

Deformation Capture and Modeling of Soft Objects

Bin Wang*[†] Longhua Wu* KangKang Yin[†] Uri Ascher[‡] Libin Liu[‡] Hui Huang*[§]

*Shenzhen VisuCA Key Lab / SIAT [†]National University of Singapore [‡]University of British Columbia

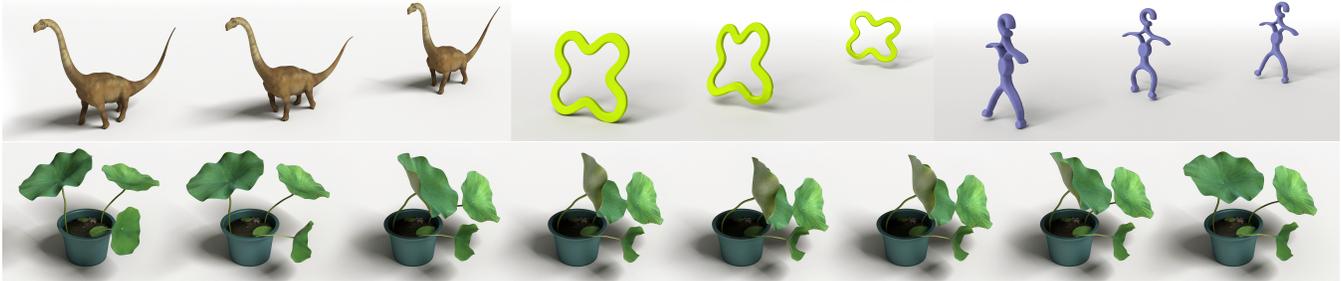


Figure 1: Our system can capture and model deformation behavior of generic soft objects from kinematic data alone. We can then synthesize new motions that satisfy user-specified constraints and respond to dynamic perturbations. Top: left - a dinosaur walking; middle - a pot holder jumping; right - a coat hanger skipping. Bottom: lotus leaves moving in an artificial wind field.

Abstract

We present a data-driven method for deformation capture and modeling of general soft objects. We adopt an iterative framework that consists of one component for physics-based deformation tracking and another for spacetime optimization of deformation parameters. Low cost depth sensors are used for the deformation capture, and we do not require any force-displacement measurements, thus making the data capture a cheap and convenient process. We augment a state-of-the-art probabilistic tracking method to robustly handle noise, occlusions, fast movements and large deformations. The spacetime optimization aims to match the simulated trajectories with the tracked ones. The optimized deformation model is then used to boost the accuracy of the tracking results, which can in turn improve the deformation parameter estimation itself in later iterations. Numerical experiments demonstrate that the tracking and parameter optimization components complement each other nicely.

Our spacetime optimization of the deformation model includes not only the material elasticity parameters and dynamic damping coefficients, but also the reference shape which can differ significantly from the static shape for soft objects. The resulting optimization problem is highly nonlinear in high dimensions, and challenging to solve with previous methods. We propose a novel splitting algorithm that alternates between reference shape optimization and deformation parameter estimation, and thus enables tailoring the optimization of each subproblem more efficiently and robustly.

*email: {wangbin, wulh}@siat.ac.cn

[†]email: kkyin@comp.nus.edu.sg

[‡]email: {ascher, libinliu}@cs.ubc.ca

[§]Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Our system enables realistic motion reconstruction as well as synthesis of virtual soft objects in response to user stimulation. Validation experiments show that our method not only is accurate, but also compares favorably to existing techniques. We also showcase the ability of our system with high quality animations generated from optimized deformation parameters for a variety of soft objects, such as live plants and fabricated models.

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

Keywords: motion capture, deformation modeling, FEM

1 Introduction

Physics-based deformable models enable realistic animation of a wide range of objects and phenomena [Nealen et al. 2006]. Estimating model parameters, however, still heavily relies on either manual tuning or tedious measurements [Terzopoulos et al. 1987]. Such approaches can hardly scale to complex models with nonlinear or heterogeneous material distributions. Moreover, these methods usually employ static shapes, i.e., the static equilibrium under gravity that can be easily observed, as the original reference shapes of the deformation models. This approximation does not work for very soft objects, for instance long plant leaves, as they deform significantly due to gravity. In addition, dynamic properties such as damping coefficients have seldom been considered previously, even though they play a critical role in achieving realistic behavior, in particular for soft objects.

Data-driven methods have recently been quite successful in constructing physics-based deformable models for cloth, human organs and faces [Otaduy et al. 2012]. However, they often require measuring the dense force-displacement relationships. Such measuring processes and hardware have to be tailored to specific types of objects being modeled, and thus are hard to generalize. We wish to build a system that can learn from real-world measurements as well, and is applicable to generic objects without requiring any expensive or specialized hardware for force actuation or measurement.

In this paper, we propose a novel data-driven deformation capture and modeling framework for generic soft objects. Our system builds deformable models from pure kinematic motion trajectories,

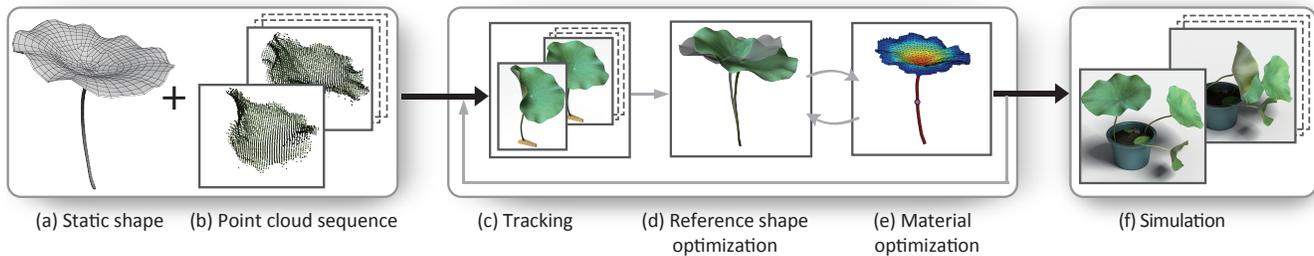


Figure 2: Schematic overview of our data-driven framework that iterates between deformation tracking and parameter optimization.

triggered by simple user interactions and captured by cheap depth sensors. The elimination of force measurement and employment of cheap sensors call for algorithms that can robustly deal with low quality kinematic input. We thus first employ a physics-based probabilistic tracking method that is robust to noise, occlusions, fast movements and large deformations. Then we formulate the deformation parameter estimation as a spacetime optimization problem, aiming to match the simulated trajectories with the tracked ones up to noise and modeling inaccuracies.

Our deformable models include not only the material elasticity parameters and the dynamic damping coefficients, but also the reference shape which introduces high dimensionality into the spacetime optimization problem. When coupled with significant nonlinearity and noisy input, our problem becomes too challenging for previous methods to solve effectively. We thus propose a novel splitting algorithm that divides the large optimization problem into subproblems that are solved separately and iteratively. Then the reference shape optimization can take advantage of the dynamics simulation by applying virtual forces, resulting in a much faster solver than conventional optimization methods.

We further employ an iterative scheme that alternates between the physics-based deformation tracking and physics-based parameter estimation, yielding a maximum-likelihood solution that converges at a reasonable speed. This scheme not only makes the final results less dependent on the initial tracking accuracy with default deformation parameters, but also facilitates motion inference of unobservable parts of the deformable objects and in turn improves the fidelity of parameter identification.

In summary, our main contributions include:

- an iterative tracking and parameter estimation framework that only requires kinematic motion capture with simple user interactions and cheap depth sensors;
- a novel splitting scheme that can solve the high-dimensional and non-convex spacetime optimization for both the deformation parameters and the reference shape;
- an augmented physics-based probabilistic tracking method that can reconstruct fast and highly deformable movements from extremely noisy and incomplete point clouds.

2 Related Work

Animation Capture We use “animation capture” to refer to several terminologies introduced in different application context and research communities, including performance capture, object tracking and animation reconstruction. With the recent advances in commercial-grade scanners, depth sensors and video cameras, and the cohort research efforts on data-driven animation, the robustness of such animation capture methods and systems has been improving

steadily. Such capture systems often take multi-view video recordings as input [Bradley et al. 2008; de Aguiar et al. 2008; Vlasic et al. 2008; Li et al. 2012], but single-view capture systems are also gaining more popularity due to their low cost and simple setups [Li et al. 2009; Wei and Chai 2010; Tevs et al. 2012; Wei et al. 2012; Helten et al. 2013; Schulman et al. 2013].

These methods demonstrate various ways of dealing with partial observation, occlusion, noise and outliers, such as involving users in the loop [Wei and Chai 2010], using geometry templates [Li et al. 2009; Schulman et al. 2013], coupling tracking with detection [Wei et al. 2012], utilizing physics constraints [Choi and Szymczak 2009], or fusing measurements from multiple sensors [Helten et al. 2013]. We have been particularly inspired by the probabilistic tracking method proposed by Schulman et al. [2013]. It can deal with highly noisy point clouds captured using cheap sensors. Furthermore, our finite element method (FEM) simulator can be easily plugged into such a physics-based tracking framework. Wuhler et al. [2013] also used linear FEM for template-based tracking of deforming surfaces. However, they use earlier tracking frames as the reference shapes for the later frames, and thus their estimated material parameters are not physically meaningful. By contrast, we stress that our total estimation process will also recover a reference shape rather than assume one is observed or known.

