Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture

Demetri Terzopoulos
Kurt Fleischer

Schlumberger Palo Alto Research
3340 Hillview Avenue, Palo Alto, CA 94304

Abstract

We continue our development of physically-based models for animating nonrigid objects in simulated physical environments. Our prior work treats the special case of objects that undergo perfectly elastic deformations. Real materials, however, exhibit a rich variety of inelastic phenomena. For instance, objects may restore themselves to their natural shapes slowly, or perhaps only partially upon removal of forces that cause deformation. Moreover, the deformation may depend on the history of applied forces. The present paper proposes inelastically deformable models for use in computer graphics animation. These dynamic models tractably simulate three canonical inelastic behaviors—viscoelasticity, plasticity, and fracture. Viscous and plastic processes within the models evolve a reference component, which describes the natural shape, according to yield and creep relationships that depend on applied force and/or instantaneous deformation. Simple fracture mechanics result from internal processes that introduce local discontinuities as a function of the instantaneous deformations measured through the model. We apply our inelastically deformable models to achieve novel computer graphics effects.

Keywords: Modeling, Animation, Deformation, Elasticity, Dynamics, Simulation

CR Categories and Subject Descriptors: G.1.8—Partial Differential Equations; I.3.5—Computational Geometry and Object Modeling (Curve, Surface, Solid, and Object Representations); I.3.7—Three-Dimensional Graphics and Realism (Animation); I.6.3 Simulation and Modeling (Applications)

 ACM-0-89791-275-6/88/008/0269 $00.75

1. Introduction

Modeling and animation based on physical principles is establishing itself as a computer graphics technique offering unsurpassed realism [1, 2]. Physically-based models of natural phenomena are making exciting contributions to image synthesis. A popular theme is the use of Newtonian dynamics to animate articulated or arbitrarily constrained assemblies of rigid objects in simulated physical environments [3–8]. The animation of continuously stretchable and flexible objects in such environments is also attracting increasing attention. It is extremely difficult to animate nonrigid objects with any degree of realism using conventional, kinematic methods. A better approach to synthesizing physically plausible nonrigid motions is to model the continuum-mechanical principles governing the dynamics of nonrigid bodies.

Initial models of flexible objects were concerned with static shape [9, 10]. Subsequent work produced models for animating nonrigid objects in simulated physical worlds [11–14]. In [11] we employ elasticity theory to model the shapes and motions of deformable curves, surfaces, and solids. Technically as well as computationally, this approach is more demanding than conventional methods for modeling free-form shape, but the results are well worth the extra effort. Our simulation algorithms have proven capable of synthesizing realistic motions arising from the complex interaction of elastically deformable models with diverse forces, ambient media, and impenetrable obstacles.

Prior work on deformable models in computer graphics treats only the case of objects undergoing perfectly elastic deformation. A deformation is termed elastic if the undeformed or reference shape restores itself completely, upon removal of all external forces. A basic assumption underlying the constitutive laws of classical elasticity theory is that the restoring force (stress) in a body is a single-valued function of the deformation (strain) of the body and, moreover, that it is independent of the history of the deformation. It is possible to quantify elastic restoring forces in terms of potential energies of deformation, a characterization that we employ in the formulation of our models. Like an ideal spring, an elastic model stores potential energy during deformation and releases the energy entirely as it recovers the reference shape. By contrast, a perfect (Newtonian) fluid stores no deformation energy,

hence it exhibits no resilience.

In the present paper, we develop computer graphics models which make inroads into the broad spectrum of *inelastic* deformation phenomena intermediate between perfectly elastic solids, on the one hand, and viscous fluids, on the other. Generally, a deformation is inelastic if it does not obey the idealized (Hookean) constitutive laws of classical elasticity. Inelastic deformations occur in real materials for temperatures and forces exceeding certain limiting values above which irreversible dislocations at the atomic level can no longer be neglected.

Why model inelastic behavior? Aside from an artistic motivation to achieve a rich variety of novel graphics effects, we wish to incorporate into our deformable models the mechanical behaviors commonly associated with high polymer solids—organic compounds containing a large number of recurring chemical structures—such as modeling clay, thermoplastic compound, or silicone putty [15]. These behaviors are responsible for the universal utility of these sorts of modeling materials in molding complex shapes (e.g, in the design of automobile bodies). We are interested in assimilating some of the natural conveniences of this traditional art into the computer-aided design environment of the future. We envision users, aided by stereoscopic and haptic input-output devices, carving "computer plasticine" and applying simulated forces to it in order to create free-form shapes interactively.

Our physically-based models incorporate three canonical genres of inelastic behavior—*viscoelasticity, plasticity,* and *fracture.* Viscoelastic material behavior includes the characteristics of a viscous fluid together with elasticity. Silicone ("Silly") putty exhibits unmistakable viscoelastic behavior; it flows under sustained force, but bounces like a rubber ball when subjected to quickly transient forces. Inelastic materials for which permanent deformations result from the mechanism of slip or atomic dislocation are known as plastic. Most metals, for instance, behave elastically only when the applied forces are small, after which they yield plastically, resulting in permanent dimensional changes. Our models can also simulate the behavior of thermoplastics, which may be formed easily into desired shapes by pressure at relatively moderate temperatures, then made elastic or rigid around these shapes by cooling. As materials are deformed beyond certain limits, they eventually fracture. Cracks develop according to internal force or deformation distributions and their propagation is affected by local variations in material properties.

Fig. 1 illustrates some of the capabilities of our inelastic models in Flatland, a restricted physical world. Flatland models are deformable planar curves capable of rigid-body dynamics or general "elastoviscoplastodynamics" (!) with possible fractures. An efficient numerical algorithm provides real-time response (on Symbolics 3600 series Lisp Machines), enabling us to interact with the models by subjecting them to user-controlled forces, aerodynamic drag, gravity, collisions, etc. (see [11] for more details on formulating forces). Fig. 1a–c shows the strobed motion of an inelastic flatland model that has zero, medium, and high viscoelasticity as it collides into frictionless walls. The strobe frames in Fig. 1d illustrate the interactive molding

