Fast Volume Preservation for a Mass-Spring System

Min Hong
Soonchunhyang University, South Korea

Sunhwa Jung, Min-Hyung Choi, and Samuel W.J. Welch
University of Colorado at Denver and Health Sciences Center

This article presents a new method to model fast volume preservation of a mass-spring system to achieve a realistic and efficient deformable object animation, without using internal volumetric meshing. With this method the simulated behavior is comparable to a finite-element-method-based model at a fraction of the computational cost.

Many modern interactive graphics applications such as computer games and surgical simulators require both reasonably high mesh resolution and higher performance to achieve a smooth and responsive interactivity and real-time animation of deformable objects. In addition, enhanced realism of physical simulation is called for to better represent real-world deformable materials such as human tissue. It's still a challenge to resolve the conflicting goals of higher physical and dynamic realism on a densely meshed structure and at the same time achieve faster computation.

Researchers have extensively studied modeling and simulating deformable objects in the past decade. Over time, this has led to two main threads:

- continuum mechanics-based techniques for an accurate simulation and
- a discrete mass-spring damper system for a responsive and efficient model.

While the continuum mechanics-based finite element method (FEM) or boundary element method (BEM) can deliver sophisticated and physically accurate analysis, usually their computational complexity is prohibitively expensive for an interactive application.

With this in mind, we decided to pursue an enhanced mass-spring model that incorporates the object level of volume preservation into a mass-spring model without a significant computational cost (for other approaches, see the sidebar "Related Work" on the next page). Contrary to the traditional continuum mechanics-based methods, our method provides a much faster real-time volume-preserving deformation, yet the quality of simulation in dynamic behavior is comparable to an FEM-based model.

Our method uses only the surface mesh and keeps the material properties intact. Our new volume-preserving constraint is well maintained within a tight error rate and example animations demonstrate behavior that's almost identical to a linear FEM-based simulation.

We can readily apply our proposed method to any existing meshed deformable model regardless of the geometric structure because the constraint doesn't affect the material properties or the topology of element connectivity. We use a deformation zone to efficiently control the distribution of volume-preserving forces for representing the local deformation of objects depending on the material properties and the duration and magnitude of external loading. Figure 1 shows the result of muscle deformations when flexing the forearm at the elbow joint. We plausibly simulate active muscle bulging and passive volume preservation incurred by

1 Muscle deformation using a volume-preserving mass-spring system.

 0272-1716/06/$20.00 © 2006 IEEE

Related Work

As fast algorithms for animating deformable objects have been introduced, others have tried multiple approaches to achieve realistic and efficient deformable object animation. Debunne et al.[1] proposed a space- and time-adaptive method according to the most interesting region for real-time deformation. Müller and Gross[2] introduced a fast method that uses the warped stiffness approach to represent some types of material properties using the finite element method (FEM). However, these methods require additional storage for precomputation and are restricted for relatively low-resolution objects.

Promayon et al.[3] used the divergence theorem to compute and maintain a constant volume using the projection method, but they require solving a third-order equation to maintain the constant volume. It might not guarantee the physically correct behavior of an object similar to the post-stabilization method,[4] because finding the projection vector to minimize the constraint drift is performed independently from the object's conforming dynamic motion. Moreover, because of the uniform distribution of the volume preservation, this method only provides the global deformation of the object and can't achieve the local deformation.

Pauly et al.[5] used a surface integral formulation to constrain deformations to be a constant volume. However, they worked with contact problems using the Boussinesq approximation from linear elasticity. These solutions assume small deformations; hence, they're appropriate for stiff bodies in contact, but they're not suitable for large deformation.

Hirota et al.[6] used a 2D projection method to calculate the volume of a free-form deformation object, but they require small meshing resolution to guarantee the accuracy of the volume calculation. They employ a computationally expensive numerical minimization approach with a penalty term that might cause numerical instability.

Zordan et al.[7] applied a volume constraint method based on the object's pressure. They calculated the object's total volume by summing the volume of a set of the pyramidal elements that consists of each surface triangle and the center of mass. However, their method is inapplicable to general geometric structures, such as torus-like objects, and users have to provide the arbitrary volumetric modulus to enforce the volume preservation.

In theory, the collective enforcement of level volume preservation on every element in an object well provides local and global deformation of objects. However, it's computationally expensive and can cause singularities on shared nodes when multiple constraints try to move the shared nodes to different directions to preserve the volume of each element.

Volume preservation has been used to represent human muscle dynamics. Nedel and Thalmann[8] modeled a mass-spring muscle with additional angular springs to preserve the muscle shape, but their approach didn't include a correct volume calculation and therefore volume preservation wasn't accurate. In addition, the newly added springs could easily affect the material properties, often resulting in a stiffer model.

Aubel and Thalmann[9] presented an ad-hoc, real-time, and realistic modeling tool for building human anatomy with three layers of muscle structure. But it didn't include volume preservation for muscle deformation.

Bourguignon and Cani[10] used additional artificial springs to the barycenter of tetrahedral elements to maintain an element level volume. But their method requires stiff volume-preserving springs to maintain the constraint, and that could often result in numerical instability. In addition, adding stiff volume-preserving springs in every element could change the object's material properties. For example, the large stiffness coefficient in the added volume-preserving springs to prevent the excessive deformation increases the object's hardness. It might help to reduce the excessive volume changes but it doesn't guarantee constant object volume during deformation. Furthermore, none of the previously mentioned mass-spring-based volume preservation methods includes the nonlinear local deformation that propagates the constant-volume characteristics only to the neighboring structures.