Deformation Modeling It is a common challenge in computer animation and physics-based simulation to assign material parameters accurately in order to yield desired deformation behavior. Manual parameter tuning cannot scale to complex models with nonlinear or inhomogeneous material distributions. With recent improvements in sensing technologies, the data-driven approach of modeling and reconstructing deformation parameters from real world measurements has offered great potential for computer graphics applications, such as fabrics, soft objects, and human organs and faces [Pai et al. 2001; Schoner et al. 2004; Becker and Teschner 2007; Wang et al. 2011; Miguel et al. 2012; Bickel et al. 2009]. Bickel et al. [2009] fit material parameters with an incremental loading strategy to better approximate the nonlinear strain-stress relationships. Wang et al. [2011] proposed a piecewise linear elastic model to reproduce the nonlinear, anisotropic stretching and bending of cloth. Miguel et al. [2012] directly optimized the nonlinear stress-strain curves based on measurements. A common weakness with previous methods is they require a dense force displacement field and known reference shapes. While Bhat et al. [2003] avoided the need for force capture by using video tracking of cloth, they still assumed a trivial cloth reference shape. By contrast, our method requires neither force displacement capture nor a priori reference shapes, making it significantly more convenient.

The spacetime optimization framework has been utilized quite often for editing and control of deformable animations [Barbič et al. 2012; Hildebrandt et al. 2012; Li et al. 2014], and occasionally for

material estimation as well [Lee et al. 2012; Li et al. 2014]. Model reduction techniques are needed to achieve interactive rates. Our problem is too challenging for these methods to solve effectively as we also wish to recover the reference shape, which introduces many more unknowns that depend nonlinearly on the dynamic positions, and our input point cloud data is much noisier than key frames or medical images. We propose a novel splitting scheme to efficiently and robustly solve our spacetime optimization problem.

Fabrication-oriented Deformation Design Fabrication-oriented design has recently been gaining attention in the computer graphics community. In such design scenarios, physical objects are not always available for measurements, and specific and strict fabrication constraints must be respected. Bickel et al. [2010] used a spatial combination of basis material layers to achieve anisotropic, inhomogeneous and nonlinear deformation behavior. Chen et al. [2014] and Derouet et al. [2013] optimized for the reference shapes to attain desired static shapes. They require either known material properties [Chen et al. 2014] or known boundary conditions [Derouet-Jourdan et al. 2013]. Similar techniques are also effective in controlling deformable characters [Coros et al. 2012] and balloon shapes when inflated [Skouras et al. 2012]. In addition to material distributions, mechanical actuation can also be optimized to control the deformable motions [Skouras et al. 2013]. Our work shares the same goal of reproducing desired deformation behavior. However, we directly optimize for both material properties and reference shape, which to the best of our knowledge no existing work has attempted so far.

3 Overview

Figure 2 shows our data-driven deformation capture and modeling framework. The system starts from capturing both the static shape and the deformable motion of a soft object. The static shape, the equilibrium geometry under gravity, is first scanned and reconstructed as a triangular mesh (Figure 2(a)). Then the dynamic motion, introduced by an initial deformation, is captured in real time using three depth sensors from different viewpoints (Figure 3(a)) and stored as a time sequence of point clouds (Figure 2(b)).

The core of our algorithm is an iterative scheme that alternates between a deformation tracking component and a parameter optimization component. Using the static shape as a template, a physics-based probabilistic method is employed to track the dynamic point cloud sequence (Figure 2(c)). The output of this component is represented as 3D coordinates of a tetrahedral mesh, which tightly encloses the surface template in the static shape. Next the parameter estimation component optimizes for the elastic material parameters, the damping coefficients, and the reference shape as well. The reference shape refers to the geometry subject to zero external forces and can be significantly different from the static shape, especially for soft objects (see Figure 4). Furthermore, the optimized deformation parameters can in turn enhance the accuracy of the physics-based tracking component (see Figure 8), which can consequently improve the deformation parameter estimation in the next iteration.

We emphasize that here we have two distinctive sets of unknowns. The first is the dynamic set of deformed positions, i.e., volumetric mesh vertex coordinates that move in a time sequence. The second is the static set of parameters that vary in space, viz. the material distribution and the reference shape. These unknowns are fewer in number, but they are not as well structured as the dynamic positions. Importantly, they are the ones that cause the nonlinearity difficulties in our optimization problem. Therefore the reconstruction of unknowns is optimized in space and time separately and iteratively (Figure 2(c-e)). The static set of parameters can be fur-

ther divided into two subsets, the material parameters and damping coefficients, and the reference shape. Due to the different nature of these two subsets of static unknowns, we again split the optimization routine into two solvers that can thus take advantage of different techniques in order to achieve better robustness and overall performance. These solvers are alternated in an iterative fashion (Figure 2(d-e)), just like the outer optimization loop.

4 Elastic Deformation Modeling

We employ the widely adopted co-rotated linear FEM to handle large deformations of soft objects [Müller et al. 2002]. To speed up the computation, we use coarse volumetric tetrahedral meshes to conduct the FEM simulation in both tracking and optimization tasks. The low-resolution volumetric mesh and the high-resolution surface mesh are coupled through a mesh embedding technique [Kim and Pollard 2011].

Throughout this article we denote the nodal positions in the reference shape by \mathbf{X} . The material parameters will be denoted by \mathbf{p} once they are defined below. The dynamic, deformed positions at a time instant t are denoted by $\mathbf{x} = \mathbf{x}(t)$. We use the subscript $-e$ to refer to quantities associated with each tetrahedral element. Then the element-wise stress-strain relationship using Hooke’s law and Cauchy’s linear strain tensor is as follows:

$$\boldsymbol{\sigma} = \mathbf{E}\boldsymbol{\epsilon} = \mathbf{E}\mathbf{B}_e(\mathbf{x}_e - \mathbf{X}_e),$$

where $\mathbf{B}_e = \mathbf{B}_e(\mathbf{X}_e)$ is a 6×12 matrix that depends on \mathbf{X}_e nonlinearly. The 6×6 matrix \mathbf{E} only depends on Young’s modulus E and Poisson’s ratio ν for isotropic materials.

Denoting the per-element rotation matrix obtained from polar decomposition by $\mathbf{R}_e = \mathbf{R}_e(\mathbf{x}_e(t), \mathbf{X}_e)$, the element-wise elastic forces using the co-rotated linear model [Müller and Gross 2004] can then be expressed as

$$\mathbf{f}_e(E, \nu, \mathbf{X}_e, \mathbf{x}_e(t)) = \mathbf{R}_e \mathbf{K}_e (\mathbf{R}_e^T \mathbf{x}_e(t) - \mathbf{X}_e), \quad (1)$$

where

$$\mathbf{K}_e = V_e \mathbf{B}_e^T \mathbf{E} \mathbf{B}_e \quad (2)$$

is the 12×12 element stiffness matrix and V_e is the element volume. Note that \mathbf{f}_e depends nonlinearly on \mathbf{X}_e through both \mathbf{K}_e and \mathbf{R}_e . We discuss in the Appendix how to calculate the Jacobian of \mathbf{f}_e with respect to \mathbf{X}_e , as this is needed for the reference shape optimization described in Section 6.

Next, we construct the equations of motion by assembling contributions from all the FEM elements. The motion of our deformable objects is described by Newton’s 2nd law, which relates forces to accelerations. Assembling the element-wise force contributions \mathbf{f}_e given in Eq (1), we obtain the following ordinary differential equations in time t :

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{R}\mathbf{K}(\mathbf{R}^T \mathbf{x} - \mathbf{X}) = \mathbf{f}_{\text{ext}}. \quad (3)$$

Here the sparse stiffness matrix \mathbf{K} is assembled from the element-wise \mathbf{K}_e defined in Eq (2). The non-zero components for each row of \mathbf{K}_e are determined by the underlying topology structure of the volumetric model. The mass matrix \mathbf{M} reflects the lumped mass at each node [Kim and Pollard 2011], which is derived from a uniform density estimated by measuring the total mass and volume of the object in advance. We use Rayleigh damping $\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$ to model the effects of friction and air drag. Finally, \mathbf{f}_{ext} stands for external forces and equals gravity in our simulation. The solution \mathbf{x} of Eq (3) depends on the unknown parameters $\mathbf{p} = (E, \nu, \alpha, \beta)$, as well as on \mathbf{X} .

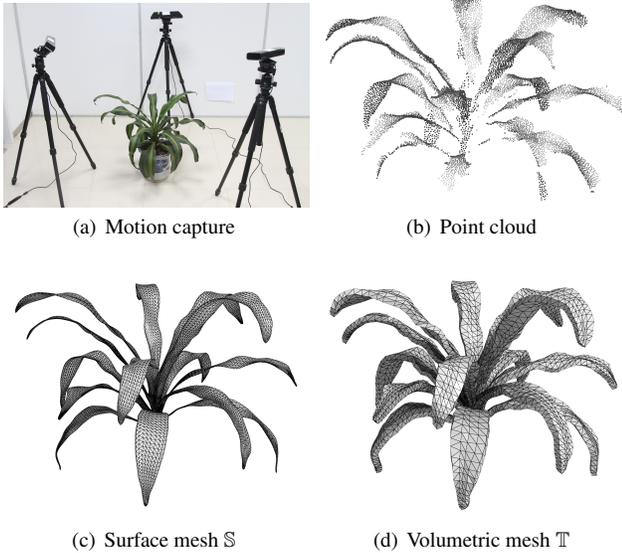


Figure 3: Our capture and tracking setup: three Kinect sensors are placed around the object (a) and the deformation point cloud sequence (b) is captured at 30Hz; a surface mesh \mathbb{S} with 15368 vertices (c) is used as a template to track captured point clouds and its corresponding volumetric mesh \mathbb{T} with 9594 nodes (d) is used for FEM simulations.

