2014 IEEE 27th International Symposium on Computer-Based Medical Systems

Patient-Specific Interactive Simulation of Compression Ultrasonography

Kresimir Petrinec, PhD
SonoSim, Inc.
Santa Monica, CA
Email: petrinec@cs.ucla.edu

Eric Savitsky, MD
David Geffen School of Medicine
University of California, Los Angeles
Email: esavitsk@ucla.edu

Demetri Terzopoulos, PhD
Department of Computer Science
University of California, Los Angeles
Email: dt@cs.ucla.edu

***Abstract*—We are developing an ultrasonography training system that promises to accelerate the broader use of ultrasound imaging in healthcare. Aiming at cheaper, more efficient, and more effective ultrasound training, a key feature of our system is the real-time, interactive simulation of a 3D virtual patient that, unlike conventional, purely geometric models of the human body, includes deformable soft tissues. Since soft-tissue deformation is an important factor in the clinical practice of ultrasound imaging, our objective in this paper is to incorporate real-time interactive soft tissue mechanics simulation into our 3D patient model. To this end, we adapt and evaluate two well-known deformable model simulation methods—mass-spring-damper systems and the finite element method—and we apply these methods to the simulation of ultrasound imaging in soft tissues, obtaining promising results on a multicore laptop computer.**

I. Introduction

Ultrasound imaging systems provide a low-cost, real-time, noninvasive and safe way to examine soft tissues inside the human body. Yet the ability of medical practitioners to understand and mentally register two-dimensional (2D) ultrasound image slices within the three-dimensional (3D) anatomy is a difficult task, so training is needed. Traditional ultrasound training methods are expensive, inefficient, and pose a major obstacle to the wide deployment of ultrasound imaging systems in routine clinical practice. In a new approach to ultrasound training, complex and expensive physical phantoms are replaced by a 3D virtual patient model, which represents the anatomy of any desired body part or organ, and ultrasonography of the virtual patient with a virtual ultrasound probe is simulated on a standard laptop computer [1], [2]. The advantage to training is not only cost effectiveness, but also the ability to emulate various disease states or conditions in different virtual patients and to visualize the underlying body structures of interest through multiple examination procedures with the virtual probe.

In practice, when an ultrasound probe is pressed against the skin, the deformation of soft tissues (e.g., skin, fat, muscles, tendons, ligaments, fascia, blood vessels, nerves, etc.) has a substantial diagnostic value. A good example is ultrasound-guided central venous catheter (CVC) placement, where the difference between vein and artery can be determined by shape and compressibility. However, traditional blind techniques, which rely on anatomical landmarks to estimate the location of vessels, result in a high rate of complications. What makes the traditional landmark approach problematic is that many factors, such as intravenous drug use, cardiac arrest, or even body type (e.g., underweight or overweight), can alter the usual anatomic relationship, so physicians require multiple attempts to cannulate the vessel of interest. On the other hand, ultrasound enables the real-time imaging of vessels during CVC placement, making it safe, fast, and easy, so it is not surprising that ultrasound-guided CVC placement is becoming a standard. Another good example where compression ultrasonography is useful is in the diagnosis of deep vein thrombosis (DVT). For example, when the probe is pressed against the skin of a normal patient, the veins collapse easily and the deformation is instantly visible in the ultrasound image. If the veins do not deform under pressure, which can also be clearly visible in the ultrasound image, that may indicate a positive finding for venous occlusion [3].

This paper describes our approach to the simulation of soft tissue compression in ultrasound imaging and presents our results. In particular, we focus on the ultrasonography training system components related to the soft-tissue compression functionality. For real-time soft-tissue simulation, the speed of the deformation algorithms is more important than their accuracy; rather than highly accurate soft-tissue deformation that may take seconds or minutes per frame to simulate, we are interested in less accurate methods that can run in real-time yet which can provide visually plausible results. To this end, we evaluate two deformable model methods for use in our simulator—mass-spring-damper systems (MSDS) and the finite element method (FEM).