The hybrid method[11] and adaptive meshing technique[12] have been proposed to accelerate simulations, but these methods aren't fast enough to achieve real-time interactive simulation, especially for complex deformable objects that require high resolution. Preprocessing methods[13,14] have been extensively researched to overcome computational obstacles but they require additional storage that depends on a predefined sampling space and the interactive region of simulation is limited by the presampled space. On the other

skin-to-skin contact (created in the border of the upper and lower arm).

Volume preservation

One reason we need volume preservation is to accurately model various human tissue characteristics, including the muscle deformation shown in Figure 1. Approximately 70 percent of the human body is based on water and the overall volume is well maintained even during a large deformation.

Volume preservation helps us accurately model the detailed motion of a muscle, skin-to-skin contact, and the creases caused by joint folding. The challenge is to model the object-level incompressibility using only the compressible springs and to embed the volume preservation constraint without changing any material properties of the original mass-spring model. It should deform freely while maintaining overall volume.

Another important objective is to use only the surface mesh to model the volume preservation, because most graphical models only employ hollow surface mesh without volumetric connectivity.

Mass-spring system

The mass-spring system is a simple physical model where the discrete concentrated mass points are connected by springs to propagate the energy. Modeling deformable structures requires an appropriate meshing scheme to represent specific physical and material characteristics. We use standard structure and sheer springs

hand, although the mass-spring system is simple and efficient, it's difficult to accurately model the behavior of deformable objects because the elasticity is modeled only with compressible springs.

Unlike fluid simulation using a point cloud with density, or an FEM-based elastic model using a volume penalty method, the mass-spring system doesn't include a mechanism to represent the volume either at the object or element level.

Chen and Zeltzer[15] applied a biomedical model of human muscle using FEM. They applied the tension-length curves to represent the muscle forces for human biceps simulation. Inexact volume preservation for human muscle deformation has also been studied for visually plausible deformation models using anatomy-based modeling.

Scheepers et al.[16] introduced the anatomy-based procedural human musculature model to define and manipulate articulated models. They analyzed the human body using artistic anatomy and focused on skin deformation. Although their work successfully described the anatomical structure and function of bones and muscles, each muscle was divided into elliptical multibelly muscles to preserve the volume and bulge of the muscle. But this method can only represent the muscle deformation using a major axes ratio (height and width).

Dong et al.[17] also introduced an anatomy-based approach to represent realistic muscle simulation using geometric modeling, deformation, and texture. This method enforces the volume preservation in a nonphysical way to reduce the problem's complexity. It's based on a large error tolerance and imprecise volume calculation.

Our proposed method guarantees volume preservation with a tight error rate (0.01 percent) and physically correct behavior at virtually no extra cost to the existing mass-spring system.

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. Müller and M. Gross, "Interactive Virtual Materials," *Proc. Graphics Interface*, A.K. Peters, 2004, pp. 239-246.
3. E. Promayon, P. Baconnier, and C. Puech, "Physically Based Deformations Constrained in Displacements and Volume," *Proc. Eurographics*, Eurographics Assoc., 1997, pp. 155-164.
4. M.B. Cline and D.K. Pai, "Post-Stabilization for Rigid Body Simulation with Contact and Constraints," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE CS Press, vol. 3, 2003, pp. 3744-3751.
5. M. Pauly, D.K. Pai, and L.J. Guibas, "Quasi-Rigid Objects in Contact," *Proc. Eurographics/ACM Siggraph Symp. Computer Animation*, Eurographics Assoc., 2004, pp. 109-119.
6. G. Hirota, R. Maheshwari, and M. Lin, "Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization," *Proc. ACM Solid Modeling*, ACM Press, 1999, pp. 234-245.
7. V. Zordan et al., "Breathe Easy: Model and Control of Simulated Respiration for Animation," *Proc. Eurographics/ACM Siggraph Symp. Computer Animation*, Eurographics Assoc., 2004, pp. 29-37.
8. L.P. Nedel and D. Thalmann, "Real-Time Muscle Deformations Using Mass-Spring Systems," *Proc. Computer Graphics Int'l*, IEEE Press, 1998, pp. 156-165.
9. A. Aubel and D. Thalmann, "MuscleBuilder: A Modeling Tool for Human Anatomy," *J. Computer Science and Technology*, vol. 19, no. 5, 2004, pp. 585-595.
10. D. Bourguignon and M. Cani, "Controlling Anisotropy in Mass-Spring Systems," *Proc. Computer Animation and Simulation*, Springer Computer Science, 2000, pp. 113-123.
11. S. Cotin, H. Delingette, and N. Ayache, "A Hybrid Elastic Model for Real-Time Cutting, Deformations, and Force Feedback for Surgery Training and Simulation," *The Visual Computer*, vol.16, no. 7, 2000, pp. 437-452.
12. E. Grinspun, P. Krysl, and P. Schröder, "CHARMS: A Simple Framework for Adaptive Simulation," *ACM Trans. Graphics*, vol. 21, no. 3, 2002, pp. 281-290.
13. M. Bro-Nielsen, "Surgery Simulation Using Fast Finite Elements," *Proc. Visualization in Biomedical Computing*, Springer-Verlag, 1996, pp. 529-534.
14. D.L. James and D.K. Pai, "DyRT: Dynamic Response Textures for Real Time Deformation Simulation with Graphics Hardware," *ACM Trans. Graphics*, vol. 21, no. 3, 2002, pp. 582-585.
15. D.T. Chen and D. Zeltzer, "Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method," *Computer Graphics*, vol. 26, no. 2, 1992, pp. 89-98.
16. F. Scheepers et al., "Anatomy-Based Modeling of the Human Musculature," *Proc. Siggraph*, ACM Press, 1997, pp. 163-172.
17. F. Dong et al., "An Anatomy-Based Approach to Human Muscle Modeling and Deformation," *IEEE Trans. Visualization and Computer Graphics*, vol. 8, no. 2, 2002, pp. 154-170.

to represent the surface of deformable objects similar to Choi and Ko.[1] Curvature-preserving characteristics are also modeled with bending springs over a curved edge. The structure springs connect direct neighbor nodes to model the objects' elastic properties. The sheer springs provide the resistance for object sheering and their connected diagonal nodes. The bending springs that connect the secondary neighbor nodes define the bending and flexural properties of the material. They also help maintain the object's rest-state shape. The bending springs preserve the surface's curvature.

