Real-Time Finite-Element Simulation of Linear Viscoelastic Tissue Behavior Based on Experimental Data

Mert Sedef, Evren Samur, and Cagatay Basdogan
Koc University

The lack of experimental data on the viscoelastic material properties of live organ tissues has been a significant obstacle in the development of realistic models. A real-time and realistic finite-element simulation of viscoelastic tissue behavior using experimental data collected by a robotic indenter offers one solution.

VR-based surgical simulators that provide realistic visual and haptic feedback to users is a promising technology for medical training.[1] The core component of a computer-based surgical simulation and training system is a realistic organ-force model. An organ-force model must display physics-based behavior while handling various types of boundary conditions and constraints. Developing real-time and realistic organ-force models is challenging not only because of the nonlinearity, rate, and time dependence of an organ's material properties but also because of organ tissues' layered and nonhomogeneous structure.

Researchers have used both linear and nonlinear finite-element methods (FEMs) to develop real-time organ-force models.[1,2] Achieving a computationally fast and stable simulation is possible using a linear static FEM model because the system's global stiffness matrix is constant and can be inverted before the real-time simulation. However, the linearity assumption isn't valid for soft tissues with complex nonlinear behavior. A linear FEM model can't accurately simulate large organ deformations. Whereas nonlinear FEM models display more realistic deformations than linear FEMs, they have a greater computational complexity because of the system's nonconstant stiffness matrix.

We can also group FEM-based organ-force models as static or dynamic based on whether we consider inertial and viscous effects. Static FEM models can't simulate time-dependent effects such as viscoelasticity. Because of modeling challenges and the high cost of real-time computations, only a few research groups have recently focused on the real-time simulation of viscoelastic tissue behavior (see the "Related Work in Finite-Element Simulation" sidebar).

We propose an end-to-end solution to real-time and realistic finite-element modeling and simulation of viscoelastic soft tissue behavior. We provide an efficient numerical scheme for solving a linear viscoelastic FEM model derived from the generalized Maxwell solid, and present methods for measuring and integrating experimental data on the viscoelastic material properties of soft tissues into the model for realistic display of visual deformations and interaction forces. Our precomputation scheme and multilayer computational architecture enable the model's real-time execution with visual and haptic feedback to the user. Researchers have applied the precomputation approach to real-time static FEM simulation in the past, but, to our knowledge, no one has extended it to the real-time linear viscoelastic FEM simulation. Our approach includes time- and rate-dependent effects, which requires considering a node's loading history in our displacement computations at each cycle of the simulation.

Linear viscoelasticity

For elastic materials, Hooke's Law applies—that is, the stress is proportional to the strain, and the elastic modulus is defined as the ratio of stress to strain. For a purely viscous material, stress is proportional to the rate of strain, and the ratio of stress to strain rate is known as viscosity. Materials that fall into neither classification are called viscoelastic materials.

In viscoelastic materials, an instantaneous elastic response is observed upon loading, followed by a slow and continuous change in the response at a decreasing rate. The rate of straining or stressing affects the time-dependent response of a viscoelastic material. For example, the longer it takes to reach the final value of stress at a constant rate of stressing, the larger the corresponding strain. For this reason, viscoelastic materials are said to keep a record of their response history and possess a memory.[3] This memory effect is evident in the constitutive relationship between the stress and strain tensors.

One way to derive a constitutive relationship for linear viscoelastic materials is to assume that we can apply a Boltzmann superposition of strain increments to viscoelas-

 Published by the IEEE Computer Society

tic materials. Consider an arbitrary strain input obtained through superposition of small strain increments:

$$\varepsilon(t) = \sum_{j=1}^{n} \Delta\varepsilon_j = \int_0^t d\left[\varepsilon(s)\right]$$

where s is any arbitrary past time between 0 and t when we apply a constant strain ε. The strain increments up to time t are related to corresponding stress increments by Hooke's law as follows:

$$\sigma(t) = \sum_{j=1}^{n} \Delta\sigma_j(t - s_j) = \sum_{j=1}^{n} E(t - s_j)\Delta\varepsilon_j$$

Each of these stress increments relax according to the time dependency of stress relaxation function $E(t)$. By taking the appropriate limit, we get the following constitutive law:

$$\sigma(t) = \int_0^t E(t-s)\frac{\partial\varepsilon(s)}{\partial s}ds \qquad (1)$$

Viscoelastic materials are typically modeled by the generalized Maxwell solid,[3,4] which is a combination of springs and dashpots. This type of model results in a prony series expression for the stress relaxation function in the form:

$$E(t) = E_\infty + \sum_{j=1}^{N} E_j e^{\left(\frac{-t}{\tau_j}\right)} \qquad (2)$$

where N is the number of Maxwell elements, E_j is the elastic coefficient (E_∞ is the long-term elastic modulus corresponding to the system's steady-state elastic response), and τ_j is the relaxation time related to the damping coefficients of dashpots as η/E_∞.

Numerical computation

Splitting the integral in Equation 1 into elastic and viscoelastic contributions[3] leads to

$$\begin{aligned}\sigma(t) &= \int_0^t E_\infty \frac{\partial\varepsilon(s)}{\partial s}ds + \int_0^t \sum_{j=1}^{N} E_j e^{\left(\frac{-t}{\tau_j}\right)}\frac{\partial\varepsilon(s)}{\partial s}ds \\ &= E_\infty\varepsilon(t) + \sum_{j=1}^{N}\int_0^t E_j e^{\left(\frac{t-s}{\tau_j}\right)}\frac{\partial\varepsilon(s)}{\partial s}ds \\ &= \sigma_0(t) + \sum_{j=1}^{N}\mathbf{h}_j(t)\end{aligned}$$

Defining the step size as $\Delta t = t_{n+1} - t_n$, where t_{n+1} and t_n are the current and previous time steps, and substituting $\varepsilon(t) = \sigma_0(t)/E_\infty$, gives us a recursive formula for internal stress variables. The transition from differential coefficient to discrete time steps yields

$$\mathbf{h}_j^{n+1} = e^{\left(-\frac{\Delta t}{\tau_j}\right)}\mathbf{h}_j^n + \gamma_j\int_{t_n}^{t_{n+1}} e^{\left(\frac{t_{n+1}-s}{\tau_j}\right)}ds\frac{\sigma_0^{n+1} - \sigma_0^n}{\Delta t}$$

Related Work in Finite-Element Simulation

Debunne et al.[1] developed a robust, adaptive method for simulating dynamic deformations of a viscoelastic object in real-time using an explicit finite-element method. Instead of merging finite-element equations in a large matrix system, explicit FEM solves each element independently through a local approximation, dramatically reducing the computational time.

Hauth et al.[2] developed a viscoelastic finite-element formulation for the visual simulation of viscoelastic deformable objects. They use a Maxwell solid with one memory parameter in their viscoelastic material model. To obtain a formulation for the shear relaxation function, they first assume a constant mechanical quality factor for the material and then find the parameters of the Prony series corresponding to the Maxwell model by matching their compliance functions while minimizing the relative error. They use mass lumping and nested tetrahedral meshes to reduce the number of real-time computations.

Schoner et al.[3] introduced a method for simulating viscoelastic solids in real-time based on a parameter estimation method derived from physical measurements of real objects. To model the viscoelastic effects, they combine a discrete Green's function matrix with particle systems by replacing the spring-like relations in the DGFM with compositions of springs and dashpots. This let them simulate viscoelastic behavior while retaining the DGFM for the primary deformation calculations. Although they've achieved real-time update rates for visual display of deformations, they used a quasi-static elastic model for calculating and reflecting interaction forces through a haptic device in real-time.

