Demetri Terzopoulos*†¶
John Platt‡
Kurt Fleischer§

*Department of Computer Science, University of Toronto, Toronto, ON M5S 1A4, Canada

†Schlumberger Laboratory for Computer Science, P.O Box 200015, Austin TX 78720, U.S.A.

‡Synaptics, Inc., 2860 Zanker Road, San Jose, CA 95134, U.S.A.

§California Institute of Technology, Pasadena, CA 91125, U.S.A.

Heating and Melting Deformable Models

SUMMARY **We develop physically-based graphics models of non-rigid objects capable of heat conduction, thermoelasticity, melting and fluid-like behaviour in the molten state. These deformable models feature non-rigid dynamics governed by Lagrangian equations of motion and conductive heat transfer governed by the heat equation for non-homogeneous, non-isotropic media. In its solid state, the discretized model is an assembly of hexahedral finite elements in which thermoelastic units interconnect particles situated in a lattice. The stiffness of a thermoelastic unit decreases as its temperature increases, and the unit fuses when its temperature exceeds the melting point. The molten state of the model involves a molecular dynamics simulation in which 'fluid' particles that have broken free from the lattice interact through long-range attraction forces and short-range repulsion forces. We present a physically-based animation of a thermoelastic model in a simulated physical world populated by hot constraint surfaces.**

KEY WORDS: Modelling Deformable models Liquids Animation Dynamics Simulation

1. INTRODUCTION

Methods for modelling non-rigid objects and their motions are attracting considerable attention in computer graphics. Deformable models[1] are physically-based models of non-rigid curves, surfaces and solids which are finding many interesting applications. The modelling and animation of cloth[2] saw the first successful application of elastic surface models.[1,3–6] Deformable 'characters' have been animated in simulated physical worlds.[7] Physically-based constraint methods have been developed for controlling deformable model animations.[8,9] 'Muscle' actuators have been incorporated into deformable models to synthesize self-locomoting snakes and worms.[10] Inelastic models, a type of 'computational modelling clay', appear promising as an interactive medium for free-form shape design in CAD/CAM.[11] Deformable models are also applicable to the physically-based simulation of facial tissue for real-time facial animation.[12]

A general formulation of deformable models based on elasticity theory was first proposed in Reference 4 and was expanded subsequently to include inelastic behaviours, such as plasticity.[8,11] In this paper, we extend deformable models further to include the simulation of thermal phenomena. In the real world, rigid and non-rigid objects absorb, radiate and conduct heat. Heat causes solid materials to soften and eventually melt into fluids.

We construct thermoelastic models whose shapes and dynamics are governed not only by the Lagrange equation of non-rigid motion that underlies our previous deformable models, but also by the heat equation, a partial differential equation that describes a range of diffusive phenomena. Our thermoelastic models interact non-rigidly with their simulated physical environment, as do prior deformable models. As soon as they come into contact with 'hot' graphics objects, however, the new models begin to conduct heat into their interiors. They exhibit thermoelastic effects—as their temperature rises, they become softer and more pliable. When the temperature exceeds the melting point, the solid models melt into simple molecular fluids. Following Greenspan[13], we take a molecular dynamics approach[14] to simulating the fluid state, in which pairs of fluid particles interact through long-range attraction forces and short-range repulsion forces.

The remainder of this paper is structured as follows: Section 2 reviews the equations of motion for elastically deformable solids, and Section 3 reviews the equation that governs conductive heat transfer in solids. In Section 4 we incorporate both differential equations to create a discrete heat-conducting deformable model. Section 5 explains how we simulate thermoelasticity and melting effects. Section 6 describes the interaction forces underlying our discrete fluid models. Section 7 explains how we impose constraints and frictional forces to control and increase the realism of our physically-based animations. Section 8 specifies the numerical time integration scheme that we have employed to create the simulation presented in Section 9, which demonstrates constrained non-rigid dynamics, friction, heating, melting and fluid behaviour. Section 10 concludes the paper with some remarks and suggestions for future work.