The mass-spring system only defines the connectivity between neighboring nodes and their relative displacement as a response to the external load without any information about the volume at the element and object levels. It represents the elasticity by using the compression and decompression of the length of a spring and the effects are propagated through the connected mesh network. Thus, an object's volume preservation is difficult to achieve in a conventional mass-spring system.

To overcome the inherent drawback of volume loss of a mass-spring system, we propose the fast volume preservation method. Instead of preserving the local volume of every element (typically a tetrahedron), our volume-preservation method maintains the global volume of a closed mesh structure of deformable objects using only surface mesh. Using only the surface mesh has some important advantages. We can add volume preservation to any existing surface mesh object with minimal modification. Adding a volumetric internal

mesh not only requires an additional lengthy remeshing procedure, but it could alter the material properties and might cause an element inversion problem.

Finding the proper coefficients for the internal mesh is not trivial. The problem size is much smaller (with only the surface mesh) than a volumetric mesh, and therefore it has a clear performance advantage. The task of our volume preservation is nicely divided into computing an overall object volume from a set of surface nodes and triangles, and enforcing the constant volume constraint at every time step of a mass-spring simulation. A deformable object's surface should be closed to provide the object's complete boundary, but there's no restriction on the convexity or topological configuration. We use the divergence theorem to represent the relationship between a triple integral over a deformable object's volume and a surface integral over the object's surface:

$$\iint \mathbf{F} \cdot \mathbf{N}\, dS = \iiint \mathrm{div}\mathbf{F}\, dV \tag{1}$$

In Equation 1, $\mathbf{F}$ is an object's vector field and $\mathbf{N}$ is a surface triangle unit normal vector. The total volume of a deformable object is estimated by summing the object's surface triangles. Assume that the vector $\mathbf{F}$ is the position vector $\mathbf{r}$, thus $3V = \iint \mathbf{r} \cdot \mathbf{N}\, dS$ can be computed by following Equation 2. We represent the surface by flat triangular patches with coordinates (x, y, z) varying linearly on these patches. It's convenient to introduce the area coordinates as L_1, L_2, and L_3 and express the surface integral as shown in Lapidus and Pinder:[2]

$$3\mathbf{V} = \int \sum_{i=1}^{3} \left(x_i L_i N_x + y_i L_i N_y + z_i L_i N_z \right) dA \tag{2}$$

Note that the unit normal vector is constant on the triangular surface patch. The integral is easily evaluated[2] using Equation 3 for integrating polynomials in L_1.

$$\int L_1^{a_1} L_2^{a_2} L_3^{a_3} dxdy = 2A \frac{a_1!a_2!a_3!}{(a_1 + a_2 + a_3 + 2)!} \tag{3}$$

Here a_1, a_2, and a_3 are nonnegative integers, and A is the area of a triangle. We have the three cases: a_1, a_2, and a_3 are $a_1=1$ $a_2=a_3=0$, $a_2=1$ $a_1=a_3=0$, and $a_3=1$ $a_1=a_2=0$. Equation 4 gives us the three integrals we need:

$$\int L_1 dxdy = \int L_2 dxdy = \int L_3 dxdy = \frac{A}{3} \tag{4}$$

Therefore, we can obtain the total volume of object V in Equation 5:

$$V = \frac{1}{3} \sum_{i=1}^{nt} \frac{A_i}{3} \left\{ \begin{array}{l} N_x (x_1 + x_2 + x_3) + N_y (y_1 + y_2 + y_3) \\ + N_z (z_1 + z_2 + z_3) \end{array} \right\} \tag{5}$$

Here i is the volume contribution of surface triangle i and nt is the number of surface triangles. This volume must remain a constant over the entire simulation, so we cast this condition into a constrained dynamic system. The constraint formulation using Lagrange multipliers results in a mixed system of ordinary differential equations (ODEs) and algebraic expressions. We write this system of equations (see Equation 6) using $3n$ generalized coordinates, $\mathbf{q}$, where n is the number of discrete masses, and the generalized coordinates are simply the Cartesian coordinates of the discrete masses:

$$\mathbf{q} = [x_1 y_1 z_1 x_2 y_2 z_2 \ldots x_n y_n z_n]^T \tag{6}$$

Because we only use one volume constraint for the volume-preserving mass-spring system, the constraint system is simple. However, we'll also describe the general formulation of a constraint equation for a multiconstraint dynamic system. Let $\Phi(\mathbf{q}, t)$ be the constraint vector made up of m components each representing an algebraic constraint. Equation 7 mathematically represents the constraint vector as

$$\Phi(\mathbf{q},t) = [\Phi^1(\mathbf{q},t)\ \Phi^2(\mathbf{q},t) \ldots \Phi^m(\mathbf{q},t)]^T \tag{7}$$

where the Φ^i are the individual scalar algebraic constraint equations and m is the number of constraints. To preserve the object's volume, the difference between V_0 (original volume) and V (current volume)—which can be calculated by Equation 5—should be 0 during the simulation, as shown in Equation 8:

$$\Phi(\mathbf{q},t) = V - V_0 = 0 \tag{8}$$

We applied the implicit constraint method[3] to maintain the volume constraint. The constraint force is added into the momentum equation to estimate the new position as follows in Equations 9 and 10:

$$\dot{\mathbf{q}}(t + \Delta t) = \dot{\mathbf{q}}(t) - \Delta t \boldsymbol{M}^{-1} \Phi_{\mathbf{q}}^{\mathbf{T}}(\mathbf{q},t) \lambda + \Delta t \boldsymbol{M}^{-1} \mathbf{F}^{\mathbf{A}}(\mathbf{q},t) \tag{9}$$