Schwartz et al.[4] developed a tensor-mass method for the finite-element simulation of nonlinear viscoelastic mechanical deformations in biological soft tissues. They introduce the material nonlinearity by locally modifying the stiffness tensors while keeping the strain tensor linear and adjusting the Lame constants. They use a Kelvin-Voigt viscoelastic element in their model and introduce a viscous force proportional to the deformation speed. They adjust and validate the simulation model's parameters using an experimental setup designed to characterize the material properties of biological tissues through in vitro experiments.

References

1. G. Debunne et al., "Dynamic Real-Time Deformations Using Space and Time Adaptive Sampling," *Proc. Siggraph*, ACM Press, 2001, pp. 31-36.
2. M. Hauth, J. Groß, and W. Straßer, "Interactive Physically Based Solid Dynamics," *Proc. Eurographics/Siggraph Symp. Computer Animation*, Eurographics Assoc., 2003, pp. 17-27.
3. J.L. Schoner, J. Lang, and H.-P. Seidel, "Measurement-Based Interactive Simulation of Viscoelastic Solids," *Computer Graphics Forum,* vol. 23, no. 3, 2004, pp. 547-556.
4. J.-M. Schwartz et al., "Modeling Liver Tissue Properties Using a Non-linear Visco-Elastic Model for Surgery Simulation," *Medical Image Analysis*, vol. 9, no. 2, 2005, pp. 103-112.

where $\gamma_j = E_j/E_\infty$ is the jth normalized elastic modulus. If we integrate the above expression analytically, we get a recursive formula in 3D tensor representation given as

$$\mathbf{h}_j^{n+1} = \mathrm{e}^{\left(-\frac{\Delta t}{\tau j}\right)}\mathbf{h}_j^n + \gamma_j \frac{1-\mathrm{e}^{\left(-\frac{\Delta t}{\tau_j}\right)}}{\frac{\Delta t}{\tau_j}}\left[\sigma_0^{n+1} - \sigma_0^n\right] \tag{3}$$

Using the internal stress variables defined by Equation 3 and the elastic contribution, we define the total stress of a linear elastic Maxwell material as

$$\sigma^{n+1} = \sigma_0^{n+1} + \sum_{j=1}^{N} \mathbf{h}_j^{n+1} \tag{4}$$

Linear static FEM formulation

We can construct a finite-element representation of an organ from 3D tetrahedral elements, each having four nodes.[5] With known displacements within the element, $\mathbf{u}$, we can determine the strains, ε, at any point using the relation

$$\varepsilon = \boldsymbol{B}\mathbf{u} \tag{5}$$

where $\boldsymbol{B}$ is a constant matrix defined by the shape functions. Assuming a linear elastic behavior, stress-strain relation is given by Hooke's law as

$$\sigma = \boldsymbol{C}_{\text{elastic}}\varepsilon \tag{6}$$

where $\boldsymbol{C}_{elastic}$ is a symmetric material stiffness matrix. Using the definition of internal force, we obtain

$$\begin{aligned}\mathbf{f}_{\text{int}} &= \int \boldsymbol{B}^{\mathrm{T}}\sigma dV \\ &= (\int \boldsymbol{B}^{\mathrm{T}}\boldsymbol{C}_{\text{elastic}}\boldsymbol{B}dV)\mathbf{u} \\ &= \boldsymbol{K}\mathbf{u}\end{aligned} \tag{7}$$

where $\boldsymbol{K} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{C}_{\text{elastic}}\boldsymbol{B}V$ is the element stiffness matrix and V is the element's volume. Because internal and external forces balance each other within the element, $\mathbf{f}_{\text{int}} = \mathbf{f}_{\text{ext}}$, we end up with the well-known linear finite-element equation given as

$$\boldsymbol{K}\mathbf{u} = \mathbf{f}_{\text{ext}} \tag{8}$$

If we calculate the stiffness matrix's inverse in advance, we can easily compute the static nodal displacements in real-time for the applied external forces ($\mathbf{u} = \boldsymbol{K}^{-1}\mathbf{f}_{\text{ext}}$). Bro-Nielsen and Cotin detail the linear FEM formulation using tetrahedral elements for static tissue simulation.[5]

Linear viscoelastic FEM formulation

To derive a similar expression for linear viscoelastic FEM formulation, we use the relations given in Equations 5 and 6 and the elastic contribution term

$$\sigma_0^{n+1} = \boldsymbol{C}_{\text{elastic}}\varepsilon^{n+1} \tag{9}$$

We can then write Equation 4 as

$$\begin{aligned}\sigma^{n+1} &= \boldsymbol{C}_{\text{elastic}}\boldsymbol{B}\mathbf{u}^{n+1} + \\ &\sum_{j=1}^{N}\left[\mathrm{e}^{\left(-\frac{\Delta t}{\tau_j}\right)}\mathbf{h}_j^n + \gamma_j A_j\left(\boldsymbol{C}_{\text{elastic}}\boldsymbol{B}\mathbf{u}^{n+1} - \boldsymbol{C}_{\text{elastic}}\boldsymbol{B}\mathbf{u}^n\right)\right] \\ &= \boldsymbol{C}_{\text{elastic}}\boldsymbol{B}\left(1 + \sum_{j=1}^{N}\gamma_j A_j\right)\mathbf{u}^{n+1} + \\ &\sum_{j=1}^{N}\mathrm{e}^{\left(-\frac{\Delta t}{\tau_j}\right)}\mathbf{h}_j^n - \boldsymbol{C}_{\text{elastic}}\boldsymbol{B}\left(\sum_{j=1}^{N}\gamma_j A_j\right)\mathbf{u}^n\end{aligned} \tag{10}$$

where

$$A_j = \frac{1-\mathrm{e}^{\left(-\frac{\Delta t}{\tau_j}\right)}}{\frac{\Delta t}{\tau_j}} \tag{11}$$

Using the definition in Equation 7, we derive an internal force expression from Equation 10:

$$\mathbf{f}_{\text{int}}^{n+1} = \boldsymbol{K}_{\mathrm{T}}\mathbf{u}^{n+1} + \mathbf{h}_{\text{hist}}^{n+1} - \boldsymbol{K}_{\text{hist}}\mathbf{u}^n$$

where $\mathbf{u}^{n+1}$ and $\mathbf{u}^n$ are the vectors of nodal displacements at current and previous time steps, $\boldsymbol{K}_{\mathrm{T}}$ is the constant tangent stiffness matrix defined as

$$\boldsymbol{K}_{\mathrm{T}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{C}_{\text{elastic}}\boldsymbol{B}V\left(1 + \sum_{j=1}^{N}\gamma_j A_j\right)$$

using Equations 8 and 11. $\boldsymbol{K}_{\text{hist}}$ is the history stiffness matrix defined as

$$\boldsymbol{K}_{\text{hist}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{C}_{\text{elastic}}\boldsymbol{B}V\sum_{j=1}^{N}\gamma_j A_j$$

and $\mathbf{h}_{\text{hist}}^{n+1}$ is the history vector at the current time step defined as

$$\mathbf{h}_{\text{hist}}^{n+1} = V\boldsymbol{B}^{\mathrm{T}}\sum_{j=1}^{N}\mathrm{e}^{\left(-\frac{\Delta t}{\tau_j}\right)}\mathbf{h}_j^n$$

We can derive $\mathbf{h}_j^{n+1}$ for each element from Equations 3 and 9 as

$$\mathbf{h}_j^{n+1} = \mathrm{e}^{\left(-\frac{\Delta t}{\tau_j}\right)}\mathbf{h}_j^n + \gamma_j A_j \boldsymbol{C}_{\text{elastic}}\boldsymbol{B}\left(\mathbf{u}^{n+1} - \mathbf{u}^n\right)$$

