

# Measurement-Based Interactive Simulation of Viscoelastic Solids

Jeffrey L. Schoner, Jochen Lang and Hans-Peter Seidel

Max-Planck-Institut für Informatik, Saarbrücken, Germany

---

## Abstract

*Animation of viscoelastic solids in entertainment and medical applications as well as scientific simulation can be improved through observations of real world objects. This paper presents a method for simulating viscoelastic solids in real-time for visual and haptic display along with a method for determining the parameters of the underlying model from automated physical measurements of real world objects. The viscoelastic model is a novel extension of the discrete Green's function matrix for linear elasticity, which combines static behavior represented by Green's functions with dynamic behavior expressed by differential equations inspired by particle systems. We describe a novel estimation method of dynamic contact behavior for heterogeneous complex objects based on these measurements. For this estimation, our method relies only on measurement data previously used in the acquisition of less realistic elastostatic models. In this way our method allows more physically accurate realism in animation of viscoelastic solids without large additional computational costs or any measurements besides those associated with related methods for elastostatic solids.*

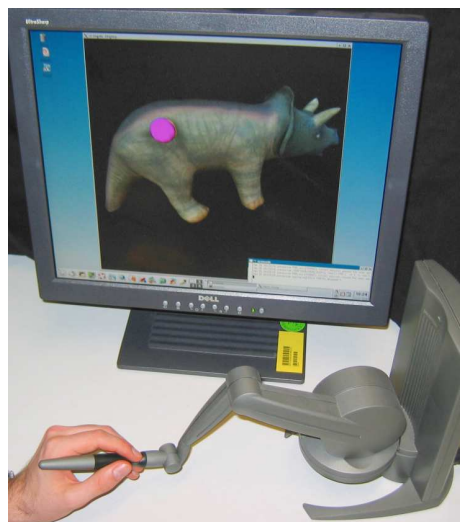
Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling  
I.3.7 [Computer Graphics]: Virtual reality, animation

---

## 1. Introduction

Physical simulation of deformable models for visual display has a long tradition in computer graphics starting with Terzopoulos et al. [TPBF87]. The focus of most of the recent and current research is on increasing the fidelity of the physical simulation while rendering the models at interactive rates [CEO\*93, JP99, DDCB01, PDA01, HGS03]. The increases in fidelity of the physical models are achieved by increasing the number of degrees of freedom in the simulation and by increasing the resolution for display. Our approach shares this focus, although we concentrate on matching simulation to observed object behavior. The realism of a simulation is limited by the realism of the underlying elastic model and even physical models are only as realistic as their material parameters fit the desired behavior. Choosing appropriate parameters is especially critical if the deformable model is to match a specific object behavior as required in medical simulation or in animation.

Traditionally the properties of a deformable solid simulation are tuned by setting global material constants such as



**Figure 1:** The screen shows a user interacting with a model of the natural material replica *triceratops* by Bullyland Inc., N.Y., USA

Young's modulus and Poisson's ratio (or equivalently Lamé constants). These constants along with the object geometry can be used to determine the behavior of an elastic solid. During physical simulation, the behavior of the solid under loading and position constraints is determined with methods from engineering such as the finite element method (FEM) or the boundary element method (BEM). One simple but not particularly precise approach for determining the material constants is to manually adjust them until the object behaves as desired. The meaning of material parameters (especially Poisson's ratio) can often be non-intuitive and is additionally complicated by the influence of geometry. A more precise procedure that has seen application notably in the simulation of human tissue for surgery training is to take mechanical measurements of a sample of the object's material [MK99, dCL99, VKSR00, BUB\*01, HGS03]. However, the use of global material properties is viable only in the case, where the object consists of a homogenous material and where a material sample can realistically be taken.

Recent work on the acquisition of the deformation of "soft" elastic solids addresses these concerns. Lang et al. [LPW02] and Pai et al. [PvdDJ\*01] use a robotic arm at the University of British Columbia Active Measurement Facility (ACME) to apply forces in various directions at points on the surface of the object. Because they measure observable object deformation directly with 3D range flow in stereo imagery of the object's surface, their approach is capable of estimating the physical deformation of heterogeneous objects. They represent the deformation as an individual discrete Green's function for each position (corresponding to vertices of a mesh), resulting in a linear approximation to the observed static deformation. However, viscoelastic effects, in particular creep and relaxation, are not modeled in their approach. Our approach leads to measurement-based models where both elastic and viscoelastic behavior vary for each point on the surface of an object.

### 1.1. Related Work

Previous work related to ours can be found in the categories of animation of deformable models, interactive elastic simulation, and determining material parameters for visual simulation. We will concentrate here on physically motivated approaches to deformable model simulation and refer the reader interested in other approaches to the survey by Gibson and Mirtich [GM97].

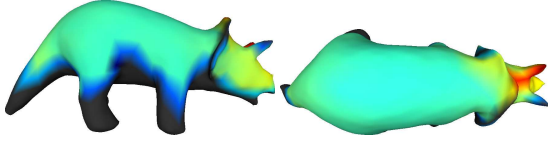
The original work of Terzopoulos et al. [TPBF87] on deformable models introduced elastic solid simulation to computer graphics. They use finite differences to obtain a dynamic model of solid deformation  $M\ddot{u} + D\dot{u} + Ku = f$  and employ a *prescribed metric tensor* for calculation of elastic energy. Later their work was extended [TF88] to viscoelastic solids defining the deformation energy as a second order *controlled-continuity spline kernel*. They use spring-dampers in a *four-element viscoelastic model* consisting of

a Kelvin-Voigt element in series with a Maxwell element (see, for example, Fung's text on biomechanics [Fun93]). The four-element model is physically motivated, but does not employ the constitutive equations of engineering simulations because of computational cost.

Simulation cost can be reduced by modal analysis as shown by Pentland and Williams [PW89]. The modal decomposition is only possible with Rayleigh damping where  $D = \alpha M + \beta K$ . Recently, James and Pai [JP02] have shown how to evaluate these modal representations in real-time on graphics hardware. Hauser et al. [HSO03] demonstrated an efficient implementation with interactive manipulation constraints. A completely different approach to reducing simulation costs is to introduce multi-level mesh hierarchies. DeBunne et al. [DDCB01] employ a set of non-nested mesh hierarchies to adapt the spatial resolution of their simulation based on the force acting on a volume element, although the synchronization of the different mesh hierarchies during dynamic deformation is difficult. Wu et al. [WDGT01] solve the synchronization by creating dynamic progressive meshes in a mixture of precomputation and on-line refinement. James and Pai [JP03] employ surface subdivision meshes representing Green's functions of an elastic solid combined with wavelet decomposition. Similarly, the approach by Hauth et al. [HGS03] uses nested tetrahedral meshes where hierarchical function approximation is performed. The general CHARMS framework for adaptive simulation by Grinspun et al. [GKS02] can also be applied to deformable model simulation. An alternative to adaptive meshes is the use of a dedicated computing cluster such as that described by Szekely et al. [SBD\*00].