2. DEFORMABLE SOLIDS

A general formulation of deformable curve, surface and solid models was proposed in References 1 and 11. We review the formulation of deformable solid models in this section.

Let $\mathbf{u} = (u_1,u_2,u_3)$ be the material coordinates of points in the solid model's material domain $\Omega = [0,1]^3$. Let the time-varying positions of material points be

$$\mathbf{x}(\mathbf{u},t) = [x_1(\mathbf{u},t), x_2(\mathbf{u},t), x_3(\mathbf{u},t)]^T \quad (1)$$

where subscripts 1, 2 and 3 denote the X, Y and Z axes in space. The position $\mathbf{x}(\mathbf{u}, t)$, velocity $\partial\mathbf{x}/\partial t$, and acceleration $\partial^2\mathbf{x}/\partial t^2$ specify the model's motion as a function of $\mathbf{u}$ and time t.

The deformable model is governed by the *Lagrange equation* of motion

$$\mu\frac{\partial^2\mathbf{x}}{\partial t^2} + \gamma\frac{\partial\mathbf{x}}{\partial t} + \delta_{\mathbf{x}}\mathscr{E} = \mathbf{f}. \quad (2)$$

This hyperbolic-parabolic partial differential equation dynamically balances the net external forces $\mathbf{f}(\mathbf{u},t)$ against (i) the inertial force due to the mass density $\mu(\mathbf{u})$ of the model, (ii) the velocity-dependent damping force with damping density $\gamma(\mathbf{u})$, and (iii) the model's internal elastic force $\delta_{\mathbf{x}}\mathscr{E}$ which attempts to restore a deformed elastic model to its natural, undeformed shape.

¶Fellow, Canadian Institute for Advanced Research

1049–8907/91/020068–06$05.00

Received 17 February 1990
Revised 12 October 1990

The elastic force is expressed as a variational derivative with respect to $\mathbf{x}$ of a non-negative deformation energy function $\mathscr{E}(\mathbf{x})$. For non-homogeneous, non-isotropic deformable solids, we proposed the functional

$$\mathscr{E}(\mathbf{x}) = \iiint_{\Omega} \|\mathbf{G} - \mathbf{G}^0\|_{\mathbf{w}}^2 \, du_1 du_2 du_3 \quad (3)$$

where $\|\cdot\|_{\mathbf{w}}$ is a weighted matrix norm, i.e. $\|\mathbf{A}\|_{\mathbf{w}}^2 = \Sigma_{i,j} w_{ij} a_{ij}^2$, where a_{ij} are the entries of matrix $\mathbf{A}$ and $w_{ij}(\mathbf{u})$ are non-negative weighting functions. Here $\mathbf{G}$ and $\mathbf{G}^0$ denote the metric tensor of the solid in its deformed and undeformed state, respectively. $\mathbf{G}$ is a 3×3 symmetric matrix with entries[15]

$$G_{ij}(\mathbf{x}) = \frac{\partial \mathbf{x}}{\partial u_i} \cdot \frac{\partial \mathbf{x}}{\partial u_j} \quad (4)$$

The functional $\mathscr{E}$ is designed to be invariant with respect to rigid-body motions of the model in space, since such motions impart no deformation. $\mathscr{E}$ is zero for the model in its natural shape and grows with increasing deformation away from the natural shape. The weighting functions w_{ij} control the rate of growth of the deformation energy and, hence, the strength of the elastic restoring forces.

Animating the deformable solid model amounts to solving an initial-boundary-value problem for (2) with (3), given appropriate conditions for $\mathbf{x}$ on the boundary $\partial\Omega$ of the material domain, and given the initial position $\mathbf{x}(\mathbf{u},0)$ and velocity $\partial\mathbf{x}/\partial t|_{\mathbf{u},0}$.