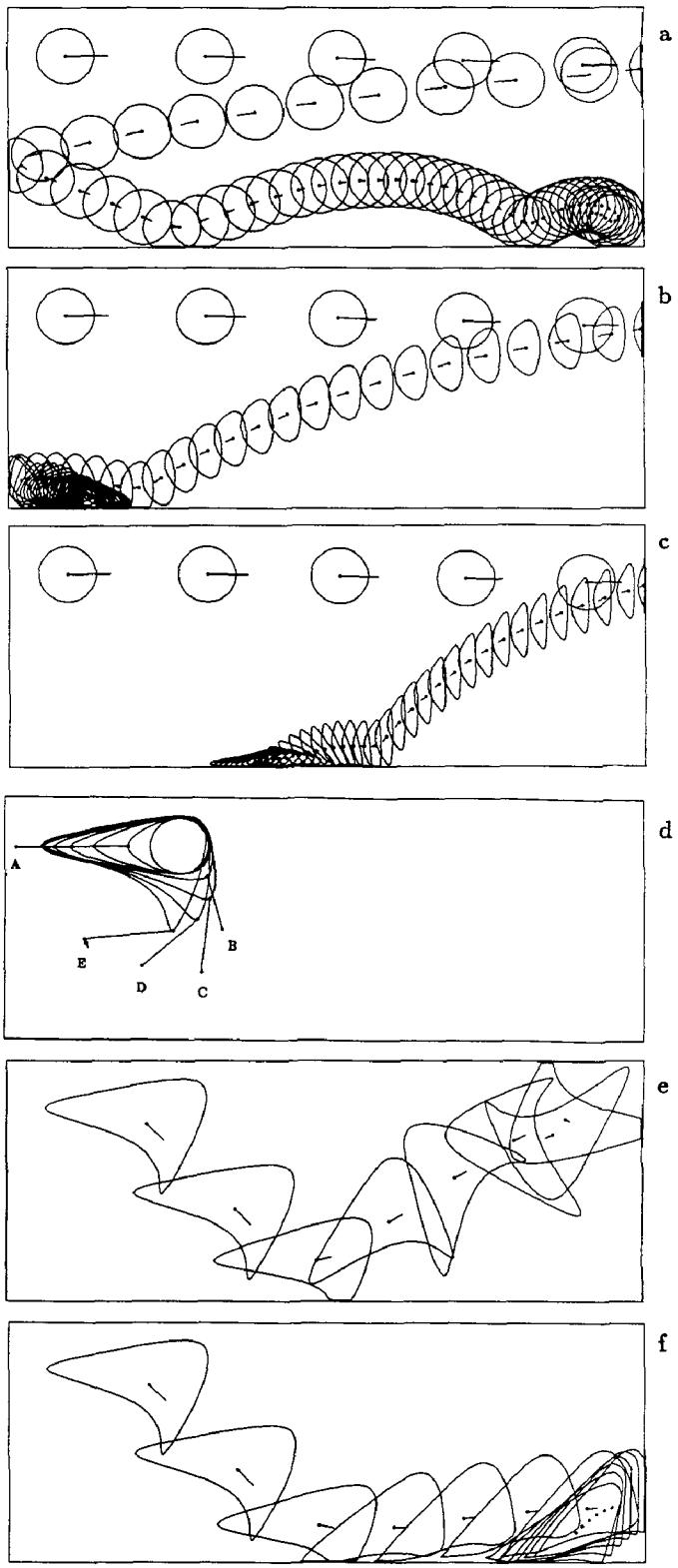

Figure 1. Simulations in Flatland. Models are strobed while undergoing motion subject to gravity, drag, collisions, and user-controlled forces. Velocity vector of the center of mass (dot) is indicated. (a) Elastic model. (b) Viscoelastic model. (c) Highly viscoelastic model. (d) A viscoelastic model is deformed. (e) Resulting shape is made elastic and bounced. (f) Same shape made viscoelastic and bounced.

of inelastic models through the application of simple forces. The user starts with a circular viscoelastic model fixed at its center. The model simulates thermoplastic material. The user applies a sustained spring force from point A. The spring (under position control from a "mouse") is shown in the figure as a line between two points. The spring force deforms the model, stretching it to the left (an effect known as stress relaxation). Next, the user releases the spring from A, then reactivates it at B and sweeps through C, D, and E, pulling the material along. The final shape is set by "cooling" the thermoplastic. The model is then made perfectly elastic and it can be bounced (Fig. 1e). Finally the model is made inelastic and bounced again (Fig. 1f). Later we present further details and examples of more complex three-dimensional inelastic models.

The inelastic models described in this paper generalize our prior elastic models and inherit their animate characteristics, thereby unifying the description of shape and motion. We show how to model inelastic deformation in the context of two varieties of deformable models which we have developed in prior papers [11, 14]. Both formulations allow elastic deformation away from a reference shape represented within the model. In our inelastic generalizations, internal viscous and plastic processes dynamically feed part of the instantaneous deformation back into the reference shape component. Simplified fracture mechanics result from internal processes which introduce local discontinuities dynamically as a function of the instantaneous deformations measured through the model.

We conclude the introduction with a perspective on our work as it relates to the engineering analysis of materials and structures. First, here is a caveat: We make no particular attempt to model specific materials accurately. Usually the general behavior of a material will defy accurate mathematical description, and engineering models tend to be complicated. Sophisticated finite element codes are available for analyzing the mechanics of nonrigid structures constructed from specific materials such as steel and concrete [16]. Computer graphics has become indispensable for visualizing the overwhelming amount of data that can be produced during the preprocessing and postprocessing stages of finite element analysis [17–19].

Although we adopt certain numerical techniques from finite element analysis, our computer graphics modeling work has a distinctly different emphasis. We have sought to develop physically-based models with associated numerical procedures that can be utilized to create realistic animations. Hence, our deformable models are convenient for computer graphics applications, where a keen concern with tractability motivates mathematical abstraction and computational expediency. This paper develops inelastic models that idealize regimes of material response under certain types of environmental conditions, whose parameters describe qualitatively familiar behaviors, such as stretchability, bendability, resilience, fragility, etc.

The organization of the remainder of the paper is as follows: Section 2 describes inelastic deformation phenomena in more detail using idealized mechanical units. Sections 3 and 4 review our basic elastic models and explain how we incorporate inelastic behaviors into the partial differential equations that govern their motions. Section 5 summarizes our implementation. Section 6 presents more simulation results and Section 7 draws conclusions.

2. Inelastic Deformation

A formal treatment of inelastic deformation is beyond the scope of this paper. For theory on viscoelasticity, plasticity, and fracture, refer to, e.g., [20–22]. The basic inelastic behaviors may be understood readily, however, in terms of assemblies of idealized uniaxial (one-dimensional) mechanical units. The ideal linear elastic unit is the spring (Fig. 2a). The spring satisfies Hooke's law—elongation or contraction e (strain) is proportional to applied tension or compression force f (stress): $ke = f$, where k is the spring constant. The elastic unit is supplemented by two other uniaxial units, the viscous and plastic units (Fig. 2b,c). By assembling these units in specific configurations, we can simulate simple, uniaxial viscoelasticity and plasticity. Our inelastically deformable models incorporate the laws governing these units, suitably generalized and extended over a multidimensional continuum.