5 Physics-based Probabilistic Tracking

The tracking algorithm needs to handle noisy input with occlusions, fast movements, and large deformations. We thus augment a robust physics-based tracking algorithm [Schulman et al. 2013] that formulates the deformation tracking as a maximum a posteriori (MAP) estimation problem, and solves it with the expectation maximization (EM) algorithm. Physics-based tracking of deformable objects offers three main advantages in our context:

- physical constraints, e.g., elasticity and collisions, can be handled by the simulation implicitly and efficiently;
- the incorporation of soft body simulation is essentially free here as we need it for the modeling and synthesis anyway;
- physical parameters, e.g., material configuration, can be optimized to improve the tracking in an iterative fashion.

More specifically, a fine surface mesh \mathbb{S} (Figure 3(c)) of the object is constructed first. Subsequently, it is used as a template for the tracking from point clouds in time, i.e., the assimilation of the point cloud observations with the dynamics described by appropriate equations of motion. A coarser version of this surface mesh is passed to TETGEN [Si 2011] to generate a lower resolution volumetric mesh \mathbb{T} (Figure 3(d)), which encloses \mathbb{S} for the FEM simulation. To deform the triangular surface mesh \mathbb{S} from the simulated tetrahedral elements of \mathbb{T} , we employ the mesh embedding technique proposed in [Capell et al. 2002; Kim and Pollard 2011].

We use three Kinect sensors (Figure 3(a)) to capture the point clouds of the deformation sequence (Figure 3(b)). For a frame consisting of N points at a given time instant, we denote by $\mathbf{c} = \mathbf{c}_{1:N}$, $1 \leq n \leq N$ the point coordinates in the point cloud, and by $\mathbf{s} = \mathbf{s}_{1:K}$, $1 \leq k \leq K$ the vertex positions in the surface mesh \mathbb{S} . Our task is to infer the values of \mathbf{s} given the point cloud

c. We enforce a prior on \mathbf{s} that penalizes potential energy of the deformable objects. The correspondence between the point cloud and the mesh vertices, however, is not known and is treated as latent variables z_{kn} , which indicate if nodal value \mathbf{s}_k of the surface mesh has contributed to the observation \mathbf{c}_n . Assuming that \mathbf{c}_n is normally distributed around \mathbf{s}_k as $\mathbf{c}_n \sim \mathcal{N}(\mathbf{s}_k, \Sigma_k)$ with an isotropic covariance matrix $\Sigma_k = \sigma^2 \mathbf{I}$, we seek the most probable vertex positions given the measurements:

$$\mathbf{s} = \arg \max_{\mathbf{s}} p(\mathbf{s}|\mathbf{c}).$$

To solve this MAP estimation problem, the EM algorithm is employed. In the first (E) half-step, the total log joint probability lower bound $\log p(\mathbf{s}, \mathbf{c})$ is computed based on the expectation over the latent variables $p(z_{kn}|\mathbf{s}, \mathbf{c})$. In the second (M) half-step, the lower bound is maximized with respect to the vertex positions:

$$\mathbf{s} = \arg \max_{\mathbf{s}} [\log p(\mathbf{c}|\mathbf{s}) + \log p(\mathbf{s})]. \quad (4)$$

The second term in Eq (4) corresponds to the potential energy of the deformable object, which can be optimized through simulating the physics-based deformable model. The first term reflects how well the model explains the measurements. To reduce the error, we introduce an artificial external force on each mesh vertex to move them closer to their corresponding sample points in the point cloud:

$$\mathbf{f}_k = \eta \sum_n p(z_{kn}) \Sigma_k^{-1} (\mathbf{c}_n - \mathbf{s}_k). \quad (5)$$

Here η is a scaling factor to control the magnitude of the artificial force, similar to the stiffness of virtual springs. Thus, our deformable object simulation in the context of tracking consists of solving Eq (3) with the external forces \mathbf{f}_{ext} given by the vector assembled from the contributions \mathbf{f}_k defined in Eq (5). Hereafter we denote the resulting solution \mathbf{s} at time instance t by $\hat{\mathbf{x}}_t$.

Interested readers are referred to [Schulman et al. 2013] for more derivation details. The main difference in our implementation is that we simulate the volumetric mesh using co-rotated linear FEM, while Schulman et al. [2013] simulate mass-spring systems with the Bullet engine. More specifically, we use mesh embedding to transfer the virtual forces from the surface mesh to the tetrahedral mesh, and map the displacements from the tetrahedral mesh back to the surface mesh. Multiple EM iterations are required to align the mesh template and the point cloud sufficiently well when large discrepancy appears between them, therefore we run the motion simulation until convergence for each EM iteration.

6 Deformation Parameter Estimation

We propose a data-driven method to estimate the unknown static physical parameters $\mathbf{p} = (E, \nu, \alpha, \beta)$ and the reference shape \mathbf{X} . Our problem is formulated as a spacetime optimization, where the following objective function \mathcal{F} measures the deviation between the simulated and captured trajectories:

$$\min_{\mathbf{p}, \mathbf{X}} \mathcal{F}(\mathbf{p}, \mathbf{X}) = \sum_t \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2. \quad (6)$$

Here we use the sum of squared distances across all frames t as a fitting measure, where $\hat{\mathbf{x}}_t$ is the output of the tracking component, and \mathbf{x}_t is the simulated positions at frame t in Eq (3).

This spacetime optimization problem is large, nonlinear and non-convex. The evaluation of the objective function is expensive, and the evaluation of gradients is far worse still, with all the frames from the motion trajectories. Conventional techniques simply cannot solve it effectively. Therefore we propose a novel splitting scheme

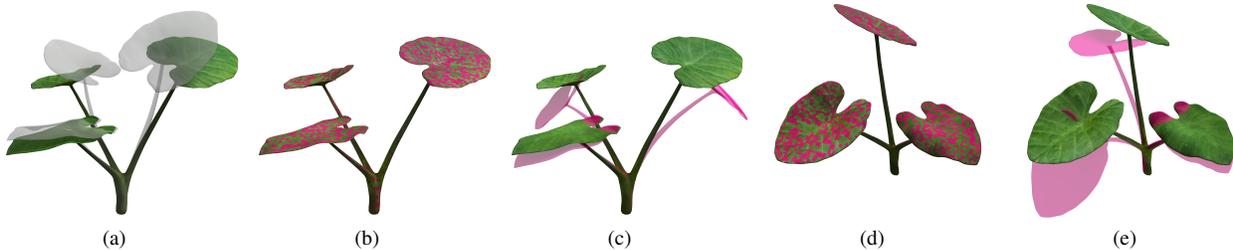


Figure 4: Reference shape optimization for a taro plant example: (a) the static shape (textured) and its optimized reference shape (gray); (b) captured static shape (textured) and simulated static shape (pink) with optimized reference shape; (c) simulated static shape without using optimized reference shape; (d) and (e) are another view of (b) and (c), respectively.

Algorithm 1 Splitting Optimization

| | |
|---|----------|
| Warm start $\mathbf{p}_0 = (E_0, \nu_0, \alpha_0, \beta_0)$; | ▷ (§6.3) |
| repeat | |
| Find \mathbf{X} that satisfies static equilibrium; | ▷ (§6.1) |
| Update \mathbf{p} to decrease trajectory deviation ; | ▷ (§6.2) |
| until Minimization of \mathcal{F} converges. | |

which alternately and iteratively optimizes \mathbf{X} and \mathbf{p} . The first half-step solves for the high-dimensional variable \mathbf{X} given the material parameters \mathbf{p} and the captured trajectories; this can be carried out rapidly using physics-based simulation with a simpler objective to maintain equilibrium for the static shape. Then the subsequent half-step computes the material parameters \mathbf{p} (whose dimension is much smaller than that of \mathbf{X}) as the minimizer of the least squares problem Eq (6) with \mathbf{X} fixed. The advantage of the splitting scheme lies essentially in the utilization of a far simpler and more efficient procedure for solving for the large number of unknowns in \mathbf{X} alone.

The overall splitting scheme is described in Algorithm 1. Within each iteration of the algorithm, we first run the reference shape optimization (Section 6.1) to guarantee that the current deformation parameters \mathbf{p} and their corresponding reference shape \mathbf{X} satisfy the physical constraints. Then we simulate the whole trajectory and update the parameters \mathbf{p} along the direction which can further decrease the trajectory deviation, using a standard downhill search method (Section 6.2). We run the scheme iteratively until it converges sufficiently. A crucial procedure for obtaining a good initial guess is described in Section 6.3.

6.1 Reference Shape Optimization

Taking the static shape as the reference shape is quite common in modeling and simulation of deformable objects. For objects with soft material or thin-long features, however, the difference between the two simply cannot be neglected if faithful reconstruction is desired. We choose static shapes with no external contacts or self-contacts, hence the gravity is the only factor that causes the difference between the static shape and the reference shape. We then simplify Eq (3) accordingly to optimize for the reference shape that best explains the observed static shape.

More specifically, we denote the static shape as \mathbf{x}^s and drop the first two terms of Eq (3):

$$\mathbf{R}\mathbf{K}(\mathbf{R}^T \mathbf{x}^s - \mathbf{X}) = \mathbf{M}\mathbf{g}. \quad (7)$$

This suggests an objective to achieve the least force residual:

$$\min_{\mathbf{X}} \mathcal{F}(\mathbf{X}) = \min_{\mathbf{X}} \left\| \mathbf{R}\mathbf{K}(\mathbf{R}^T \mathbf{x}^s - \mathbf{X}) - \mathbf{M}\mathbf{g} \right\|^2. \quad (8)$$

For a volumetric model with N nodes, the dimension of the search space for \mathbf{X} is $3N$. Conventional optimization techniques are simply not fast enough for this high dimensional problem in our context, thus we adopt the virtual force idea again [Schulman et al. 2013], which has led to Eq (5), to approximate the optimization of \mathbf{X} by an artificial-time simulation. Specifically, we apply the current force residual as virtual forces to each node, and rely on the mock-physics simulation to update the reference shape. The simulation is stopped when the force residual is sufficiently small in norm and the simulation stabilizes around an equilibrium. Readers can also draw similarity between the virtual force idea and the well-known active contours method from the computer vision community [Kass et al. 1988]. The major advantage of the simulation-based method is its robustness and speed. Furthermore, since optimizing Eq (6) alone cannot guarantee the equilibrium constraint of Eq (7), we need to optimize for the reference shape for each iteration of the splitting optimization algorithm.