3. CONDUCTIVE HEAT TRANSFER

Heat is thermal energy. The associated potential function is temperature θ. The basic (macroscopic) conductive heat transfer phenomena are:

1. The amount of heat required to raise the temperature of a small material sample $\Delta\theta$ degrees is proportional to $\Delta\theta$ and the mass of the sample. The proportionality factor σ is called the specific heat and is a property of the material.
2. Heat is conducted from high temperature to low temperature. More specifically, the rate of heat conduction per unit area is inversely proportional to the gradient of the temperature. The proportionality factor c, known as the thermal conductivity, is another property of the material.

3.1 The heat equation

The *heat equation* describes the diffusion of heat in materials. In the case of solids, the equation governs the temperature distribution $\theta(\mathbf{u},t)$. Assuming mass density $\mu(\mathbf{u})$ and specific heat σ, and introducing the gradient operator in material co-ordinates $\nabla = [\partial/\partial u_1, \partial/\partial u_2, \partial/\partial u_3]^T$, we can write the general heat equation as

$$\frac{\partial}{\partial t}(\mu\sigma\theta) - \nabla \cdot (\mathbf{C}\nabla\theta) = q \quad (5)$$

where $q(\mathbf{u},t)$ is the rate of heat generation (or loss) per unit volume in the solid and $\mathbf{C}(\mathbf{u})$ is a 3×3 symmetric matrix, known as the thermal conductivity matrix.

It is always possible to determine locally a principal co-ordinate system wherein $\mathbf{C}$ becomes a diagonal matrix with the three principal thermal conductivities $c_1(\mathbf{u})$, $c_2(\mathbf{u})$, $c_3(\mathbf{u})$ along the main diagonal. If the principal axes happen to coincide globally with the material co-ordinates, the heat equation simplifies to

$$\frac{\partial}{\partial t}(\mu\sigma\theta) + \frac{\partial}{\partial u_1}\left(c_1 \frac{\partial\theta}{\partial u_1}\right) + \frac{\partial}{\partial u_2}\left(c_2 \frac{\partial\theta}{\partial u_2}\right) + \frac{\partial}{\partial u_3}\left(c_3 \frac{\partial\theta}{\partial u_3}\right) = q \quad (6)$$

For a homogeneous and isotropic material, $\mathbf{C}(\mathbf{u}) = c(\mathbf{u})\mathbf{I}$, where $\mathbf{I}$ is the identity matrix, and the heat equation reduces to its most familiar form

$$\frac{\partial}{\partial t}(\mu\sigma\theta) + c\nabla^2\theta = q \quad (7)$$

where $\nabla^2 = \partial^2/\partial u_1^2 + \partial^2/\partial u_2^2 + \partial^2/\partial u_3^2$ is the Laplacian.

3.2. Boundary conditions

The heat equation is a parabolic partial differential equation. Its solution in the material domain Ω of a deformable solid requires conditions on the domain's boundary $\partial\Omega$. Through boundary conditions we can describe the gain (loss) of heat by our model from (to) the outside world. The following boundary conditions are useful:

1. *Dirichlet condition*, i.e. specified temperature:

$$\theta = \bar{\theta}, \text{ on } \partial\Omega \quad (8)$$

where $\bar{\theta}$ is the given boundary temperature function.

2. *Newton condition*, i.e. specified normal component of heat flow $\eta = -(\mathbf{C}\nabla\theta)\cdot\hat{\mathbf{n}}$ and radiative heat loss on the boundary:

$$-(\mathbf{C}\nabla\theta)\cdot\hat{\mathbf{n}} - \rho\theta = \bar{\eta}, \text{ on } \partial\Omega \quad (9)$$

where $\hat{\mathbf{n}}$ is the unit normal function on the boundary, $\bar{\eta}$ is the specified normal component of flow, and ρ is a specified (non-negative) radiation coefficient. We obtain the *Neumann condition* for the special case $\rho = 0$.

3. *Mixed conditions*, Dirichlet, Neumann or Newton conditions may be applied on different portions of $\partial\Omega$.

4. THE DISCRETE MODEL

The Lagrange equation (2) together with the heat equation (5) govern the continuous deformable model. To simulate the equations in the material domain, we must discretize Ω. We can apply local discretization techniques, such as the finite-element or finite-difference methods.[16]

We divide the domain into finite-element subdomains. A convenient approach is to tessellate Ω into hexahedra whose vertices are occupied by nodes which represent point masses or particles. The deformation of each hexahedron is dictated by a discrete approximation to the deformation potential energy (3).

