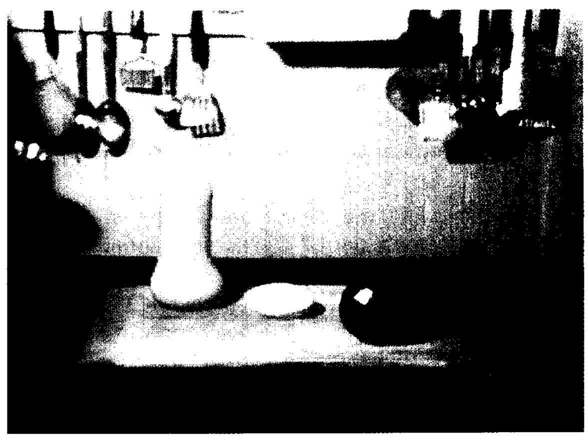
Deformable Models

Physically Based Models with Rigid and Deformable Components

Demetri Terzopoulos and Andrew Witkin
Schlumberger Palo Alto Research

In an earlier work we proposed a class of physically based models suitable for animating flexible objects in simulated physical environments.[1] Our original formulation works well in practice for models whose shapes are moderately to highly deformable, but it tends to become numerically ill conditioned as we increase the rigidity of the models.

This article develops an alternative formulation of deformable models. We decompose deformations into a reference component, which may represent an arbitrary shape, and a displacement component allowing deformation away from this reference shape. The reference component evolves according to the laws of rigid-body dynamics. Equations of nonrigid motion based on linear elasticity govern the dynamics of the displacement component. With nonrigid and rigid dynamics operating in unison, this hybrid formulation yields well-conditioned discrete equations, even for complicated reference shapes, particularly as the rigidity of models is increased beyond the stability limits of our prior formulation. We illustrate the application of our deformable models to a physically based computer animation project.



The animation of graphics objects often requires the coordinated motion of multiple geometric primitives, each involving multiple variables such as position, orientation, and scale. Conventional computer animation is kinematic. To synthesize convincing motions, the animator must specify the variables at each instant in time while satisfying kinematic constraints. A standard scheme for rendering the task less onerous is to spline trajectories automatically through key frames. Often the results are not entirely satisfactory—motions, especially as they increase in complexity, tend to acquire unnatural qualities. In short, creating natural-looking animation kinematically requires patience and expertise.[2]

Dynamic animation goes beyond kinematic animation, offering unsurpassed realism through the use of fundamental physical principles. Users can create realistic motions by applying forces to dynamic, physically based models in simulated physical worlds, while numerical procedures automatically generate time-varying values for the simulation variables in accordance with the laws of Newtonian mechanics.[3-9]

Unlike conventional, purely geometric models, phys-

An earlier version of this article appeared in *Graphics Interface 88*.

0272-1716/88/1100-0041$01.00

ically based models exhibit a naturally animate response to applied forces, as do objects in the real world. Physically based models encourage computer animators to think more like choreographers, who tend to concentrate on abstract qualities of motion (such as timing, rhythm, and style) and remain rather unconcerned with the kinematic details of routines, knowing that physics will dictate the low-level motions of dancers. To "choreograph" physically based models, we control the dynamic simulation through its physical parameters, initial conditions, and applied forces, which may be mediated by constraints imposed on the simulation variables through time.

Deformable models

Physically based simulation is indispensable when animating continuously flexible objects. In our earlier work, we proposed a class of physically based models that describe the shapes and motions of deformable curve, surface, and solid primitives.[1] These primitives simulate "elastic materials" such as string, rubber, cloth, paper, metal, or sponge. Our results demonstrate complex, realistic motions arising from the interaction of deformable models with ambient media and impenetrable obstacles. Attempts to recreate these free-form motions kinematically—that is, without making use of the physical principles underlying the dynamics of nonrigid bodies—would seem contrived and unreasonably tedious.

The deformable models in our earlier study[1] are based on elasticity theory.[10] The (Lagrangian) equations of nonrigid motion are expressed in terms of position functions in Euclidean three space. These functions are parametric in the material (intrinsic) coordinates of the model—they explicitly locate each of its points in space as a function of time. The partial differential equations of motion include a nonlinear elastic force associated with the deformable body. We designed this force to be invariant with respect to rigid-body motion, since such motions impart no deformation. Nonlinearity results because the elastic force attempts to restore the shape of the deformed body to a prescribed undeformed or rest shape. This (generally free-form) shape is defined by as many nonvanishing fundamental tensors as may be necessary to specify it up to a rigid-body transformation (e.g., for a deformable curve, the required tensors reduce to the familiar arc-length, curvature, and torsion functions along the prescribed undeformed curve).

The advantage of nonlinear elasticity is that it is in principle the most accurate way to characterize the behavior of certain elastic phenomena, such as large deformations of shells. However, the nonlinear formulation can lead to serious practical difficulties in the numerical implementation of deformable models for animation. It turns out that the discrete equations involved become increasingly ill conditioned as we try to increase the rigidity of the model or the complexity of the rest shapes. Sophisticated and computationally costly algorithms are needed to integrate ill-conditioned, nonlinear, time-varying partial differential equations robustly.

Decomposition into reference and displacement components

Linear elasticity theory appears attractive in formulating deformable models for computer animation, since it avoids most of the complexities of the nonlinear theory. Interestingly, our nonlinear formulation[1] reduces to a linear model when the rest shape has trivially zero fundamental tensors, i.e., when it is collapsed to a point. This is clearly too restrictive. Another possibility, which we have attempted with limited success, is to linearize the equations and approximate nonlinear effects as explicit, external forces. Unfortunately, the explicit forces tend to degrade the stability of our time-integration algorithms.

In this article, we define for computer animation deformable models that enjoy the benefits of linear elasticity. Rather than being represented explicitly by position functions, the new model incorporates two types of dependent functions: functions that determine a reference configuration for the body in three space, and functions that determine the displacements of material points away from the reference configuration. When necessary, the three-space positions of points can be determined by adding the displacement component to the reference component.

The elastic behavior of the deformable model manifests itself only in the displacement component, which defines the deformation mode of the model. The deformation mode is governed by linear elasticity, and zero displacement implies an arbitrary shape determined by the reference component. But since the reference component represents a prescribed set of reference positions in three space, the position and attitude of the rest shape will remain fixed. For the deformable model to permit a free motion mode in addition to an elastic mode, we allow the reference component to evolve over time according to the laws of rigid-body dynamics.[11]

Thus, we obtain a *hybrid* model that includes both rigid and deformation dynamics. With regard to numerical implementation, this hybrid formulation of deformable models offers an important benefit—it leads to discrete equations that remain well conditioned as we make the model more rigid.

The remainder of this article is organized as follows: The next section describes the geometric representation underlying the hybrid formulation. Then the equations of motion governing the hybrid model and the energy of linear elastic deformation are developed. We describe our numerical solution, and finally we present an application of our deformable models to a physically based animation project.

Geometric representation

Let u be the intrinsic or material coordinates of points in a body Ω. For a solid body, $u=(u_1,u_2,u_3)$ has three coordinates. For a surface $u=(u_1,u_2)$, and for a curve $u=(u_1)$. In the three cases, respectively, and without loss of generality, Ω will be the unit interval $[0,1]$, the unit square $[0,1]^2$, and the unit cube $[0,1]^3$.