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \Delta t \dot{\mathbf{q}}(t + \Delta t) \tag{10}$$

Here $\mathbf{F}^{\mathbf{A}}$ are applied as gravitational and spring forces acting on the discrete masses, $\boldsymbol{M}$ is a $3n \times 3n$ diagonal matrix containing discrete nodal masses, λ is an $m \times 1$ vector containing the Lagrange multipliers, and $\Phi_{\mathbf{q}} = \partial V / \partial q$ is the $m \times 3n$ Jacobian matrix. For more information on how to calculate this, please see the "Additional Information" sidebar. To be treated implicitly, the constraint equations should be calculated at $t + \Delta t$:

$$\Phi(\mathbf{q}(t + \Delta t), t + \Delta t) = 0 \tag{11}$$

Equation 11 is now approximated using a truncated first-order Taylor series in Equation 12:

$$\Phi(\mathbf{q}, t) + \Phi_{\boldsymbol{q}}(\mathbf{q}, t)(\mathbf{q}(t + \Delta t) - \mathbf{q}(t)) + \Phi_{\boldsymbol{t}}(\mathbf{q}, t)\Delta t = 0 \tag{12}$$

Here the subscripts q and t indicate partial differentiation with respect to q and t, respectively. Substituting $\dot{\mathbf{q}}(t + \Delta t)$ from Equation 9 into 10 we obtain

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \Delta t \dot{\mathbf{q}}(t) + \Delta t \left\{ \Delta t \boldsymbol{M}^{-1} \mathbf{F}^{\mathbf{A}}(\mathbf{q},t) - \Delta t \boldsymbol{M}^{-1} \Phi_{\mathbf{q}}^{\mathbf{T}}(\mathbf{q},t) \lambda \right\} \tag{13}$$

Substituting this result from Equation 13 into Equation 12 thus eliminates results in Equation 14 with the remaining unknown:

$$\Phi_{\mathbf{q}}(\mathbf{q},t)\boldsymbol{M}^{-1}\Phi_{\mathbf{q}}^{\mathbf{T}}(\mathbf{q},t)\lambda=\frac{1}{\Delta t^2}\Phi(\mathbf{q},t)+\frac{1}{\Delta t}\Phi_t(\mathbf{q},t)+\Phi_{\mathbf{q}}(\mathbf{q},t)\left(\frac{1}{\Delta t}\dot{\mathbf{q}}(t)+\boldsymbol{M}^{-1}\mathbf{F}^{\mathbf{A}}(\mathbf{q},t)\right) \quad (14)$$

If the only constraint is the volume-preservation constraint, then in Equation 14

$$\Phi_{\mathbf{q}}(\mathbf{q},t)\mathrm{M}^{-1}\Phi_{\mathbf{q}}^{\mathbf{T}}(\mathbf{q},t)$$

and

$$\frac{1}{\Delta t^2}\Phi(\mathbf{q},t)+\frac{1}{\Delta t}\Phi_t(\mathbf{q},t)+\Phi_{\mathbf{q}}(\mathbf{q},t)\left(\frac{1}{\Delta t}\dot{\mathbf{q}}(t)+\boldsymbol{M}^{-1}\mathbf{F}^{\mathbf{A}}(\mathbf{q},t)\right)$$

are scalars. Therefore, we can calculate λ by simple division and no linear system needs to be solved.

Local deformation

Simply maintaining an overall volume can produce undesirable global deformation because the volume-preserving condition is evenly distributed over the surface, showing a balloon-like behavior. This global volume preservation might represent homogeneous material well, but it's unsuitable for material with complex properties such as a combination of anisotropy, viscosity, and nonlinear deformation. For example, when we push the deformable object with a finger in Figure 2, we can deform the object three different ways: compressible, global, and local deformation.

To provide fast but visually realistic animations of local deformation, we used a weight vector redistribution scheme. This scheme provides a flexible approach to obtain visual reality and isn't intended to enforce differential incompressibility. We implemented a weight vector on the surface node to distribute the effects of the volume preservation. The weight vector $[w_1, w_2, w_3, \ldots w_n]$ exists for an individual object and W_i, the weight of node i, is assigned to every surface node and multiplied into the constraint terms in the momentum equations to scale the constraint forces. The sum of the weights is kept equal to the number of surface nodes. For example, in a case where the weight of each node is 1, all the nodes in the object respond to the change of volume equally and this represents global object deformation. If a node has zero weight, then the volume preservation constraint has no effect on the particular node.