Constitutive equations with a linear small strain tensor are employed by Bro-Nielsen and Cotin [BNC96]. In order to achieve interactive simulation rates with their full continuum mechanics model, they use the FEM with Guyan reduction [Guy65], mapping the volumetric elastostatic behavior to the surface. James and Pai [JP99] achieve this instead by employing the BEM. These approaches may also be viewed as finding the Green's functions of the boundary value problem of linear elasticity [JP01]. The realism of these models is limited by the use of the linear small strain tensor, by the constitutive equations (from a simple linear 3D Hookean spring) and by quasi-static simulation. The linear small strain tensor causes the object to distort if the deformation is large. The linear constitutive relationship parameterized by Young's modulus and Poisson's ratio (or equivalently by Lamé constants) only accurately approximates real material for small strain. Finally, the quasi-static simulation cannot produce dynamic effects such as relaxation (force decreasing over time while a deformation is maintained) and creep (deformation creeping back to its rest state after the force has been removed).

O'Brien and Hodgins [OH99] employ the rotationally invariant large deformation (Green's) strain tensor and a



**Figure 2:** False color coding of the triceratops model showing compliance across its surface (red is high, blue low).

strain-rate tensor in their FEM solution scheme but their approach is not interactive. Wu et al. [WDGT01], Debunne et al. [DDCB01] and Picinbono et al. [PDA01] employ these tensors in their respective interactive simulation. Picinbono et al. also render not only isotropic but also *transversely isotropic* materials. Complex non-linear material laws are employed by Hauth et al. [HGS03] based on the mechanical quality concept of springs in their interactive simulation. Teran et al. [TBHF03] make use of the finite volume method (FVM) and a quasi-incompressible, transversely isotropic, hyperelastic constitutive model to simulate skeletal muscles. These increasingly realistic simulation techniques require material parameters which are equally realistic to match the real-world behavior to be modeled. Measurement of material parameters has been recognized as an important step, especially in surgery simulation [MK99, dCL99, VKSR00, BUB\*01, HGS03]. At best, material parameter sampling leads to a realistic description of a sample. However, soft tissues, soft toys, animation characters and other complex objects do not consist of a single material. What is needed to achieve realism is a test of material parameters throughout the volume of the object. An alternative approach has been suggested by Lang et al. [LPW02, PvdDJ\*01, LPS03] by observing the deformation behavior of a complex object directly and fitting the discrete model to these observations. Their measurement-based models allow for a per-vertex variation of material behavior, although they only model linear quasi-static behavior. Their models are not capable of showing relaxation and creep which is a quite noticeable deficiency. In this paper, we show how to model and estimate these viscoelastic effects from their measurement method.

## 2. Elasticity Basics

A discrete Green's function matrix (DGFM) linearly relates surface traction (contact force over area) and displacement of nodes on a discrete mesh for a given boundary configuration. In this paper, we employ linear triangular surface elements for which the vertices of the triangular mesh are these nodes. Each of the  $n$  vertices has either a displacement constraint or a traction constraint. Following the notation used by James and Pai [JP01], from the set  $\Lambda$ , consisting of all the vertices in the mesh, two exclusive subsets are defined:  $\Lambda_u$ , which consists of the vertices with displacement constraints and  $\Lambda_v$ , which consists of those with traction con-

constraint on $i$	constraint on $j$	
	traction	displacement
traction	tract. $\rightarrow$ displ.	displ. $\rightarrow$ displ.
displacement	tract. $\rightarrow$ tract.	displ. $\rightarrow$ tract.

**Table 1:** The four possible relationships between two vertices  $i$  and  $j$  defined by a DGFM entry  $\Xi_{i,j}$ . Entries take the form known constraint value applied to  $j \rightarrow$  unknown constraint value produced on  $i$ . Relations in red correspond to those like a linear spring.

straints. The constraint values are three-dimensional vectors. Let  $\bar{\mathbf{u}}_i$  denote either the known displacement constraint or  $\bar{\mathbf{p}}_i$  the known traction constraint on vertex  $i$ , depending on the corresponding constraint type of  $i$ . Given these constraints, we solve for the unknown traction  $\mathbf{p}_i$  for displacement constrained nodes and for the unknown displacement  $\mathbf{u}_i$  for traction constrained nodes. These values can be combined into  $n$ -dimensional block vectors,

$$\mathbf{v}_i = \begin{cases} \mathbf{p}_i & : i \in \Lambda_u \\ \mathbf{u}_i & : i \in \Lambda_v \end{cases}$$

and

$$\bar{\mathbf{v}}_i = \begin{cases} \bar{\mathbf{u}}_i & : i \in \Lambda_u \\ \bar{\mathbf{p}}_i & : i \in \Lambda_v \end{cases}$$

These  $n$  length block vectors are related by

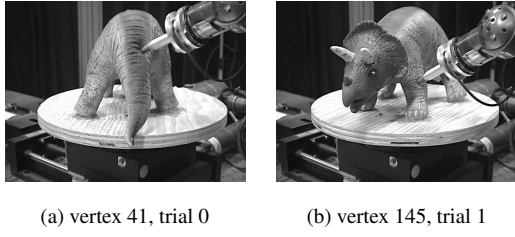
$$\Xi \bar{\mathbf{v}} = \mathbf{v}$$

where  $\Xi$  is an  $n$ -by- $n$  DGFM containing 3-by-3 block entries. The element from the  $i$ -th row and  $j$ -th column of the matrix forms the additive influence the given  $j$ -th element of  $\bar{\mathbf{v}}$  has on the unknown  $i$ -th element of  $\mathbf{v}$ .