II. System Overview

Our ultrasound training system is illustrated by the block diagram shown in Fig. 1. The system consists of six major components—(1) a database of real-patient volumetric ultrasound data, (2) a soft-tissue deformation simulator, (3) a slice interpolation module, (4) a skin deformation simulator, (5) a probe motion tracking subsystem, and (6) an interactive user interface—each of which are described below:

Real-patient volumetric ultrasound database: The database contains a number of data sets acquired either with a 3D ultrasound probe, or data reconstructed from 2D ultrasound B-scans. The data, which is typically stored in DICOM format, is preprocessed in such a way that only raw 3D data with spacial information is preserved. All other information, such as patient ID, name, age, or pathology, is stripped from the data sets. In addition to real-patient data, each data set contains segmentation information.

1063-7125/14 $31.00 © 2014 IEEE
DOI 10.1109/CBMS.2014.76

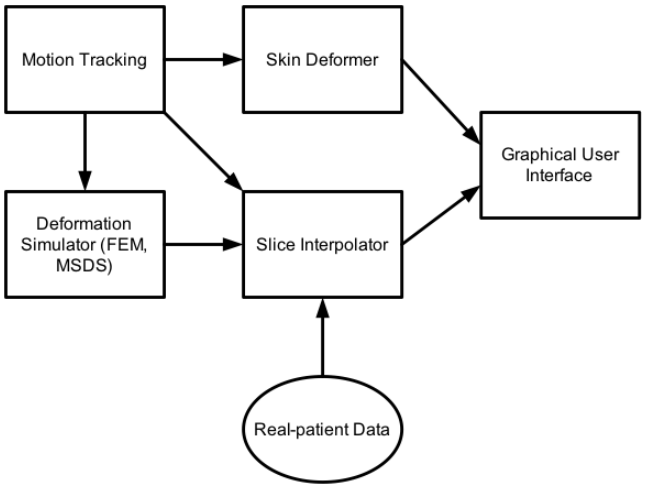


Fig. 1. Ultrasound training system overview.

Soft-tissue deformation simulator: The soft-tissue deformation simulator encapsulates two deformable model simulation methods—mass-spring-damper systems (MSDS) and the finite element method (FEM). A unique interface to the simulation libraries permits the real-time switching and comparison of the methods. Given the position and orientation of a collider (usually the ultrasound probe), the simulator handles the collision, and computes the new state of the nodes in the mesh.

Slice interpolation module: The slice interpolation module samples and interpolates the ultrasound image over the deformed soft-tissue mesh. The soft-tissue model in its undeformed state is determined by static real-patient data captured by applying a linear ultrasound probe. Using the probe position and orientation, probe beam parameters, volumetric data, and the mesh in the undeformed and deformed state, the module synthesizes a 2D ultrasound image that is displayed on the screen.

Skin deformation simulator: The skin deformation simulator uses an MSDS to simulate a visually pleasing surface deformation. The edges of the triangular mesh are replaced with springs. To provide resistance and to ensure that the surface returns to its initial position when the collider is removed, all nodes of the mesh are anchored with zero-length springs.

Probe Motion tracking system: The probe motion tracking system determines the position and orientation of the virtual ultrasound probe. Different systems may be used; for example, a single camera optical tracking system, a 6-DOF mouse, or a haptic device.

Interactive user interface: The user interface consists of an interactive 3D display with a deformable virtual body, virtual probes, and a simulated ultrasound image slice; and dialogs for setting the FEM, MSDS, and simulation parameters. The user can rotate and zoom the display, or move and rotate the probe with a mouse. The user can also select a case using a case list dialog.

The body of the virtual patient (Fig. 2) has a rigid skeleton, deformable skin, and simulated deformable structures embedded into the tissue (isosurfaces of vessels and nerves). The triangular mesh of the skin deforms when the virtual probe (collider) attempts to penetrate, and recovers its original shape when the collider is not in contact. Isosurfaces of the segmented regions of real-patient data are embedded into the mesh and follow the deformation of the mesh. The user can look under the skin and examine how the segmented anatomical structures move. For example, unlike arteries, veins under pressure will deform and even completely collapse.