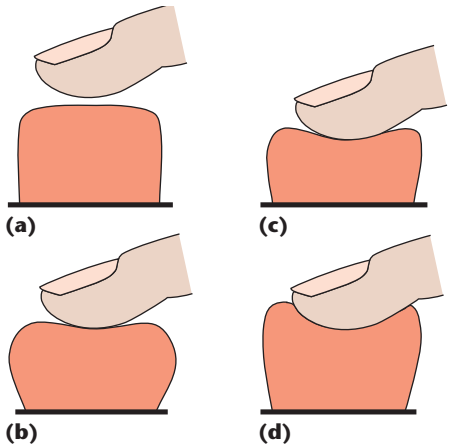
Figure 3 shows different muscle deformations with given weight vector distributions (shown in the middle of the figure). Figure 3a shows the initial state of three

Additional Information

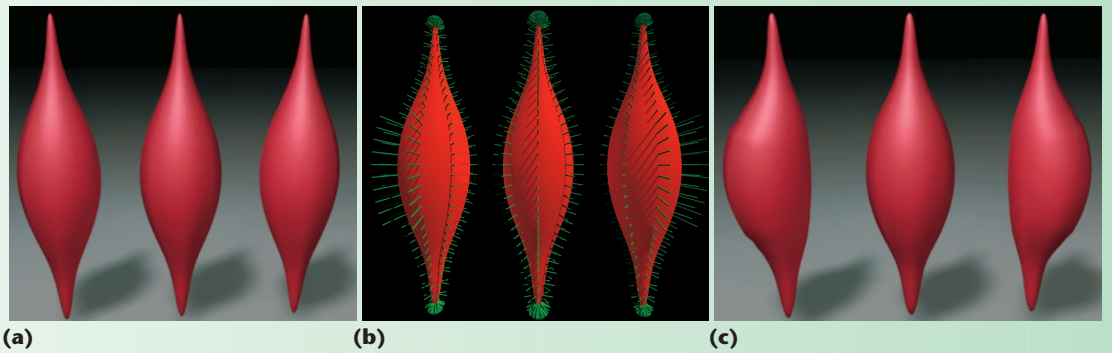
If the triangle surface is defined with three nodes, (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), the formulas for the Jacobian $\partial V/\partial q$ are calculated by using the equations noted in Figure A:

$$\frac{\partial V}{\partial x_1}=\frac{1}{2}(y_3z_2-y_2z_3) \quad \frac{\partial V}{\partial y_1}=\frac{1}{2}(-x_3z_2+x_2z_3) \quad \frac{\partial V}{\partial z_1}=\frac{1}{2}(x_3y_2-x_2y_3)$$
$$\frac{\partial V}{\partial x_2}=\frac{1}{2}(-y_3z_1+y_1z_3) \quad \frac{\partial V}{\partial y_2}=\frac{1}{2}(x_3z_1-x_1z_3) \quad \frac{\partial V}{\partial z_2}=\frac{1}{2}(-x_3y_1+x_1y_3)$$
$$\frac{\partial V}{\partial x_3}=\frac{1}{2}(y_2z_1-y_1z_2) \quad \frac{\partial V}{\partial y_3}=\frac{1}{2}(-x_2z_1+x_1z_2) \quad \frac{\partial V}{\partial z_3}=\frac{1}{2}(x_2y_1-x_1y_2)$$

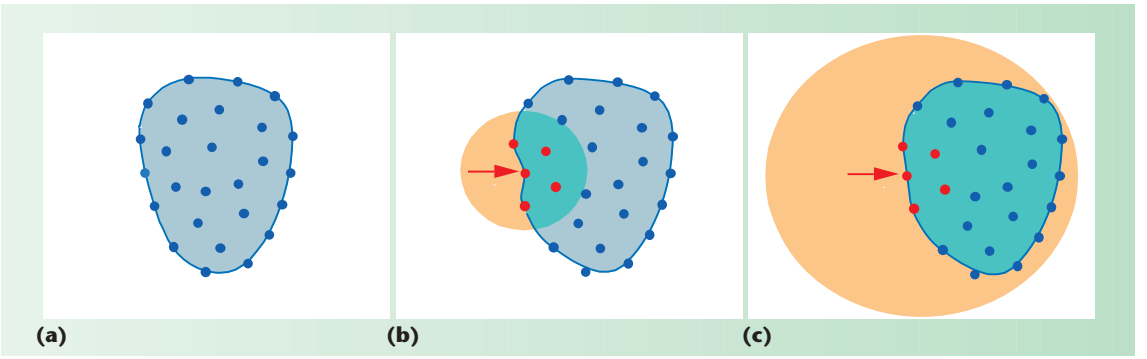
A Equations used to calculate the formulas for the Jacobian $\partial V/\partial q$.



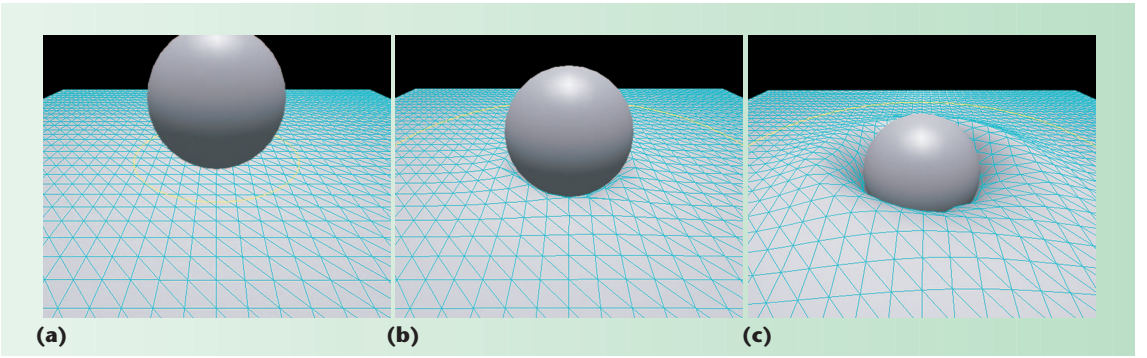
2 (a) Pressing the deformable object with a finger leads to three different deformation types: (b) compressible, (c) global, and (d) local deformation.