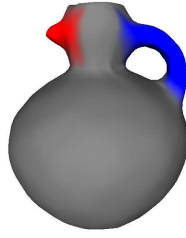
### 2.1. Relating the DGFM to Linear Springs

A simple linear elastic system characterized by a DGFM can be conceptually viewed as a set of linear springs; mathematically, a DGFM relates the known and unknown quantities at all  $n$  vertices through linear combination. The relationship between two entries of the DGFM falls into one of four possible cases depending on the boundary conditions of the nodes (see Table 1). Of the four cases, only two directly map to a linear spring because they relate traction and displacement. Furthermore, during simulation we are only interested in obtaining the displacement for the vertices that are force constrained. Although the traction present on displacement constrained vertices can be calculated, the quantity is not needed for (visual and haptic) display purposes. Therefore, the displacement of vertex  $i$  for these conditions is given by

$$\mathbf{u}_i = \sum_j \Xi_{i,j} \mathbf{p}_j. \quad (1)$$



**Figure 3:** *Triceratops* model being probed for measurements at ACME.



**Figure 4:** “Bird Jar” with a color coding showing assigned artificial Rayleigh damping coefficients. The body of the jar (in grey) has no damping, the handle (in blue) a medium amount, and the spout (in red) a high amount.

Rewriting the inside of this summation with respect to force instead of traction reveals the relationship to stiffness as,

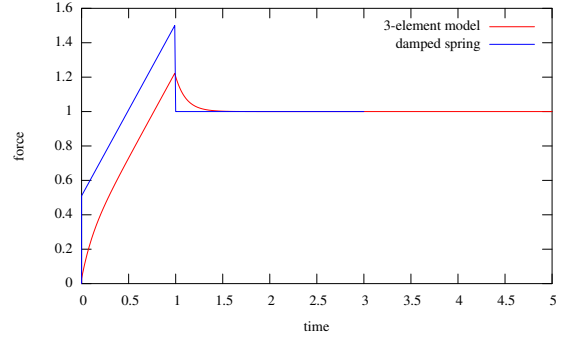
$$\mathbf{f}_{i,j} = \underbrace{\alpha_j \mathfrak{E}_{i,j}^{-1}}_{\mathbf{K}_{i,j}} \mathbf{u}_{i,j} \quad (2)$$

where  $\mathbf{K}_{i,j}$  is known as the stiffness tensor and  $\alpha_j$  is the area of the  $j$ -th node.

### 3. Viscoelastic Model

In a quasi-static elastic model (as given by a DGFM) deformation purely depends on the current boundary conditions; their history or their rate of change do not influence the simulation result. This is not sufficient to model the viscoelastic effects of relaxation and creep. In this section, a flexible approach to viscoelasticity that combines the advantages of a DGFM (precomputability, accurate global behavior and simple global parameters for control) with those of particle systems (ability to model secondary motion) is presented.

The underlying idea for our viscoelastic model is to replace the spring-like relations in the DGFM with compositions of springs and dashpots. This provides the ability to simulate viscoelastic behavior while retaining the DGFM for primary deformation calculation. The actual arrangement of springs and dampers is arbitrary and various reasonable choices to model viscoelasticity exist. The general goals for useful models are:



**Figure 5:** Sample force loading curve for the Kelvin-Voigt model with  $k = 1.0$  and  $b = 0.5$  (in blue) and the 3-element damper spring model with  $k_1 = 1.5$ ,  $k_2 = 3.0$ , and  $b = 0.5$  (in red). This response is driven by applying a constant velocity to both models from  $t = 0$  to  $t = 1$ .

1. The behavior resulting from the model should look and feel more realistic than a simple elastic model.
2. When static, the model should behave exactly as the DGFM. Mathematically, this can be written more precisely with the new relation  $\mathbf{g}$  as,

$$\forall i, \forall \mathbf{c}_k \in \mathbb{R}^3, \lim_{t \rightarrow \infty, \forall \bar{\mathbf{p}}_k, \bar{\mathbf{p}}_k \rightarrow \mathbf{c}_k} \sum_j \mathbf{g}(\mathfrak{E}_{i,j}, t, \bar{\mathbf{p}}_j) = \sum_j \mathfrak{E}_{i,j} \bar{\mathbf{p}}_j,$$

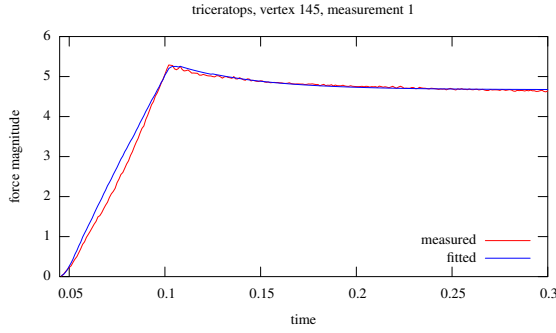
where  $t$  is time.

3. The model should be adjustable by a few intuitive key parameters. Manual tuning of all  $n^2$  relations (or a large subset thereof) should not be required. Nevertheless, it should remain possible to tune all relations in an automated manner, such as one that relies on measurements acquired physically.
4. Simulating the object with the model should be fast enough to satisfy interactive haptic and visual rates (approximately 1000Hz and 30Hz, respectively).

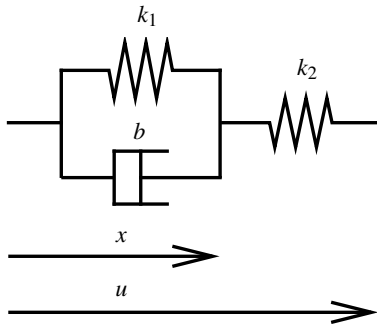
Perhaps the simplest viscoelastic model is based on the Kelvin-Voigt model, where a damper and spring are used in parallel [Fun93]. However, this model does not accurately fit our physically acquired measurements (see Figures 5 and 6). Instead, we have chosen to use a slightly more complicated 3-element model (see Figure 7 for a schematic) that accurately captures the observed behavior. With this model, the one dimensional relationship between force and displacement is given by the ordinary differential equation,

$$f = k_2(u - x) = k_1x + b\dot{x}, \quad (3)$$

where  $f$  is the force,  $u$  the total displacement,  $x$  the displacement of the damped spring,  $k_1$  and  $k_2$  the two spring constants and  $b$  the damping coefficient. Using an implicit first-



**Figure 6:** Force loading curve for vertex 145, measurement 1 of the triceratops model along with best fit curve from 3-element damper-spring model.



**Figure 7:** Schematic of the 3-element model: two springs and one dashpot.

order finite differences scheme gives,

$$k_1 x + b \dot{x} = k_1 x(t+h) + b \left( \frac{x(t+h) - x(t)}{h} \right),$$

where  $t$  is the current time and  $h$  is the step size, Equation 3 can be solved for force in terms of displacement or displacement in terms of force.