Now, if we consider the force balance at each time step,

$$\mathbf{f}_{\text{int}}^{n+1} = \mathbf{f}_{\text{ext}}^{n+1}$$

we get a general expression for linear viscoelasticity

$$\boldsymbol{K}_{\mathrm{T}}\mathbf{u}^{n+1} = \mathbf{f}_{\text{ext}}^{n+1} - \mathbf{f}_{\text{hist}}^{n+1} \tag{12}$$

where the object's internal load history vector at the current time step is

$$\mathbf{f}_{\text{hist}}^{n+1} = \mathbf{h}_{\text{hist}}^{n+1} - \boldsymbol{K}_{\text{hist}}\mathbf{u}^{n}$$

Solving Equation 12 for $\mathbf{u}^{n+1}$, we get an expression for the nodal displacements of a viscoelastic object under the influence of internal load history and the external load at the current time step as

$$\mathbf{u}^{n+1} = \boldsymbol{K}_{\text{T}}^{-1}\left[\mathbf{f}_{\text{ext}}^{n+1} - \mathbf{f}_{\text{hist}}^{n+1}\right] \tag{13}$$

This expression is similar to the static finite-element equation except for the time-dependent history term's effect.

Figure 1 shows the numerical scheme for solving linear viscoelastic finite-element equations. Given an external load, the pseudocode in Figure 1 solves for the nodal displacements.

Measuring and characterizing the material properties of soft tissues

One of the main obstacles in developing realistic organ-force models is the lack of data on the material properties of live organ tissues. Measuring and characterizing in vivo organ properties is a highly challenging task, but is a requirement for realistic organ-force modeling. Organ-force models with incorrect material properties will adversely affect training in VR-based surgical simulator systems.

The research on tissue mechanics is extensive, but most of the earlier experiments took place in a laboratory environment (in vitro studies) under well-defined boundary and loading conditions. Typically, tissue samples taken from an organ of interest are transferred to a laboratory for measurement in a chemical solution. Because researchers carefully decide the sample geometry and experimental conditions in advance, they can easily obtain stress and strain values from the measurement data. However, mechanical properties of soft tissues change with time and the results obtained through in vitro measurements don't represent actual tissue properties.

We developed a robotic indenter for minimally invasive measurement of live tissue properties in a living body (Figure 2a on the next page).[6] The system includes a robotic arm (Sensable Technologies' Phantom haptic device, model 1.0), a force sensor (ATI Industrial Automation's Nano 17), and a long probe that has a round tip with a 2-millimeter radius. We attach the force sensor to the proximal end of the probe, which we insert through a surgical trocar (that is, a port for inserting surgical instruments to access the internal organs during a minimally invasive surgery). We developed a proportional, integral, derivative (PID) controller to command the robotic arm such that the probe tip follows a given path in 3D space with a desired velocity during the experiments. We also developed a GUI to record current time, displacement, and force data in a text file after each experiment.

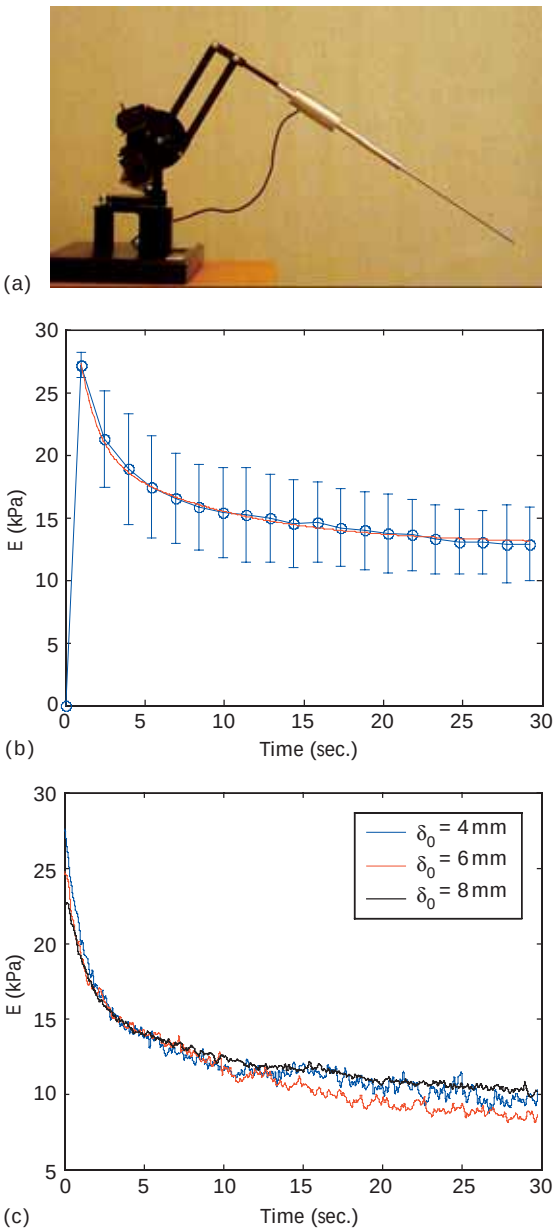
Using the robotic indenter, we conducted stress relaxation experiments with three pigs and successfully measured the viscoelastic material properties of pig liver under four different loading conditions. We indented each pig's liver to depths of 2, 4, 6, and 8 mms in one second and held the indenter there for 30 seconds to record the liver's force response as a function of time—that is, the force relaxation function, $F(t)$. We used the stress relaxation function to estimate the viscoelastic material properties of the pig liver.

A. Initializations
- A.I. Given: Δt
- A.II. Given, for $j = 1, \ldots, N$: γ_j and τ_j
- A.III. Given, for $n = 0, \ldots, \text{T}_{\text{end}}$: $\mathbf{f}_{\text{ext}}^{n}$
- A.IV. Zero, for $e = 1, \ldots, N_e$ and $j = 1, \ldots, N$: ${}_e\mathbf{h}_j^0 \leftarrow 0$

B. Precalculations
- B.I. Form, for $j = 1, \ldots, N$: $A_j = \dfrac{1 - e^{-\frac{\Delta t}{\tau_j}}}{\frac{\Delta t}{\tau_j}}$
- B.II. Form, for $e = 1, \ldots, N_e$: ${}_e\boldsymbol{K}_{\text{T}} = {}_e\boldsymbol{B}^{\text{T}}\boldsymbol{C}_{\text{elastic}}\,{}_e\boldsymbol{B}\,{}_eV\left(1 + \sum_{j=1}^{N}\gamma_j A_j\right)$; ${}_e\boldsymbol{K}_{\text{hist}} = {}_e\boldsymbol{B}^{\text{T}}\boldsymbol{C}_{\text{elastic}}\,{}_e\boldsymbol{B}\,{}_eV\sum_{j=1}^{N}\gamma_j A_j$
- *B.III. Assemble :* $\boldsymbol{K}_{\text{T}} = A_{e=1}^{N_e}\left({}_e\boldsymbol{K}_{\text{T}}\right)$
- *B.IV. Take inverse :* $\boldsymbol{K}_{\text{T}}^{-1} = I\left(\boldsymbol{K}_{\text{T}}\right)$
- *B.V. Solve :* $\mathbf{u}^0 = \boldsymbol{K}_{\text{T}}^{-1}\mathbf{f}_{\text{ext}}^0$
- *B.VI. Disassemble,* $e = 1, \ldots, N_e$: ${}_e\mathbf{u}^0 = A^{-1}(\mathbf{u}^0)$