The positions of points in the body relative to an inertial frame of reference Φ in Euclidean three space are given by a time-varying vector-valued function of the material coordinates

$$\mathbf{x}(\mathrm{u},t) = [x_1(\mathrm{u},t), x_2(\mathrm{u},t), x_3(\mathrm{u},t)]', \qquad (1)$$

where the prime denotes transposition. We write a three-space vector in bold and its elements in italic.

We represent a deformable body as the sum of a reference component

$$\mathbf{r}(\mathrm{u}) = [r_1(\mathrm{u}), r_2(\mathrm{u}), r_3(\mathrm{u})]' \qquad (2)$$

and a displacement component

$$\mathbf{e}(\mathrm{u},t) = [e_1(\mathrm{u},t), e_2(\mathrm{u},t), e_3(\mathrm{u},t)]'. \qquad (3)$$

Both components can be conveniently expressed in body coordinates; that is, relative to a body frame ϕ (see Figure 1) whose origin coincides with the body's center of mass

$$\mathbf{c}(t) = \int_\Omega \mu(\mathrm{u})\mathbf{x}(\mathrm{u},t)\,d\mathrm{u}, \qquad (4)$$

where $\mu(\mathrm{u})$ is the mass density of the deformable body (Ω is assumed to be the domain of integration for integrals with respect to u and is henceforth suppressed). We denote the positions of mass elements in the body relative to ϕ by

$$\mathbf{q}(\mathrm{u},t) = \mathbf{r}(\mathrm{u}) + \mathbf{e}(\mathrm{u},t). \qquad (5)$$

The (noninertial) body frame ϕ is conveyed along with the body in accordance with the laws of rigid-body dynamics.[11] Define the linear and angular velocities of ϕ relative to Φ as

$$\mathbf{v}(t) = \frac{d\mathbf{c}}{dt}; \qquad \boldsymbol{\omega}(t) = \frac{d\boldsymbol{\theta}}{dt}, \qquad (6)$$

where $d\boldsymbol{\theta}$ is a quantity whose magnitude equals the infinitesimal rotation angle and whose direction is along the instantaneous axis of rotation of ϕ relative to Φ. Then, the velocity of mass elements of the model relative to Φ, given their velocities $\dot{\mathbf{e}}(\mathrm{u},t) = \partial\mathbf{e}/\partial t$ relative to ϕ, is

$$\dot{\mathbf{x}}(\mathrm{u},t) = \mathbf{v}(t) + \boldsymbol{\omega}(t) \times \mathbf{q}(\mathrm{u},t) + \dot{\mathbf{e}}(\mathrm{u},t). \qquad (7)$$

In this article, overstruck dots denote time derivatives d/dt or $\partial/\partial t$, as appropriate.

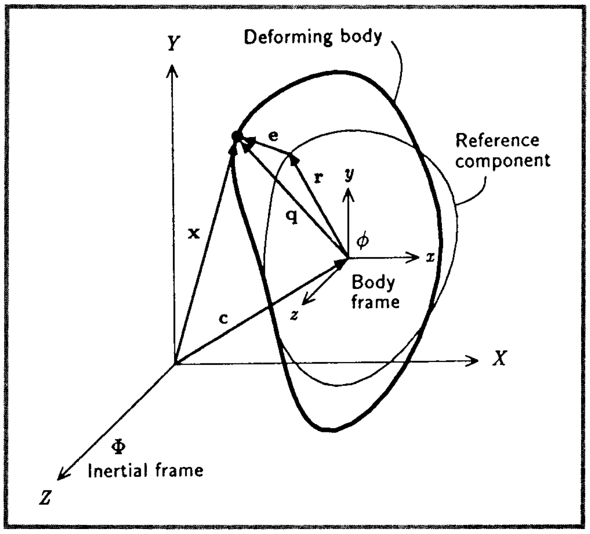


Figure 1. Geometric representation. Shape is decomposed into reference and displacement (or deformation) components.

Equations of motion

A deformable model is described completely by the positions $\mathbf{x}(\mathrm{u},t)$, velocities $\dot{\mathbf{x}}(\mathrm{u},t)$, and accelerations $\ddot{\mathbf{x}}(\mathrm{u},t)$ of its mass elements as functions of material coordinates u and time t. When these functions are expressed in the inertial frame Φ directly from Equation 1 (without making reference to a body frame ϕ as in Equation 7) the Lagrange equation of motion governing $\mathbf{x}(\mathrm{u},t)$ takes on a relatively simple form:[1]

$$\mu\ddot{\mathbf{x}} + \gamma\dot{\mathbf{x}} + \delta_{\mathbf{x}}\mathcal{E} = \mathbf{f}, \qquad (8)$$

where $\mu(\mathrm{u})$ is the mass density, $\gamma(\mathrm{u})$ is the damping density (here a scalar, but generally a matrix), and $\mathbf{f}(\mathbf{x},t)$ represents the net external forces. This is a partial differential equation (because of the dependence of $\delta_{\mathbf{x}}\mathcal{E}$ on $\mathbf{x}$ and its partial derivatives with respect to u—see below). Given appropriate conditions for $\mathbf{x}$ on the boundary of Ω and initial conditions $\mathbf{x}(\mathrm{u},0)$, $\dot{\mathbf{x}}(\mathrm{u},0)$, we have a well-posed initial-boundary-value problem (second order in time and of the hyperbolic-parabolic type).

The external forces $\mathbf{f}$ are dynamically balanced against the force terms intrinsic to the deformable model, which are found on the left-hand side of Equation 8. The first term is the inertial force due to the model's distributed mass as it resists acceleration. The second term is a velocity-dependent (viscous) damping force that dissipates the kinetic energy of the body's mass elements as they move through a viscous ambient medium. The third term is the elastic force due to the deformation of the model away from its natural reference shape.

The elastic force is conveniently expressed as $\delta_{\mathbf{x}}\mathcal{E}$, a variational derivative[12] of a deformation energy $\mathcal{E}(\mathbf{x})$ associated with the model. The nonnegative functional $\mathcal{E}$ measures the potential energy associated with an instantaneous elastic deformation of the body. Its value increases with the magnitude of the deformation.

We used Equation 8 in our earlier study,[1] where it proved workable for models with moderately high flexibilities. However, our experiments with increasingly rigid models show a rapid deterioration of the numerical conditioning of the associated discrete equations: To increase rigidity using this formulation, we must increase the magnitude of nonquadratic terms in $\mathcal{E}$, consequently making Equation 8 more seriously nonlinear.

Instead we apply Equation 7, and reformulate the equation of motion to treat rigid-body motion explicitly. This permits us to use a purely quadratic elastic functional $\mathcal{E}$. The numerical conditioning of this hybrid formulation will improve as the model becomes more rigid, tending in the limit to well-conditioned, rigid-body dynamics, while the capability of modeling nonrigid bodies is retained. However, Equation 8 remains preferable for extremely nonrigid models, such as very stretchy rubber sheets, where the hybrid formulation may yield unrealistic results because of the simple linear force coupling the deformation to a rigid reference shape.