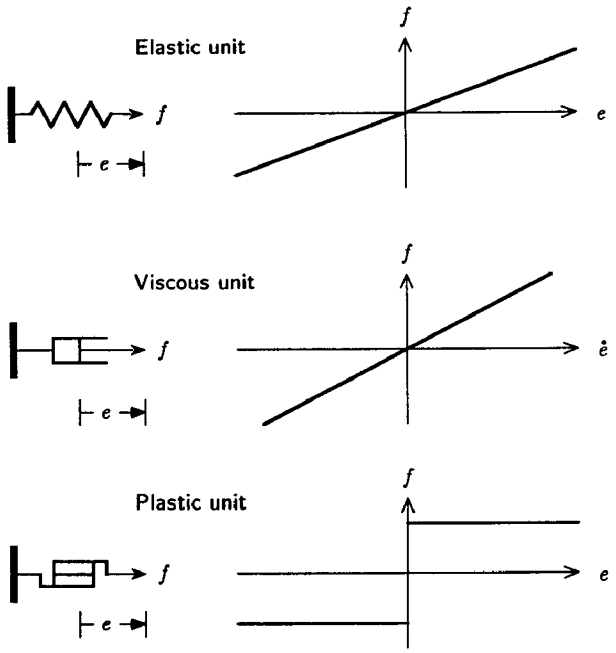

Figure 2. Uniaxial deformation units and their response to applied forces. (a) Elastic spring. (b) Viscous dashpot. (c) Plastic slip unit.

2.1. Viscoelasticity

Viscoelasticity is a generalization of elasticity and viscosity. It is characterized by the phenomenon of creep which manifests itself as a time dependent deformation under constant applied force. In addition to instantaneous deformation, creep deformations develop which generally increase with the duration of the force. Whereas an elastic model, by

definition, is one which has the memory only of its reference shape, the instantaneous deformation of a viscoelastic model is a function of the entire history of applied force. Conversely, the instantaneous restoring force is a function of the entire history of deformation.

The ideal linear viscous unit is the dashpot (Fig. 2b). The rate of increase in elongation or contraction e is proportional to applied force f: $\eta\dot{e} = f$, where η is the viscosity constant (the overstruck dot denotes a time derivative). The elastic and viscous units are combined to model linear viscoelasticity, so that the internal forces depend not just on the magnitude of deformation, but also on the rate of deformation. Fig. 3a illustrates a four-unit viscoelastic model, a series assembly of the so called Maxwell and Voigt viscoelastic models. The stress-strain relationship for this assembly has the general form

$$a_2\ddot{e} + a_1\dot{e} + a_0 e = b_2\ddot{f} + b_1\dot{f} + b_0 f, \qquad (1)$$

where the coefficients depend on the spring and viscosity constants. The response of the models to an applied force (Fig. 3b) is shown graphically in Fig. 3c.

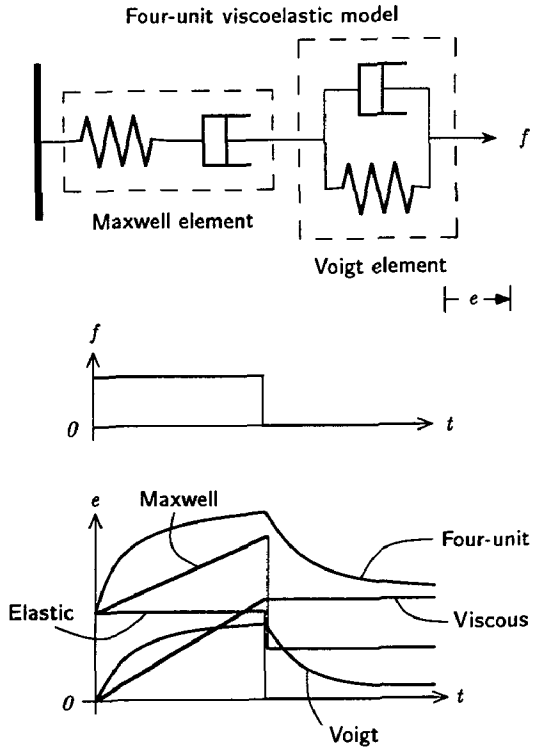

Figure 3. Uniaxial viscoelastic model. (a) The four-element model is a series connection of a Maxwell viscoelastic unit and a Voigt viscoelastic unit. (b) Force applied to the model. (c) Response of various components.

2.2. Plasticity

In plasticity, unique relationships between displacement and applied force do not generally exist. The ideal plastic unit is the slip unit (Fig. 2c). It is capable of arbitrary elongation or contraction as soon as the applied force exceeds a yield force. During plastic yield, the apparent instantaneous elastic constants of the material are smaller than those in the elastic state. Removal of applied force causes the material to unload elastically with its initial elastic constants. This behavior may be termed elastoplastic.

Viscoplasticity, a generalization of plasticity and viscosity, can be modeled by assembling dashpots with plastic units. Analogously, elastoplasticity generalizes elasticity and plasticity and is modeled by assembling springs with plastic units. Fig. 4b presents graphically the response of a simple elastoplastic model (Fig. 4a). The model is linearly elastic from O to A. After reaching the yield point A, the model exhibits linear work hardening. Upon unloading from B, the elastic region is defined by force amplitude $f_B - f_C = 2f_A$. Subsequent loads now move the model along BC. Loading past point B causes further plastic deformation along BE. The reverse plastic deformation occurs along CD. After a closed cycle in force and displacement OABCDO, the model returns to its initial state and subsequent behavior is not affected by the cycle.

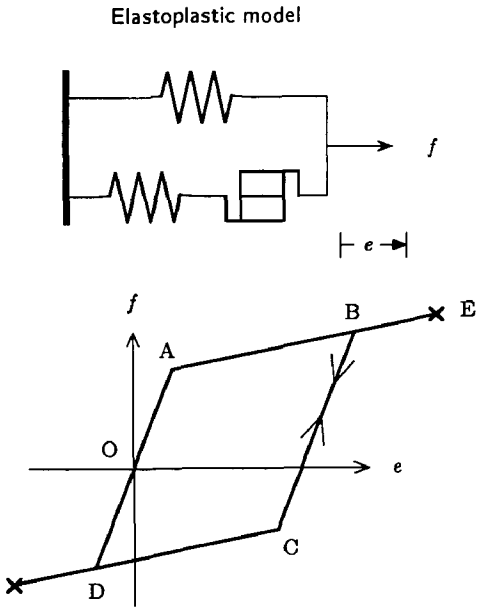

Figure 4. Uniaxial elastoplastic model. (a) The three-unit model. (b) Response to applied force (see text).

2.3. Fracture

Solid materials cannot sustain arbitrarily large stresses without failure, as is represented at point E in the elastoplastic model of Fig. 4. Beyond this limiting elongation, the elastoplastic model fractures. Fractures are localized position discontinuities that arise due to the breaking of atomic bonds in materials. They usually initiate from stress singularities that arise at corners of irregularities or cavities present in solids. Solids exhibit three modes of fracture opening: a tensile mode and two shear modes, one planar and one normal to a plane.

As fractures develop they release internal potential energy of deformation (strain energy). For fractures to propagate through the material, the energy release rate as the fracture lengthens must be greater than a critical value. For brittle materials such as glass, fractures will develop unstably if the energy released is equal to the energy needed to create the free surface associated with the fracture. In this case, minor variations in material properties in the continuum can greatly influence the propagation. For materials like steel, however, the effects of plasticity at fracture tips must be taken into account. We do not consider this effect; its mathematical treatment is under development in the large body of literature on fracture mechanics (see [22]).