Figure 4(a) presents the reference shape we recovered from a taro plant model: it looks intuitive and reasonable. The simulated stable state under gravity using our derived reference shape can match the captured static shape with high accuracy; see both Figures 4(b) and 4(d). In contrast, upon comparing Figures 4(b) vs. 4(c) and Figures 4(c) vs. 4(e), obvious sagging is observed when directly using the captured static shape as the reference shape to do the simulation.

To achieve fast and stable simulation for the above physics-based optimization method, we use an implicit ODE solver when updating the reference shape. To this end we need to compute the Jacobian of the elastic forces with respect to the reference shape $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}$. The derivation of this Jacobian is quite involved; see Appendix.

6.2 Deformation Parameter Fitting

We require motion trajectories of the soft object to estimate its deformation parameters \mathbf{p} that includes the elasticity parameters and damping coefficients. We trigger the motion by simple interactions such as twisting and pulling, after which the soft object typically oscillates around and gradually settles down at its initial static equilibrium. Such motion and deformation can be reproduced by simulating the fitted deformable model with appropriate material properties starting from the same initial state. Considering that the reference shape \mathbf{X} can be separately obtained by the method introduced in the previous section, the objective function in Eq (6) can be rewritten as:

$$\mathcal{F}(\mathbf{p}) = \sum_{t,k} \|\mathbf{x}_{tk} - \hat{\mathbf{x}}_{tk}\|^2 \quad (9)$$

where t is the frame index and k is the node index.

To minimize $\mathcal{F}(\mathbf{p})$, we employ the gradient-free Nelder-Mead

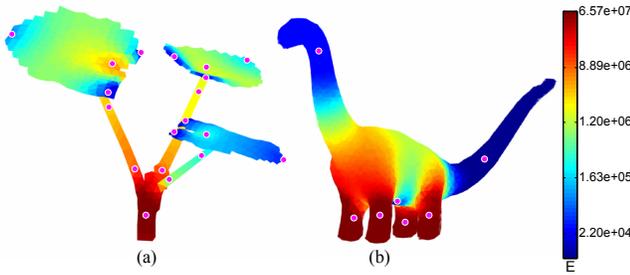


Figure 5: Visualization of Young’s moduli E for a taro plant (a) and a synthetic dinosaur model (b), obtained by blending the estimated values at the marked control points.

algorithm, *a.k.a.*, the downhill simplex method [Nocedal and Wright 2006]. Gradient-based optimization methods require a rather complicated calculation of derivatives, involving the adjoint method [Giles and Pierce 2000; McNamara et al. 2004]. This in turn depends on the noisy trajectories \hat{x} that do not strictly satisfy physical constraints.

For a homogeneous material, the vector $\mathbf{p} = (E, \nu, \alpha, \beta)$ is only four dimensional. However, homogeneous materials typically cannot faithfully model real world soft objects, as their material distributions are usually more complex. Our optimization algorithm above can handle multiple materials without much change, although the more material parameters we use (*i.e.*, the larger \mathbf{p} is), the slower the convergence. Xu et al. [2015] tackled this problem by introducing a reduced material space. We adopt a different approach that strategically places a few control points on the geometry, and then linearly blends the material parameters of these control points to obtain the material specification for each tetrahedron. Figure 5 illustrates the control point placements and Young’s moduli E for two models. For the taro example, control points are placed along the axial direction and the normalized axial distances are used as blending weights. For the synthetic dinosaur model, we manually placed seven control points on the extremities, and the blending weights are calculated using harmonic coordinates. Similar to skinning, we could also involve artists in placing the control points and painting the blending weights for more complex structures.

6.3 Warm Start

Figure 6 shows a typical landscape of our objective function $\mathcal{F}(\mathbf{p})$, which contains degenerated areas, large plateaus, and narrow valleys with multiple local minima. Searching for the true solution of the Young moduli E on such a landscape is extremely costly and error prone, especially for models of multiple materials, due to the curse of dimensionality. The efficiency and robustness of the parameter estimation can be greatly improved, if it starts from a good initial guess in the vicinity of the solution valley. Here we propose an effective warm start strategy that utilizes modal analysis.

In modal analysis, small deformation displacements $\mathbf{u} = \mathbf{x} - \mathbf{X} \in \mathbb{R}^n$ are represented using linear combinations of natural vibration modes of the structure as $\mathbf{u} = \Phi \mathbf{z}$, where each column of $\Phi = [\phi_1, \phi_2, \dots, \phi_k]$ represents one deformation mode and $\mathbf{z} \in \mathbb{R}^k$ are their corresponding amplitudes. The modes can be obtained by solving the generalized eigenvalue problem $\mathbf{K}\phi_i = \lambda_i \mathbf{M}\phi_i$, and represent the natural characteristic displacements that the elastic object can undergo [Sifakis and Barbic 2012]. The natural vibration frequency $\sqrt{\lambda_i}$ is implicitly determined by Young’s modulus through the stiffness matrix \mathbf{K} . Intuitively speaking, softer materials vibrate slower. Figure 7 illustrates the frequency $\sqrt{\lambda_1}$ of the first mode for a synthetic bar with three different E values.

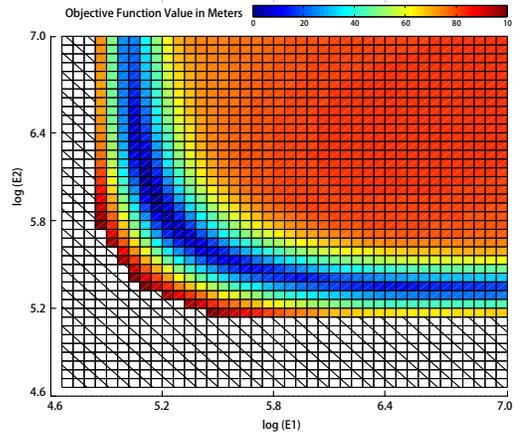


Figure 6: The landscape of the objective function $\mathcal{F}(\mathbf{p})$ of a synthetic bar with two material control points. The striped area marks where the FEM simulation with virtual forces is degenerated and cannot obtain a solution for \mathbf{X} .

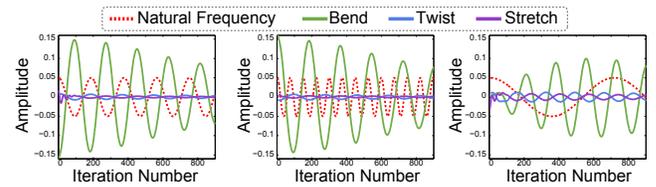


Figure 7: Frequency comparison using the synthetic bar example with Young’s modulus set to 6.8×10^5 (ground truth), 4.0×10^6 and 5.0×10^4 , respectively. The red dashed lines represent the natural frequency of the first eigen mode, while the other lines show three captured trajectories projected onto the first mode.

Therefore, if the estimated E value is closer to the ground truth, the frequency $\sqrt{\lambda_1}$ derived from modal analysis should be close to that of the captured data. We uniformly sample a total of fifty E values in the range of $[1e4, 1e7]$, and compute their corresponding reference shape \mathbf{X} as described in Section 6.1. With known E and \mathbf{X} , we can derive \mathbf{K} and perform the general eigen-decomposition to estimate $\sqrt{\lambda_1}$ and ϕ_1 . Then we extract the displacements from the captured trajectories and project them onto the estimated first mode ϕ_1 , as depicted by the solid lines in Figure 7. The corresponding vibration frequency of the real data ω_1 is then estimated by averaging the periods of multiple cycles of the projected displacements. We use the value E that produces the best match between $\sqrt{\lambda_1}$ and ω_1 as the initial guess to further improve upon using the optimization method described in Section 6.2. Interestingly, Li et al. [2014] recently exploited a related frequency matching idea to directly construct the stiffness matrix by uniformly scaling the eigenvalues to match the first non-zero eigenmode frequency with a user-defined animation period.