According to (4), the diagonal terms of the metric tensor, $G_{ii}, i = 1,2,3$, dictate lengths in the solid along the co-ordinate directions u_i, whereas the off-diagonal terms, G_{ij}, $i \neq j$, express angles between directions u_i and u_j. Within an infinitesimal material volume $du_1 du_2 du_3$, the integrand in (3) aims to restore the distances and angles to their reference values, as measured by $\mathbf{G}^0$. The magnitude of the $w_{ij}(\mathbf{u})$ within the volume determine the strength of the restoring forces.

We assemble finite-length, non-linear spring units along the twelve edges and diagonally across the six faces of the hexahedral element in order to restore the distances and angles expressed in $\mathbf{G}$ (Figure 1). Spring l will have its own natural length L_l, set according to $\mathbf{G}^0$, to determine the natural shape of the element, as well as stiffness K_l, dictated by the w_{ij}, to determine its deformation properties.

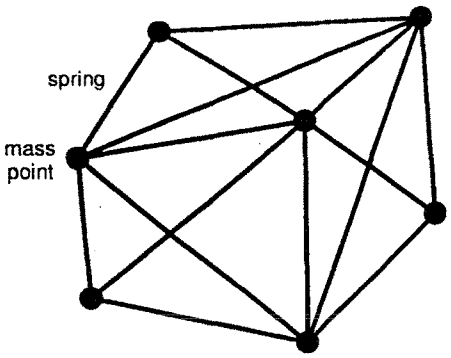

Figure 1. A hexahedral assembly of particles and springs

Next, we assemble the hexahedral elements to cover Ω, such that adjacent elements share nodes and springs on common faces. We index the nodes in the resulting 3D lattice by k. The nodal position variables $\mathbf{x}_k$ specify the 3-space locations of the particles, and the variables $\mathbf{v}_k$, their velocities.

We also associate a temperature variable θ_k with each node. The nodal temperature variables are governed by the heat equation for the case of a non-homogeneous, non-isotropic conductive medium. A convenient approach to discretizing the partial derivatives with respect to u_i in the $\nabla\cdot(\mathbf{C}\nabla\theta)$ term in (5), given our finite-element model, is to associate a particular value of heat resistance R_l per unit length to each spring. Assuming that spring l connects node i to node j, its conductance is $C_l = (R_l\|\mathbf{x}_i - \mathbf{x}_j\|)^{-1}$. A conducting spring will tend to equalize the temperatures of the two nodes it connects. The finite-element assembly approximates the general heat equation (5) over the discrete lattice.

If we permit heat conduction only along the material co-ordinate axes (by zeroing the conductivities of the springs running diagonally across element faces), then the finite-element assembly will approximate equation (6). In this case, one can show that the resulting discrete equations consist of central finite difference expressions for the terms involving partial derivatives with respect to material co-ordinates in (6).

5. THERMOELASTICITY AND MELTING

Real materials typically soften when heated, a phenomenon known as thermoelasticity.

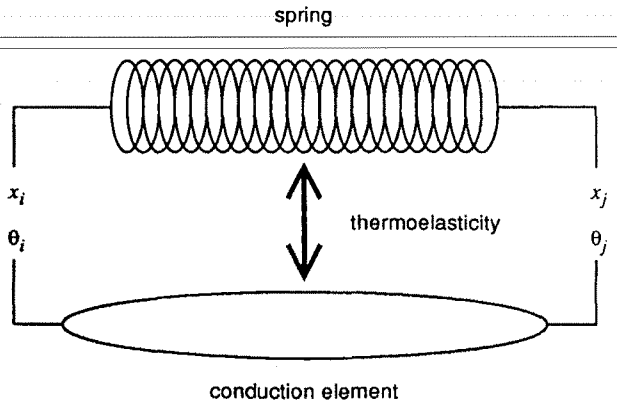

Figure 2. A thermoelastic unit

Eventually, materials melt as the temperature increases. It is straightforward to simulate thermoelasticity and melting in our heat-conducting deformable models—we establish a relationship between the temperature variables θ_k and the stiffnesses K_l of the spring units in the discrete model (Figure 2).