This one dimensional case is extended into three dimensions by replacing the force and displacement scalars with vectors and the scalar model parameters with 3-by-3 tensors. Displacement in terms of force for the 3-element model at the  $i$ -th vertex can be written in three dimensions as,

$$\mathbf{u}_i(t+h) = \sum_j \mathbf{u}_{i,j}(t+h) \quad (4)$$

$$\mathbf{u}_{i,j}(t+h) = \mathbf{K}_{2(i,j)}^{-1} \mathbf{f}_j(t+h) + \mathbf{x}_{i,j}(t+h), \quad (5)$$

$$\mathbf{x}_{i,j}(t+h) = \left( \mathbf{K}_{1(i,j)} + \frac{1}{h} \mathbf{B}_{i,j} \right)^{-1} \left( \mathbf{f}_j(t+h) + \frac{1}{h} \mathbf{B}_{i,j} \mathbf{x}_{i,j}(t) \right) \quad (6)$$

and force in terms of displacement as,

$$\mathbf{f}_i(t+h) = \sum_j \mathbf{f}_{i,j}(t+h)$$

$$\mathbf{f}_{i,j}(t+h) = \mathbf{K}_{2(i,j)} (\mathbf{u}_j(t+h) - \mathbf{x}_{i,j}(t+h)) \quad (7)$$

$$\mathbf{x}_{i,j}(t+h) = \left( \mathbf{K}_{1(i,j)} + \mathbf{K}_{2(i,j)} + \frac{1}{h} \mathbf{B}_{i,j} \right)^{-1} \left( \mathbf{K}_{2(i,j)} \mathbf{u}_j(t+h) + \frac{1}{h} \mathbf{B}_{i,j} \mathbf{x}_{i,j}(t) \right), \quad (8)$$

where the  $(i, j)$  subscripted quantities are the stiffness tensors ( $\mathbf{K}_{1(i,j)}$  and  $\mathbf{K}_{2(i,j)}$  corresponding to  $k_1$  and  $k_2$  in the 1D model, respectively), damping tensors ( $\mathbf{B}_{i,j}$  corresponding to  $b$ ), and vector-valued state information ( $\mathbf{x}_{i,j}(t)$ ) for the effect of node  $j$  on node  $i$ . While the above description is a general characterization of the 3-element model, there are a number of assumptions that can be introduced to make control or acquisition of the above viscoelastic parameters simpler and more manageable.

### 3.1. Using Stiffness Ratios

While  $k_1$  and  $k_2$  as scalars in the one dimensional case serve their simple, but specific purpose clearly, in three dimensions the corresponding stiffness tensors each have 9 degrees of freedom controlling not only magnitude but also direction. One simplification is to partially preserve one dimensionality in the three dimensional case by assuming that both  $\mathbf{K}_{1(i,j)}$  and  $\mathbf{K}_{2(i,j)}$  act in the same direction—the direction of stiffness in general,  $\mathbf{K}_{i,j}$ . With this assumption, the stiffness tensors can be rewritten as  $\mathbf{K}_{1(i,j)} = k_1 \mathbf{K}_{i,j}$  and  $\mathbf{K}_{2(i,j)} = k_2 \mathbf{K}_{i,j}$ , where  $k_1$  and  $k_2$  control the magnitude of the tensors. We would additionally like to ensure that the stiffness generated by these tensors is equal to the overall stiffness. Such a constraint can be written mathematically as  $k_1 k_2 = k_1 + k_2$  (derived from the formula for the behavior of two springs in series). To unify the above conditions, we introduce the stiffness ratio, defined as  $r = \frac{k_2}{k_1}$ . Combining these last three relationships, only one parameter ( $r$ ) is necessary:  $k_1 = \frac{1+r}{r}$  and  $k_2 = 1+r$ .

Most importantly, the use of stiffness ratios allows for a simple connection to the underlying DGFm. Because DGFm entries are simply inverted, area-compensated stiffness tensors (refer back to the underbraced part of Equation 2), we can write  $\mathbf{K}_{1(i,j)}$  and  $\mathbf{K}_{2(i,j)}$  as,

$$\mathbf{K}_{1(i,j)} = \frac{1+r_{i,j}}{r_{i,j}} \alpha_j \mathbf{\Xi}_{i,j}^{-1}, \quad (9)$$

$$\mathbf{K}_{2(i,j)} = (1+r_{i,j}) \alpha_j \mathbf{\Xi}_{i,j}^{-1}, \quad (10)$$

where  $\alpha_j$  is the area (with regard to traction) of the  $j$ -th node.

### 3.2. Using Rayleigh Damping

Rayleigh damping assumes that each damping tensor is equal to the corresponding stiffness tensor multiplied by a

non-negative scalar,

$$\mathbf{B}_{i,j} = b_{i,j} \mathbf{K}_{i,j} = b_{i,j} \alpha_j \mathbf{\Xi}_{i,j}^{-1}. \quad (11)$$

Substituting this along with Equations 9 and 10 into Equations 5 and 6 results in simplified equations for displacement in terms of force, node areas, a DGFm, and various viscoelastic parameters:

$$\begin{aligned} \mathbf{u}_{i,j}(t+h) &= \frac{\alpha_j}{1+r_{i,j}} \mathbf{\Xi}_{i,j} \mathbf{f}_j + \mathbf{x}_{i,j}(t+h) \\ \mathbf{x}_{i,j}(t+h) &= \frac{hr_{i,j}}{h+hr_{i,j}+b_{i,j}r_{i,j}} \\ &\quad \left( \alpha_j \mathbf{\Xi}_{i,j} \mathbf{f}_j + \frac{b_{i,j}}{h_{i,j}} \mathbf{x}_{i,j}(t) \right). \end{aligned} \quad (12)$$

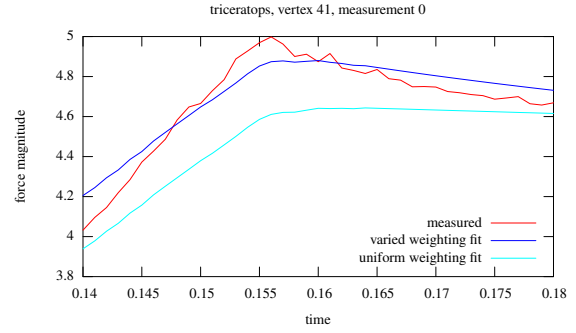
For brevity, we leave out the results of a similar substitution into Equations 7 and 8.