To obtain the equations of motion for the unknown functions $\mathbf{v}$, $\boldsymbol{\omega}$, and $\mathbf{e}$ under the action of an applied force $\mathbf{f}$, we transform the kinetic energy that governs the deformable body, using Lagrangian mechanics. Assuming small deformations, this yields three coupled differential equations:

$$m\dot{\mathbf{v}} + \frac{d}{dt}\int \mu\dot{\mathbf{e}}\,du + \int \gamma\dot{\mathbf{x}}\,du = \mathbf{f}^{\mathbf{v}}, \tag{9a}$$

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) + \frac{d}{dt}\int \mu\mathbf{q}\times\dot{\mathbf{e}}\,du + \int \gamma\mathbf{q}\times\dot{\mathbf{x}}\,du = \mathbf{f}^{\boldsymbol{\omega}}, \tag{9b}$$

$$\mu\ddot{\mathbf{e}} + \mu\dot{\mathbf{v}} + \mu\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\mathbf{q}) + 2\mu\boldsymbol{\omega}\times\dot{\mathbf{e}} + \mu\dot{\boldsymbol{\omega}}\times\mathbf{q} + \gamma\dot{\mathbf{x}} + \delta_{\mathbf{e}}\mathcal{E} = \mathbf{f}. \tag{9c}$$

Here, $m = \int \mu du$ is the total mass of the body, and the inertia tensor $\mathbf{I}$ is a 3×3 symmetric matrix with entries

$$I_{ij} = \int \mu(\delta_{ij}\mathbf{q}^2 - q_i q_j)\,du, \tag{10}$$

where $\mathbf{q} = [q_1, q_2, q_3]'$ and δ_{ij} is the Kronecker delta. The applied force $\mathbf{f}(u,t)$ contributes to elastic deformation, as well as to a net translational force $\mathbf{f}^{\mathbf{v}}(t)$ and net torque $\mathbf{f}^{\omega}(t)$ acting on the center of mass:

$$\mathbf{f}^{\mathbf{v}} = \int \mathbf{f}\,du; \qquad \mathbf{f}^{\boldsymbol{\omega}} = \int \mathbf{q}\times\mathbf{f}\,du. \tag{11}$$

We derive the above system of equations in Appendix A.

Let us examine Equations 9 in detail. Equations 9a and 9b describe $\mathbf{v}$ and $\boldsymbol{\omega}$, the translational and rotational motion of the body's center of mass. Together these ordinary differential equations describe the motion of the body frame ϕ relative to the inertial frame Φ. The partial differential Equation 9c describes, relative to ϕ, the deformation $\mathbf{e}$ of the model from its reference shape $\mathbf{r}$.

The first term of Equation 9a represents the net inertial force experienced by the center of mass due to the total moving mass of the body as if it were concentrated at $\mathbf{c}$. The second term represents an inertial force due to the net displacement motion of mass elements about the reference component $\mathbf{r}$. The third term gives the net damping force of the moving mass elements. An analogous interpretation in terms of inertial torques holds for Equation 9b. The first two terms give the inertial torques resulting from the body's moment of inertia about $\mathbf{c}$ and the net angular momentum from the displacement motion of mass elements, while the third term gives the net damping torque of the elements.

Equation 9c indicates several inertial forces experienced by individual mass elements as the body deforms in Φ. The first term represents the simple inertial force of a mass element. The second term gives the inertial force due to the linear acceleration of the center of mass. The next three terms give the centrifugal force on mass elements due to the rotation of ϕ, the Coriolis force due the velocity of the mass elements in ϕ, and the transverse force on these elements due to the angular acceleration of ϕ. The penultimate term gives the damping force on individual mass elements. In the next section we examine the final term, which represents the elastic force resulting from the deformation of elements away from the reference component.

Elastic deformation

The elastic force due to deformational displacement $\mathbf{e}(u,t)$ away from the reference component $\mathbf{r}(u)$ is represented in Equation 9c by $\delta_{\mathbf{e}}\mathcal{E}$, a variational derivative with respect to $\mathbf{e}$ of an elastic potential energy functional $\mathcal{E}$. The general form of $\mathcal{E}$ is

$$\mathcal{E}(\mathbf{e}) = \int E(u, \mathbf{e}, \mathbf{e_u}, \mathbf{e_{uu}}, \ldots)\,du, \tag{12}$$

an integral over material coordinates of an elastic energy density E, which depends on $\mathbf{e}$ and its partial derivatives with respect to material coordinates.

In our earlier study,[1] the elastic functional for a solid deformable model was of the form $\mathcal{E}(\mathbf{x}) = \int |\mathbf{G} - \mathbf{G}^0|^2 du$, a squared normed difference between the first-order or metric tensors (matrices) $\mathbf{G}(\mathbf{x})$ of the deformed body and $\mathbf{G}^0$ of the undeformed body. Elastic functionals for surface and curve models involve additional squared difference terms of second- and third-order tensors. The collection of tensors associated with the undeformed body describes its shape up to rigid-body motions, and $\mathcal{E}$ quantifies the model's actual deformation away from this rigid shape. Thus the reference component is incorporated into the energy functional that is invariant with respect to rigid-body motion. Such invariance is necessary in the simple equation of motion (Equation 8).

The virtue of the new equations of motion (Equations 9) is that they make fewer demands on $\mathcal{E}$. Because rigid motion is represented explicitly, $\mathcal{E}$ no longer need be invariant with respect to such motion. All that is required is that $\mathcal{E}=0$ when $\mathbf{e}=\mathbf{0}$ and that $\mathcal{E}$ increase monotonically with increasing $\mathbf{e}$, as measured by some reasonable norm.

We use a class of controlled-continuity generalized spline kernels.[13] These splines are given in the form of Equation 12, with the potential energy density defined by

$$E = \frac{1}{2}\sum_{m=0}^{p}\sum_{|j|=m}\frac{m!}{j_1!\ldots j_d!}w_j(\mathrm{u})\left|\partial_j^m \mathbf{e}\right|^2, \tag{13}$$

where $j=(j_1,\ldots,j_d)$ is a multi-index with $|j|=j_1+\ldots+j_d$, where d is the material dimensionality of the model ($d=1$ for curves, $d=2$ for surfaces, and $d=3$ for solids), and where the partial derivative operator

$$\partial_j^m = \frac{\partial^m}{\partial u_1^{j_1}\ldots\partial u_d^{j_d}}. \tag{14}$$

Thus, E is a weighted combination of partial derivatives of $\mathbf{e}$ of all orders up to p. Generally, the smoothness of the allowable deformation increases with increasing p. The weighting functions $w_j(\mathrm{u})$ in Equation 13 control the material properties of the deformable model over the material coordinates.

In the interior of the material domain Ω, the variational derivative of $\mathcal{E}$ with the spline density of Equation 13 is

$$\delta_{\mathbf{e}}\mathcal{E} = \sum_{m=0}^{p}(-1)^m\Delta_{w_m}^m\mathbf{e}, \tag{15}$$

where

$$\Delta_{w_m}^m = \sum_{|j|=m}\frac{m!}{j_1!\ldots j_d!}\partial_j^m\left(w_j(\mathrm{u})\partial_j^m\right) \tag{16}$$

is a spatially weighted, iterated Laplacian operator of order m. The operator is modified at the boundary in accordance with the boundary conditions.[13]

Numerical solution

The equations of motion (Equations 9 through 11 with Equations 15 and 16) are continuous in material coordinates and time. To simulate the deformable model numerically, we discretize the equations using finite-element or finite-difference approximation methods.[14,15] First we discretize with respect to material coordinates to obtain semidiscrete equations of motion, a large system of simultaneous ordinary differential equations.

The second step is to integrate the semidiscrete system through time, thus simulating the dynamics of deformable models. We use a semi-implicit time integration procedure that evolves the elastic displacements and rigid-body dynamics from given initial conditions. In essence, the evolving deformation yields a recursive sequence of (dynamic) equilibrium problems, each requiring the solution of a *sparse* linear system whose dimensionality is proportional to the number of nodes making up the discrete model.