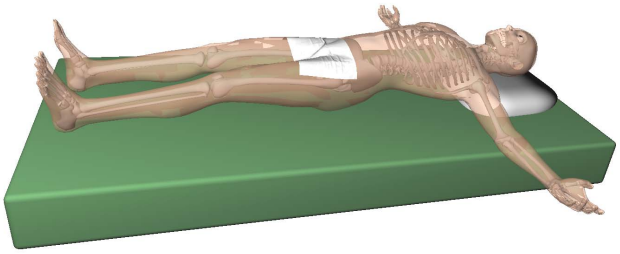


Fig. 2. A virtual patient—the Ultimate Human Model, male cgCharacter.

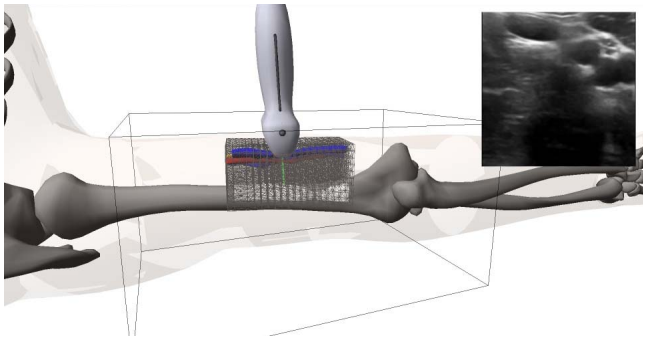


Fig. 3. Deformable soft-tissue mesh under the skin, with embedded vessels.

The arrows in the block diagram (Fig. 1) indicate the information flow. The virtual ultrasound probe applies force to mesh nodes within a spherical collision volume. The motion tracker, which tracks the virtual probe, interacts with the tissue deformation simulator and the skin deformer, which modifies the nodes and surface normals of the skin mesh. The deformed mesh is available to the slice interpolator, along with the probe position/orientation from the probe motion tracker and case-specific real-patient data from the database. The slice interpolator projects slice pixels into the deformed mesh, determines corresponding points in the undeformed state, and sets pixels intensities to voxel intensities interpolated from the volumetric real-patient data. The interpolated ultrasonogram slice is shown in the graphical user interface display as a 2D image.

A snapshot of our simulator in Fig. 3 shows the skeleton of our virtual patient, a virtual ultrasound probe, a mesh with the embedded isosurfaces of vessels and nerves, and a simulated ultrasound B-scan. The probe applies pressure on the skin and soft-tissue, and the embedded vein is compressed. Simultaneously, the resulting simulated 2D ultrasound B-scan shows vein deformation.

III. Simulation of Compression Ultrasonography

To develop a model for the real-time simulation of compression ultrasonography, we must select a suitable geometric

representation and mathematical model of deformation, as well as a fast solution algorithm. We need a robust model with consistent and predictable behavior that can be realistically simulated with visually convincing results. To achieve those goals, we evaluated two well-known deformable model simulation methods: mass-spring-damper systems (MSDS) and the finite element method (FEM) with a quasistatic solution of isotropic linear elastic materials with Cauchy strain.

The finite element method is a well-known method for the simulation of deformable solids [4]. We use the FEM simulator in OpenTissue, an open-source library for physics-based animation ([5] evaluates alternative real-time physics simulation systems). OpenTissue contains a collection of algorithms and data structures written in an object-oriented style and optimized for interactive modeling and simulation. Interactive simulation speed is achieved by using quasi-static stress-strain simulation; i.e., an iterative solver using the conjugate-gradient method.

Mass-spring-damper systems are easy to implement and have a low computational complexity (the appendix provides additional details). We use explicit Euler numerical time integration, which is fast albeit unstable for large time steps. To ensure stability, we choose appropriately small time steps and we overdamp the MSDS; i.e., we set the spring and damping constants so that the system remains stable even when a large external force is applied.