3. Basic Deformable Models

This section briefly reviews two formulations of deformable models, a *primary formulation* and a *hybrid formulation*, each of which can serve as a foundation for modeling inelastic behavior. In both formulations u denotes the intrinsic or material coordinates of points in a body Ω. For a solid body $\mathrm{u} = (u_1, u_2, u_3)$ has three coordinates. For a surface $\mathrm{u} = (u_1, u_2)$ and for a curve $\mathrm{u} = (u_1)$. In these 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 primary formulation of deformable models [11] describes deformations using the positions $\mathbf{x}(\mathrm{u},t)$ of points in the body relative to an inertial frame of reference Φ in Euclidean 3-space (Fig. 5). Position is a 3-component vector-valued function of the material coordinates and time. Deformations are measured away from a reference shape which is represented in differential geometric form. For elastic deformations, this representation gives rise to internal deformation energies $\mathcal{E}(\mathbf{x})$ which produce restoring forces that are invariant with respect to rigid motions in Φ.

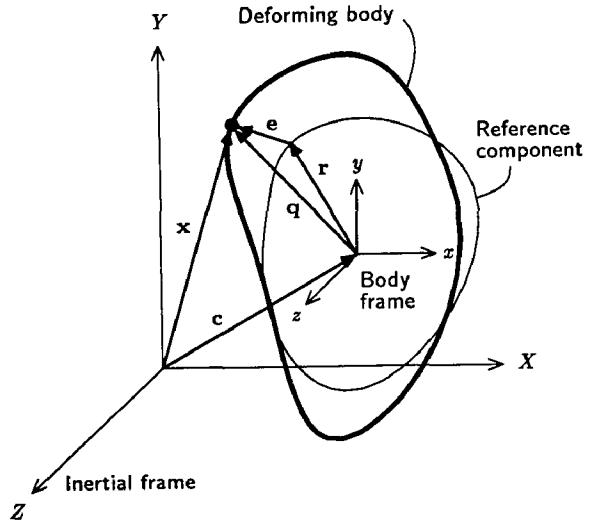

Figure 5. Geometric representation of deformable models.

The hybrid formulation [14] represents the same deformable body as the sum of a reference component $\mathbf{r}(\mathrm{u},t)$ and a deformation component $\mathbf{e}(\mathrm{u},t)$. Both components are expressed relative to a reference frame ϕ whose origin coincides with the body's center of mass $\mathbf{c}(t)$ and which translates and rotates along with the deformable body (Fig. 5). We denote the positions of mass elements in the body relative to ϕ by

$$\mathbf{q}(\mathrm{u},t) = \mathbf{r}(\mathrm{u},t) + \mathbf{e}(\mathrm{u},t). \tag{2}$$

We measure deformations with respect to the reference shape $\mathbf{r}$ represented in parametric form. Elastic deformations are again representable by an energy $\mathcal{E}(\mathbf{e})$, but this energy depends on the position of ϕ. Hence, for the deformable model to have a rigid-body motion mode in addition to an elastic mode, the reference component must be evolved over time according to the laws of rigid-body dynamics [23]. We obtain a model with explicit deformable and rigid characteristics; hence the name "hybrid."

Appendix A gives the equations of motion for both formulations. The primary and hybrid formulations offer different practical benefits at extreme limits of deformable behavior. The primary formulation handles free motions implicitly, but at the expense of a nonquadratic energy functional $\mathcal{E}(\mathbf{x})$ (nonlinear restoring forces). The equation of motion (9) with such a functional is numerically solvable without much difficulty for extremely nonrigid models such as rubber sheets, but the numerical conditioning deteriorates with increasing rigidity due to exacerbated nonlinearity. The hybrid formulation permits the use of a quadratic energy functional $\mathcal{E}(\mathbf{e})$ (linear restoring forces). Despite their greater complexity, the equations of motion (13) offer a significant practical advantage for fairly rigid models with complex reference shapes. Conditioning improves as the model becomes more rigid, tending in the limit to well-conditioned, rigid-body dynamics. See [14] for more details.

4. Incorporating Inelastic Behavior

This section describes how we incorporate inelastic behavior using the hybrid formulation of deformable models and also briefly indicates how we obtain similar effects using the primary formulation. First we will specify the internal restoring forces that govern deformation. Recall that the hybrid formulation expresses this deformation $\mathbf{e}(\mathrm{u},t)$ with respect to a reference component $\mathbf{r}(\mathrm{u},t)$. We obtain viscoelastic, plastic, and fracture behavior by designing internal processes that lawfully update $\mathbf{r}$ and modify material properties according to applied force and instantaneous deformation.

In the hybrid equations of motion (13), the restoring force due to deformational displacement $\mathbf{e}(\mathrm{u},t)$ is represented in (13c) by $\delta_{\mathbf{e}}\mathcal{E}$, a variational derivative [24] with respect to $\mathbf{e}$ of an elastic potential energy functional $\mathcal{E}$. The general form of $\mathcal{E}$ is

$$\mathcal{E}(\mathbf{e}) = \int_{\Omega} E(\mathrm{u}, \mathbf{e}, \mathbf{e}_{\mathrm{u}}, \mathbf{e}_{\mathrm{uu}}, \ldots)\, d\mathrm{u}, \tag{3}$$

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.

A convenient choice for $\mathcal{E}$ is the controlled-continuity generalized spline kernels [25]. These splines are of the form (3) with the integrand defined by

$$E = \frac{1}{2}\sum_{m=0}^{p}\sum_{|j|=m}\frac{m!}{j_1!\ldots j_d!}w_j\left|\partial_j^m \mathbf{e}\right|^2, \tag{4}$$

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

$$\partial_j^m = \frac{\partial^m}{\partial u_1^{j_1}\ldots\partial u_d^{j_d}}. \tag{5}$$

Thus, E is a weighted combination of partial derivatives of $\mathbf{e}$ of all orders up to p, with the weighting functions $w_j(\mathrm{u})$ in (4) controlling the elastic properties of the deformable model over u. The allowable deformation becomes smoother for increasing p.

The variational derivative in Ω of $\mathcal{E}$ with the spline density (4) is

$$\delta_{\mathbf{e}}\mathcal{E} = \sum_{m=0}^{p}(-1)^m\Delta_{w_m}^m\mathbf{e}, \tag{6}$$

where

$$\Delta_{w_m}^m = \sum_{|j|=m}\frac{m!}{j_1!\ldots j_d!}\partial_j^m\left(w_j\partial_j^m\right) \tag{7}$$

is a spatially-weighted iterated Laplacian operator of order m. For convenience, we use cyclic boundary conditions on Ω and we introduce predetermined fractures to create free boundaries as necessary. To create a free surface, for example, we start with a torus and section it around the large and small circumference to obtain a single sheet.