We can use iterative methods to solve these linear systems, as well as direct matrix factorization methods.[16] Because of the linear elastic energy density (Equation 13), the system matrix remains constant; hence a direct-solution method need factorize it only once at the beginning, then simply resolve the vector on the right-hand side at each time step, thus saving significant computation.

Our implementations to date use the second-order ($p=2$) controlled-continuity spline model. Appendix B presents implementation details for the case of surfaces.

Animation examples

To create animation we simulate numerically the differential equations of motion. After each time step (or every few time steps) in the simulation, we render the models' state data to create successive frames of the animation. We have implemented curve, surface, and solid deformable models in two and three dimensions on Symbolics 3600 series Lisp Machines, which provide an excellent prototyping environment. The subsequent sections describe two of our physically based animations.

Flatland

Our Lisp Machines lack the computational power to support real-time interaction with surface and solid models of modest size (having more than about 100 nodes). However, we can interact with simple hybrid models within a 2D world called Flatland. Flatland models are planar deformable curves (displayed as wireframes), capable of rigid-body dynamics and elastodynamics (see Figure 2). The models may be subjected to user forces controlled via a mouse, gravity, impulsive forces due to collisions with obstacles, etc.[1] The simulations illustrated in the figure involve a 50-node discrete model.

Cooking with Kurt

We have used the hybrid formulation of deformable models developed in this article to create the physically based animation "Cooking with Kurt," which intimately combines natural and synthetic images.[17] The action begins with live video of Kurt Fleischer walking into his kitchen and placing several large vegetables on a cutting board. The vegetables "come to life" in what appears to

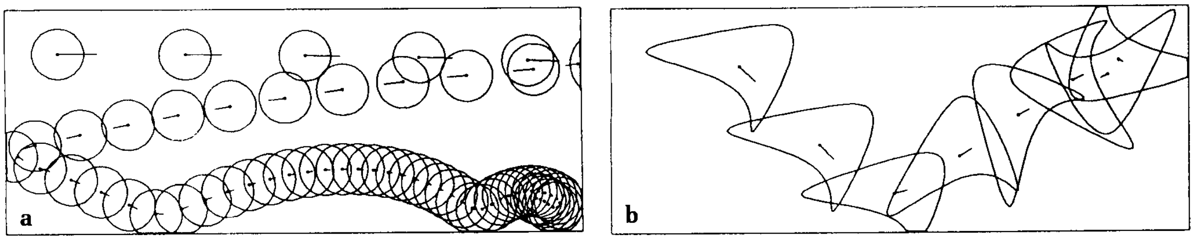


Figure 2. Flatland animations. Models are "strobed" while undergoing motion subject to gravity, aerodynamic drag, and collisions against frictionless walls. Velocity vector of the center of mass (dot) is indicated. Figure 2a shows a deformable hoop; Figure 2b, an arbitrary shape.

be the physical kitchen table environment. They bounce, slide, roll, tumble, and collide with one another and with the tabletop, cutting board, and back wall. Figure 3 shows selected still frames. "Cooking with Kurt" demonstrates that animating physically based models in a simulated physical environment offers a powerful alternative to the common practice of creating animation by key framing and spline interpolation (in-betweening).

Another novel feature of this animation project is the use of recently developed computer vision techniques[18-20] to reconstruct the 3D reference shapes **r** of the deformable vegetable models from a 2D natural gray-level image (the video frame shown in Figure 3b). These techniques exploit the "modeling clay" properties of deformable models. They provide systematic ways of transforming raw image data into synthetic force fields that sculpt deformable models into shapes consistent with the imaged objects.

Given a viewpoint into the 3D space occupied by our synthetic physical world, user-assisted optimization methods served to position three planar surfaces such that they project correctly into the tabletop, cutting board, and back wall visible in the background image (identical to Figure 3b but lacking the vegetables). This approach also served in adjusting surface colors and albedos, and in positioning the synthetic light source so that the rendering of the vegetable models and their shadows (cast on the planar surfaces) is consistent with the image of the scene. Thus we could undetectably excise the real vegetables from the scene and matte in our synthetic animate copies (see Figure 3c). We used our modeling testbed system[21] to render the models into the background image.

Once the shapes of the real vegetables had been captured in the reference components, the models were animated by numerically simulating their discrete equations of motion (their deformable shells are represented by about 500 variables). The animate vegetable models exhibit deformations, linear and angular accelerations, collisions, and other physically realistic motions. The deformation parameters (w_i in Equation 26) were chosen to make the vegetable models appear rather "turgid." The forces that determine the motions include "rocket thruster" driving forces, "servo control" forces for following choreographed paths and maintaining attitude, and interaction forces that arise from friction and collision among the models and planar surfaces in the simulated kitchen table environment.

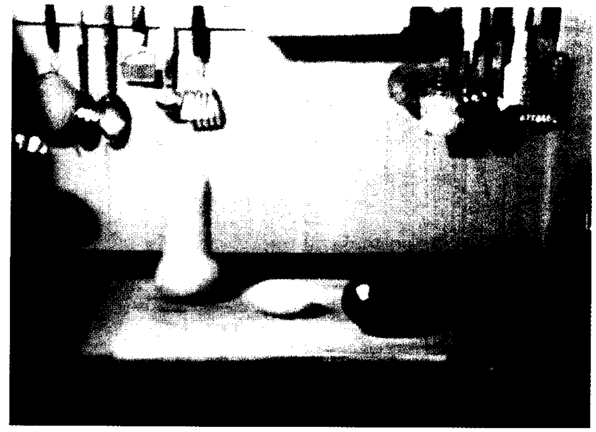
Conclusion

We have proposed novel physically based models for use in computer graphics animation. Our hybrid deformable models unify rigid and nonrigid dynamics. By incorporating a reference component with explicit (six-degree-of-freedom) dynamic equations, we are able to exploit a relatively simple linear theory to model free-form elastic deformations. Reduction in computational effort, ease in animating flexible objects having complicated natural shapes, and good conditioning of the numerical equations with increasing rigidity are among the benefits accrued. Moreover, our hybrid formulation makes it especially convenient to model the inelastic deformations characteristic of modeling clay.[22] The hybrid formulation complements our earlier work in elastically deformable models and it significantly extends our ability to create realistic animations of nonrigid objects in simulated physical environments. ■

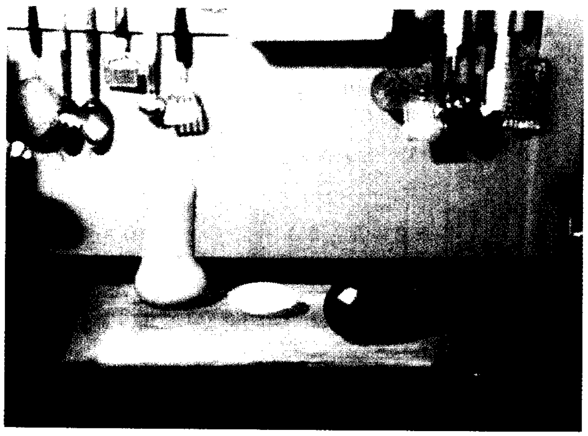
Acknowledgments

We wish to thank our fellow goop scientists: Michael Kass assisted with the derivation in Appendix A. Kurt Fleischer and Michael Kass played major roles in creating "Cooking with Kurt." John Platt made valuable contributions to our earlier work on deformable models.