#### 4. Acquisition of Viscoelastic Behavior

In order to determine the parameters for the 3-element model described in the previous section, we rely on data sets originally used to acquire a simple DGFm. The basic DGFm acquisition procedure is to have the probe apply a series of surface displacements in various directions and observe the behavior for each trial visually (by taking stereo images before and after) and mechanically (by measuring force and displacement at the tip of the probe during contact). The probe locations correspond to vertices of the triangular mesh utilized in the DGFm. The data is then used to estimate a DGFm using various data fitting techniques [LPW02]. The force measurements from the probe are taken with a frequency of 1000Hz which matches the haptically relevant mechanical vibration. It is through these temporally precise measurements that we determine viscoelastic parameters for measured vertices. Essentially, we start with a series of measurements taken at one vertex location from several directions. We also have data available which is recorded at different probe velocities. One trial is a function of force against time taken for one vertex, in one direction, and at a specific velocity. Unfortunately, the stereo-imagery is not recorded frequently enough to allow for the observation of time-dependent behavior. However, this limitation comes from the specific hardware setup in ACME and methods for high frequency image-based acquisition of displacement exist [GOM].

##### 4.1. The Estimation Process

We use a multi-step approach to estimate our three-element models for the relevant entries of the DGFm. Beforehand, the static relationship (the DGFm) is estimated with the method of Lang et al. [LPW02]. The first stage in our process is to fit the magnitude force profile of each trial to our one-dimensional 3-element viscoelastic model of Equation 3. Next, based on these individual one-dimensional models, we construct appropriate stiffness and damping tensors.



**Figure 8:** One dimensional fitting of a noisy vertex 41, measurement 0 using a uniform weighting and a varied weighting least squares minimization function, zoomed in on curve peak. uniform weighting fitted parameters:  $k_1 = 8.53 \times 10^{-6}$ ,  $k_2 = 0.447$ ,  $b = 1.19$ ; varied weighting fitted parameters:  $k_1 = 1.81$ ,  $k_2 = 0.528$ ,  $b = 0.137$

The one-dimensional fitting is performed with Nelder-Mead Simplex discrete optimization [NM65]. Figure 8 shows the result of the one-dimensional fitting. Notice how if measurements are weighted equally over time, the damping is reduced in an attempt to better fit the front of the curve at the detriment of the peak and tail. This occurs because during constant velocity loading the recorded force profile has a slope  $\leq 1$  while the model can only produce slopes  $\geq 1$ . This mismatch is due to the fact, that the data fitting assumes constant contact area while at the beginning of the measurement the contact area of the probe is increasing as the deformation increases. Using a least squares minimization function, which weights the curve peak and tail more highly than the front, the optimization algorithm is led to disregard the mismatched time steps. We have found that weighting the tail 100 times more and the 3 points around the peak 1000 times more than the front produces good results for all our measurements.

Next, stiffness and damping parameters are determined for each vertex, where one dimensional fittings were performed. For the three-element model, we have found it necessary to fit measurements taken at different speeds separately. For damping parameters, there are two possibilities: fitting a full 3-by-3 tensor and fitting a Rayleigh damping constant.

In the tensor case, we seek to find the values for  $\mathbf{B}$  that minimize  $r^2(\mathbf{B}) = \sum_{i,j} \mathbf{R}_{i,j}^2$ , where

$$\mathbf{R} = \mathbf{B} \underbrace{\begin{pmatrix} \hat{u}_{x,1} & \cdots & \hat{u}_{x,n} \\ \hat{u}_{y,1} & \cdots & \hat{u}_{y,n} \\ \hat{u}_{z,1} & \cdots & \hat{u}_{z,n} \end{pmatrix}}_{\mathbf{P}} - \underbrace{\begin{pmatrix} b_1 \hat{f}_{x,1} & \cdots & b_n \hat{f}_{x,n} \\ b_1 \hat{f}_{y,1} & \cdots & b_n \hat{f}_{y,n} \\ b_1 \hat{f}_{z,1} & \cdots & b_n \hat{f}_{z,n} \end{pmatrix}}_{\mathbf{Q}}$$

and where  $b_1 \cdots b_n$  are the  $n$  measured 1D damping coeffi-

cients,  $\hat{\mathbf{u}}_1 \cdots \hat{\mathbf{u}}_n$  are the  $n$  normalized 3D displacement vectors used during measurement, and  $\hat{\mathbf{f}}_1 \cdots \hat{\mathbf{f}}_n$  are the  $n$  normalized 3D force vectors measured. Conceptually, we are searching for the damping tensor that most closely mimics all of the one dimensional damping scalars for each corresponding measurement. We further assume that the directions of the measured displacement  $\hat{u}_i$  and force  $\hat{f}_i$  are the same as the directions of displacement and force at the damping element of the model. The optimal value for  $\mathbf{B}$  can be calculated by solving the three systems,

$$\forall i \in \{1, 2, 3\}, \mathbf{P}\mathbf{P}^T \mathbf{B}_i = \left( \mathbf{Q}\mathbf{P}^T \right)_i$$

where the subscripts represent the specified row of the matrix used as a column vector. If the matrix is singular, this is a sign that there are insufficient measurements to obtain a reasonable solution. For such vertices, the parameter determination is abandoned. In our implementation, we also use damped singular value decomposition (DSVD) [Han94] on  $\mathbf{P}\mathbf{P}^T$  before solving the system, to smooth the solution in cases where the matrix is nearly singular.

In the case of Rayleigh damping, a scalar damping coefficient is determined by taking the determined tensor and then fitting it to the known stiffness tensor ( $\mathbf{K}$  derived from  $\mathbf{\Xi}_{i,i}$ , where  $\mathbf{\Xi}$  is the previously acquired DGFM and  $i$  the vertex number being processed). This is done by minimizing  $r^2(b) = \sum_{i,j} (b\mathbf{K}_{i,j} - \mathbf{B}_{i,j})^2$ . The solution is given by

$$b = \frac{\sum_{i,j} \mathbf{K}_{i,j} \mathbf{B}_{i,j}}{\sum_{i,j} \mathbf{K}_{i,j}}. \quad (13)$$