For a surface with $p = 2$ (the highest order of p that we have used to date), the variational derivative of (18) is

$$\begin{aligned}\delta_{\mathbf{e}}\mathcal{E}(\mathbf{e}) =& w_{00}\mathbf{e} - \frac{\partial}{\partial u_1}\left(w_{10}\frac{\partial \mathbf{e}}{\partial u_1}\right) - \frac{\partial}{\partial u_2}\left(w_{01}\frac{\partial \mathbf{e}}{\partial u_2}\right)\\ &+ \frac{\partial^2}{\partial u_1^2}\left(w_{20}\frac{\partial^2 \mathbf{e}}{\partial u_1^2}\right) + 2\frac{\partial^2}{\partial u_1\partial u_2}\left(w_{11}\frac{\partial^2 \mathbf{e}}{\partial u_1\partial u_2}\right)\\ &+ \frac{\partial^2}{\partial u_2^2}\left(w_{02}\frac{\partial^2 \mathbf{e}}{\partial u_2^2}\right),\end{aligned} \tag{8}$$

where $\mathrm{u} = (u_1, u_2)$ are the surface's material coordinates. The function w_{00} penalizes the total magnitude of the deformation; w_{10} and w_{01} penalize the magnitude of its first partial derivatives; w_{20}, w_{11}, and w_{02} penalize the magnitude of its second partial derivatives; etc.

The controlled-continuity spline kernel (4) allows our models to simulate the piecewise continuous deformations characteristic of fractures, creases, curvature discontinuities, etc. The distributed parameter functions w_j offer local continuity control throughout the material domain Ω. Discontinuities in the deformation of order $0 \le k < p$ will occur freely at a material point u_0 when $w_j(\mathrm{u}_0)$ is set to 0 for $|j| > k$ [25].

When the stresses or deformations exceed preset fracture limits, we locally nullify the w_j to introduce discontinuities. We have experimented with several simple schemes for propagating fractures in our models; for instance, at each time step we can insert a position discontinuity (order $k = 0$) at the material point u_* at which there occurs the greatest elastic displacement beyond the limiting elongation over Ω. The yield limit may vary greatly over material coordinates in real materials, especially if there happen to be localized weaknesses, say, from imperfections. We have experimented successfully with yield functions that vary stochastically around some mean yield limit. Promising variations on this theme abound.

As a simple case of viscoelasticity, consider the Maxwell unit depicted in Fig. 3. We allow $\mathbf{e}(\mathrm{u}, t)$, as governed by (6), to play the role of a multidimensional elastic spring in the continuum generalization of this unit, while $\mathbf{r}(\mathrm{u}, t)$ plays the role of the dashpot. The viscous behavior of the dashpot is simulated by an internal process which evolves the reference component as follows: $\dot{\mathbf{r}}(\mathrm{u}, t) = (1/\eta(\mathrm{u}))\mathbf{e}(\mathrm{u}, t)$. We extend this to simulate the four-element viscoelastic model shown in the figure, according to (1). Thus, the viscoelastic process establishes a feedback path from $\mathbf{e}$ into $\mathbf{r}$. During each time interval, a portion of the instantaneous elastic displacement is transferred into the reference component, thereby maintaining a deformation history. This is analogous to the incremental strain theory or flow theory of elasticity. More complex viscoelastic behaviors are produced readily by introducing nonlinear functions into the feedback loop. Bizarre yet interesting behavior—such as negative viscosity—is possible by choosing physically unrealizable parameters.

We have incorporated a multidimensional extension of the uniaxial elastoplastic model of Fig. 4. Here, the reference component $\mathbf{e}$ absorbs the extension of the plastic unit as soon as the applied force exceeds the yield limit. In the multidimensional case, we can incorporate a *yield condition* which can either be dependent on the stresses internal to the model (such as the Tresca or von Mises yield conditions [21]) or on the internal deformation $\mathbf{e}$. The model behaves elastically until the yield condition is exceeded locally. Then the material parameters w_j are reduced locally to simulate linear strain hardening.

The primary formulation of elastically deformable models involves deformation energy functionals that contain fundamental tensors of curves, surfaces, and solids (see [11]). For example, the elastic functional for a solid model was of the form $\mathcal{E}(\mathbf{x}) = \int_\Omega |\mathbf{G}-\mathbf{G}^0|_{\mathbf{W}}^2\, d\mathrm{u}$, 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. The weighted norm $|\cdot|_{\mathbf{W}}$ provides functions $w_i(\mathrm{u})$ that determine material properties. The approach for introducing inelastic behavior is essentially the same as for the hybrid model: We evolve the metric tensor $\mathbf{G}^0$ (and other tensors in $\mathcal{E}(\mathbf{x})$ associated with the undeformed body) according to the model's internal stresses or deformations. For plasticity and fracture, this includes dynamic adjustments to the material property functions.

5. Implementation Overview

Our implementation of inelastic models is built on a substrate of numerical algorithms that we have developed for simulating elastically deformable models [11, 14]. This sec-

tion provides an overview of the solution methodology. We refer the reader to our prior papers for mathematical details and discussion.

The first step is to discretize the continuum equations (9) or (13c) in material coordinates (these partial differential equations are of the hyperbolic-parabolic type, second-order in time and, so far, up to fourth-order in material coordinates). This step, known as semidiscretization, may be performed using finite-difference or finite-element methods on a discrete mesh of nodes [26]. The result is 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. At each time step (or every few time steps) the resulting simulation data may be rendered to create successive frames of the animation. We use a semi-implicit time integration procedure which evolves the elastic displacements (and rigid-body dynamics in the hybrid model) 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 comprising the discrete model.

The size of these linear systems can vary greatly depending on the application. The simulations presented in the next sections range from hundreds to tens of thousands of state variables. Since deformable models involve so many variables (very many more than for typical rigid or articulated body simulations) it is crucial to choose the applicable numerical solution methods judiciously in order to achieve efficiency ([27] is a nice survey of standard numerical techniques).

For up to moderately-sized problems, we have used direct methods; specifically, a Choleski-type matrix factorization procedure with forward-reverse resolution. We use an efficient, profile storage scheme [28] which exploits the sparsity of the linear system (a sparse stiffness matrix results from discretizing the variational derivative $\delta_{\mathbf{e}}\mathcal{E}$ using finite-element or finite-difference approximations; e.g., discretizing (8) using central differences yields equations having at most 13 nonzero coefficients). For large problems involving surfaces or solids, we must resort to iterative methods such as successive over-relaxation (SOR) or the conjugate gradient (CG) method. We have also made use of an alternating-direction-implicit method (ADI) which iterates fast, one-dimensional Choleski solvers [27]. Multigrid methods based on SOR have served well in the largest of our simulations [29].

6. Simulation Examples

The Flatland simulations in Fig. 1 involve a 50-node discrete model (100 deformation equations) and the Choleski solution on the hybrid equations of motion in two dimensions. The collisions are computed by a simple projection method which does not conserve the area of the model. We have animated both physically realizable and unrealizable behaviors in Flatland, including buckling and collapse under load, swelling after impact, etc. It should be possible to animate such inelastic dynamics in real-time in three dimensions on a supercomputer.

