graphics (SIGGRAPH 2009)* 28, 3, 39:1–39:8.
- YIN, K. AND PAI, D. K. 2003. Footsee: an interactive animation system. In *Symp. on Computer Animation (SCA)*. 329–338.
- ZHANG, H. 2004. Discrete combinatorial laplacian operators for digital geometry processing. In *Proceedings of SIAM Conference on Geometric Design and Computing*. Nashboro Press, 575–592.
- ZHANG, H., VAN KAICK, O., AND DYER, R. 2010. Spectral mesh processing. *Computer graphics forum* 29, 6, 1865–1894.
- ZHOU, C., CHEN, Y., YANG, Z., AND KHOSHNEVIS, B. 2013. Digital material fabrication using mask-image-projection-based stereolithography. *Rapid Prototyping Journal* 19, 3, 153–165.

Received April 2014; accepted November 2014