To handle cases of multiple materials, we utilize coordinate descent to sequentially and cyclically update the estimation for each material parameter. Furthermore, as linear modal coordinates are inadequate for large deformations, we perform the eigen-analysis in the rotation-strain space. Interested readers are referred to [Huang et al. 2011] for more implementation details.

| | | E | | | | | | | | α | β |
|---------------------|-----------------------|--------|--------|--------|--------|--------|--------|--------|--------|----------|---------|
| Bar (1 ctrl pts) | ground truth | 6.8e+5 | - | - | - | - | - | - | - | 2.0e-2 | 1.0e-3 |
| | estimated(1 ctrl pts) | 6.7e+5 | - | - | - | - | - | - | - | 1.9e-2 | 1.6e-3 |
| | estimated(8 ctrl pts) | 7.2e+5 | 7.2e+5 | 6.3e+5 | 6.7e+5 | 6.5e+5 | 6.6e+5 | 7.0e+5 | 6.8e+5 | 2.0e-2 | 1.3e-3 |
| Bar (8 ctrl pts) | ground truth | 1.0e+5 | 1.0e+6 | 1.0e+4 | 6.8e+5 | 2.0e+6 | 7.0e+4 | 1.0e+7 | 3.0e+4 | 2.0e-2 | 1.0e-3 |
| | estimated(8 ctrl pts) | 1.0e+5 | 1.0e+6 | 1.2e+4 | 6.6e+5 | 2.1e+6 | 6.6e+4 | 1.0e+7 | 1.9e+4 | 1.9e-2 | 6.0e-4 |
| Dinosaur | ground truth | 2.0e+5 | 1.0e+4 | 1.0e+5 | 1.0e+6 | 1.0e+6 | 1.0e+6 | 1.0e+6 | - | 2.0e-2 | 1.0e-3 |
| | estimated(7 ctrl pts) | 2.0e+5 | 9.9e+3 | 9.5e+4 | 1.0e+6 | 1.0e+6 | 1.0e+6 | 1.0e+6 | - | 1.9e-2 | 0.4e-3 |

Table 1: Material optimization and damping coefficients estimation for three synthetic examples: the bar in Figure 9 with one and eight material control points, and the dinosaur in Figure 5(b) with seven control points.

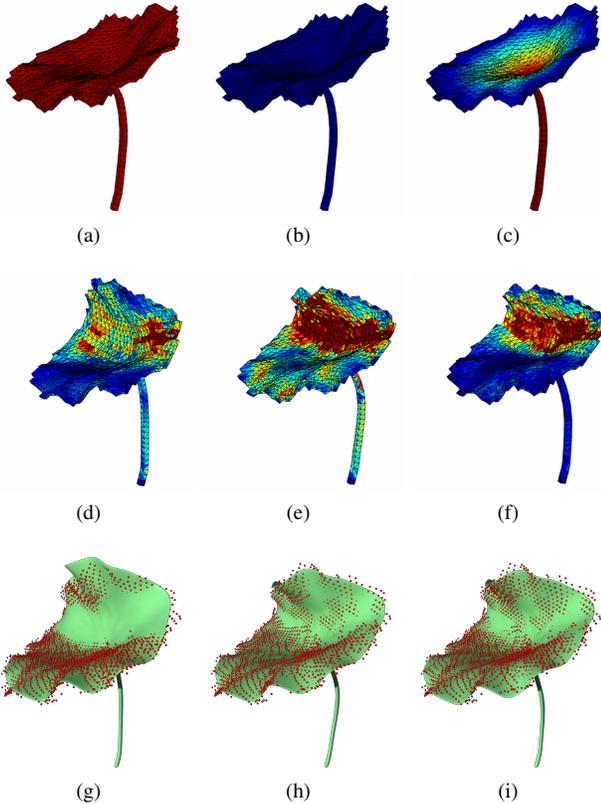


Figure 8: Comparison of tracking results using user-specified material properties vs. optimized ones. (a-c) soft, hard and optimized heterogeneous material distribution as input; (d-f) stress distribution for a frame with large deformation from the output; (g-i) tracking result overlaid with the point cloud data.

7 Results

Deformation Capture and Tracking We use one Artec Eva scanner (<http://www.artec3d.com/>) to firstly construct the static surface geometry of an object of interest. Then we use three Kinect sensors, synchronized and calibrated by the Artec Studio, to capture the point clouds of the object’s motion and deformation at 30Hz, as shown in Figure 3(a). We introduce an initial deformation to the soft object by simple manipulations, and then capture the subsequent motion as the object oscillates and returns to its static equilibrium. The captured point clouds are noisy and incomplete due to occlusions and limited sensory coverage of the sensors.

For the first iteration, we simply use the static shape as the reference shape and user-specified material properties for the simulation-

| Model | |  Plant |  Beam |  Phone holder |
|--------------|---------|---|--|--|
| ΔX | Mean(m) | 8.88e-5 | 2.59e-4 | 1.70e-3 |
| Δx^s | Mean(m) | 6.35e-5 | 2.22e-5 | 2.25e-9 |
| ANM | Time(s) | 9.27 | 3.25 | 17.99 |
| Ours | Time(s) | 12.51 | 6.03 | 21.94 |

Table 2: Comparison of our reference shape optimization with the ANM solver of Chen et al. [2014], in terms of both accuracy and performance. The 3D models were normalized first before we compute the average differences between the shapes. Courtesy of [Chen et al. 2014] for the images and data.

based point cloud tracking. The subsequent iterations of tracking use the result of the deformation parameter optimization from the previous iteration. We compare the tracking results in Figures 8 (g)–(i), using a uniformly soft material, a uniformly hard material, and the nonuniform optimized material, as shown in Figures 8 (a)–(c) respectively. This example demonstrates that the tracking algorithm can generate visually good tracking results for a large range of the elasticity parameters, e.g., at the first iteration with user-specified materials. However, harder materials may not track the data well in certain regions, as shown in Figure 8(g); while softer materials tend to over-fit to noise, as shown in Figure 8(h). This example also clearly demonstrates the advantage of the iterative scheme that alternates the tracking and parameter estimation. Using the optimized elastic parameters for tracking, both the deformation and the stress field can be reconstructed more reliably.

Reference Shape Optimization We validate our reference shape optimization method by comparing its results with those of the state-of-the-art method of Chen et al. [2014]. They used an Asymptotic Numerical Method (ANM) to solve for the reference shape that can generate a desired target shape under known external forces. We compare the difference between the reference shapes ΔX derived from the ANM method and ours, and also the difference between the simulated static shapes Δx^s , as shown in Table 2. We also compare the performance of our simulation-based optimization to the ANM method, and the computation timing is comparable. This is rather satisfactory as we do not require any specialized optimization solver, given that our mock-physics simulation is needed for the modeling and simulation and thus comes essentially free. It is also much easier to incorporate other physical constraints, such as avoiding interpenetrations, using our method.

Material Parameter Estimation To validate the accuracy of our material parameter estimation algorithm, we first use synthetic data generated by forward simulations with known elasticity parameters. The first is a dinosaur model with seven material control points as shown in Figure 5(b). The second is a bar model as shown in Figure 9. We fix one end of the bar on the wall and extend the bar

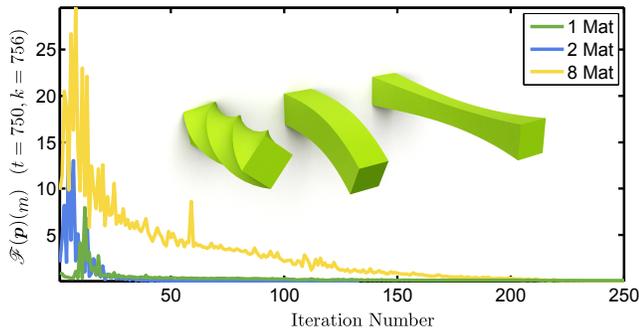


Figure 9: Convergence characteristics of our deformation parameter estimation algorithm for a synthetic bar example with three material configurations (one, two and eight material control points). The trajectories contain 750 frames and the volumetric mesh has 756 nodes.

horizontally. Three initial deformed configurations, i.e., twisting, bending, and stretching, are used to generate simulated motion trajectories, which are subsequently used as input to our optimization algorithm. Figure 9 shows the convergence characteristics of three tests with the ground truth trajectories generated using one, two and eight material control points randomly placed, respectively. Table 1 further shows the numerical values of the estimated parameters with respect to the ground truth. Generally speaking, the more material control points we use in generating the ground truth, the more iterations it takes to estimate the parameters, but our solutions are all in the right order of magnitude as the ground truth. We also perform a test to see if our algorithm may overfit homogeneous materials and result in bogus heterogeneity. We deliberately fit eight material control points to a bar of homogeneous material, i.e., a bar with only one control point. The result in the third row of Table 1 shows all the eight fitted values are within $\pm 7\%$ of the ground truth.

We also test our algorithm with two real-world objects shown in Figure 10: a silicon pot holder, and an elastomer hanger. The values for E obtained from our method are $7.0e + 6$ and $5.6e + 6$, respectively. The results are validated through both static loading tests and dynamic load releasing tests. More specifically, we fix the objects on one end either horizontally or vertically, and attach different weights on the other end as shown in Figure 10. Table 3 lists the averaged node distances between the captured and the simulated equilibrium states upon loading of the external weights. For the dynamic test, the external weights are suddenly released and the vibrations of the soft objects are simulated and compared with the ground truth video. We encourage the readers to watch this side by side comparison in the accompanying video.

We have learned material parameters for a few live plants from induced natural vibrations after being pulled and then released by a user. As the ground truth values of elasticity parameters for plants are typically not available [Barbič and Zhao 2011], we evaluate our results by simulating the learned models from estimated initial state of the captured motions. Side-by-side comparisons show high fidelity of the simulated motions with respect to the captured motions. We have also synthesized a series of plant motions and deformations subject to novel external perturbations. The first row of Figure 11 and the second row of Figure 1 show a lotus leaf responding to falling raindrops and swinging in an artificial wind field, respectively. The estimated Young moduli smoothly decrease from the center of the leaf to the edge, thus resulting in realistic folding of the leaves. The second row of Figure 11 shows a taro plant disturbed by user interactions. And the last row of Figure 11 shows a dracaena fragrans plant moving in an artificial wind field. Note that dracaena fragrans leaves actually demonstrate anisotropic proper-

ties which we currently do not model. However, the synthesized motions still convey the characteristic deformation of the plant to a certain extent. We model all the plants with multiple material control points. Modeling plants as homogeneous material also works but will result in larger objective function values when the optimization converges and larger discrepancies between the tracked and reconstructed trajectories.