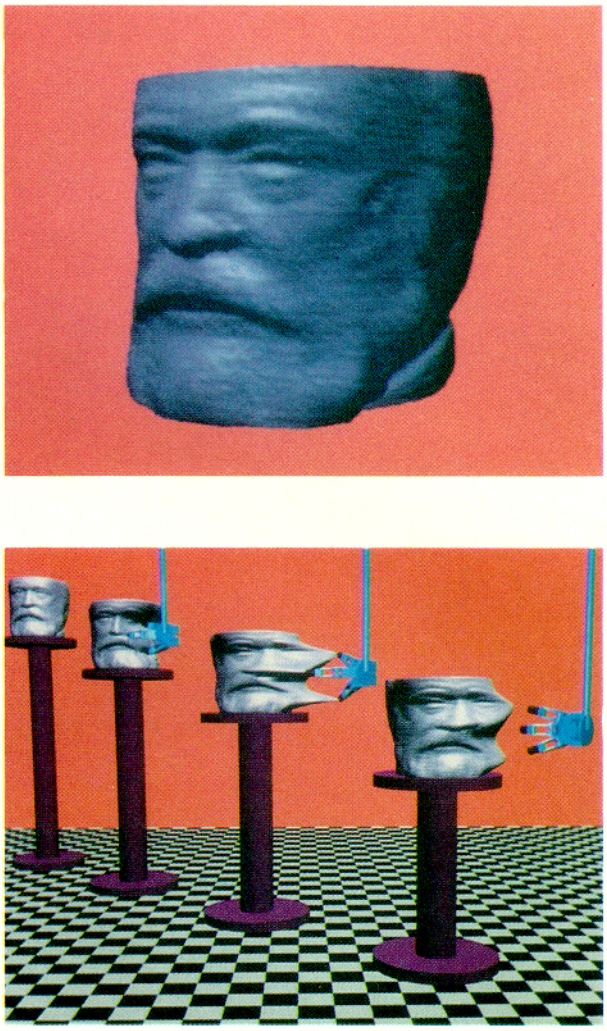

Figure 6. Hugo. (a) A "plasticine" bust of Victor Hugo. (b, back to front) Grabby hand pinches; grabby hand pulls; deformed Hugo.

Next, we demonstrate a physically-based interaction with a 3D simulated "plasticine" bust (Fig. 6a). Employing the hybrid formulation, we initialized the reference component of the model with sampled three-dimensional data (made available by the University of Utah [30]) from a laser scanned sculpture of Victor Hugo. Fig. 6b shows first the undeformed model, followed by a simulation of a robot hand pinching the deformable material with sticky fingers, pulling, then releasing to show the residual plastic deformation. Because of the relatively large size of the discrete model (180×127 mesh; 68580 equations), we applied a multigrid solution method similar to the one described in [31].

The last two examples simulate fracture propagation in surfaces. We used the primary deformable model for-

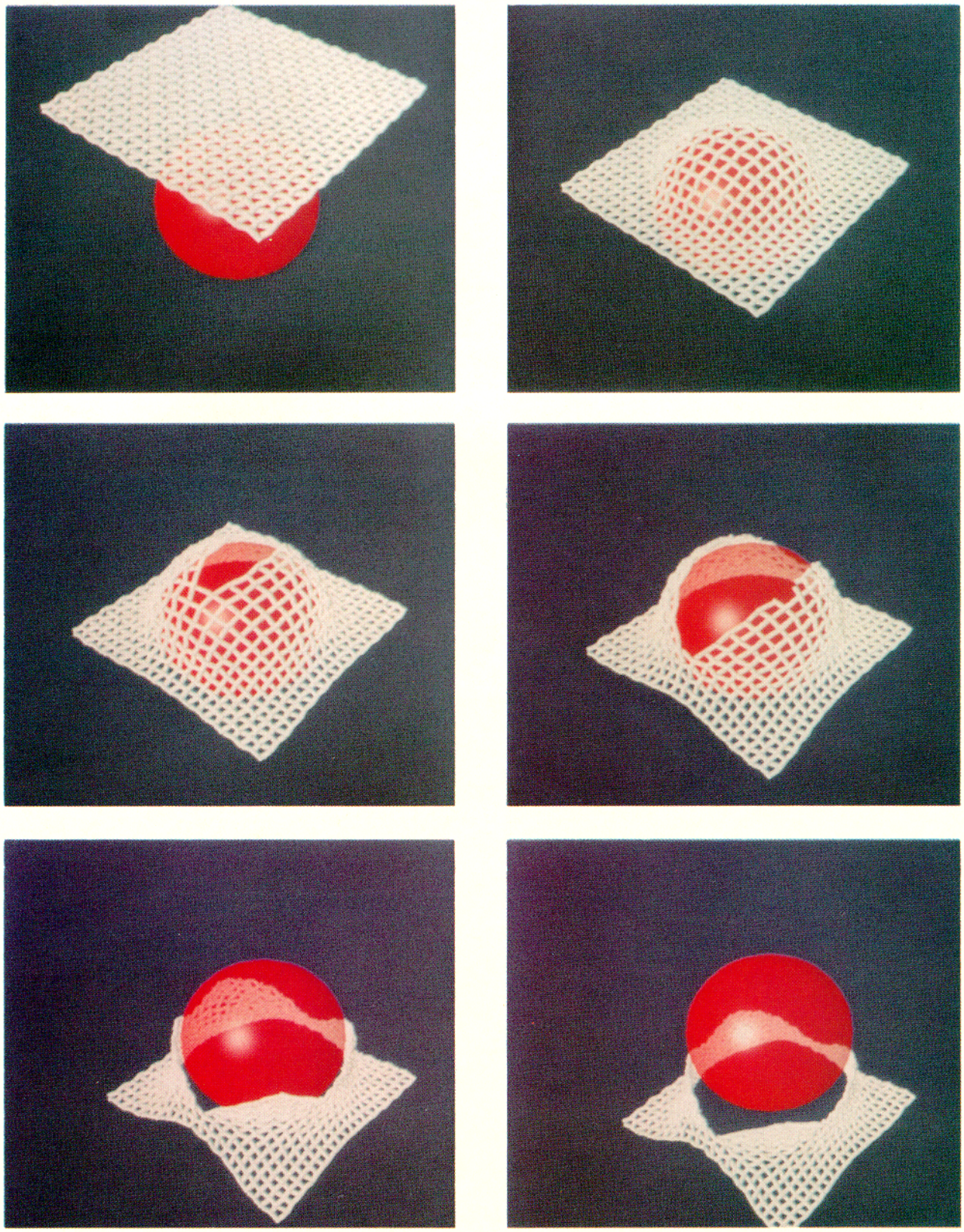

Figure 7. A net falling over a spherical obstacle. Fractures develop and propagate as the deformation exceeds the elastic limit.

mulation and the ADI solution method to run these simulations. Fig. 7 presents an animation of a net (23 × 23 mesh; 1587 equations) falling over an impenetrable obstacle in a gravitational field, in the spirit of the flying carpet animation in [11]. The difference here is that the "fibers" of the mesh are subject to fracture limits based on the deformation in the material. When a fiber stretches beyond the fracture limit it is broken by the fracture process which inserts a discontinuity as described in Section 4. The yield limit is uniform over the mesh, which causes linear tears as one might obtain with cloth.