A. Comparison Between FEM and MSDS

In order to evaluate the two methods for ultrasound image simulation in deformable tissue, we must make sure that both simulations have approximately the same dynamic response, which presents a problem. First we create an FEM system and set its material properties to produce a stable, over-damped dynamic response. We then apply Algorithm 1 to create an MSDS with the same topology and number of nodes as the FEM and with a similar dynamic response.

Algorithm 1: Convert an FEM to an MSDS.

input : *Tets* – List of FEM tetrahedra
output: *Springs* – List of MSDS springs

forall the *tetrahedra* T *in Tets* **do**
  forall the *edges* E *in* T **do**
    Let p_A and p_B be the nodes of edge E;
    Let S_E be a spring between p_A and p_B;
    Set the spring constant $k \leftarrow 0.002 * T_{\text{Young}}$;
    Set the damping constant $\gamma \leftarrow 0.1 * k$;
    if S_E *exists in Springs* **then**
      Update the spring constant of S_E: $k_{S_E} \leftarrow k_{S_E} + k$;
      Update the damping constant of S_E: $\gamma_{S_E} \leftarrow \gamma_{S_E} + \gamma$;
    else
      Add S_E with spring constant k and damping constant γ to *Springs*;
    end
  end
end

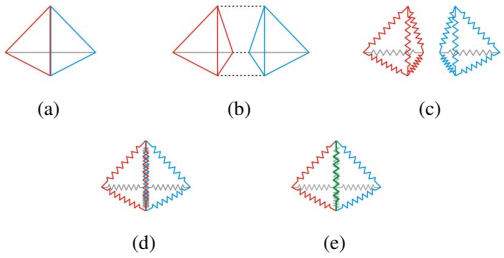
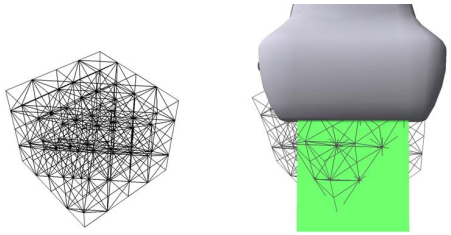
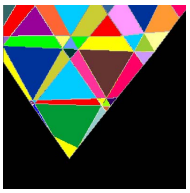


Fig. 4. Converting FEM to MSDS.



(a) Undeformed mesh. (b) Mesh deformed by probe.



(c) Slice cut from deformed mesh.

Fig. 5. Slice cutting.

The algorithm converts all the edges of the FEM mesh tetrahedra into springs. Spring constants are set to the Young modulus of the tetrahedra multiplied by a constant (0.002), which we empirically found to yield a similar dynamic response. Damping constants are set to the spring constant multiplied by a factor (0.1), which we empirically found results in stable, over-damped dynamic response. Duplicate springs are merged by summing spring and damping constants. Fig. 4 illustrates the process on two adjacent tetrahedra with connected faces.

IV. Ultrasound Image Synthesis Using the Deformed Mesh

Given a regular mesh and volumetric data embedded in the mesh, the challenge is to synthesize 2D images (slices) from the deformed mesh in real time. For example, Fig. 5(a) shows a regular $5 \times 5 \times 5$ mesh in undeformed configuration. It comprises 125 nodes and 320 conforming tetrahedra. Fig. 5(b) shows the mesh deformed when a probe applies pressure on the top side of mesh. A slice plane cuts a number of tetrahedra in the deformed mesh. The cut consists of triangles and rectangles, as shown in various colors in Fig. 5(c).

For each pixel in the slice, we can easily find the corresponding voxel intensity. First, one can find the 3D position of the pixel in the deformed mesh. One can precompute the slice transformation and apply it to the pixel. Then, one must find which tetrahedron contains the pixel, compute its barycentric

coordinates, and use them to find its 3D coordinates in the undeformed mesh. Finally, one must sample the volumetric data to obtain the voxel intensity and set the pixel to that intensity. Unfortunately, doing this for every pixel is too inefficient for real-time applications.