To simulate softening, we make a thermoelastic unit whose stiffness varies inversely with the temperature averaged over the two nodes it connects: $\theta^a = (\theta_i+\theta_j)/2$. The variation may be non-linear, e.g. we can initiate thermoelastic behaviour when θ^a exceeds a specified threshold θ^s.

To simulate melting, we fuse the thermoelastic unit whose average temperature exceeds the melting point θ^m, by setting its stiffness K_l to zero.

We incorporate the following thermoelasticity/melting law:

$$K_l = \begin{cases} K_l^0 & , \text{ if } \theta^a \le \theta^s \\ K_l^0 - \nu(\theta^a - \theta^s) & , \text{ if } \theta^s < \theta^a < \theta^m \\ 0 & , \text{ if } \theta^a \ge \theta^m \end{cases} \quad (10)$$

where K_l^0 is the zero-degree stiffness and ν is a positive constant. The second case in (10) defines the thermoelastic region, which states that the elastic force will be linearly related to the displacement minus a component which is proportional to the temperature. This is known as the Duhamel–Neumann law of thermoelasticity.[17]

6. A DISCRETE FLUID MODEL

When all the thermoelastic units that bond a particle to other particles in the lattice have fused, the particle breaks free from the deformable solid. It can then interact freely with other particles, as do molecules in a fluid at the microscopic level.[18]

Greenspan[13] investigated various N-body systems of this sort as discrete models of solid, liquid and gaseous media. Recent computer animations of 'fluids', due to Miller, are apparently based on similar ideas.[19] Over the years, much attention has been given in the physics and chemistry literature to the development of discrete fluid models involving aggregate molecular dynamics in which the molecules are subject to various interaction potentials.[14] A basic technique is to model long-range attraction and short-range repulsion forces between pairs of particles according to potentials of the Lennard–Jones type, which lead to forces involving inverse powers of particle separation distance d.[18]

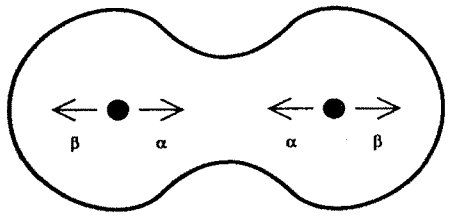

Figure 3. Fluid particles both attract and repel each other

Following Reference 13, we choose a force which has a component of attraction that behaves like αd^{-a} and a component of repulsion that behaves like βd^{-b}, where a and b are non-negative parameters with $0 \le a \le b$ (Figure 3).

Specifically, let particle i have mass m_i and be located at $\mathbf{x}_i(t)$ at time t. Let particle j have mass m_j and be located at $\mathbf{x}_j(t)$ at time t. Let $d_{ij}(t) = \|\mathbf{x}_i - \mathbf{x}_j\|$ be the separation of the two particles. Then we define the force on particle i exerted by particle j as

$$\mathbf{g}_{ij}(t) = m_i m_j (\mathbf{x}_i - \mathbf{x}_j)\left(-\frac{\alpha}{(d_{ij}+\zeta)^a} + \frac{\beta}{(d_{ij})^b}\right) \quad (11)$$

where α and β are non-negative parameters that determine the strength of the attraction and repulsion components of the force, and ζ is a positive measure of how close the particles are allowed to be.

To model inter-particle collisions, we can define

$$\beta = \beta'\left(\frac{d_{ij}}{r_i}\right)^{-p} \quad (12)$$

where β' is a non-negative repulsion strength, r_i is the non-negative collision radius of particle i, and

$$p = \begin{cases} \ge 0 & , \text{ when } d_{ij} < r_i \\ 0 & , \text{ otherwise} \end{cases} \quad (13)$$

is the collison exponent, whose effect is to increase the repulsion force during collision.