3 Weight vector distribution for muscle segments. (a) The initial state of three muscles, (b) weight vectors to represent asymmetric bulging and contracting for three muscles, and (c) results after the simulation.



4 Local and global deformation zones. (a) Initial configuration of an object, (b) local deformation, and (c) global deformation.



5 Redistributing the volume preservation constraint force over time simulates the local and nonlinear deformation.

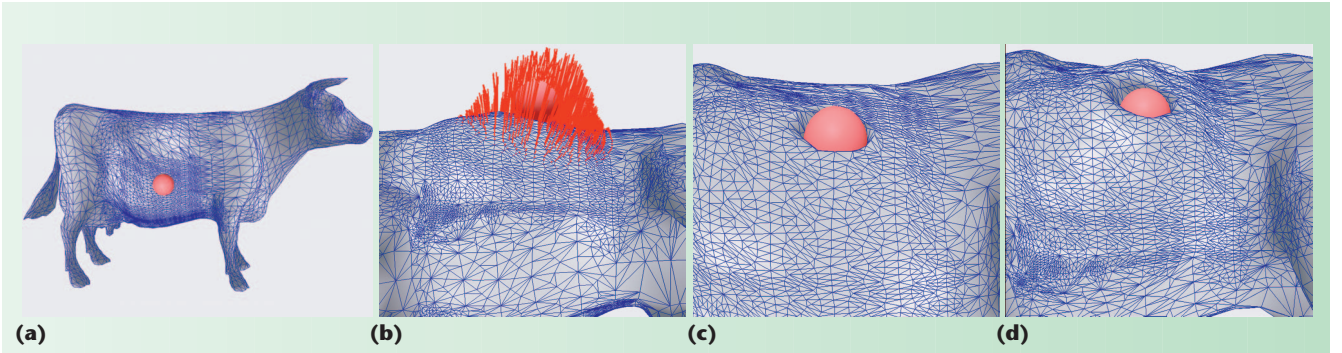
muscles. Figure 3b shows three muscle segments with different weight vectors (where the weight vector is redistributed more on the left side, evenly on the entire surface, and more on the right side of the muscle, respectively) to demonstrate the asymmetric bulging and contracting behavior of the muscle models in Figure 3c.

The material properties and characteristics of the external load cause local and nonlinear deformation. To model the local deformation, the weight vector distribution zone is automatically created based on the direction, duration, and amount of external force as well as the target material's various material properties. The deformation zone with a user-specified threshold value determines the scope and propagation pattern of the weight vector redistribution when the external load is applied to the object. This threshold value defines the radius of the deformation zone, which proportionally increases by the increment of duration and amount of external forces. Figure 4 illustrates the effects of the different deformation zones. The amount and duration of external force mainly determines the zone's radius. The nodes inside the zone are affected by the weight vectors while the excluded nodes remain intact.

In Equation 15, the volume preservation constraint force is redistributed among the nodes within the deformation zone using a weight distribution function. Here

$$W_i = \left(\cos\left(\frac{\|p_i - c\|}{r} \right) + 1 \right) / 2 \qquad (15)$$

where p_i is the position of node i, c is the position where the external force is loaded, and r is the radius for the deformation zone. This r is defined as being equally proportional to the duration and the amount of external force. The weight distribution function also guarantees a smooth



6 These pictures show the interactions between the deformable cow and a spherical probe. (a) Deformable cow model. (b) Weight vector for local deformation. (c) Global deformation. (d) Local deformation. Note that in (c), the volume-preserving condition is evenly distributed for the entire cow model so that the effect of volume preservation is hardly noticeable. In (d), the picture shows apparent local bulging from a fast and strong contact from a spherical probe.

transition from the affected area to the nonaffected area. To avoid artificial energy loss or addition from the application of the weight distribution function, all final weight vectors are normalized to make the sum of weight equal to n. Figure 5 illustrates the diffusion of deformation according to the external load's duration. The deformation zone's radius (the yellow circle) is increased due to the duration of the external load. During this simulation, the percentage of relative volume errors stays within 0.01 percent.

Because the Lagrange multiplier is multiplied by the weight vector, some errors will occur in the volume constraint. Our extensive empirical studies show that the effect is small enough that we gain the benefit of control of local bulging at a negligible computational cost with a small constraint error. To confirm the accuracy of the proposed local deformation method, we tested our method using a high-resolution (14,136 triangles) deformable cow model (see Figure 6a). The red lines demonstrate the weight vector of local deformation in Figure 6b when the spherical probe contacts and pushes the abdomen of the deformable cow. This example shows the distinct deformations between the global (see Figure 6c) and local deformation with volume preservation. The local bulging of the contact area shown in Figure 6d and the relative volume error of local deformation for the complex cow model is also less than 0.01 percent.