C. Timestep loop calculations
- $n = 0, \ldots, \text{T}_{\text{end}} - 1$
- **********
- C.I. Calculate, for $e = 1, \ldots, N_e$: ${}_e\mathbf{h}_{\text{hist}}^{n+1} = {}_eV\,{}_e\boldsymbol{B}^{\text{T}}\sum_{j=1}^{N}e^{\left(-\frac{\Delta t}{\tau_j}\right)}\,{}_e\mathbf{h}_j^n$; ${}_e\mathbf{f}_{\text{hist}}^{n+1} = {}_e\mathbf{h}_{\text{hist}}^{n+1} - {}_e\boldsymbol{K}_{\text{hist}}\,{}_e\mathbf{u}^n$
- C.II. Assemble : $\mathbf{f}_{\text{hist}}^{n+1} = A_{e=1}^{Ne}\left({}_e\mathbf{f}_{\text{hist}}^{n+1}\right)$
- C.III. Solve : $\mathbf{u}^{n+1} = \boldsymbol{K}_{\text{T}}^{-1}\left[\mathbf{f}_{\text{ext}}^{n+1} - \mathbf{f}_{\text{hist}}^{n+1}\right]$
- C.IV. Disassemble, for $e = 1, \ldots, N_e$: ${}_e\mathbf{u}^{n+1} = A^{-1}(\mathbf{u}^{n+1})$
- C.V. Calculate, for $e = 1, \ldots, N_e$ and for $j = 1, \ldots, N$: ${}_e\mathbf{h}_j^{n+1} = e^{-\frac{\Delta t}{\tau_j}}\,{}_e\mathbf{h}_j^n + \gamma_j A_j\boldsymbol{C}_{\text{elastic}}\,{}_e\boldsymbol{B}\left({}_e\mathbf{u}^{n+1} - {}_e\mathbf{u}^n\right)$
- **********
- Next n

1 Pseudocode for calculating the nodal displacements of our linear viscoelastic finite-element model.



2 (a) The components of our robotic system for minimally invasive measurement and characterization of soft tissue behavior. (b) The stress relaxation behavior of pig livers for the indentation depth of 4 millimeters. The red colored curve is the Prony series approximation of the experimental data. (c) The stress relaxation function for different depths and loading rates.

The stress relaxation function defined by the generalized Maxwell solid results in a Prony series representation, given in Equation 2. We determined the coefficients of the Prony series for $N = 2$ (that is, the viscoelastic material properties E_∞, E_1, E_2, τ_1, τ_2) via curve fitting to the experimental relaxation data (see Figure 2b). For this purpose, we first obtained the shear relaxation function, $G(t)$, from the experimental force relaxation function, $F(t)$, using the small deformation assumption[7] ($G(t) = 3F(t)/16\delta\sqrt{R\delta}$, where δ is the indentation depth and R is the indenter tip's radius). Next, we obtained the stress relaxation function using the relation $E(t) = 2G(t)(1 + \nu)$, where ν is Poisson's ratio and equal to 0.5 for soft tissue (Figure 2b). Finally, we obtained the normalized values of elastic moduli, $\gamma = E_j/E_\infty$, used in our numerical computations (see Figure 1) from the averaged values of short-term (E_1 and E_2) and long-term moduli (E_∞) of three pigs. The results of the stress relaxation experiments suggest that a pig liver exhibits almost linear viscoelastic response. As Figure 2c shows, the experimental stress relaxation data for different loading rates overlap.

Real-time simulation

Our simulation system's hardware components include a computer monitor, for displaying visual interactions between the model and a virtual pointer, and the Phantom haptic device for simulating force interactions (see Figure 3a). Whereas we use a 3D volumetric model (made of tetrahedral elements) of human liver in our finite-element computations, we use a triangular surface representation constructed from the volumetric model's surface nodes in our graphical simulations. We wrote the underlying code in MS Visual C++. We use Open Inventor (a scene graph API) to display the object's graphical rendering and visual deformations, and the Ghost v. 4.0 driver to give haptic feedback to the user (Figure 3b).

Haptic rendering

The haptic rendering loop is much more demanding than the graphical rendering loop in displaying 3D objects. For rendering rigid objects, we must update the interaction forces between a haptic probe and a 3D object at 1 kHz. This rate is lower for rendering deformable objects, but still leaves us a short time for executing the underlying physics-based model to calculate the nodal displacements and interaction forces. In haptic rendering of static FEM deformations, we can calculate the nodal displacements in real-time using the FEM equation ($U = K^{-1}F$), assuming that we store the stiffness matrix's inverse in advance. However, in rendering viscoelastic deformations, we must consider the loading history's effect in displacement computations (see Equation 13).

This small change in the FEM formulation results in a significant increase in the number of computations. In fact, the direct implementation of the pseudocode in Figure 1 is computationally too expensive to execute in real-time and not suitable for haptic simulation. In particular, the number of computations in section C of the psuedocode is a major bottleneck.

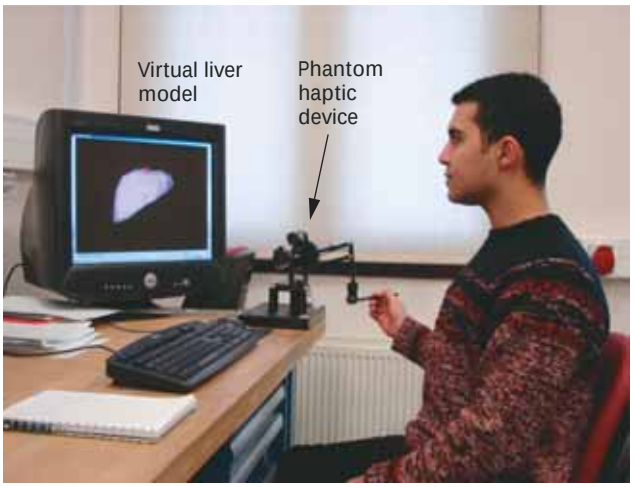
For example, the pseudocode in Figure 1 executes the computation of nodal displacements of a 3D cube consisting of $N_v = 51$ vertices, $N_{dof} = 153$ degrees of freedom, and $N_e = 136$ tetrahedrons in approximately 1.8 seconds on a Pentium IV 2.4-GHz dual-processor PC for an external force applied to the nodes on the cube's top surface for 1 second ($\Delta t = 1$ msec). For a relatively finer model, consisting of $N_v = 380$ vertices, $N_{dof} = 1{,}140$ degrees of freedom, and $N_e = 1{,}659$ tetrahedrons, the same computation takes 40 seconds. In particular, section C.III contains a complex multiplication of an $N_{dof} \times N_{dof}$ full

matrix with an $N_{dof} \times 1$ full vector. That complexity is directly proportional to the number of vertices in the mesh. Overall, the number of computations is $O\left(N_{dof}^2\right)$.