Performance Table 4 shows the statistics of our deformable models and a breakdown of the computational cost as measured on an 8-core 3.50GHz Intel Xeon E5-2637 desktop. The performance of the parameter estimation component correlates with the number of tetrahedral elements, the number of material control points, and the number of frames in the motion trajectory. Considering the high dimensionality of the parameter space and the inherent difficulty of the spacetime optimization tasks, we deem our optimization algorithm robust and reasonably fast.

8 Discussion

We have presented the first complete system that can (i) capture deformation of generic soft objects in high fidelity with low-cost depth sensors; and (ii) estimate plausible deformation parameters from these pure kinematic motion trajectories, without requiring any force-displacement measurements as is common in traditional methods. Using the learned deformation models, new motion and deformation can be synthesized at interactive rates to respond to dynamic perturbations or satisfy user-specified constraints.

The two main components of our framework are the physics-based deformation tracking and the spacetime optimization of deformation parameters. The tracking component builds upon a state-of-the-art tracking algorithm that implicitly handles physical constraints and thus is robust to noisy and partial point cloud measurements. Our motion-based parameter estimation strategy requires no tedious force-displacement measurement procedures, and is able to reconstruct dynamic properties such as damping coefficients as well. We run these two components in an iterative fashion so that the results of parameter optimization can in turn enhance the tracking performance. Due to self-occlusions and hardware limitations, the captured point clouds typically are partial and noisy. The learned elastic model facilitates more faithful inference of the missing part of the soft objects, such as the thin stem of the lotus leaf. It also helps eliminate over-fitting and achieve better stress distribution for large deformations.

Our system’s ability to recover both reference shape and deformation parameters simultaneously is a major novelty of our method. Estimating either the rest shape or material properties alone is difficult, and doing both together is even more difficult. We were inspired by the reduced model reference shape optimization of Coros et al. [2012], but we wanted to optimize the full geometry of the reference shape while simultaneously optimizing material parameters. We solve the spacetime optimization problem by a novel splitting scheme that separates the reference shape optimization and the material fitting. The reference shape optimization takes advantage of the dynamics simulation by applying virtual forces, which is not only fast but also robust to noisy data. We reduce the dimensionality of the material fitting problem by introducing a user-in-the-loop skinning scheme, where users manually mark out a few material control points. The results clearly show that both our captured and synthesized deformations are accurate and realistic.

Limitations and Future Work Our template-based tracking requires a high quality surface mesh as the initial shape, which we manually create from scanned point clouds. Automatic surface re-

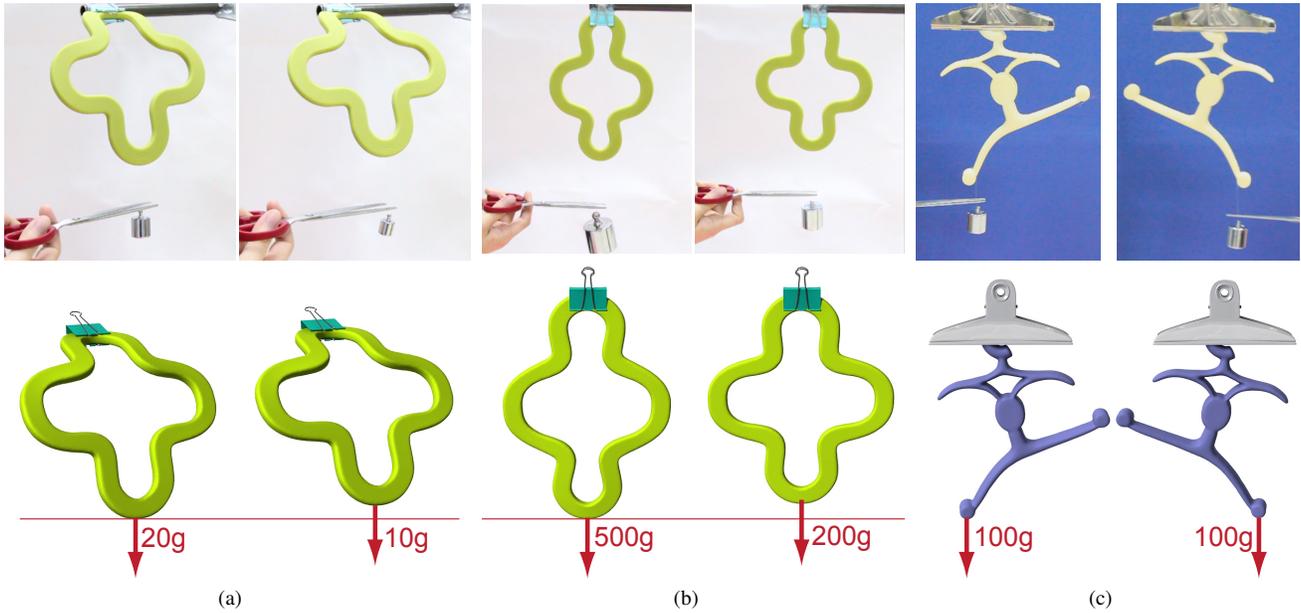


Figure 10: Static loading tests: a silicon pot holder is bent (a) and pulled (b) by external weights; the holder is fixed at one end horizontally in (a) and vertically in (b). An elastomer hanger (c) is tilted on one side by external weights.

| | Pot Holder (Vertical) | | | Pot Holder (Horizontal) | | | Hanger (Vertical) | | |
|------------|-----------------------|--------|--------|-------------------------|--------|--------|-------------------|---------|---------|
| Weight (g) | 0 | 20 | 10 | 0 | 500 | 200 | 0 | 100 (L) | 100 (R) |
| Mean (m) | 1.9e-2 | 5.6e-2 | 4.0e-2 | 7.0e-5 | 2.8e-2 | 7.9e-3 | 1.2e-4 | 3.3e-2 | 3.5e-2 |

Table 3: Averaged node distances between the captured and the simulated equilibrium upon loading of different external weights. Models were normalized first before the distance computation.

| Model | #verts. | #tets. | #nodes | #frames | track (m) | #ctrls | #iter | optimization (h) |
|-------------------|---------|--------|--------|-----------------|-----------|-----------|-------|------------------|
| Bar | 452 | 3000 | 756 | 250 / 250 / 250 | - | 1 / 2 / 8 | - | 0.2 / 0.5 / 1.5 |
| Dinosaur | 19537 | 16270 | 4867 | 523 | - | 7 | - | 3 |
| Pot holder | 12212 | 7488 | 2316 | 81 | 13 | 1 | 2 | 0.7 |
| Hanger | 12837 | 3445 | 1314 | 44 | 9 | 1 | 2 | 0.5 |
| Lotus | 10802 | 6174 | 2197 | 234 | 25 | 2 | 3 | 1.0 |
| Dracaena fragrans | 1876 | 3244 | 1203 | 269 | 7 | 3 | 3 | 0.2 |
| Taro plant | 5832 | 6218 | 2397 | 239 | 38 | 13 | 3 | 2.0 |

Table 4: Performance statistics measured on an 8-core 3.50GHz Intel Xeon E5-2637 desktop. From left to right, the number of mesh vertices (#verts), the number of tetrahedral elements (#tets), the number of volumetric mesh nodes (#nodes), the number of frames of the captured point cloud data (#frames), tracking time in minutes (track), the number of material control points (#ctrls), the number of iterations of tracking and parameter estimation (#iter), and parameter optimization timing in hours (optimization). For Lotus and Dracaena fragrans, we only modeled a single leaf; while for Taro we modeled the whole plant with three leaves.

construction methods, such as Poisson reconstruction, can be used in the future. The corresponding volumetric mesh is built by TetGen automatically, on which we manually specify boundary conditions such as attachment points.

Methods using kinematic input rather than force measurements are fundamentally constrained by the capture frequency in handling stiff materials. Due to the limitations of our low-cost hardware that operates at 30Hz, high frequency vibrations of stiff objects cannot be captured faithfully. Thus our tracking and simulation results will miss high frequency vibrations presented in some of the ground truth videos, such as the dynamic load releasing test. The reconstruction accuracy of the mass damping coefficient in the Rayleigh damping model can be affected. Nevertheless, commonly tested previous examples like pillows, cloth and human organs are well-within the stiffness range that our method handles well.

The coarser volumetric meshes employed in our FEM simulation introduces artificial stiffness due to lack of degrees of freedom,

which causes the optimization to favor softer materials. This problem is more severe when the relative volume difference between the surface mesh and the enclosing tetrahedral mesh becomes larger. Thus, a finer discretization can easily return more accurate estimation of the parameters. In the future, it would be interesting to explore non-linear FEM models to reduce this dependency of the estimation accuracy on discretization resolution as well as to enhance the robustness of the system. Since we have not noticed visual deficiency, we have favored coarse meshes, even though it may be important to use a finer tetrahedral discretizations for medical applications. But for medical applications where accuracy is more important, we would recommend using a finer tetrahedral mesh.