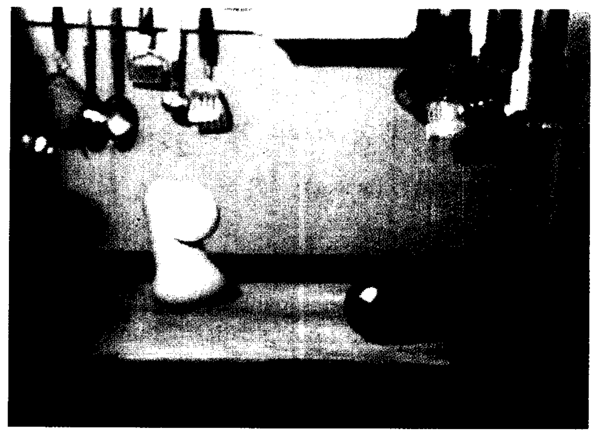
a



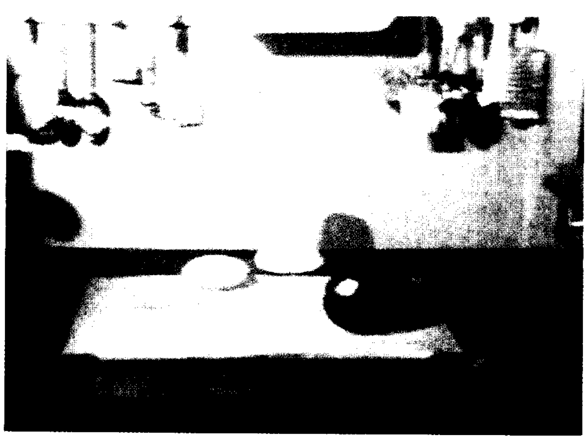
b



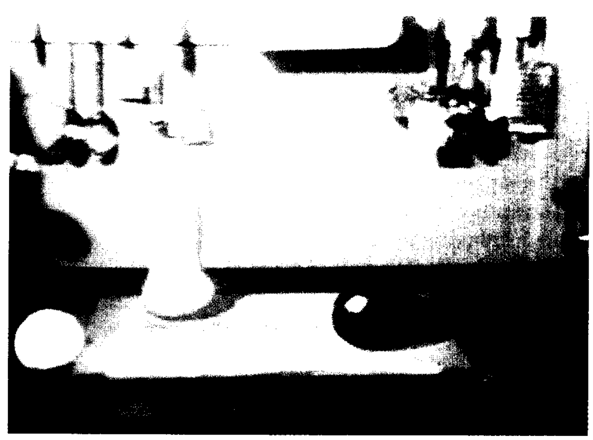
c



d



e



f

Figure 3. Selected still frames from "Cooking with Kurt"[17] showing live action and animation of 3D deformable models in a simulated physical world: (a) Kurt Fleischer with real vegetables, (b) real vegetables on a cutting board, (c) reconstructed deformable vegetable models matted into the background scene, (d) elastic collision, (e) bouncing (note deformed base on large gourd), (f) rolling.

Appendix A: Derivation of the equations of motion

We apply Lagrangian mechanics to the kinetic energy that governs deformable models:

$$\mathcal{T} = \int \frac{\mu}{2}\dot{\mathbf{x}}\cdot\dot{\mathbf{x}}\,du, \tag{17}$$

where $\dot{\mathbf{x}}(u,t)$ is the instantaneous velocity of mass elements. The energy can be rewritten, using Equation 7, in terms of the geometric representation of Figure 1 as

$$\mathcal{T} = \int T\,du = \int \frac{\mu}{2}(\mathbf{v}+\boldsymbol{\omega}\times\mathbf{q}+\dot{\mathbf{e}})\cdot(\mathbf{v}+\boldsymbol{\omega}\times\mathbf{q}+\dot{\mathbf{e}})\,du. \tag{18}$$

Expanding,

$$\mathcal{T} = \sum_{k=1}^{6}\mathcal{T}_k = \sum_{k=1}^{6}\int T_k\,du, \tag{19}$$

where the integrands are

$$\begin{aligned} T_1 &= \frac{\mu}{2}\mathbf{v}\cdot\mathbf{v}; \quad T_2 = \mu\mathbf{v}\cdot\dot{\mathbf{e}}; \quad T_3 = \frac{\mu}{2}\dot{\mathbf{e}}\cdot\dot{\mathbf{e}};\\ T_4 &= \frac{\mu}{2}(\boldsymbol{\omega}\times\mathbf{q})\cdot(\boldsymbol{\omega}\times\mathbf{q}); \quad T_5 = \mu\boldsymbol{\omega}\times\mathbf{q}\cdot\dot{\mathbf{e}};\\ T_6 &= \mu\mathbf{v}\cdot\boldsymbol{\omega}\times\mathbf{q}. \end{aligned} \tag{20}$$

Velocity-dependent energy dissipation may be incorporated in terms of the (Raleigh) dissipation functional

$$\mathcal{F} = \int F\,du = \int \frac{\gamma}{2}\dot{\mathbf{x}}\cdot\dot{\mathbf{x}}\,du, \tag{21}$$

where $\gamma(u)$ is a damping density. Note that Equation 21 has the same form as Equation 19 with γ replacing μ. Hence we express

$$\mathcal{F} = \sum_{k=1}^{6}\mathcal{F}_k = \sum_{k=1}^{6}\int F_k\,du$$

in the representation of Figure 1, where the associated integrands F_1 to F_6 are readily obtained from Equation 20 by replacing μ with γ.

Using the functionals $\mathcal{T}$ and $\mathcal{F}$ and observing that $\mathcal{E}$ does not depend on $\mathbf{v}$ and $\boldsymbol{\omega}$, the equations of motion can be expressed as

$$\begin{aligned} \delta_{\mathbf{c}}\mathcal{T} + \delta_{\mathbf{v}}\mathcal{F} &= \mathbf{f}^{\mathbf{v}},\\ \delta_{\boldsymbol{\theta}}\mathcal{T} + \delta_{\boldsymbol{\omega}}\mathcal{F} &= \mathbf{f}^{\boldsymbol{\omega}},\\ \delta_{\mathbf{e}}\mathcal{T} + \delta_{\dot{\mathbf{e}}}\mathcal{F} + \delta_{\mathbf{e}}\mathcal{E} &= \mathbf{f}, \end{aligned} \tag{22}$$

where the δ operators denote variational derivatives with respect to the subscripted functions. The generalized forces associated with $\mathbf{v}$, $\boldsymbol{\omega}$, and $\mathbf{e}$ are $\mathbf{f}^{\mathbf{v}}$, $\mathbf{f}^{\omega}$, and $\mathbf{f}$.