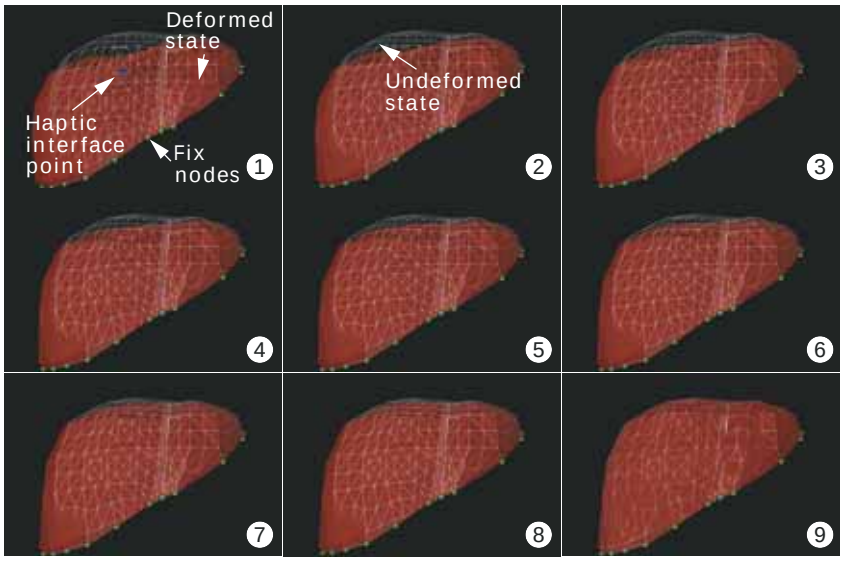
Implementation details

To calculate the nodal displacements and interaction forces in real-time, we exploit the linearity and superposition principles. Before the real-time simulation, we record the displacement and force response of each surface node and its neighboring nodes to a unit step force and a unit step displacement, respectively. During the real-time simulation, we use the prerecorded responses of the contacted node and its neighbors to calculate the resultant nodal displacements and interaction forces. Although researchers have implemented precomputation for real-time simulation of static FEM, extending this approach to a viscoelastic FEM simulation isn't straightforward because nodes' displacement and force response are rate and time dependent. However, this rate- and time-dependent behavior is important for displaying the object's true viscoelastic nature. In our real-time simulations, for example, the user feels the viscoelastic model's force relaxation response when the haptic probe penetrates the model and stays in place for a while. The user can also visually observe the nodes' slowly changing recovery displacements when the probe is removed.

To reduce the number of computations, we assume that the nodes' recovery response lasts for 30 seconds. (The force relaxation response of a pig's liver for different loading rates lasts approximately 30 seconds, as Figure 2b shows.) In addition, we assume that the loading influences only the nodes around the contacted node within a finite radius of influence (ROI). To calculate a node's displacement at any instant, we superimpose the effect of all past penetrations up to that instant. This requires us to simultaneously access the prerecorded data of previously contacted nodes and their neighbors. Accomplishing this task in a single cycle of haptic loop is highly challenging. Therefore, at this stage, we calculate the displacement response of each surface node based on the superposition principle at a rate of 100 Hz while updating the haptic loop at 1 kHz. Between two consecutive cycles of the displacement calculations ($\Delta t = 10$ msec), we feed the prerecorded force response of the contacted node to the haptic device for 10 msec.



(a)

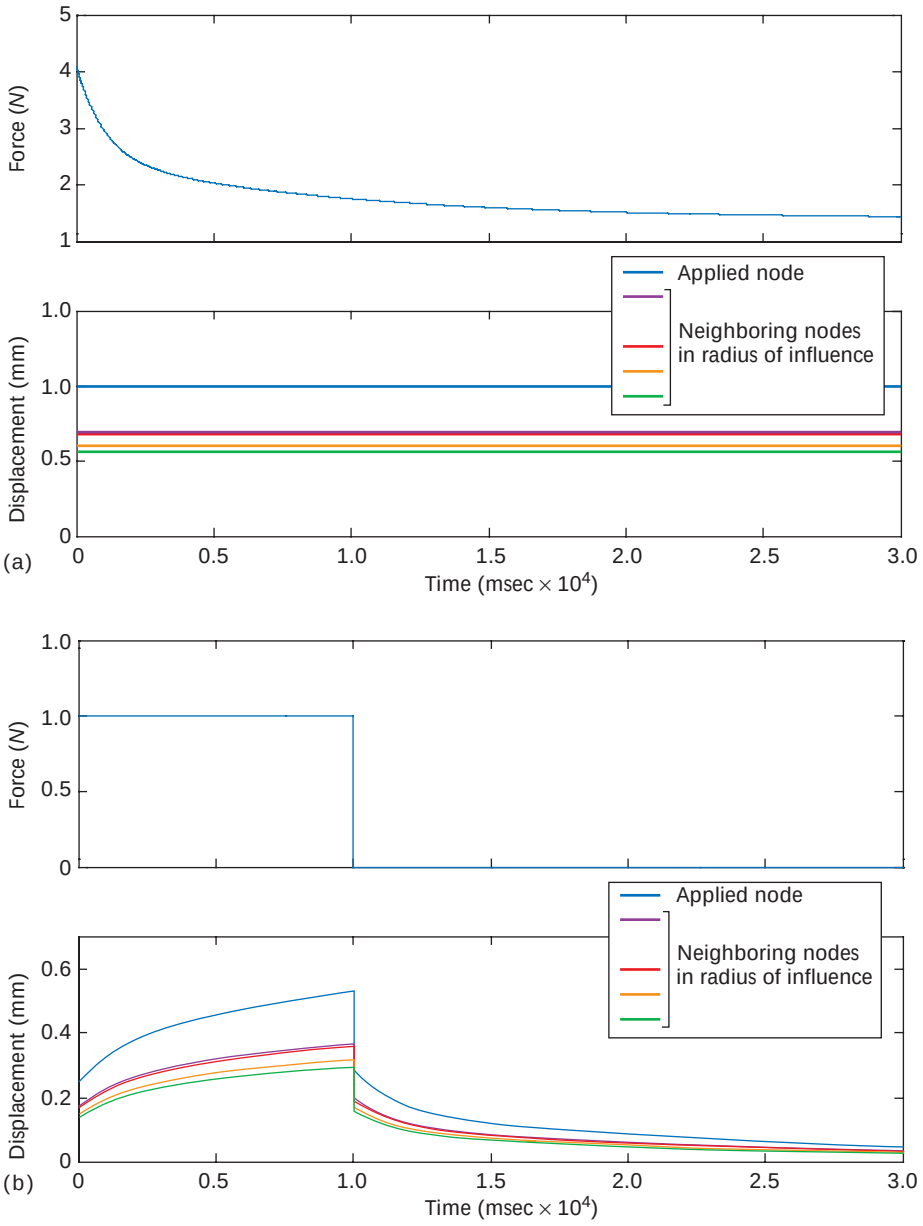


(b)

3 (a) The simulation system's components include a computer monitor for displaying visual deformations and a haptic device for displaying reaction forces. (b) A series of snapshots showing the viscoelastic relaxation of liver model in response to the force applied briefly to a surface node.

Prerecording phase. We first determine each surface node's neighbors within an ROI. We then record two sets of data for each surface node in the model. The first set stores the surface node's force response and its neighboring nodes' displacement response to a unit step displacement applied to it for 30 seconds. The second set stores 30 seconds of each surface node's and its neighboring nodes' recovery displacement response when we apply a unit step force to each surface node for 10 msec (recall that the prerecorded displacements are superimposed at 100 Hz). If a collision occurs during the real-time interactions, we use both data sets to calculate the nodal displacements and reaction forces. If no collision occurs, we use the second data set only to calculate and display the nodal relaxations.

To construct the first data set, we conduct virtual stress-relaxation experiments with the viscoelastic FEM model. In a typical stress-relaxation experiment conducted with an actual tissue sample, we apply a step displacement to the sample and record the force relaxation until a steady state force value is reached. Because force is the only input to our viscoelastic FEM model, and the nodal displacements are returned as output (see Equation 13), we perform a reverse operation to obtain each node's force-relaxation response to a unit step displacement. We individually apply the Prony series representation of the experimental force relax-



4 (a) We scale the Prony series representation of the experimental force-relaxation data (top) and apply it to a 3D cube surface node such that its displacement response is exactly 1 mm (bottom). (b) We apply a unit step force to the same node for 10 seconds (top) to obtain its resultant recovery displacement and that of neighboring nodes (bottom). The largest displacement is naturally observed at the applied node. (We apply the force for 10 seconds instead of 10 msec to display the creep response more clearly in the figure.)

ation data to each surface node of the viscoelastic FEM model to calculate the corresponding step-displacement profile at the applied node and its neighboring nodes. We then scale the force-relaxation curve at the applied node such that its nodal displacement is exactly 1 mm (see Figure 4a). We record the first 10 msec of the scaled force data as the *force-response* of the applied node and record the constant-valued nodal displacements of its neighbors (less than 1 mm) as the applied node's *displacement-response*.