Goksel and Salcudean [6] proposed an algorithm for fast ultrasound image synthesis. They exploit the fact that mesh tetrahedra are much bigger than image pixels, so numerous image pixels are enclosed by a tetrahedron cut by the image. They build a data structure where for each pixel they store the index of the tetrahedron which intersects the image. In the first pass, they look for all intersections of faces with the image. A pixel may be intersected by multiple faces of the tetrahedra. To assign the correct element to the pixel, they topologically sort all tetrahedra cross sections, top to bottom, or column by column. Subsequently, they use a scan-line approach to find tetrahedra of pixels with unassigned tetrahedra.

In our approach, we first create a list of tetrahedra that intersect bounding boxes of deformed tetrahedra. Then, we process slice pixels row-by-row. In each row, we take a pixel P_a at the far left and P_z at the far right. If they fall into same tetrahedron, then we linearly sample voxels from interval $[P_a, P_z]$ in the undeformed configuration. Otherwise, we take a pixel P_m in the middle and recursively apply the same process to intervals $[P_a, P_{m-1}]$ and $[P_m, P_z]$ while $a < z-1$. To speed up the search for tetrahedra, we split the slice into R regions, and for each region we create a list of tetrahedra crossing the region. We analyzed the trade-off between the number of regions and the speed of this algorithm, and we empirically found that we achieve the best results when $R = 32$.

To summarize, the major differences between our approach and the approach of [6] is that we scan and set the image pixels row-by-row, and we use row subdivision to find and set pixels values within the same tetrahedron. Our simpler approach is easier to implement because it does not require topological sorting.

V. Deformable Skin Simulation

Pressing the ultrasound probe against the skin may cause significant, visible deformation to the skin and underlying tissue. It is desirable that the skin of the virtual body also be deformable so as to simulate compression as realistically as possible.

In our initial implementation, we embedded the skin of the virtual patient directly into the soft-tissue deformation mesh. When we applied pressure to the mesh, it caused the skin to bulge on the edges where the skin intersected the mesh. Even though the deformation of the skin inside the mesh looked satisfactory, the abrupt changes in the skin exposed the rectangular shape of the mesh, which was not visually pleasing.

To prevent sudden and unnatural skin deformation around the intersection with the tissue mesh, we decided to simulate the skin separately from the tissue. The skin is simulated as a MSDS with a single-layer mesh. We use the same collision model for both systems; i.e., when the probe is in contact, the collider, which approximates the shape of the probe, applies a repulsion force to the nodes of both the skin mesh and the soft-tissue mesh. As a result, the skin deforms nicely under contact

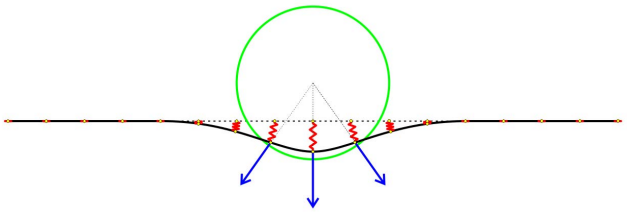


Fig. 6. A cross-section of a surface (black) deformed by a sphere (green). External forces (blue) are applied to surface nodes in collision. Zero-length springs (red) pull the surface back to its initial state (dashed line).

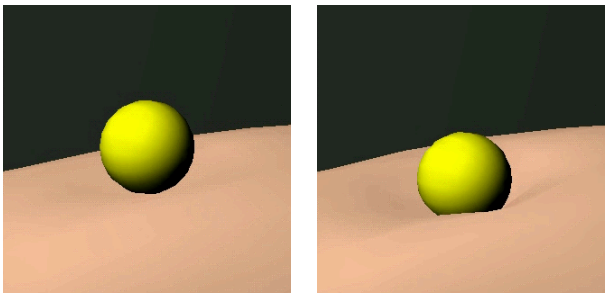


Fig. 7. Example interaction between the skin and a spherical object.

with the probe and there are no unrealistic deformations where the skin crosses the tissue mesh.