In view of the time derivatives contained in the functional terms $\mathcal{T}_k$ and $\mathcal{F}_k$, we have

$$\begin{aligned} \delta_{\mathbf{c}}\mathcal{T} + \delta_{\mathbf{v}}\mathcal{F}_k &= \sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\mathbf{v}} + \frac{\partial\mathcal{F}_k}{\partial\mathbf{v}},\\ \delta_{\boldsymbol{\theta}}\mathcal{T} + \delta_{\boldsymbol{\omega}}\mathcal{F}_k &= \sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\boldsymbol{\omega}} + \frac{\partial\mathcal{F}_k}{\partial\boldsymbol{\omega}},\\ \delta_{\mathbf{e}}\mathcal{T} + \delta_{\dot{\mathbf{e}}}\mathcal{F}_k &= \sum_{k=1}^{6}\frac{\partial}{\partial t}\frac{\partial T_k}{\partial\dot{\mathbf{e}}} - \frac{\partial T_k}{\partial\mathbf{e}} + \frac{\partial F_k}{\partial\dot{\mathbf{e}}}. \end{aligned} \tag{23}$$

Now, $\int\mu\mathbf{q}du=\mathbf{0}$, since it is simply the center of mass and lies at the origin of the body frame ϕ; hence $\mathcal{T}_6=T_6=0$. The above sums may be derived term by term:

$$\begin{aligned} \sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\mathbf{v}} &= \frac{d(m\mathbf{v})}{dt} + \frac{d}{dt}\int\mu\dot{\mathbf{e}}\,du + \mathbf{0}+\mathbf{0}+\mathbf{0}+\mathbf{0},\\ \sum_{k=1}^{6}\frac{\partial\mathcal{F}_k}{\partial\mathbf{v}} &= \mathbf{v}\int\gamma\,du + \int\gamma\dot{\mathbf{e}}\,du\mathbf{0}+\mathbf{0}\\ &\quad + \mathbf{0} + \boldsymbol{\omega}\times\int\gamma\mathbf{q}\,du,\\ \sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\boldsymbol{\omega}} &= \mathbf{0}+\mathbf{0}+\mathbf{0}+\frac{d\mathbf{L}}{dt} - \frac{d}{dt}\int\mu\mathbf{q}\times\dot{\mathbf{e}}\,du + \mathbf{0},\\ \sum_{k=1}^{6}\frac{\partial\mathcal{F}_k}{\partial\boldsymbol{\omega}} &= \mathbf{0}+\mathbf{0}+\mathbf{0}+\int\gamma\mathbf{q}\times(\boldsymbol{\omega}\times\mathbf{q})\,du\\ &\quad - \int\gamma\mathbf{q}\times\dot{\mathbf{e}}\,du - \mathbf{v}\times\int\gamma\mathbf{q}\,du,\\ \sum_{k=1}^{6}\frac{\partial}{\partial t}\frac{\partial T_k}{\partial\dot{\mathbf{e}}} &= \mathbf{0} + \mu\dot{\mathbf{v}} + \frac{d(\mu\dot{\mathbf{e}})}{dt} + \mathbf{0}\\ &\quad + \mu(\dot{\boldsymbol{\omega}}\times\mathbf{q} - \boldsymbol{\omega}\times\dot{\mathbf{e}}) + \mathbf{0},\\ \sum_{k=1}^{6}\frac{\partial T_k}{\partial\mathbf{e}} &= \mathbf{0}+\mathbf{0}+\mathbf{0}+\mu\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\mathbf{q}) + \mu\boldsymbol{\omega}\times\dot{\mathbf{e}} + \mathbf{0},\\ \sum_{k=1}^{6}\frac{\partial F_k}{\partial\dot{\mathbf{e}}} &= \mathbf{0} + \gamma\mathbf{v} + \gamma\dot{\mathbf{e}} + \mathbf{0} + \gamma\dot{\boldsymbol{\omega}}\times\mathbf{q} + \mathbf{0}. \end{aligned} \tag{24}$$

The term

$$\mathbf{L} = \frac{\partial}{\partial\boldsymbol{\omega}}\int\frac{\mu}{2}(\boldsymbol{\omega}\times\mathbf{q})\cdot(\boldsymbol{\omega}\times\mathbf{q})\,du = \int\mu\mathbf{q}\times(\boldsymbol{\omega}\times\mathbf{q})\,du$$

is known as the angular momentum of the deformable body as it rotates rigidly about the center of mass. It can be shown that

$$\mathbf{L} = \int\mu(\boldsymbol{\omega}\mathbf{q}\cdot\mathbf{q} - \mathbf{q}\mathbf{q}\cdot\boldsymbol{\omega})\,du = \mathbf{I}\boldsymbol{\omega} \tag{25}$$

where $\mathbf{I}$ is the inertia tensor whose components are given in Equation 10.

Insert Equations 24 and 25 into Equations 22; then Equations 23 yield the equations of motion (Equations 9).

Appendix B: Implementation details

To illustrate the implementation, we consider the case of surfaces. Curves (solids) involve a straightforward restriction (extension) of the two-parameter equations developed in this section. Letting $\mathrm{u}=(u_1,u_2)=(u,v)$ be the surface's material coordinates and letting $p=2$ in Equations 15 and 16 will yield the variational derivative

$$\delta_{\mathbf{e}}\mathcal{E} = w_{00}\mathbf{e} - (w_{10}\mathbf{e}_u)_u - (w_{01}\mathbf{e}_v)_v + (w_{20}\mathbf{e}_{uu})_{uu} + 2\,(w_{11}\mathbf{e}_{uv})_{uv} + (w_{02}\mathbf{e}_{vv})_{vv}\,, \tag{26}$$

where the subscripts denote partial derivatives with respect to material coordinates. The functions $w_{ij}(\mathrm{u})$ locally control the partial derivatives of deformational displacement $\mathbf{e}$ of the model. Specifically, w_{00} controls the local magnitude of the deformation, w_{10} and w_{01} control its local variations, while w_{20}, w_{11}, and w_{02} control its local curvatures.

Semidiscretization

We illustrate the semidiscretization step using standard finite-difference approximations. The unit square domain $\Omega = 0 \leq u,v \leq 1$ of the surface is discretized as a regular $M \times N$ discrete mesh Ω^h of nodes. The internode spacings are $h_1 = 1/(M-1)$ and $h_2 = 1/(N-1)$ in the u and v coordinate directions respectively. Nodes are indexed by integers $[m,n]$ where $0 \leq m \leq M$ and $0 \leq n \leq N$. We approximate the (continuous) vector functions of (u,t) in Equations 9 through 11 by arrays of (continuous-time) vector-valued nodal variables: $\mathbf{r}[m,n] = \mathbf{r}(mh_1,nh_2)$, $\mathbf{e}_t[m,n] = \mathbf{e}(mh_1,nh_2,t)$, and $\mathbf{f}_t[m,n] = \mathbf{f}(mh_1,nh_2,t)$. We will suppress the time-dependence notation until the next section, where we consider integration through time.

The discrete elastic force requires approximating from the nodal variables $\mathbf{e}[m,n]$ the first and second partial derivatives of $\mathbf{e}$ with respect to material coordinates u and v. We define the forward first-difference operators

$$\begin{aligned} D_{10}^{+}(\mathbf{e})[m,n] &= (\mathbf{e}[m+1,n] - \mathbf{e}[m,n])/h_1 \\ D_{01}^{+}(\mathbf{e})[m,n] &= (\mathbf{e}[m,n+1] - \mathbf{e}[m,n])/h_2 \end{aligned} \tag{27}$$

and the backward first-difference operators

$$\begin{aligned} D_{10}^{-}(\mathbf{e})[m,n] &= (\mathbf{e}[m,n] - \mathbf{e}[m-1,n])/h_1 \\ D_{01}^{-}(\mathbf{e})[m,n] &= (\mathbf{e}[m,n] - \mathbf{e}[m,n-1])/h_2. \end{aligned} \tag{28}$$