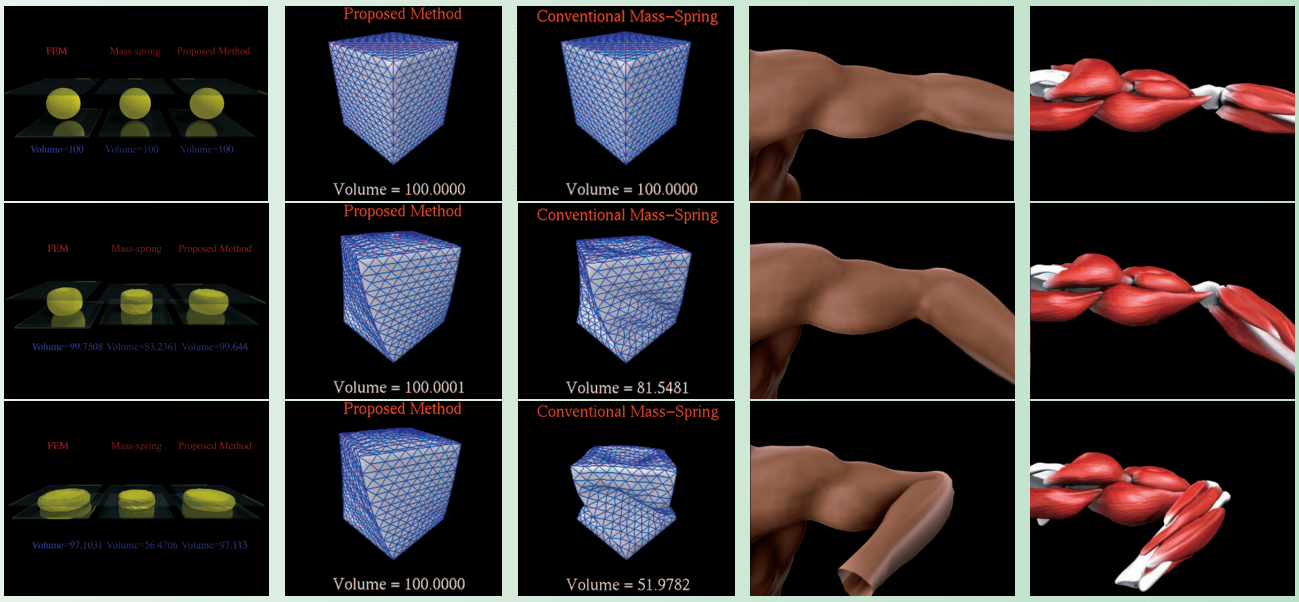
Table 1. Performance comparison of the mass-spring with the proposed volume preservation, conventional mass-spring system, and linear finite element method.

Animation	No. of Elements (Triangles/ Tetrahedrons)	CPU Seconds per Frame		
		Mass-Spring with Volume Preservation	Conventional Mass-Spring	Linear Finite Element Method
Coarse model	2,004/2,691	0.0016	0.0013	0.22
Intermediate model	1,766/3,528	0.0039	0.0036	0.63
Refined model	9,800/52,633	0.013	0.0094	2.59

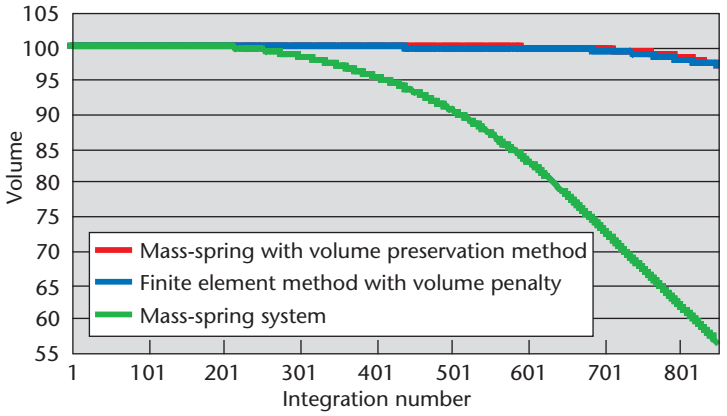
Implementation and results

We conducted a series of comparative simulations of a simple deformable ball squeezed by two planes at three different mesh resolutions. All tests were performed on a 2.0-gigahertz Intel Pentium 4 single processor with 768 Mbytes of memory. Table 1 shows the performance summary of a ball-squeezing simulation tested with three methods: the proposed volume-preserving mass-spring system, a conventional mass-spring system, and a linear FEM using the element level volume penalty as shown in Figure 7a. The proposed volume preservation method requires almost the same computational cost as the conventional mass-spring system but the behavior is close to the FEM-based volume penalty method, which increases the material's stiffness by adding artificial elements in the system.

In Table 1, unlike the proposed method and conventional mass-spring system, which use surface triangles, FEM should use tetrahedrons for an entire ball. We used exactly the same number of surface triangles for com-



7 Snapshots of example animations. (a) Ball-squeezing simulation using a linear finite element method, a conventional mass-spring system, and a proposed incompressible mass-spring system. (b) Cube twisting with proposed incompressible mass-spring system. (c) Cube twisting with conventional mass-spring system. (d) Muscle deformation with skin. (e) Muscle deformation without skin.



8 Percentage of volume preservation for the three methods.

Table 2. Performance measurement for example simulations.

Simulation		No. of Triangles	Maximum Constraint Error	Frames per CPU Second
Cube (Twist)		1,536	0.001	83.3
Cube (Local)		19,200	0.01	21.3
Cow		14,136	0.01	20.7
Muscle deformation	Muscle_1	680	0.01	32.34
	Muscle_2	520		
	Muscle_3	760		
	Muscle_4	760		
	Muscle_5	1,064		
	Muscle_6	440		
	Muscle_7	280		
	Muscle_8	840		
	Muscle_9	680		
	Muscle_10	680		
	Muscle_11	660		
	Muscle_12	1,380		

parison. Again, because the proposed method uses only the surface mesh for deformable objects, our method is fast, simple, and free from element inversion[4] and stiffness problems.