Additionally, a stiffness ratio needs to be determined. This is done by minimizing  $r^2(k_l) = \sum_{i,j} \mathbf{R}_{i,j}^2$  separately for both  $l = 1$  and  $l = 2$ , where

$$\mathbf{R} = k_l \mathbf{K} \underbrace{\begin{pmatrix} \hat{u}_{x,1} & \cdots & \hat{u}_{x,n} \\ \hat{u}_{y,1} & \cdots & \hat{u}_{y,n} \\ \hat{u}_{z,1} & \cdots & \hat{u}_{z,n} \end{pmatrix}}_{\mathbf{P}'} - \underbrace{\begin{pmatrix} k_{l,1} \hat{f}_{x,1} & \cdots & k_{l,n} \hat{f}_{x,n} \\ k_{l,1} \hat{f}_{y,1} & \cdots & k_{l,n} \hat{f}_{y,n} \\ k_{l,1} \hat{f}_{z,1} & \cdots & k_{l,n} \hat{f}_{z,n} \end{pmatrix}}_{\mathbf{Q}'}$$

and where  $k_{l,1}, \dots, k_{l,n}$  are the  $n$  1D stiffness coefficients corresponding to  $k_l$ . Because  $r^2(k_l)$  is in the same form as the Rayleigh damping coefficient problem ( $r(k_l)^2 = \sum_{i,j} (k_l \mathbf{P}'_{i,j} - \mathbf{Q}'_{i,j})^2$ ), Equation 13 can be used to find the minimal solutions here as well. After an optimal  $k_1$  and  $k_2$  have been determined, the stiffness ratio is defined to be  $r = \frac{k_2}{k_1}$ . If  $k_1$  is zero, we set the damping to zero (either tensor or Rayleigh damping coefficient) and set the ratio to an arbitrary non-zero value. This has the effect that no damping will occur at this vertex, but elastic behavior will continue as normal.

## 4.2. Handling Non-Diagonal Entries

The above procedures only produce one set of viscoelastic parameters for each vertex due to measurement limitations

discussed at the beginning of this section. The values occupy the diagonal entries of all of the parameter structures, because they measure self effects. It is necessary to somehow spread the known parameters to the other non-diagonal entries of the parameter structures. The solution we chose to employ is to simply use the same parameter values across an entire column (for the values measured at vertex  $k$ ,  $b_{\cdot,k} = b_k$  or  $\mathbf{B}_{\cdot,k} = \mathbf{B}_k$ , and  $r_{\cdot,k} = r_k$ ). This has the effect of causing the activated vertices to drive the damping behavior at all the non-activated vertices.

## 5. Rendering

During simulation Equation 4 (or a simplification such that derived in Section 3.2) is evaluated once per time step with the current applied tractions and time step size to obtain the current displacements for all vertices. In practice, a number of optimizations and precomputation steps need to be performed to get good real time performance while simulating the model.

The two major optimizations used in a quasi-static elastic simulation that uses a DGFM are taking advantage of the sparse nature of the traction vector in the matrix-vector multiplication (see Equation 1) and not computing traction values for displacement constrained vertices. Taking these into account, the complexity of evaluation is  $O(V_d V_t)$  as opposed to  $O(V_d V_h)$  in the case of a full multiplication where  $V_d$  is the number of vertices in the display mesh,  $V_t$  the number that have a traction currently applied to them (in our implementation this is generally 3), and  $V_h$  the number of moveable vertices in the haptic mesh. In the viscoelastic simulation, this same complexity is not achievable, because although a vertex may no longer have a traction applied, it may still contribute to the deformation (for example, it is currently retracting). However, resorting to the full  $O(V_d V_h)$  is not necessary. To handle this issue, we maintain a list of vertices, which are active (they either currently have a traction applied or are not at rest). The list starts out empty and vertices are added at the beginning of each time step, which have a traction currently applied to them. Any vertices that were active in the last time step that do not have a traction applied in the current time step are examined. If their displacement effect ( $\mathbf{x}_{i,j}$ ) has dropped below an arbitrary visual threshold (we use  $10^{-3}$ ), they become inactive and are not used in the computation until they are activated again by a traction. This leads to a complexity of  $O(V_d V_a)$  where  $V_a$  is the number of active vertices. Generally,  $V_a$  will not be particularly large as long as damping is not terribly high and will tend to a small constant (in our implementation, again 3) when only one triangle is activated for an extended period of time.

In the general three-element case (without Rayleigh damping), the inverted term in Equation 6 needs to be pre-computed to allow real-time interaction with our models. Unfortunately, this can only be done when the time step size is held constant. As can be seen in Equation 12, this is not the

case with Rayleigh damping. Furthermore, even if the time step size can be held constant, the general case requires two matrix-vector multiplications and a vector addition, while Rayleigh damping requires only two scalar-vector multiplications and one vector addition, making it still a bit faster. Furthermore, in both the general case and in the case with Rayleigh damping, if the force applied drops to zero for a given vertex, an entire column can be computed using a simplified formula. In the general case, without a force term, the formula for  $\mathbf{u}_i$  collapses to  $\sum_j \mathbf{x}_{i,j}$ , while  $\mathbf{x}_{i,j}$  itself collapses (assuming possible precomputations are done) to just one matrix-vector multiplication. Similarly, in the case of Rayleigh damping,  $\mathbf{u}_i$  is also just  $\sum_j \mathbf{x}_{i,j}$  and  $\mathbf{x}_{i,j}$  becomes a singular scalar-vector multiplication.

We built our simulator to function with and without a haptic device. Due to software threading issues, performance concerns, and collision detection complexity, we chose to simulate a quasi-static elastic model on the device, while showing a viscoelastic model on the screen.

To obtain visually more attractive results, we use subdivision in two places. Non-square DGFMs are used to accommodate different haptic and display meshes. The display mesh is simply one displaced subdivision level finer than the haptic mesh. Additionally, we perform two additional levels of Loop subdivision [Loo87] on the display mesh at run-time for each time step. Because our contact model applies tractions to the three vertices composing the triangle in the coarse haptic mesh currently in contact with the haptic proxy, an artifact of three contact depressions can sometimes be seen. To eliminate this, we perform an additional precomputation step, where the vertices on the fine mesh which are enclosed by this triangle on the coarse mesh are found. At run-time, the enclosed vertices of the currently depressed triangle receive an average of the deformation present on neighboring coarse vertices instead of their normal deformation computed directly using the viscoelastic model.