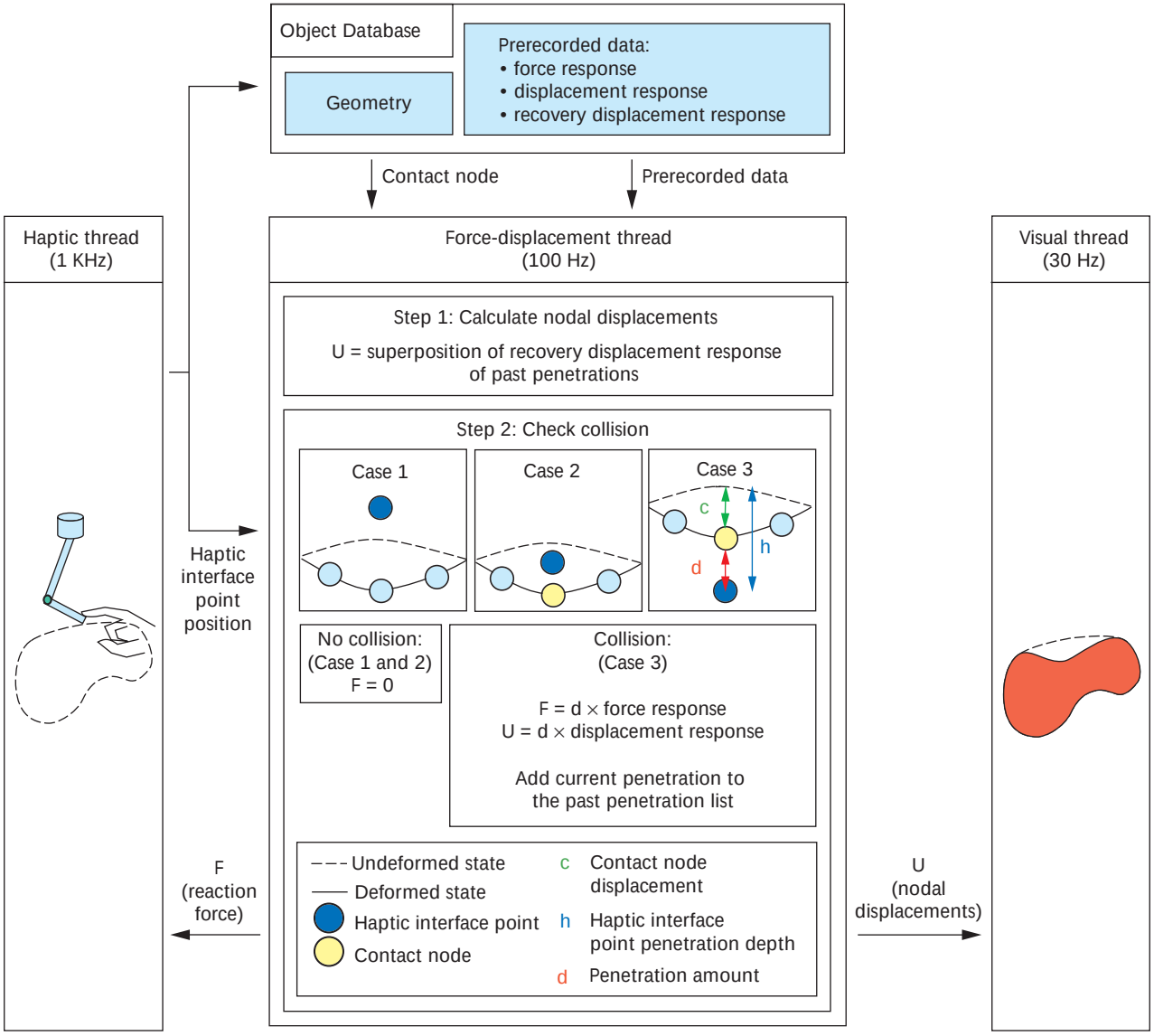
To construct the second data set, we conduct virtual creep experiments with the viscoelastic FEM model. We apply a unit step force to each surface node of the model for 10 msec and record the corresponding displacement response of the applied node and its neighboring nodes for 30 seconds as the applied node's *recovery-displacement-response* (see Figure 4b).

Real-time computation phase. Our real-time computational architecture for simulating linear viscoelasticity consists of three threads running asynchronously, as Figure 5 illustrates:

- The *haptic thread*, updated at 1 kHz, acquires the haptic probe's new position as the user manipulates it. If the force-displacement thread detects a collision, the haptic thread reflects the calculated interaction forces to the user through the haptic device.
- The *force-displacement thread*, updated at 100 Hz, performs collision detection and calculates the collision response based on the superposition principle.
- The *visual thread*, updated at 30 Hz, graphically renders the haptic interface point (HIP) and the model deformations.

At each cycle of the force-displacement thread, we determine the object's current (deformed) state. For that purpose, we calculate a node's displacement at the current time step by first scaling each past penetration's recovery-displacement-response with the magnitude of reaction force calculated for that penetration and then superimposing the scaled displacement responses. Because we assume that a node's recovery-displacement-response converges to a constant value after 30 seconds and the force-displacement thread is updated at 100Hz, the maximum number of superposition operations due to the past penetrations is at most 3,000.

We then check the collisions between the HIP's current position and the object's current state. If the HIP is outside the object, no force is displayed to the user and the superimposed nodal displacements are sent to the visual thread for graphical rendering. If a collision occurs, we calculate the penetration vector as the difference between the current positions of the HIP and the *contact node*. (Our geometric database returns the undeformed model's nearest surface node to the HIP as the contact node.) We calculate the reaction force for the next 10 msec by scaling the contact node's force-response by the penetration vector. We send this force profile to the haptic thread to be displayed to the user



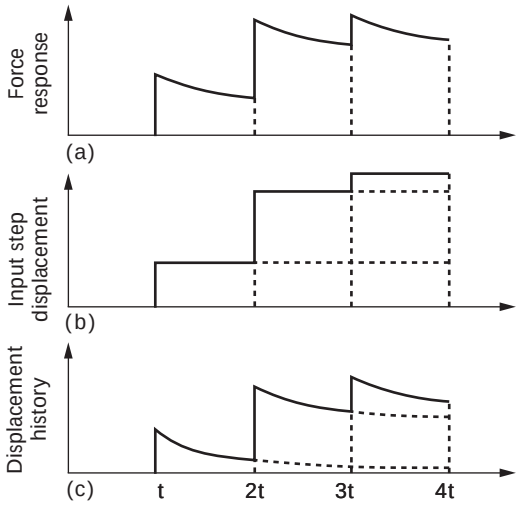
5 Flow chart of our precomputation approach.

through the haptic device until the next force-displacement thread cycle. To determine the nodal displacements due to the effect of the current penetration only, we set the contact node's position to the HIP's current position and scale the contact node's displacement response by the penetration vector's magnitude. To calculate the neighboring nodes' final displacements, we add their displacements due to the current penetration to their displacements due to past penetrations (see Figure 6 on the next page). In addition, we add information about the current penetration (that is, contact node, time of occurrence, and reaction force) to the database of past penetrations for use in calculations in upcoming cycles. Finally, we send the nodal displacements to the visual thread for graphical rendering.