The total force on particle i due to all other particles is

$$\mathbf{g}_i(t) = \sum_{j \ne i} \mathbf{g}_{ij}. \quad (14)$$

Then, the discrete version of the Lagrange equation (2) gives the equations of motion for the ensemble of particles

$$m_i \frac{\partial^2 \mathbf{x}_i}{\partial t^2} + \gamma_i \frac{\partial \mathbf{x}_i}{\partial t} + \mathbf{g}_i = \mathbf{f}_i, \; i = 1, \ldots, N \quad (15)$$

7. CONSTRAINTS AND FRICTION

The various parameters of our physically-based model afford control over its animation, as do the initial conditions of the simulation. Moreover, it is possible to control the animation through physically-based constraints. We have applied several constraint mechanisms to our non-rigid models, just as Barzel and Barr[20] have done for the animation of rigid and articulated models.

We use reaction constraints[8,9] to expel the particles of an evolving solid or fluid model out of any impenetrable obstacles in the scene (Figure 4). Reaction constraints cancel force components normal to the surface of an obstacle that would take particles into an obstacle, and substitute forces which induce critically damped motion that converts penetration into mere contact.

It is simple to express reaction constraints for objects constructed of planar polygonal patches. Let $P(\mathbf{x}) = ax+by+cz+d$ be the plane equation of a polygon and let $Q(\mathbf{v}) = av_x+bv_y+cv_z$. If a particle with mass m_i has penetrated the obstacle through a patch, then the reaction force on the point acts normal to the polygon and is proportional to

$$\mathbf{f}_R = m_i\left(\frac{P(\mathbf{x})}{\tau^2} + \frac{2Q(\mathbf{v})}{\tau}\right)\hat{\mathbf{n}} \quad (16)$$

where τ is the time constant of the critically damped motion and where $\hat{\mathbf{n}} = [a,b,c]/\|a,b,c\|$ is the unit inward normal. In the absence of friction, the component of force tangent to the polygon remains unchanged.

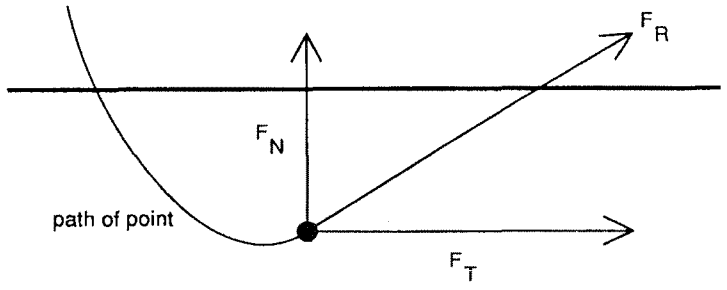

Figure 4. Reaction constraints expel particles from impenetrable obstacles

friction effects lend a greater degree of realism to the animation of physically based models.[21] A simple treatment of friction involves adding a force which opposes the velocity of a particle. More realistically, however, a particle will stick to a surface until the force on the particle exceeds a threshold known as the static friction.

We use a standard friction model (see also Reference 9). Consider a particle in contact with a polgon and experiencing a net force $\mathbf{f}$ (before modification by reaction constraints). The normal force is $\mathbf{f}_N = (\mathbf{f} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$. The tangential force prior to applying friction is

$$\mathbf{f}_T = \mathbf{f} - \mathbf{f}_N \tag{17}$$

If the tangential force is less than the static friction, then the particle begins to stick and quickly comes to a halt ($\mathbf{v}_T = \mathbf{f}_T = \mathbf{0}$), otherwise a kinetic frictional force acts tangentially to the surface to retard sliding. The static and kinetic frictions are proportional to the magnitude of the normal force into the surface. The coefficient of static friction ξ is always larger than the coefficient of kinetic friction κ. The tangential force modified by friction is therefore

$$\mathbf{f}' = \begin{cases} -\dfrac{m}{\tau}\mathbf{v}_T & , \text{ if } \|\mathbf{f}_T\| < \xi\|\mathbf{f}_N\| \\ \mathbf{f}_T - \kappa\|\mathbf{f}_N\|\mathbf{v}_T, & \text{otherwise} \end{cases} \tag{18}$$

where $\mathbf{v}_T = \mathbf{v} - (\mathbf{v}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}}$ is the tangential velocity, and τ is the time constant for halting the motion in the static friction case.