Using Equations 27 and 28, the forward and backward cross-difference operators are

$$\begin{aligned} D_{11}^{+}(\mathbf{e})[m,n] &= D_{10}^{+}(D_{01}^{+}(\mathbf{e}))[m,n], \\ D_{11}^{-}(\mathbf{e})[m,n] &= D_{10}^{-}(D_{01}^{-}(\mathbf{e}))[m,n], \end{aligned} \tag{29}$$

and the central second-difference operators are

$$\begin{aligned} D_{20}(\mathbf{e})[m,n] &= D_{10}^{-}(D_{10}^{+}(\mathbf{e}))[m,n], \\ D_{02}(\mathbf{e})[m,n] &= D_{01}^{-}(D_{01}^{+}(\mathbf{e}))[m,n]. \end{aligned} \tag{30}$$

Using the above difference operators, we discretize Equation 26 as follows:

$$\begin{aligned} \delta_{\mathbf{e}}\mathcal{E} \approx\; & w_{00}\mathbf{e}[m,n] \\ & - D_{10}^{-}(w_{10}D_{10}^{+}\mathbf{e})[m,n] - D_{01}^{-}(w_{01}D_{01}^{+}\mathbf{e})[m,n] \\ & + D_{20}(w_{20}D_{20}\mathbf{e})[m,n] + 2D_{11}^{-}(w_{11}D_{11}^{+}\mathbf{e})[m,n] \\ & + D_{02}(w_{02}D_{02}\mathbf{e})[m,n]. \end{aligned} \tag{31}$$

Free (natural) boundary conditions are introduced by nullifying the value of difference operators found inside parentheses in Equation 31. Such conditions are appropriate at the boundaries of Ω^h, where these operators would attempt to access nodal variables $\mathbf{e}[m,n]$ outside the discrete domain. Similarly, fractures are introduced by nullifying the values of any difference operators accessing nodal variables on opposite sides of such discontinuities.

If the nodal variables making up the grid functions $\mathbf{e}[m,n]$ are collected into an MN-dimensional vector $\underline{\mathbf{e}}$, the discrete approximation (Equation 31) may be written in the grid vector form $\mathbf{K}\underline{\mathbf{e}}$ where $\mathbf{K}$ is an MN-dimensional square matrix. Because of the local nature of the finite-difference discretization, $\mathbf{K}$, known as the stiffness matrix, has the desirable computational properties of sparseness and bandedness.

The discrete mass and damping densities are grid functions $\mu[m,n]$ and $\gamma[m,n]$ respectively. Let $\mathbf{M}$ be the mass matrix, a diagonal MN-dimensional square matrix with the $\mu[m,n]$ variables as diagonal entries, and let $\mathbf{C}$ be the damping matrix constructed analogously from $\gamma[m,n]$.

Using Equation 31, the equations of motion (Equations 9) can be expressed in semidiscrete form by the following system of coupled ordinary differential equations:

$$m\frac{d\mathbf{v}}{dt} = \mathbf{g}^{\mathbf{v}}, \tag{32a}$$

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) = \mathbf{g}^{\boldsymbol{\omega}}, \tag{32b}$$

$$\mathbf{M}\ddot{\underline{\mathbf{e}}} + \mathbf{C}\dot{\underline{\mathbf{e}}} + \mathbf{K}\underline{\mathbf{e}} = \underline{\mathbf{g}}^{\mathbf{e}}, \tag{32c}$$

where

$$\mathbf{g}^{\mathbf{v}} = h_1 h_2 \left(\sum_{m,n} \underline{\mathbf{f}} - \frac{d}{dt} \sum_{m,n} \mu \dot{\underline{\mathbf{e}}} - \sum_{m,n} \gamma \dot{\underline{\mathbf{x}}} \right), \quad \textbf{(33a)}$$

$$\mathbf{g}^{\boldsymbol{\omega}} = h_1 h_2 \left(\sum_{m,n} \underline{\mathbf{q}} \times \underline{\mathbf{f}} - \frac{d}{dt} \sum_{m,n} \mu \underline{\mathbf{q}} \times \dot{\underline{\mathbf{e}}} - \sum_{m,n} \gamma \underline{\mathbf{q}} \times \dot{\underline{\mathbf{x}}} \right), \quad \textbf{(33b)}$$

$$\underline{\mathbf{g}}^{\mathbf{e}} = \underline{\mathbf{f}} - \mu \dot{\mathbf{v}} - \mu \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \underline{\mathbf{q}}) - 2\mu \boldsymbol{\omega} \times \dot{\underline{\mathbf{e}}} + \mu \dot{\boldsymbol{\omega}} \times \underline{\mathbf{q}}. \quad \textbf{(33c)}$$

Note that the integrals in Equations 9 through 11 are approximated by sums over nodal variables. Some of the terms in Equation 9 have been brought to the right-hand side to simplify the final step of the solution.

Numerical integration through time

To simulate the dynamics of our model, we integrate the semidiscrete system (Equation 32) through time. Dividing an open-ended interval from $t=0$ into time steps Δt, the integration procedure computes a sequence of approximations at times $\Delta t, 2\Delta t, \ldots, t, t+\Delta t, \ldots$. Each time step requires the solution of two algebraic equations for $\mathbf{v}$ and $\boldsymbol{\omega}$, which describe the rigid motion of the body frame ϕ, in tandem with a linear algebraic system for the displacement component $\underline{\mathbf{e}}$.

Substituting the discrete time approximation $\dot{\mathbf{v}} \approx (\mathbf{v}_{t+\Delta t} - \mathbf{v}_t)/\Delta t$ into Equation 32a, we obtain the integration procedure

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \frac{\Delta t}{m} \mathbf{g}_t^{\mathbf{v}} \quad \textbf{(34)}$$

for the linear velocity of ϕ at the next time instant. Similarly, we obtain from Equation 32b

$$\boldsymbol{\omega}_{t+\Delta t} = \mathbf{I}_{t+\Delta t}^{-1} \left(\mathbf{I}_t \boldsymbol{\omega}_t + \Delta t \mathbf{g}_t^{\boldsymbol{\omega}} \right) \quad \textbf{(35)}$$

for the angular velocity of ϕ for the next time instant. At each time step, the body is translated by $\mathbf{d} = \Delta t \mathbf{v}_t$, and rotated by an angle of $\theta = \Delta t |\boldsymbol{\omega}_t|$ about the unit vector $\mathbf{a} = [a_1, a_2, a_3]' = \boldsymbol{\omega}_t / |\boldsymbol{\omega}_t|$ using the matrix

$$\mathbf{R} = \begin{pmatrix} a_1 a_1 \operatorname{vers}\theta + \cos\theta & a_1 a_2 \operatorname{vers}\theta - a_3 \sin\theta & a_1 a_3 \operatorname{vers}\theta + a_2 \sin\theta \\ a_2 a_1 \operatorname{vers}\theta + a_3 \sin\theta & a_2 a_2 \operatorname{vers}\theta + \cos\theta & a_2 a_3 \operatorname{vers}\theta - a_1 \sin\theta \\ a_3 a_1 \operatorname{vers}\theta - a_2 \sin\theta & a_3 a_2 \operatorname{vers}\theta + a_1 \sin\theta & a_3 a_3 \operatorname{vers}\theta + \cos\theta \end{pmatrix}, \quad \textbf{(36)}$$

where $\operatorname{vers}\theta = (1 - \cos\theta)$ denotes the versine of θ. The reference and displacement components are transformed as follows:

$$\mathbf{r}_{t+\Delta t} = \mathbf{R}\mathbf{r}_t + \mathbf{d}; \qquad \mathbf{e}_{t+\Delta t} = \mathbf{R}\mathbf{e}_t. \quad \textbf{(37)}$$