Time complexity of our real-time computation phase is mainly governed by the nodal displacement calculations. Because we superimpose the recovery-displacement-response of all penetrations occurring up to the current time step, the complexity is O(number of penetrations × number of neighboring nodes of each penetration), where the number of neighbors of a node grows quadratically as the underlying mesh becomes finer.

Validation

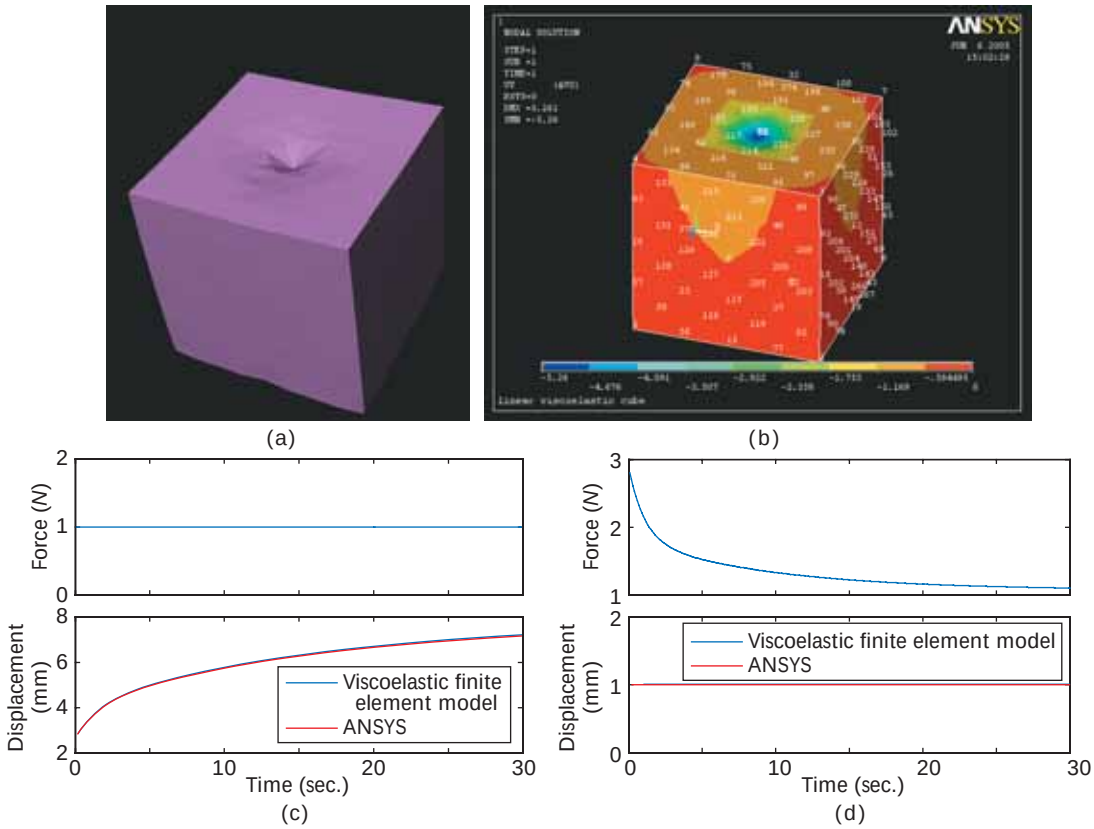
To validate our viscoelastic FEM model and the proposed precomputation approach, we conducted simulation experiments with a 3D cube consisting of $N_v = 51$ vertices, $N_{dof} = 153$ degrees of freedom, and $N_e = 136$ tetrahedrons. The nodes at the cube's bottom surface are constrained to have zero displacements in the vertical y-direction. To validate our linear viscoelastic FEM, we performed three different compression tests with the 3D cube. We then repeated the same compression tests using ANSYS finite-element package and compared the results (see Figure 7). In the first test, we applied 2.0 N to the center node on the cube's top surface. The displacement response calculated using our viscoelastic FEM model and the one calculated using ANSYS show a perfect agreement up to the second digit after the dec-



6 We compute the (a) reaction force displayed to the user by scaling the contact node's force response with magnitude of the (b) applied step displacement. (c) We scale recovery effects of past penetrations by the magnitude of the reaction force and superimpose them to each other to compute the contact node's displacement history.

imal point (see Figures 7a and 7b). In the second test, we applied 1.0 *N* to all nodes on the top surface for 30 seconds. Figure 7c shows the resultant displacement responses of our viscoelastic FEM and the ANSYS model. In the third test, we applied 1-mm displacement to all nodes on the top surface for 30 seconds using ANSYS to obtain their force-relaxation response. We then apply this response to the same nodes of our viscoelastic FEM model to reconstruct their unit step-displacement profiles, as Figure 7d shows. The results of all compression tests conducted with our viscoelastic model perfectly match those of ANSYS.

After demonstrating that our viscoelastic model works accurately, we performed a real-time test with the haptic device to validate the proposed precomputation approach. Figure 8 shows the results. First, we indented the center node on the cube's top surface to a certain depth using the haptic device and held it there for a while. Then, we released it and indented one of the neighboring nodes. We recorded the nodal displacements and force response of all the nodes on the top surface (see Figure 8a and 8b). To compare our precomputation approach with the direct solution of the linear viscoelastic FEM model, we supplied the reaction forces recorded during the real-time interactions to our



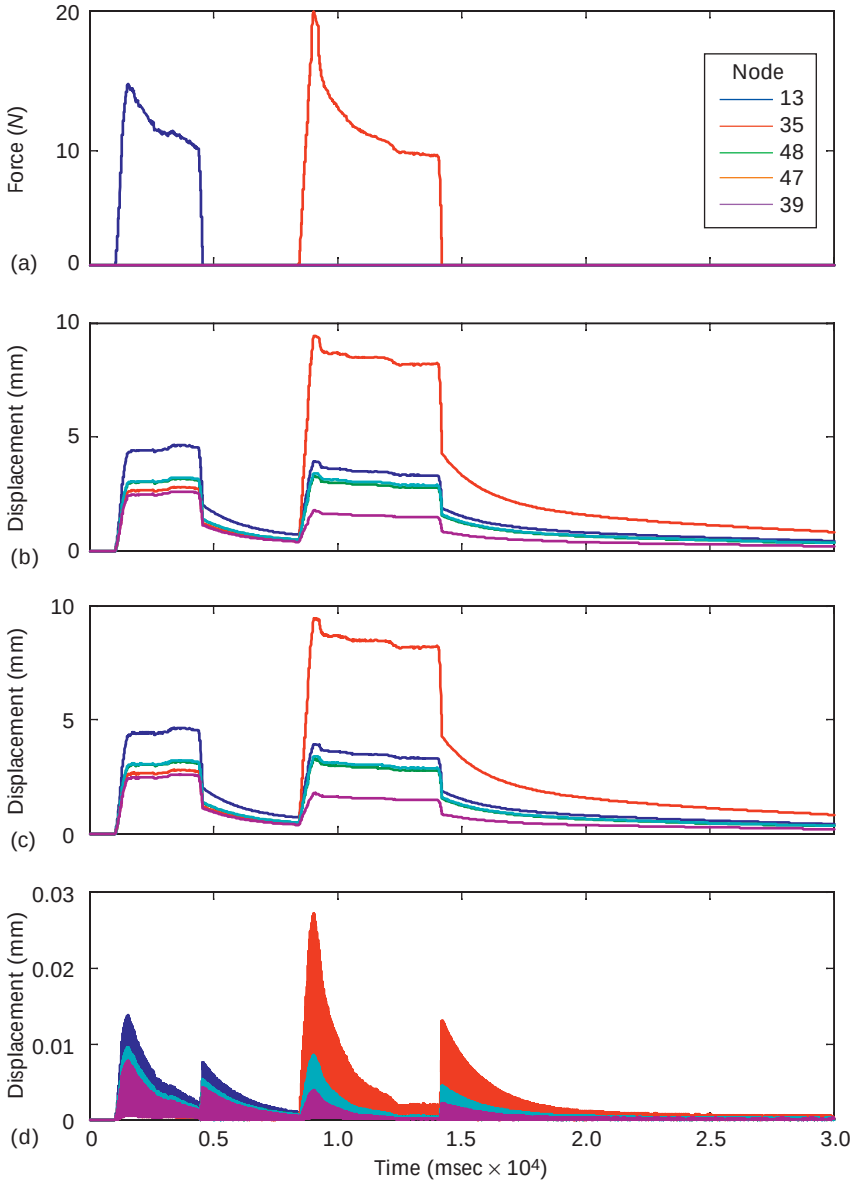
7 Validation test results. In the first test, the displacement response calculated using (a) our viscoelastic FEM and (b) that obtained using ANSYS show a perfect agreement up to the second digit after the decimal point (5.26 mm). (c) In the second test, the displacement response of node 13 calculated using our viscoelastic FEM and the response obtained using ANSYS shows a perfect agreement to the unit step force input. (d) In the third test, we recorded the force response data of node 13 calculated by ANSYS to unit step displacement input and supplied it to our viscoelastic FEM as input to reconstruct the step-displacement profile, which shows a perfect agreement with the input step-displacement profile given to ANSYS.