## 6. Results

The viscoelastic parameter estimation code was written in Python [Pyt], while the simulator was written in a mixture of Python, C, and C++. All of the precomputation code was written in Python, but utilizes Numeric Python [ADH\*01], a series of Python routines written in C for working with large arrays of homogenous data efficiently. Parts of the display and simulation code, which needed to be particularly fast were written in C as Python modules. The haptics code, which interfaces with SensAble's™ GHOST SDK was written in C++ as a Python module. Additionally, for linear algebra routines we utilized LAPACK [ABB\*99] with an ATLAS BLAS [WPD01]. Both the simulation and parameter estimation programs were run on a 2.8GHz Pentium 4 processor with 1.5GB of RAM and an ATI FireGL™ X1 GPU. The haptics device used was Sensable's™ PHAN-ToM® Desktop.

	triceratops	tiger
total vertices	193	149
moveable vertices	149	95
low-speed measurements		
total trials	269	-
vertices successfully fitted	76	-
high-speed measurements		
total trials	989	855
vertices successfully fitted	79	67
total fitting time	33 min	7 min

**Table 2:** Fitting information for the two measured objects: the triceratops and the tiger.

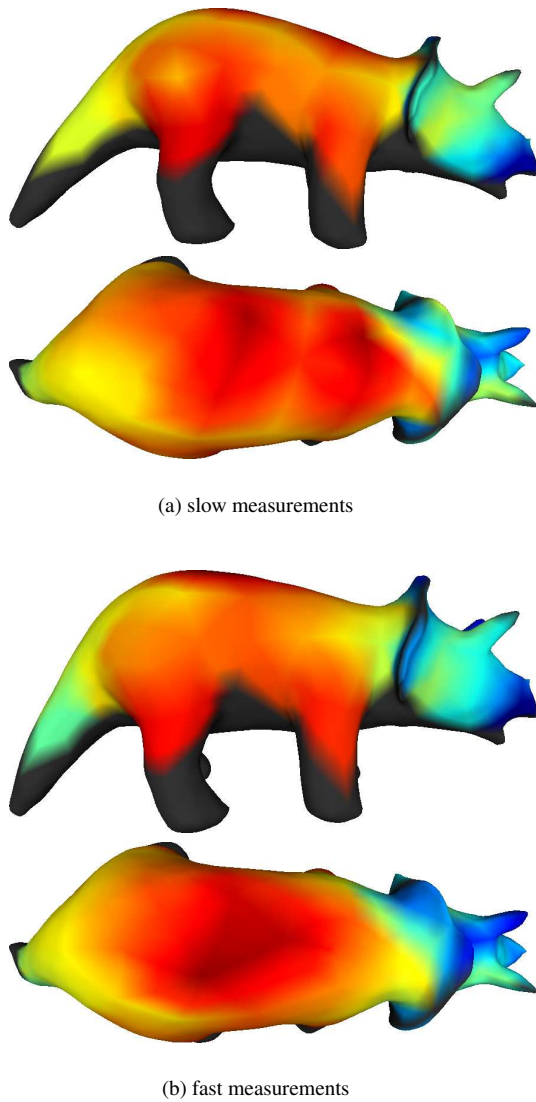
	triceratops		tiger	
display subdiv. level	0	1	0	1
total vertices	193	766	149	590
Frame Rates in Hz				
-haptics, -subdiv.	580	320	600	390
-haptics, +subdiv.	160	60	250	80
+haptics, -subdiv.	>100	>100	>100	>100
+haptics, +subdiv.	50	25	>100	35

**Table 3:** Approximate frame rates for viscoelastic simulation of the triceratops and tiger models. When using the haptic device, the supporting software limits the display frame rate to 100Hz.

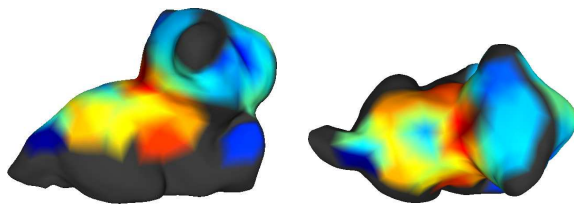
We had measurements for two objects: a rubber triceratops and a tiger plush toy. Table 2 gives important numbers for their acquired models, measurements, and fitting performance. For both viscoelastic and quasi-static elastic simulations, we achieve similar sufficient frame rates, which can be seen in Table 3. Figures 9 and 10 show the magnitude of the acquired damping for the triceratops and tiger, respectively. In the case of the triceratops, the figures show that the body exhibits more viscoelastic behavior than the head or the tail, which seems to be consistent with our own personal observations of the physical object.

## 7. Conclusions

We have successfully extended the procedures for globally estimating a quasi-static elastic model to estimate a more complex viscoelastic model. Furthermore, we have accomplished this without any new types of measurements and without significant simulation speed penalty. Future work could be done with regard to obtaining measurements at



**Figure 9:** Triceratops with false color coding (red high, blue low) from damping magnitude for both measurement velocities.



**Figure 10:** Tiger with false color coding from damping magnitude.

non-diagonal entries through use of high speed video. Also, methods for rigorously verifying the results by comparing the behavior of the actual object and its model could be developed. Finally, the use of models besides the 3-element model could result in more realism or allow measurements taken at different velocities to be used together for more robust estimation.

### Acknowledgements

In particular, we would like to thank Dinesh K. Pai for allowing us access to the UBC ACME facility and John Lloyd for help in acquiring the measurements for the triceratops.

### References

- [ABB\*99] ANDERSON E., BAI Z., BISCHOF C., BLACKFORD S., DEMMEL J., DONGARRA J., DU CROZ J., GREENBAUM A., HAMMARLING S., MCKENNEY A., SORENSEN D.: *LAPACK Users' Guide*, third ed. Society for Industrial and Applied Mathematics, Philadelphia, 1999. 8
- [ADH\*01] ASCHER D., DUBOIS P. F., HINSEN K., HUGUNIN J., OLIPHANT T.: *Numerical Python Users' Guide*. Tech. Rep. UCRL-MA-128569, Lawrence Livermore National Laboratory, 2001. 8
- [BNC96] BRO-NIELSEN M., COTIN S.: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum* 15, 3 (1996), 57–66. 2
- [BUB\*01] BROUWER I., USTIN J., BENTLEY L., SHERMAN A., DHRUV N., TENDICK F.: Measuring in vivo animal soft tissue properties for haptic modeling in surgical simulation. *Medicine Meets Virtual Reality* (2001), 69–74. 2, 3
- [CEO\*93] COVER S., EZQUERRA N., O'BRIAN J., ROWE R., GADAXZ T., PALM E.: Interactively deformable models for surgery simulation. *IEEE Computer Graphics & Applications* 13, 6 (1993), 68–75. 1
- [dCL99] D'AULIGNAC D., CAVUSOGLU M., LAUGIER C.: Modeling the human thigh for a realistic echographic simulator with force feedback. In *Proceedings of the International Conference on Medical Image Computer-Assisted Intervention* (Cambridge, United Kingdom, Sept. 1999), pp. 1191–1198. 2, 3
- [DDCB01] DEBUNNE G., DESBRUN M., CANI M.-P., BARR A. H.: Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of SIGGRAPH* (2001), ACM Press, pp. 31–36. 1, 2, 3
- [Fun93] FUNG Y.: *Biomechanics: Mechanical Properties of Living Tissues*, second ed. Springer-Verlag, New York, 1993. 2, 4
- [GKS02] GRINSPUN E., KRYSL P., SCHRÖDER P.: Charms: a simple framework for adaptive simulation. In *Proceedings of SIGGRAPH* (2002), ACM Press, pp. 281–290. 2