The skin of the virtual body is a surface that consists of a number of triangular faces. The faces share edges and vertices with adjacent faces. Using MSDS, the vertices of the triangles are represented with particles and edges of the triangles are represented with springs. The virtual probe, when in contact with the surface, applies external repulsion forces to particles in collision, pushing them away.

However, the whole surface will drift away from its initial state if it is not properly constrained. One way to prevent the drift is by anchoring the surface to the skeleton with a mesh of springs. Terzopoulos and Waters [7] create a physically-based 3D model of the human face with three layers of mass-spring elements representing a muscle layer and two skin layers (dermis and epidermis). The bottom surface of the muscle layer is constrained by bone and facial expressions are controlled by muscle contractions. A benefit of this approach is that the surface deforms together with the muscles and skull; e.g., when the mandible moves, the skin follows it.

In our work, the skeleton is rigid and the skeletal muscles are not active, so modeling the skin with an underlying mesh fixed in bone would introduce unnecessary complexity and require more processing time. Instead, we prevent the remainder of the skin from drifting by anchoring the entire surface to its initial position using zero-length springs attached to all particles. As these springs stretch, they allow the surface to deform away from a colliding object and they ensure that the surface returns to its initial undeformed state when the external forces are removed. For bodies with rigid skeletons, this enables simple and efficient implementation while yielding realistic simulation.

Fig. 6 demonstrates the interaction between a sphere and a deformable surface. An external repulsion force is applied to

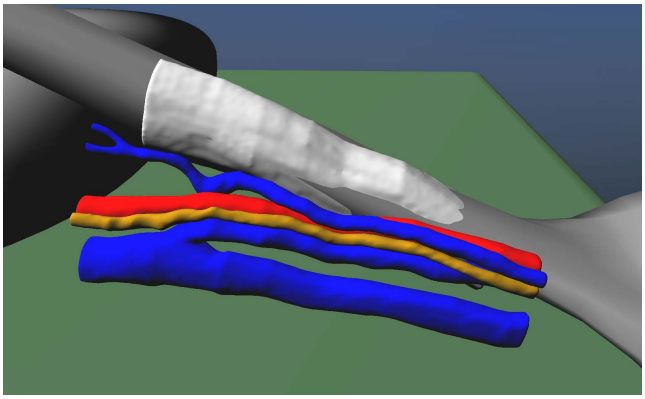


Fig. 8. Isosurfaces of ultrasound data roughly aligned with the skeleton of the virtual patient model.

TABLE I. SIMULATION TIMES (IN MILLISECONDS).

Mesh size	20x10x10	24x12x12	28x14x14	32x16x16
Number of nodes	2,000	3,456	5,488	8,192
Number of tetrahedra	7,695	13,915	22,815	34,875
MSDS (FEM) simulation	10 (30)	17 (60)	27 (100)	40 (150)
Collision response	0.1	0.2	0.2	0.3
Volume sampling	14	16	20	23

TABLE II. MSDS (FEM) FRAME RATES (IN FRAMES PER SECOND).

Mesh size	20x10x10	24x12x12	28x14x14	32x16x16
Sim. without vis.	20 (15)	16 (10)	15 (6.8)	12 (5.0)
Sim. with vis.	12 (8.6)	10 (7.4)	8.6 (5.5)	8.5 (4.3)

the surface particles that are in collision with the sphere. Zero-length springs resist the external forces, causing the surface to deform away from the sphere. This is also demonstrated in Fig. 7. A part of the surface stays in contact. The large repulsion force will push the surface away from the sphere, but it may cause unstable behavior (oscillations or divergence).

VI. SIMULATION RESULTS AND DISCUSSION

All the results presented in this section were simulated and rendered in real time on a laptop computer with a quad-core, 2.3 GHz Intel® Core™ i7 CPU and an NVIDIA® GeForce® GT 650M GPU.