Figure 8 shows the volume changes of a squeezed ball in Figure 7a over time in percentage of the original volume. For comparison, it includes the three methods' results. We used a relatively big volume penalty coefficient (the further increase resulted in instability) for the FEM to well preserve the ball's global volume. The proposed method's weight vector was evenly distributed at unity, so the constraint forces that preserve the global volume weren't modified. The conventional mass-spring system couldn't maintain the ball's volume and resulted in quite different compressing behavior. In FEM, choosing an appropriate volume penalty coefficient isn't trivial, because a big volume penalty coefficient could demolish the simulation and the small volume penalty coefficient might not be enough to maintain an object's volume. Figure 8 shows that our method and the FEM produce almost identical results for volume preservation.

Often a large global deformation under a degenerative twisting of an object could be problematic for the linear FEM. Zhuang and Canny[5] applied a fast nonlinear Green strain using preprocessing of the LU factorization of a small number of large matrices, but still it isn't enough to achieve a real-time simulation for complex deformable objects.

In Figures 7b and 7c, we used a cube and tested our method and a conventional mass-spring system similar to Zhuang and Canny.[5] The cube's bottom is fixed on the floor and its top is twisted using a high-amplitude external force. In Figure 7b, constraint forces that preserve the global volume aren't modified. The proposed method successfully simulated the large deformations without any distortions and our simulated animation is close to the result of Zhuang and Canny.[5] Table 2 shows the performance measurement for example simulations used in this article, including a collision handling.

Figures 7d and 7e show the results of muscle deformations with and without skin when flexing the forearm at the elbow joint. All 12 muscle segments are simulated with the volume-preserved mass-spring system and it's done at an interactive rate. This brachial flexion example demonstrates that the proposed method can enhance the realism of local deformation. The deformation zone modifies the Lagrange multiplier to represent the plausible shape change of muscle deformation and to model the anisotropy of human muscle.

Conclusion and discussion

Despite the simplicity and efficiency of a mass-spring system, it includes visually apparent flaws when modeling deformable objects because of the lack of effective volume preservation. We present a fast and robust volume preservation technique using an implicit constraint enforcement scheme and a deformation zone technique to achieve the local volume-preserved deformation by automatically distributing a weight vector.

Because the computational cost of the volume preservation is virtually negligible, we can achieve a realistic animation similar to an FEM at a fraction of the cost. In addition, our approach is independent from the geometric structure of an object and underlying physical model. Our focus is on the fast object level of volume preservation for a mass-spring system and this approach can make a significant impact on many applications such as games, surgical simulation in virtual environments, human tissue modeling, character animation, and other VR-based applications that require real-time interaction.

The current deformation model is a passive simulation where external forces initiate motions. In the future, more studies on the mechanical properties of human tissue and muscle as well as advanced self-actuated muscle deformation schemes are needed to further enhance the visual realism. ■

Acknowledgments

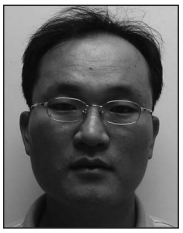
This research was partially supported by National Science Foundation Career Award ACI-0238521.

References

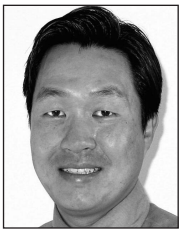
1. K.J. Choi and H.S. Ko, "Stable but Responsive Cloth," *ACM Trans. Graphics*, vol. 21, no. 3, 2002, pp. 604-611.

2. L. Lapidus and G.F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, Wiley, 1982.
3. M. Hong et al., "Effective Constrained Dynamic Simulation Using Implicit Constraint Enforcement," *Proc. 2005 IEEE Int'l Conf. Robotics and Automation*, IEEE Press, 2005, pp. 4531-4536.
4. G. Irving, J. Teran, and R. Fedkiw, "Invertible Finite Elements for Robust Simulation of Large Deformation," *Proc. Eurographics/ACM Siggraph Symp. Computer Animation*, ACM Press, 2004, pp. 131-140.
5. Y. Zhuang and J. Canny, "Real-Time Global Deformations," *Algorithmic and Computational Robotics*, A.K. Peters, 2001, pp. 97-107.

Min Hong *is a full-time lecturer in the Division of Computer Science and Engineering at Soonchunhyang University, South Korea. His research interests include physically based modeling and simulation for medical applications. Hong received an MS in computer science from the University of Colorado at Boulder and a PhD in bioinformatics from the University of Colorado at Denver and Health Sciences Center. Contact him at mhong@sch.ac.kr.*



Sunhwa Jung *is a PhD student in computer science and information systems at the University of Colorado at Denver and Health Sciences Center. Jung received an MS in computer science from the University of Colorado at Denver and Health Sciences Center in 2005. His research interests are in physically based simulation and collision detection and resolution. Contact him at sjung@ouray.cudenver.edu.*



Min-Hyung Choi *is the director of the Computer Graphics and Virtual Environments Laboratory, and an associate professor of computer science and engineering at the University of Colorado at Denver and Health Sciences Center. Choi received his MS and PhD from the University of Iowa. His research interests are in computer graphics, scientific visualization, and human–computer interaction with an emphasis on physically based modeling and simulation for medical and bioinformatics applications. Contact him at min.choi@cudenver.edu.*

Samuel W.J. Welch *is an associate professor of mechanical engineering at the University of Colorado at Denver and Health Sciences Center. He received his MS and PhD in aerospace engineering from the University of Colorado at Boulder. His research interests are in computational fluids and heat transfer and computational dynamics. Contact him at sam.welch@cudenver.edu.*

Article submitted: 28 Oct. 2005; revised: 22 Feb. 2006; accepted: 4 Apr. 2006.