- [GM97] GIBSON S., MIRTICH B.: *A survey of deformable modeling in computer graphics*. Tech. Rep. TR-97-19, Mitsubishi Electric Research Laboratory, Cambridge, MA, USA, Nov. 1997. 2
- [GOM] GOMMBH: High-speed tire deformation analysis using the grating method. available at <http://www.gom.com>. Braunschweig, Germany. 6
- [Guy65] GUYAN R. J.: Reduction of stiffness and mass matrices. *AIAA Journal* 3, 2 (1965), 380. 2
- [Han94] HANSEN P.: Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms* 6, 1/2 (1994), 1–35. 7
- [HGS03] HAUTH M., GROSS J., STRASSER W.: Interactive physically based solid dynamics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), Eurographics Association, pp. 17–27. 1, 2, 3
- [HSO03] HAUSER K. K., SHEN C., O'BRIEN J. F.: Interactive deformation using modal analysis with constraints. In *Proceedings of Graphics Interface* (June 2003), A K Peters, pp. 247–256. 2
- [JP99] JAMES D. L., PAI D. K.: ArtDefo - accurate real time deformable objects. In *Proceedings of ACM SIGGRAPH* (Los Angeles, Aug. 1999), ACM Press, pp. 65–72. 1, 2
- [JP01] JAMES D. L., PAI D. K.: A unified treatment of elastostatic and rigid contact simulation for real time haptics. *Haptics-e* 2, 1 (Sept. 2001). 2, 3
- [JP02] JAMES D. L., PAI D. K.: DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. In *Proceedings of ACM SIGGRAPH* (San Antonio, July 2002), ACM Press, pp. 582–585. 2
- [JP03] JAMES D. L., PAI D. K.: Multiresolution green's function methods for interactive simulation of large-scale elastostatic objects. *ACM Transactions on Graphics (TOG)* 22, 1 (2003), 47–82. 2
- [Loo87] LOOP C. T.: Smooth subdivision surfaces based on triangles, 1987. Master's thesis, University of Utah, Department of Mathematics. 8
- [LPS03] LANG J., PAI D. K., SEIDEL H.-P.: Scanning large-scale articulated deformations. In *Graphics Interface* (June 2003), pp. 265–272. 3
- [LPW02] LANG J., PAI D. K., WOODHAM R. J.: Acquisition of elastic models for interactive simulation. *The International Journal of Robotics Research* 21, 8 (Aug. 2002), 713–733. 2, 3, 6
- [MK99] MAASS H., KÜHNAPFEL U.: Noninvasive measurement of elastic properties of living tissue. In *13th Int. Congress on Comp. Assisted Radiology (CARS)* (1999), pp. 865–870. 2, 3
- [NM65] NELDER J., MEAD R.: A simplex method for function minimization. *Computer Journal* 7 (1965), 308–313. 6
- [OH99] O'BRIEN J. F., HODGINS J. K.: Graphical modeling and animation of brittle fracture. In *Proceedings of ACM SIGGRAPH* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 137–146. 2
- [PDA01] PICINBONO G., DELINGETTE H., AYACHE N.: Non-linear and anisotropic elastic soft tissue models for medical simulation. In *International Conference on Robotics and Automation* (Seoul, South Korea, May 2001), pp. 1371–1376. 1, 3
- [PvdDJ\*01] PAI D. K., VAN DEN DOEL K., JAMES D. L., LANG J., LLOYD J. E., RICHMOND J. L., YAU S. H.: Scanning physical interaction behavior of 3d objects. In *Proceedings of SIGGRAPH* (2001), ACM Press, pp. 87–96. 2, 3
- [PW89] PENTLAND A., WILLIAMS J.: Good vibrations: model dynamics for graphics and animation. In *Proceedings of SIGGRAPH* (1989), pp. 215–222. 2
- [Pyt] Python 2.3.2. <http://www.python.org>. 8
- [SBD\*00] SZEKÉLY G., BRECHBUHLER C., DUAL J., ENZLER R., J. HUG R. H., IRONMONGER N., KAUER M., MEIER V., NIEDERER P., RHOMBERG A., SCHMID P., SCHWEITZER G., THALER M., VUSKOVIC V., TROSTER G., HALLER U., BAJKA M.: Virtual reality-based simulation of endoscopic surgery. *Prescence* 9, 3 (2000), 313–333. 2
- [TBHF03] TERAN J., BLEMKER S., HING V. N. T., FEDKIW R.: Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), Eurographics Association, pp. 68–74. 3
- [TF88] TERZOPOULOS D., FLEISCHER K.: Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In *Computer Graphics, Annual Conference Series* (Atlanta, USA, Aug 1988), ACM SIGGRAPH, pp. 269–278. 2
- [TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. In *Proceedings of ACM SIGGRAPH* (Anaheim, July 1987), ACM Press. 1, 2
- [VKSR00] VUSKOVIC V., KAUER M., SZÉKELY G., REIDY M.: Realistic force feedback for virtual reality based diagnostic surgery simulators. In *Proceedings of the IEEE International Conference on Robotics and Automation* (San Francisco, 2000), pp. 1592–1598. 2, 3
- [WDGT01] WU X., DOWNES M., GOTEKIN T., TENDICK F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Eurographics* (Manchester, UK, 2001), Chalmers A., Rhyne T.-M., (Eds.), Computer Graphics Forum, 20(3), pp. C349–C358. 2, 3
- [WPD01] WHALEY R. C., PETITET A., DONGARRA J. J.: Automated empirical optimizations of software and the ATLAS project. *Parallel Computing* 27, 1–2 (2001), 3–35. 8