We created regular tetrahedral meshes for the FEM and MSDS off-line as follows: First, we segmented volumetric ultrasound data using a semi-automatic segmentation method. Then, we used marching cubes [8], a common technique for isosurface extraction. Isosurfaces produced by the marching cubes algorithm appear faceted. To reduce faceting, we refined the surfaces by using Gaussian filtering. Fig. 8 shows an example visualization of the isosurfaces of vessels, nerves, and bone in the upper arm. It took about 2 seconds to create five isosurfaces with a total of 243224 triangles.

In our experimental study, we used a time step of 0.01 sec for the time integration, and we ran 20 iterations per frame; i.e., the FEM ran 20 conjugate gradient iterations per frame and the MSDS ran 20 explicit Euler iterations per frame. The parameters of the MSDS were empirically set to match the dynamics of the FEM as described in Section III-A.

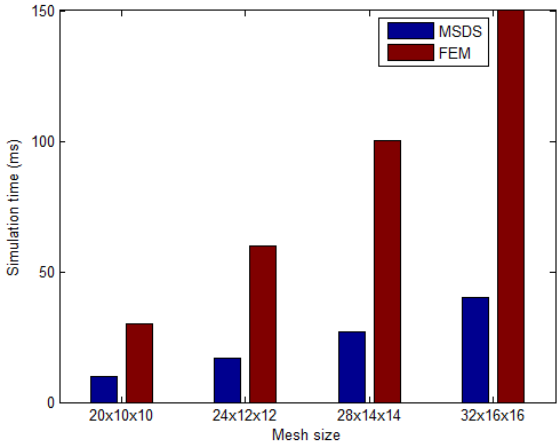


Fig. 9. Comparison of MSDS and FEM simulation times.

In the first experiment, we synthesized ultrasound images in deformable tissue using the MSDS and FEM. Table I reports the average time to simulate mesh deformation, compute probe collision, and sample the B-scan image using four different mesh sizes. Fig. 9 shows a bar graph comparing the deformation simulation times. The results show that the deformation simulation time grows linearly with the number of nodes in the mesh for both the MSDS and the FEM. The collision detection/response took less than one percent of the time, which is negligible. The volume sampling time also increases with the mesh size, but the ratio of the volume sampling time versus the deformation simulation time declines as the number of nodes increases.

In the second experiment, we investigated how the visualization of segmented data (e.g., vessels, nerves, bones, etc.) affects the overall simulation time for different mesh sizes, with and without the visualization of segmented data (Fig. 8) embedded in the mesh and deforming along with it. Table II reports the average simulation frame rates with and without visualization. Our results demonstrate that our method yields visually pleasing simulation of soft-tissue deformation in real time; i.e., the simulation of a system with approximately 35,000 tetrahedra runs at more than 10 frames per second.

VII. CONCLUSION

We have presented an advanced real-time interactive ultrasound imaging simulator incorporating soft-tissue deformation that runs on a PC laptop computer. Our approach promises to advance ultrasonography training efficacy by providing a realistic and cost-effective method of enhancing ultrasound simulation verisimilitude through real-time soft-tissue simulation. For this purpose, we applied mass-spring-damper systems (MSDS) and the finite element method (FEM) to the simulation of compression ultrasonography, and we evaluated their performance. We achieved real-time interactive simulation rates on a laptop computer by carefully adapting our code to run efficiently on multicore processors. Our simulation results show that MSDS can achieve approximately the same dynamic response as the FEM but more than three times faster for the same mesh size. In our system, the simulations of the skin and

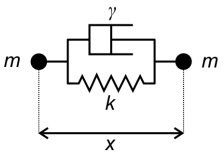


Fig. 10. Voigt model.

the other soft-tissues are separated in order to avoid artifacts and achieve realistic, visually pleasing skin deformation. The skin of the virtual body model deforms at interactive rates when in contact with the virtual ultrasound probe, and the user can visualize in real time how the vessels embedded within the soft tissue deform subject to external compression forces applied to the skin by the virtual ultrasound probe.