Fig. 8 shows surface models (30 × 30; 2700 equations) which are sheared by opposing forces. In these examples, we perturbed the fracture tolerance around the material's mean tolerance stochastically in order to introduce some unpredictability in the propagation of fractures.

We rendered the color images in this section using the modeling testbed system described in [32].

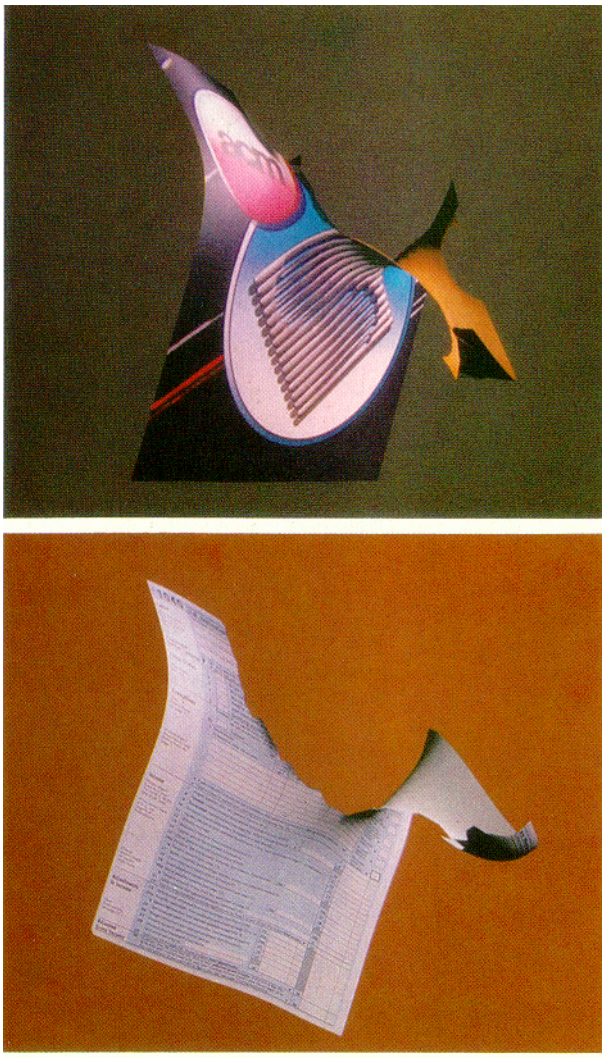

Figure 8. (a) January 12, 1988. (b) April 15, 1988.

7. Conclusion

We have developed physically-based models of objects capable of inelastic deformation for use in computer graphics. We have applied these dynamic models to create interesting viscoelasticity, plasticity, and fracture effects. Our models are designed to be computationally tractable for the purposes of animation. This paper has only touched upon the vast volume of accumulated facts about the mechanics of materials. The modeling of inelastic deformation remains open for further exploration in the context of computer graphics.

Acknowledgements

We thank Rob Howe for providing the CAD model of the robot hand.

A. 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 a function of material coordinates u and time t. In this appendix, overstruck dots denote time derivatives d/dt or $\partial/\partial t$ as appropriate.

Lagrange's equations of motion [23] for $\mathbf{x}$ in the inertial frame Φ take on a relatively simple form [11]:

$$\mu\ddot{\mathbf{x}} + \gamma\dot{\mathbf{x}} + \delta_{\mathbf{x}}\mathcal{E} = \mathbf{f}. \tag{9}$$

During motion, the net external forces $\mathbf{f}(\mathbf{x},t)$ balance dynamically against the inertial force due to the mass density $\mu(\mathrm{u})$, the velocity dependent damping force with damping density $\gamma(\mathrm{u})$ (here a scalar, but generally a matrix), and the internal restoring force. The latter is expressed as a variational derivative $\delta_{\mathbf{x}}$ [24] of a nonnegative deformation energy $\mathcal{E}(\mathbf{x})$ whose value increases monotonically with the magnitude of the deformation. Eq. (9) is a partial differential equation (due to 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.

In the hybrid formulation of the deformable model deformation is decomposed into a reference component $\mathbf{r}(u,t)$ and a deformation component $\mathbf{e}(u,t)$ in a noninertial frame ϕ located at the model's center of mass (See Fig. 5.)

$$\mathbf{c}(t) = \int_{\Omega} \mu(\mathrm{u})\mathbf{x}(\mathrm{u},t)\,d\mathrm{u}. \tag{10}$$

The orientation of ϕ relative to Φ is $\boldsymbol{\theta}(t)$. Given

$$\mathbf{v}(t) = \dot{\mathbf{c}}(t); \qquad \boldsymbol{\omega}(t) = \dot{\boldsymbol{\theta}}(t), \tag{11}$$

respectively the linear and angular velocity of ϕ relative to Φ, the velocity of mass elements relative to Φ is

$$\dot{\mathbf{x}}(\mathrm{u},t) = \mathbf{v}(t) + \boldsymbol{\omega}(t) \times \mathbf{q}(\mathrm{u},t) + \dot{\mathbf{e}}(\mathrm{u},t), \tag{12}$$

where $\mathbf{q}$ is given by (2).

In [14] we apply Lagrangian mechanics based on the kinetic and potential energies which govern our model to transform (9) into three coupled, partial differential equations for the unknown functions $\mathbf{v}$, $\boldsymbol{\omega}$ and $\mathbf{e}$ under the action of an applied force $\mathbf{f}(\mathrm{u},t)$. These equations are given

by

$$\frac{d}{dt}(m\mathbf{v}) + \frac{d}{dt}\int_\Omega \mu\dot{\mathbf{e}}\,d\mathbf{u} + \int_\Omega \gamma\dot{\mathbf{x}}\,d\mathbf{u} = \mathbf{f}^{\mathbf{v}}, \quad (13a)$$

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) + \frac{d}{dt}\int_\Omega \mu\mathbf{q}\times\dot{\mathbf{e}}\,d\mathbf{u} + \int_\Omega \gamma\mathbf{q}\times\dot{\mathbf{x}}\,d\mathbf{u} = \mathbf{f}^{\boldsymbol{\omega}}, \quad (13b)$$

$$\frac{d}{dt}(\mu\dot{\mathbf{e}}) + \mu\dot{\mathbf{v}} + \mu\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\mathbf{q}) + 2\mu\boldsymbol{\omega}\times\dot{\mathbf{e}} + \mu\dot{\boldsymbol{\omega}}\times\mathbf{q} + \gamma\dot{\mathbf{x}} + \delta_{\mathbf{e}}\mathcal{E} = \mathbf{f}^{\mathbf{e}}. \quad (13c)$$