viscoelastic FEM model as input to obtain the nodal displacements (see Figure 8c). As Figure 8d shows, the maximum error between the displacement values calculated through superposition and those calculated through the direct solution of the linear viscoelastic FEM model is less than 1/100 of a millimeter for nodal displacements of a few millimeters.

Discussion and future work

Only a limited number of studies on real-time viscoelastic simulation of tissue behavior using finite elements for medical training applications exist. Such studies are difficult because the displacement response of viscoelastic objects is time-dependent and influenced by the loading rate. In addition, most of the existing viscoelastic soft tissue models aren't realistic because they aren't based on the measured material properties. This study aims to close the gap between the real-time simulation and realistic modeling of soft tissue behavior.

Through our stress-relaxation experiments, we determined the viscoelastic material coefficients of a pig liver via curve fitting to the experimental stress-relaxation data. The long-term elastic modulus obtained through this process ($E_\infty = 12.879 \pm 2.95$ kPa) corresponds to the effective linear elastic modulus of pig livers and show a good agreement with the values that Ottensmeyer obtained.[8] However, there's a significant variation in the material properties of pig livers reported in the literature.[6] One possible cause of this variation is the difference between the measurement devices (for example, hand-held probes versus robotics-based approaches) and the techniques (for example, in vitro versus in vivo). The material properties of soft tissues obtained through in vivo measurements are obviously closer to the actual properties, but the results should still be interpreted with caution because tissue response is location and direction dependent. In addition, the small indentation assumption used in our analysis affects the computation of viscoelastic material coefficients. Gefen and Marguilis argue that the small indentation assumption is valid up to 4-mm indentations made by a 2-mm radius round probe.[7] Therefore, we used the stress relaxation data of 4-mm indentations in our analysis and simulations. However, because pig livers shows linear viscoelastic response (see Figure 2b), we could have used the stress relaxation function corresponding to any other penetration depth in our analysis and simulations.



8 (a) To validate the precomputation approach, we applied external forces to cube nodes 13 and 48 using the haptic probe during the real-time simulations. We compared (b) the displacement response obtained by the superposition approach and (c) the response obtained directly from the viscoelastic FEM to quantify (d) the modeling error.

We also implemented a precomputation approach based on the superposition principle for the real-time simulation of the viscoelastic FEM model. The direct implementation of the pseudocode given in Figure 1 isn't feasible for the real-time haptic simulation because the number of computations is $O(N_{dof}^2)$. However, the suggested precomputation approach's complexity is proportional to the number of past penetrations multiplied by the number of neighboring nodes around the contact node of each penetration. Previous researchers have applied the precomputation approach to real-time static FEM simulation, but, to our knowledge, no one has extended it to the real-time linear viscoelastic FEM simulation. The major difference from the static case is the inclusion of time- and rate-dependent effects, which requires us to consider nodes' loading histories in our displacement computations at each cycle of the simulation. For the implementation, we recorded the force and displacement responses of surface nodes of the viscoelastic FEM model to a unit step displacement and force before the real-time simulation, and used the prerecord-

ed data during the real-time simulation to compute time-dependent nodal displacements and forces. We've validated the model and the proposed precomputation approach using ANSYS finite-element package.

In the current implementation, we precalculate and record the displacement-response, force-response, and recovery-displacement-response of each surface node in the form of individual data points. This consumes significant memory storage space. In the future, we plan to use curve-fitting methods to store the prerecorded data as a set of coefficients rather than individual data points. ■

Acknowledgments

The Scientific and Technological Research Council of Turkey (TUBITAK) supported this work under contract MAG-104M283.

References

1. C. Basdogan et al., "Haptics in Minimally Invasive Surgical Simulation and Training," *IEEE Computer Graphics and Applications*, vol. 24, no.2, 2004, pp. 56-64.
2. C. Basdogan, C. Ho, and M.A. Srinivasan, "Virtual Environments for Medical Training: Graphical and Haptic Simulation of Common Bile Duct Exploration," *IEEE/ASME Trans. Mechatronics*, vol. 6, no. 3, 2001, pp. 267-285.
3. M. Kaliske and H. Rothert, "Formulation and Implementation of Three-Dimensional Viscoelasticity at Small and Finite Strains," *Computational Mechanics*, vol. 19, 1997, pp. 228-239.
4. Y.C. Fung, *Biomechanics: Mechanical Properties of Living Tissues*, 2nd ed, Springer-Verlag, 1993.
5. M. Bro-Nielsen and S. Cotin, "Real-Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation," *Computer Graphics Forum* (Eurographics), vol. 5, no. 3, 1996, pp. 57-66.
6. E. Samur et al., "A Robotic Indenter for Minimally Invasive Characterization of Soft Tissues," *Proc. 19th Int'l Conf. Computer-Assisted Radiology and Surgery*, vol. 1281, Elsevier, 2005, pp. 713-718.
7. A. Gefen and S.S. Margulies, "Are In Vivo and In Situ Brain Tissues Mechanically Similar?" *J. Biomechanics*, vol. 37, 2004, pp. 1339-1352.
8. M.P. Ottensmeyer, "Minimally Invasive Instrument for In Vivo Measurement of Solid Organ Mechanical Impedance," doctoral dissertation, Dept. of Mechanical Engineering, Mass. Inst. of Technology, 2001.

Mert Sedef *is a graduate student in the Department of Computational Sciences and Engineering at Koc University, Turkey. His research interests include virtual surgery, biomedical modeling and simulation, virtual environments, and computer haptics and graphics. Sedef has a BS in computer engineering from Koc University. Contact him at lsedef@ku.edu.tr.*

Evren Samur *is a graduate student at the Robotic Systems Laboratory at the Ecole Polytechnique Fédérale de Lausanne, Switzerland. His research interests include robotics, measurement and characterization of soft tissue properties, and biomedical modeling and simulation. Samur has an MS in mechanical engineering from Koc University. Contact him at evren.samur@epfl.ch.*

Cagatay Basdogan *is a member of the faculty in the Departments of Mechanical Engineering and Computational Sciences and Engineering at Koc University. His research interests include haptic interfaces, robotics, virtual environments, medical simulation, and biomechanical modeling. Basdogan has a PhD in mechanical engineering from Southern Methodist University, Dallas. Contact him at cbasdogan@ku.edu.tr.*