APPENDIX

A. Mass-Spring-Damper Systems

Mass-spring-damper systems (MSDS) are the most intuitive and simplest deformable models to implement. They comprise lumped-mass particles connected together with massless springs and dampers—viscoelastic elements, also called Voigt elements (Fig. 10). The force s at time t exerted by a Voigt element is

$$s(t) = -k(x - x_0) - \gamma \frac{dx}{dt}, \tag{1}$$

where k is the spring constant, x_0 is the original (natural) length of the spring, x is its current length, and γ is the damping constant of the damper. The net force acting on each particle stems from its connections to neighboring particles, plus external forces, including gravity, friction, collision forces, etc.

In accordance with Newton's second law of motion, the movement in 3D of N particles is computed by numerically solving the system of second-order ordinary differential equations [7]

$$m_i \frac{d^2\mathbf{x}_i}{dt^2} + \mathbf{s}_i = \mathbf{f}_i; \quad i = 1, \ldots, N, \tag{2}$$

where m_i is the mass of the i-th particle, whose position is $\mathbf{x}_i(t)$, the total spring-damper force acting on the particle is $\mathbf{s}_i$, and $\mathbf{f}_i$ is total external force. These differential equations can be simulated by applying explicit, semi-implicit, or implicit numerical integration methods.

The simplest numerical integration method, albeit limited in its stability, is the explicit Euler method. Starting from given initial particle positions $\mathbf{x}_i^0$ and velocities $\mathbf{v}_i^0$ at time $t = 0$, the accelerations $\mathbf{a}_i$, velocities $\mathbf{v}_i$, and positions $\mathbf{x}_i$ of the particles are computed at every time step, $t = 0, \Delta t, 2\Delta t, \ldots$, as follows:

$$\mathbf{a}_i^t = \frac{1}{m_i}\left(\mathbf{f}_i^t - \mathbf{s}_i^t\right), \tag{3}$$

$$\mathbf{v}_i^{t+\Delta t} = \mathbf{v}_i^t + \Delta t \mathbf{a}_i^t, \tag{4}$$

$$\mathbf{x}_i^{t+\Delta t} = \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+\Delta t}. \tag{5}$$

The spring constants in MSDS are often chosen arbitrarily, and little can be said about the material being modeled. Although generalized springs may be employed to preserve areas and volumes, it is difficult to incorporate continuous material properties. Even though they are not as accurate as competing methods, MSDS are acceptable for many CGI applications in motion pictures and games. They have been successfully employed in cloth animation, facial animation, simulation of soft materials and organic active bodies, etc. (see, e.g., [9]).

REFERENCES

[1] K. Petrinec, C. Hein, and E. Savitsky, "Patient-specific cases for an ultrasound training simulator," in *Studies in Health Technology and Informatics*, vol. 143, 2011.

[2] K. Petrinec, "Patient-specific interactive ultrasound image simulation with soft-tissue deformation," Ph.D. dissertation, Computer Science Department, University of California, Los Angeles, 2013.

[3] J. G. Crisp, L. M. Lovato, and T. B. Jang, "Compression ultrasonography of the lower extremity with portable vascular ultrasonography can accurately detect deep venous thrombosis in the emergency department," *Ann Emerg Med*, vol. 56, no. 6, pp. 601–610, Dec 2010.

[4] O. Zienkiewicz and R. Taylor, *The Finite Element Method: Volume 1, Fifth Edition*. Butterworth-Heinemann, 2000.

[5] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems," in *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia (GRAPHITE '07)*. New York, NY, USA: ACM, 2007, pp. 281–288.

[6] O. Goksel and S. Salcudean, "B-mode ultrasound image simulation in deformable 3-D medium," *IEEE Transactions on Medical Imaging*, vol. 28, no. 11, pp. 1657 –1669, Nov. 2009.

[7] D. Terzopoulos and K. Waters, "Physically-based facial modeling, analysis, and animation," *The Journal of Visualization and Computer Animation*, vol. 1, no. 2, pp. 73–80, 1990.

[8] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[9] A. Nealen, M. Mueller, R. Keiser, E. Boxerman, and M. Carlson, "Physically Based Deformable Models in Computer Graphics," *Computer Graphics Forum*, vol. 25, no. 4, pp. 809–836, Dec. 2006.