Recovering dynamic models from pure kinematic data is challenging. At this stage we choose a simple and popular elasticity model, which is adequate to produce realistic results that are visually indistinguishable from our capture. More advanced elastic models, such as those presented in [Bickel et al. 2009; Bickel et al. 2010; Wang et al. 2011; Miguel et al. 2012], would likely produce more

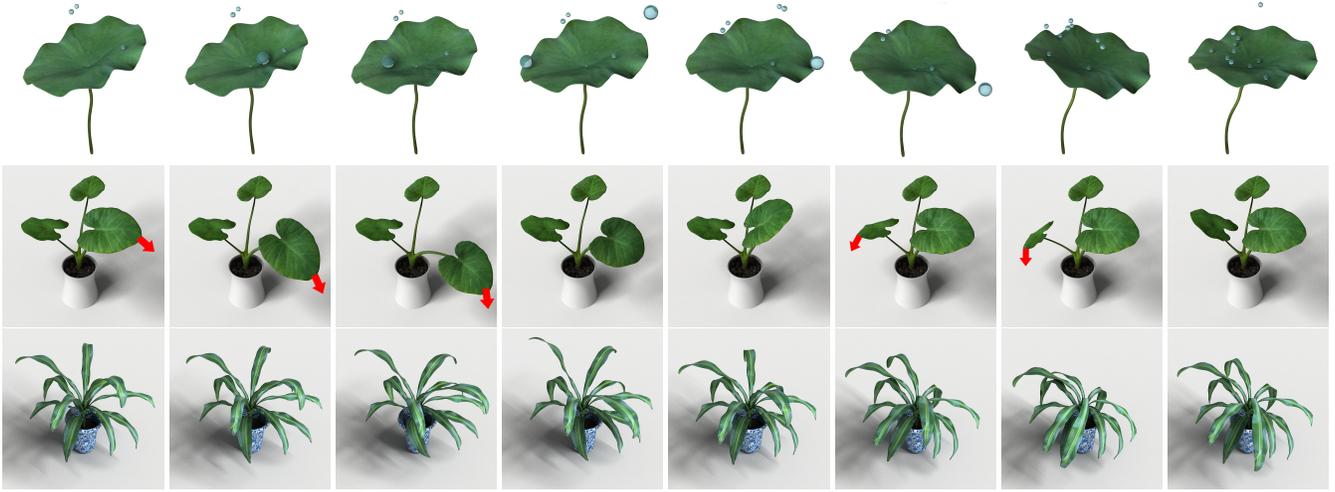


Figure 11: *Top: A single lotus leaf responds to falling raindrops. Second row: A taro plant responds to user interactions. Bottom: A dracaena fragrans shrub moving in wind.*

accurate results on more examples.

Currently, we use a gradient-free downhill simplex method for parameter fitting. Even though the Nelder-Mead algorithm occasionally performs better than gradient-based methods on problems such as ours that contain many local minima, it is worthwhile to experiment with gradient-based methods in the future.

In addition, we would like to extend our parameter optimization algorithm to accommodate more complicated situations, such as motion trajectories that are rich in contacts and collisions. Derouet et al. [2013] is a good starting point to investigate this problem. Finding an automatic skinning method for choosing material control point locations and weights is also an interesting area of future work. James and Twigg [2005] investigated this problem for models with underlying skeletal structures and one would need to extend it to deal with general soft objects. We also wish to re-derive the Reference Shape Jacobian to handle anisotropic materials.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. We are grateful to the authors of [Chen et al. 2014] for sharing their data and fabricated models for our validation and comparison experiments. We also thank Jiacheng Ren, Jiangtao Shen, Keng Hua Sing, Francois Faure and Matthieu Nesme for their help and thoughtful discussions at the various stages of developing this project. This work is supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 (MOE2011-T2-2-152); Microsoft Research Asia Collaborative Research Program (FY14-RES-OPP-002); NSFC (61402459, 61379090, 61331018); National 973 Program (2014CB360503); and Shenzhen Innovation Program (CXB201104220029A, JCYJ20140901003939034, JCYJ20130401170306810, JSGG20130624154940238).

Appendix: Reference Shape Jacobian

Exact computation of the stiffness matrix $\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ for co-rotated linear FEM has been described in the literature [Barbič and Zhao 2011; McAdams et al. 2011; Barbič 2012]. The Jacobian of the elastic forces with respect to the undeformed reference shape $\mathbb{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{X}}$, however, is quite complicated and not studied before. We therefore detail its derivation in this appendix. All the calculations are performed with respect to each tetrahedral element, but

we omit the subscript $-e$ to avoid messy notations. Let

$$\mathbf{X} = [X_1^T, X_2^T, X_3^T, X_4^T]^T \in \mathbb{R}^{12} \quad (10)$$

be the nodal position of a tetrahedral element in the undeformed shape, and $\mathbf{x} = [x_1^T, x_2^T, x_3^T, x_4^T]^T \in \mathbb{R}^{12}$ be the nodal position in the deformed shape. The corotated linear model formulates the nodal elastic force $\mathbf{f} = [f_1^T, f_2^T, f_3^T, f_4^T]^T \in \mathbb{R}^{12}$ as

$$\mathbf{f} = V \mathbf{R} \mathbf{B}^T \mathbf{E} \mathbf{B} (\mathbf{R}^T \mathbf{x} - \mathbf{X}).$$

Here V is the volume of the element, and \mathbf{E} is a constant 6×6 matrix that corresponds to the material’s elastic parameters.

The matrix $\mathbf{R} = \text{diag}(R, R, R, R)$ is a block-diagonal matrix. R is the rigid rotation of the element computed by the polar-decomposition of the deformation gradient $F (= \partial \mathbf{x} / \partial \mathbf{X})$, which is just the top-left 3×3 block of

$$\bar{F} = V_s V_m^{-1} = V_s B_m$$

$$V_s = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad V_m = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (11)$$

and $B_m = V_m^{-1}$ are all 4×4 matrices. The 6×12 matrix \mathbf{B} only depends on \mathbf{X} and is just a rearrangement of B_m . Denoting the (i, j) th element of B_m as B_{ij} , we have:

$$\mathbf{B} = [\bar{B}_1 \ \bar{B}_2 \ \bar{B}_3 \ \bar{B}_4] \quad \bar{B}_i = \begin{bmatrix} B_{i1} & 0 & 0 \\ 0 & B_{i2} & 0 \\ 0 & 0 & B_{i3} \\ B_{i2} & B_{i1} & 0 \\ 0 & B_{i3} & B_{i2} \\ B_{i3} & 0 & B_{i1} \end{bmatrix}$$

Now we are ready to derive the reference shape jacobian

$$\mathbb{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{X}}.$$

The q -th column of \mathbb{K} , represented by \mathbf{k}_q , is the derivative with respect to the q -th component of \mathbf{X} . According to Eq (10), it is actually the i -th component of node j , which we denote as X_{ij} , with i and j satisfying $q = 3(j-1) + i$, $1 \leq i \leq 3$, $1 \leq j \leq 4$. By applying the chain rule, we have:

$$\begin{aligned} \mathbf{k}_{3(j-1)+i} &= \frac{\partial \mathbf{f}}{\partial X_{ij}} \\ &= V' \mathbf{R} \mathbf{B}^T \mathbf{E} \mathbf{B} (\mathbf{R} \mathbf{x} - \mathbf{X}) + V \mathbf{R}' \mathbf{B}^T \mathbf{E} \mathbf{B} (\mathbf{R} \mathbf{x} - \mathbf{X}) \\ &\quad + V \mathbf{R} \mathbf{B}'^T \mathbf{E} \mathbf{B} (\mathbf{R} \mathbf{x} - \mathbf{X}) + V \mathbf{R} \mathbf{B}^T \mathbf{E} \mathbf{B}' (\mathbf{R} \mathbf{x} - \mathbf{X}) \\ &\quad + V \mathbf{R} \mathbf{B}^T \mathbf{E} \mathbf{B} (\mathbf{R}' \mathbf{x} - \mathbf{e}_{3(j-1)+i}) \end{aligned}$$

Here the prime symbol ($'$) represents the derivative ($\partial/\partial X_{ij}$), $\mathbf{e}_q \in \mathbb{R}^{12}$ is the q -th standard basis vector of \mathbb{R}^{12} .

We assume that the nodes of an element are sorted to satisfy $\det(V_m) > 0$; then the element volume can be computed by $V = \frac{1}{6} \det(V_m)$. From the derivative of a determinant [Petersen and Pedersen 2006]:

$$\frac{\partial V}{\partial V_m} = \frac{1}{6} \frac{\partial \det(V_m)}{\partial V_m} = \frac{1}{6} \det(V_m) V_m^{-T} = V B_m^T.$$

According to the definition of V_m in Eq (11), we have

$$V' = \frac{\partial V}{\partial X_{ij}} = \frac{\partial V}{\partial (V_m)_{ij}} = V B_{ji}$$

The matrix \mathbf{B} is a rearrangement of B_m , which is the inverse of V_m . From the derivative of an inverse:

$$\frac{\partial B_{kl}}{\partial X_{ij}} = -B_{ki} B_{jl}$$

we have

$$\begin{aligned} \mathbf{B}' &= [\bar{B}'_1 \ \bar{B}'_2 \ \bar{B}'_3 \ \bar{B}'_4] \\ \bar{B}'_k &= \frac{\partial \bar{B}_k}{\partial X_{ij}} = - \begin{bmatrix} B_{ki} B_{j1} & 0 & 0 \\ 0 & B_{ki} B_{j2} & 0 \\ 0 & 0 & B_{ki} B_{j3} \\ B_{ki} B_{j2} & B_{ki} B_{j1} & 0 \\ 0 & B_{ki} B_{j3} & B_{ki} B_{j2} \\ B_{ki} B_{j3} & 0 & B_{ki} B_{j1} \end{bmatrix} \end{aligned}$$

The computation of \mathbf{R}' is much more involved. By applying the chain rule again, we have:

$$\mathbf{R}' = \frac{\partial \mathbf{R}}{\partial X_{ij}} = \sum_{k,l} \frac{\partial \mathbf{R}}{\partial F_{kl}} \frac{\partial F_{kl}}{\partial X_{ij}} \quad (12)$$

Gradients of polar decomposition have been derived in [Barbič and Zhao 2011; McAdams et al. 2011; Barbič 2012]. We refer the reader to [Barbič 2012] for more details.