Next, substituting the discrete-time approximations $\ddot{\underline{\mathbf{e}}} \approx (\underline{\mathbf{e}}_{t+\Delta t} - 2\underline{\mathbf{e}}_t + \underline{\mathbf{e}}_{t-\Delta t})/\Delta t^2$ and $\dot{\underline{\mathbf{e}}} \approx (\underline{\mathbf{e}}_{t+\Delta t} - \underline{\mathbf{e}}_{t-\Delta t})/2\Delta t$ into Equation 32c, we obtain the procedure

$$\mathbf{A}\underline{\mathbf{e}}_{t+\Delta t} = \underline{\mathbf{g}}_t, \quad \textbf{(38)}$$

where the constant matrix

$$\mathbf{A} = \mathbf{K} + \left(\frac{1}{\Delta t^2} \mathbf{M} + \frac{1}{2\Delta t} \mathbf{C} \right) \quad \textbf{(39)}$$

and the effective force vector

$$\underline{\mathbf{g}}_t = \underline{\mathbf{g}}_t^{\mathbf{e}} + \left(\frac{1}{\Delta t^2} \mathbf{M} + \frac{1}{2\Delta t} \mathbf{C} \right) \underline{\mathbf{e}}_t + \left(\frac{1}{\Delta t} \mathbf{M} - \frac{1}{2\Delta t} \mathbf{C} \right) \tilde{\underline{\mathbf{e}}}_t, \quad \textbf{(40)}$$

with

$$\tilde{\underline{\mathbf{e}}}_t = (\underline{\mathbf{e}}_t - \underline{\mathbf{e}}_{t-\Delta t})/\Delta t. \quad \textbf{(41)}$$

Note that Equations 34 through 41 specify a semi-implicit recursive procedure that evolves the rigid-body dynamics and elastic displacements from given initial conditions $\mathbf{v}_0$, $\boldsymbol{\omega}_0$, $\underline{\mathbf{e}}_0$, and $\dot{\underline{\mathbf{e}}}_0$. In particular, the displacement $\underline{\mathbf{e}}$ evolves as a time sequence of static equilibrium problems is solved. Each is a sparse linear system (Equation 38) of size proportional to the number of nodes making up the discrete model.

We have employed iterative methods, such as the successive overrelaxation or the conjugate-gradient method, as well as direct methods, such as Choleski factorization, to solve the sparse linear systems (Equation 38).[15,16] Since $\mathbf{A}$ is a sparse matrix (because of Equation 31, each equation will have at most 13 nonzero coefficients), we implement the direct method using an efficient profile storage scheme. A more detailed description of our linear equation solvers is beyond the scope of this article. Further computational savings can be had by neglecting some of the interaction terms in the right-hand sides of

Equation 33; for example, the centrifugal force may be neglected unless large angular velocities $\boldsymbol{\omega}$ are expected, while the Coriolis force may be neglected unless significant $\dot{\mathbf{e}}$ is expected.

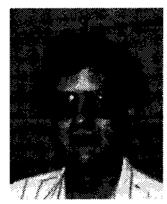
References

1. D. Terzopoulos et al., "Elastically Deformable Models," *Computer Graphics* (Proc. SIGGRAPH), Vol. 21, No. 4, July 1987, pp. 205-214.
2. J. Lassiter, "Principles of Traditional Animation Applied to 3D Computer Animation," *Computer Graphics* (Proc. SIGGRAPH), Vol. 21, No. 4, July 1987, pp. 35-44.
3. W.W. Armstrong and M. Green, "The Dynamics of Articulated Rigid Bodies for Purposes of Animation," *The Visual Computer*, Vol. 1, No. 4, 1985, pp. 231-240.
4. J. Wilhelms and B.A. Barsky, "Using Dynamic Analysis to Animate Articulated Bodies Such as Humans and Robots," *Proc. Graphics Interface 85*, Canadian Information Processing Soc., Toronto, 1985, pp. 97-104.
5. M. Girard and A.A. Maciejewski, "Computational Modeling for the Computer Animation of Legged Figures," *Computer Graphics* (Proc. SIGGRAPH), Vol. 19, No. 3, July 1985, pp. 263-270.
6. C.R. Feynman, *Modeling the Appearance of Cloth*, master's thesis, MIT, Cambridge, Mass, 1986.
7. C.M. Hoffmann and J.E. Hopcroft, "Simulation of Physical Systems from Geometric Models," *IEEE J. Robotics and Automation*, Vol. 3, No. 3, June 1987, pp. 194-206.
8. P.M. Issacs and M.F. Cohen, "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions, and Inverse Dynamics," *Computer Graphics* (Proc. SIGGRAPH), Vol. 21, No. 4, July 1987, pp. 215-224.
9. A. Barr et al., *Topics in Physically-Based Modeling*, SIGGRAPH 87 course notes, Vol. 17, ACM, New York, 1987.
10. L.D. Landau and E.M. Lifshitz, *Theory of Elasticity*, Pergamon Press, London, 1959.
11. H. Goldstein, *Classical Mechanics*, Addison-Wesley, Reading, Mass, 1980.
12. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. I, Interscience, London, 1953.
13. D. Terzopoulos, "Regularization of Inverse Visual Problems Involving Discontinuities," *IEEE Trans. PAMI*, Vol. 8, No. 4, July 1986, pp. 413-424.
14. L. Lapidus and G.F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, John Wiley and Sons, New York, 1982.
15. O.C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, London, 1977.
16. G. Dahlquist and A. Bjorck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J., 1974.
17. K. Fleischer et al., "Cooking with Kurt" (a computer-animated video), Schlumberger Palo Alto Research, Palo Alto, Calif., 1987.
18. D. Terzopoulos, "On Matching Deformable Models to Images: Direct and Iterative Solutions," *Topical Meeting on Machine Vision, Technical Digest Series*, Vol. 12., Optical Soc. America, Washington, D.C., 1987, pp. 160-167.
19. D. Terzopoulos, A. Witkin, and M. Kass, "Symmetry-Seeking Models and 3D Object Reconstruction," *Int'l J. Computer Vision*, Vol. 1, No. 3, Oct. 1987, pp. 211-221.
20. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision*, Vol. 1, No. 4, Dec. 1987, pp. 321-331.
21. K. Fleischer and A. Witkin, "A Modeling Testbed," *Proc. Graphics Interface 88*, Canadian Information Processing Soc., Toronto, 1988, pp. 127-137.
22. D. Terzopoulos and K. Fleischer, "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture," *Computer Graphics* (Proc. SIGGRAPH), Vol. 22, No. 4, Aug. 1988, pp. 269-278.

Demetri Terzopoulos is Program Leader, Visual Modeling, at Schlumberger Palo Alto Research and is an associate of the Canadian Institute for Advanced Research. Previously he was a research scientist at the MIT Artificial Intelligence Laboratory. His areas of interest include computer vision and computer modeling and animation.

Terzopoulos received a PhD in artificial intelligence from MIT in 1984, as well as an MEng in electrical engineering in 1980 and a BEng in honours electrical engineering in 1978, both from McGill University.

Terzopoulos can be reached at Schlumberger Palo Alto Research, 3340 Hillview Avenue, Palo Alto, CA 94304.



Andrew Witkin is an associate professor of computer science and robotics at Carnegie Mellon University. Previously he managed the Vision and Modeling group at Schlumberger Palo Alto Research. His areas of interest include computer vision and computer animation.

Witkin received his PhD in psychology from MIT in 1980, and his BA in psychology from Columbia College in 1975.

Witkin can be contacted at the Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.