Here $m = \int_\Omega \mu\,d\mathbf{u}$ is the total mass of the body, and the time-varying, 3×3 symmetric matrix $\mathbf{I}$ with entries $I_{ij} = \int_\Omega \mu(\delta_{ij}\mathbf{q}^2 - q_i q_j)\,d\mathbf{u}$, where $\mathbf{q} = [q_1, q_2, q_3]$ and δ_{ij} is the Kronecker delta, is known as the inertia tensor. The applied force transforms to a deformational term $\mathbf{f}^{\mathbf{e}}(\mathbf{u},t) = \mathbf{f}(\mathbf{u},t)$, as well as net translational $\mathbf{f}^{\mathbf{v}}(t) = \int_\Omega \mathbf{f}(\mathbf{u},t)\,d\mathbf{u}$ and net torque $\mathbf{f}^{\boldsymbol{\omega}}(t) = \int_\Omega \mathbf{q}(\mathbf{u},t)\times\mathbf{f}(\mathbf{u},t)\,d\mathbf{u}$ terms on the center of mass.

The ordinary differential equations (13a) and (13b) describe $\mathbf{v}$ and $\boldsymbol{\omega}$, the translational and rotational motion of the body's center of mass. The terms on the left hand sides of these equations pertain to the total moving mass of the body as if concentrated at $\mathbf{c}$, the total (vibrational) motion of the mass elements about the reference component $\mathbf{r}$, and the total damping of the moving mass elements. The partial differential equation (13c) describes (relative to ϕ) the deformation $\mathbf{e}$ of the model away from $\mathbf{r}$. Each term is a dynamic per-mass-element force: (i) the basic inertial force, (ii) the inertial force due to linear acceleration of ϕ, (iii) the centrifugal force due to the rotation of ϕ, (iv) the Coriolis force due the velocity of the mass elements in ϕ, (v) the transverse force due to the angular acceleration of ϕ, (vi) the damping force, and (vii) the restoring force due to deformation away from $\mathbf{r}$.

References

1. **Barr, A., Barzel, R., Haumann, D., Kass, M., Platt, J., Terzopoulos, D., and Witkin, A.,** Topics in physically-based modeling, ACM SIGGRAPH '87 Course Notes, Vol. 17, Anaheim, CA, 1987.
2. **Fournier, A., Bloomenthal, J., Oppenheimer, P., Reeves, W.T., and Smith, A.R.,** The modeling of natural phenomena, ACM SIGGRAPH '87 Course Notes, Vol. 16, Anaheim, CA, 1987.
3. **Armstrong, W.W., and Green, M.,** "The dynamics of articulated rigid bodies for purposes of animation," *The Visual Computer*, **1**, 1985, 231–240.
4. **Wilhelms, J., and Barsky, B.A.,** "Using dynamic analysis to animate articulated bodies such as humans and robots," *Proc. Graphics Interface '85*, Montreal, Canada, 1985, 97–104.
5. **Girard, M., and Maciejewski, A.A.,** "Computational modeling for the computer animation of legged figures," *Computer Graphics*, **19**, 3, 1985, (Proc. SIGGRAPH), 263–270.
6. **Barzel, R., and Barr, A.,** Dynamic Constraints, 1987, in [1].
7. **Hoffmann, C.M., and Hopcroft, J.E.,** "Simulation of physical systems from geometric models," *IEEE Journ. Robotics and Automation*, **RA-3**, 3, 1987, 194–206.
8. **Issacs, P.M., and Cohen, M.F.,** "Controlling dynamic simulation with kinematic constraints, behavior functions, and inverse dynamics," *Computer Graphics*, **21**, 4, 1987, (Proc. SIGGRAPH) 215–224.
9. **Weil, J.,** "The synthesis of cloth objects," *Computer Graphics*, **20**, 4, 1986, (Proc. SIGGRAPH), 49–54.
10. **Feynman, C.R.,** Modeling the Appearance of Cloth, MSc thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1986.
11. **Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K.,** "Elastically deformable models," *Computer Graphics*, **21**, 4, 1987, (Proc. SIGGRAPH) 205–214.
12. **Haumann, D.,** Modeling the physical behavior of flexible objects, 1987, in [1].
13. **Weil, J.,** "Animating cloth objects," *unpublished manuscript*, 1987.
14. **Terzopoulos, D., and Witkin, A.,** "Physically-based models with rigid and deformable components," *Proc. Graphics Interface '88*, Edmonton, Canada, June, 1988.
15. **Alfrey, T.,** *Mechanical Behavior of High Polymers*, Interscience, New York, NY, 1947.
16. **Kardestuncer, H., and Norrie, D.H., (ed.),** *Finite Element Handbook*, McGraw–Hill, New York, NY, 1987.
17. **Christiansen, H.N.,** "Computer generated displays of structures in vibration," *The Shock and Vibration Bulletin*, **44**, 2, 1974, 185–192.
18. **Christiansen, H.N., and Benzley, S.E.,** "Computer graphics displays of nonlinear calculations," *Computer Methods in Applied Mechanics and Engineering*, **34**, 1982, 1037–1050.
19. **Shephard, M.S., and Abel, J.F.,** Interactive computer graphics for CAD/CAM, 1987, in [16], Section 4.4.3.
20. **Christensen, R.M.,** *Theory of viscoelasticity, 2nd ed.*, Academic Press, New York, NY, 1982.
21. **Mendelson, A.,** *Plasticity—Theory and Application*, Macmillan, New York, NY, 1968.
22. **Sih, G.C.,** *Mechanics of Fracture*, Martinus Nijhoff, The Hague, 1981.
23. **Goldstein, H.,** *Classical Mechanics*, Addison–Wesley, Reading, MA, 1950.
24. **Courant, R., and Hilbert, D.,** *Methods of Mathematical Physics*, Vol. I, Interscience, London, 1953.
25. **Terzopoulos, D.,** "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **PAMI-8**, 1986, 413–424.
26. **Lapidus, L., and Pinder, G.F.,** *Numerical Solution of Partial Differential Equations in Science and Engineering*, Wiley, New York, NY, 1982.
27. **Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T.,** *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1986.
28. **Zienkiewicz, O.C.,** *The Finite Element Method; Third edition*, McGraw–Hill, London, 1977.
29. **Hackbusch, W.,** *Multigrid Methods and Applications*, Springer–Verlag, Berlin, 1985.
30. **Hansen, C., and Henderson, T.,** UTAH Range Database, Dept. of Computer Science, University of Utah, Salt Lake City, Utah, TR No. UUCS-86-113, 1986.
31. **Terzopoulos, D.,** "Multilevel computational processes for visual surface reconstruction," *Computer Vision, Graphics, and Image Processing*, **24**, 1983, 52–96.
32. **Fleischer, K., and Witkin, A.,** "A modeling testbed," *Proc. Graphics Interface '88*, Edmonton, Canada, June, 1988.