Finally, to calculate the second term of the right hand side of Eq (12), we need

$$\begin{aligned} \bar{F}' &= \frac{\partial \bar{F}}{\partial (V_m)_{ij}} = V_s \frac{\partial B_m}{\partial (V_m)_{ij}} \\ &= -V_s \begin{bmatrix} B_{1i} B_{j1} & B_{1i} B_{j2} & B_{1i} B_{j3} & B_{1i} B_{j4} \\ B_{2i} B_{j1} & B_{2i} B_{j2} & B_{2i} B_{j3} & B_{2i} B_{j4} \\ B_{3i} B_{j1} & B_{3i} B_{j2} & B_{3i} B_{j3} & B_{3i} B_{j4} \\ B_{4i} B_{j1} & B_{4i} B_{j2} & B_{4i} B_{j3} & B_{4i} B_{j4} \end{bmatrix} \end{aligned}$$

Then

$$\frac{\partial F_{kl}}{\partial X_{ij}} = \frac{\partial \bar{F}_{kl}}{\partial (V_m)_{ij}} = (\bar{F}')_{kl}$$

References

- BARBIČ, J., AND ZHAO, Y. 2011. Real-time large-deformation substructuring. *Proc. of SIGGRAPH* 30, 3, 91:1–91:8.
- BARBIČ, J., SIN, F., AND GRINSPUN, E. 2012. Interactive editing of deformable simulations. *ACM Trans. on Graphics* 31, 4, 70:1–70:8.
- BARBIČ, J. 2012. Exact corotational linear FEM stiffness matrix. Tech. rep., Technical report, University of Southern California.
- BECKER, M., AND TESCHNER, M. 2007. Robust and efficient estimation of elasticity parameters using the linear finite element method. In *Proc. Simulation und Visualization*, 15–28.
- BHAT, K. S., TWIGG, C. D., HODGINS, J. K., KHOSLA, P. K., POPOVIĆ, Z., AND SEITZ, S. M. 2003. Estimating cloth simulation parameters from video. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 37–51.
- BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. on Graphics* 28, 3, 89:1–89:9.
- BICKEL, B., BÄCHER, M., OTADUY, M. A., AND LEE, H. R. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Trans. on Graphics* 29, 4, 63:1–63:10.
- BRADLEY, D., POPA, T., SHEFFER, A., HEIDRICH, W., AND BOUBEKEUR, T. 2008. Markerless garment capture. *ACM Trans. on Graphics* 27, 3, 99:1–99:9.
- CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. on Graphics* 21, 3, 586–593.
- CHEN, X., ZHENG, C., XU, W., AND ZHOU, K. 2014. An asymptotic numerical method for inverse elastic shape design. *ACM Trans. on Graphics* 33, 4, 95:1–95:11.
- CHOI, J., AND SZYMCAK, A. 2009. Fitting solid meshes to animated surfaces using linear elasticity. *ACM Trans. on Graphics* 28, 1, 6:1–6:10.
- COROS, S., MARTIN, S., THOMASZEWSKI, B., SCHUMACHER, C., SUMNER, R., AND GROSS, M. 2012. Deformable objects alive! *ACM Trans. on Graphics* 31, 4, 69:1–69:9.
- DE AGUIAR, E., STOLL, C., THEOBALT, C., AND HANS-PETER SEIDEL, N. A., AND THRUN, S. 2008. Performance capture from sparse multi-view video. *ACM Trans. on Graphics* 27, 3, 98:1–98:10.
- DEROUET-JOURDAN, A., BERTAILS-DESCOUBES, F., DAVIET, G., AND HOLLLOT, J. 2013. Inverse dynamic hair modeling with frictional contact. *ACM Trans. on Graphics* 32, 6, 159:1–159:10.
- GILES, M. B., AND PIERCE, N. A. 2000. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion* 65, 393–415.
- HELTEN, T., MÜ, M., SEIDEL, H.-P., AND THEOBALT, C. 2013. Real-time body tracking with one depth camera and inertial sensors. In *Proc. Int. Conf. on Computer Vision*, 1105–1112.
- HILDEBRANDT, K., SCHULZ, C., VON TYCOWICZ, C., AND POLTHIER, K. 2012. Interactive spacetime control of deformable objects. *ACM Trans. on Graphics* 31, 4, 71:1–71:8.

- HUANG, J., TONG, Y., ZHOU, K., BAO, H., AND DESBRUN, M. 2011. Interactive shape interpolation through controllable dynamic deformation. *IEEE Trans. Visualization & Computer Graphics* 17, 7, 983–992.
- JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Trans. on Graphics* 24, 3, 399–407.
- KASS, M., WITKIN, A., AND TERZOPOULOS, D. 1988. Snakes: Active contour models. *Int. J. Computer Vision* 1, 4, 321–331.
- KIM, J., AND POLLARD, N. S. 2011. Fast simulation of skeleton-driven deformable body characters. *ACM Trans. on Graphics* 30, 5, 121:1–121:19.
- LEE, H.-P., FOSKEY, M., NIETHAMMER, M., KRAJCEVSKI, P., AND LIN, M. C. 2012. Simulation-based joint estimation of body deformation and elasticity parameters for medical image analysis. *IEEE Trans. on Medical Imaging* 31, 11, 2156–2168.
- LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. on Graphics* 28, 5, 175:1–175:10.
- LI, H., LUO, L., VLASIC, D., PEERS, P., POPOVIĆ, J., PAULY, M., AND RUSINKIEWICZ, S. 2012. Temporally coherent completion of dynamic shapes. *ACM Trans. on Graphics* 31, 1, 2:1–2:11.
- 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. on Graphics* 33, 4, 108:1–108:10.
- MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. on Graphics* 30, 4, 37:1–37:12.
- MENAMARA, A., TREUILLE, A., POPOVIC, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *ACM Trans. on Graphics* 23, 3, 449–456.
- MIGUEL, E., BRADLEY, D., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., OTADUY, M. A., AND MARSCHNER, S. 2012. Data-driven estimation of cloth simulation models. *Computer Graphics Forum* 31, 2, 519–528.
- MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proc. Int. Conf. on Graphics Interface*, 239–246.
- MÜLLER, M., DORSEY, J., MCMILLAN, L., JAGNOW, R., AND CUTLER, B. 2002. Stable real-time deformations. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 49–54.
- NEALEN, A., MLLER, M., KEISER, R., BOXERMAN, E., CARLSON, M., AND AGEIA, N. 2006. Physically based deformable models in computer graphics. *Computer Graphics Forum* 25, 4, 809–836.
- NOCEDAL, J., AND WRIGHT, S. J. 2006. *Numerical Optimization*. Springer. 2nd Ed.
- OTADUY, M. A., BICKEL, B., BRADLEY, D., AND WANG, H. 2012. Data-driven simulation methods in computer graphics: cloth, tissue and faces. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, 12:1–12:96.
- 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. 87–96.
- PETERSEN, K. B., AND PEDERSEN, M. S. 2006. The matrix cookbook. Tech. rep., MIT.
- SCHONER, J., LANG, J., AND SEIDEL, H.-P. 2004. Measurement-based interactive simulation of viscoelastic solids. *Computer Graphics Forum* 23, 3, 547–556.
- SCHULMAN, J., LEE, A., HO, J., AND ABBEEL, P. 2013. Tracking deformable objects with point clouds. In *Proc. IEEE Int. Conf. on Robotics & Automation*.
- SI, H. 2011. Tetgen, a quality tetrahedral mesh generator and a 3d delaunay triangulator. Tech. rep., WIAS.
- SIFAKIS, E., AND BARBIC, J. 2012. FEM simulation of 3D deformable solids: A practitioner’s guide to theory, discretization and model reduction. In *ACM SIGGRAPH Courses*, 20:1–20:50.
- SKOURAS, M., THOMASZEWSKI, B., BICKEL, B., AND GROSS, M. 2012. Computational design of rubber balloons. *Computer Graphics Forum* 31, 2, 835–844.
- SKOURAS, M., THOMASZEWSKI, B., COROS, S., BICKEL, B., AND GROSS, M. 2013. Computational design of actuated deformable characters. *ACM Trans. on Graphics* 32, 4, 82:1–82:10.
- TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. *Proc. of SIGGRAPH 21*, 4, 205–214.
- TEVS, A., BERNER, A., WAND, M., IHRKE, I., BOKELOH, M., KERBER, J., AND SEIDEL, H.-P. 2012. Animation cartography - intrinsic reconstruction of shape and motion. *ACM Trans. on Graphics* 31, 2, 12:1–12:15.
- VLASIC, D., BARAN, I., MATUSIK, W., AND POPOVIC, J. 2008. Articulated mesh animation from multi-view silhouettes. *ACM Trans. on Graphics* 27, 3, 97:1–97:9.
- WANG, H., O’BIEN, J. F., AND RAMAMOORTHY, R. 2011. Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. on Graphics* 30, 4, 71:1–71:12.
- WEI, X., AND CHAI, J. 2010. Videomocap: Modeling physically realistic human motion from monocular video sequences. *ACM Trans. on Graphics* 29, 4, 42:1–42:10.
- WEI, X., ZHANG, P., AND CHAI, J. 2012. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. on Graphics* 31, 6, 188:1–188:12.
- WUHRER, S., LANG, J., TEKIEH, M., AND SHU, C. 2013. Finite element based tracking of deforming surfaces. *Graphical Models* 77, 1, 1–17.
- XU, H., LI, Y., CHEN, Y., AND BARBIC, J. 2015. Interactive material design using model reduction. *ACM Trans. on Graphics* 34, 2, 18:1–18:14.