8. NUMERICAL TIME-INTEGRATION

To simulate the dynamics of our models we provide the initial positions $\mathbf{x}_i^0$ and velocities $\mathbf{v}_i^0$ of particle i for $i = 1, \ldots, N$. At each subsequent time step, $\Delta t, 2\Delta t, \ldots, t, t + \Delta t, \ldots$, we evaluate the current accelerations, new velocities, and new positions using the explicit Euler time-integration procedure:

$$\begin{aligned} \mathbf{a}_i^t &= \frac{\mathbf{f}_i^t}{m_i} \\ \mathbf{v}_i^{t+\Delta t} &= \mathbf{v}_i^t + \Delta t \mathbf{a}_i^t \\ \mathbf{x}_i^{t+\Delta t} &= \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+\Delta t} \end{aligned} \tag{19}$$

The quantity $\mathbf{f}_i$ is the total force acting on particle i. This includes a sum of the damping force $-\gamma_i \mathbf{v}_i$, the elastic forces from the (discretized) third term in (2), the fluid interaction forces from the third term in (15), the external forces on the right-hand sides of these equations, as well as all modifications made to these forces in order to apply constraints and friction as was described in the previous section.

9. A GOOP-TO-GLOP* SIMULATION

Figure 5 presents a selection of frames from an animation involving the physically-based techniques developed in this paper. The scenario is to drop a thermoelastic solid into a 'funnel', and by heating the funnel to first soften the solid, then to melt it until it dribbles onto the hot floor underneath. The model simulated in Figure 5 consists of only 250 particles. Although large enough to yield an interesting animation, this model is much too coarse to match the accuracy of sophisticated physical models intended for the analysis of specific real-world solids and fluids. We therefore refer affectionately to our overly coarse simulated solids as 'goop' and the simulated fluids into which they melt as 'glop'. By using more particles in our models (at increased computational cost), however, we may achieve increasingly accurate approximations to real-world solids and fluids under certain physical conditions.

Figure 5(a) is a bird's eye view of three planes in a funnel-like arrangement over a ground plane. The planes present obstacles to the deformable models, and contact with their surfaces produces friction. We applied the techniques described in Section 7 to produce these planar, physically-based constraints. The bluish-green colour of the planes indicates that they are cold (0°).

Figure 5(b) is a frame early in the simulation which shows a frontal view of a white piece of goop dropping, due to gravity, into the mouth of the funnel. The goop is a heat-conducting deformable model discretized on a $5 \times 5 \times 10$ lattice of nodes. The nodes in the model were rendered as 'blobbies'.[22] The blobby rendering technique associates an exponential potential function with each node and efficiently ray traces an isopotential surface of the resulting field. We chose an exponential decay rate such that neighbouring nodes of the model fuse together into a plump, continuous form.

Figures 5(c) and (d) show the goop colliding first with the left surface, and finally coming to rest in the funnel. We used significant static friction to make the walls of the funnel quite sticky, as indicated by the deformation in Figure 5(a). Up to the simulation time of Figure 5(c), the goop was cold (0°).

Next, the funnel surfaces were heated to a temperature of 5° (Figure 5(e)). Figures 5(e) and (f) show the goop conducting heat. The temperature begins to rise at the corners of the goop where it comes in contact with the hot surfaces, then spreads throughout the interior. The heat diffuses into the solid through Dirichlet boundary conditions (Section 3.2) which are automatically introduced at nodes in contact with the funnel. To visualize the temperature distribution, the larger θ_i, the more intensely red we colour the surrounding blobby.

Next, in Figure 5(g), the temperature of the funnel has been set to 7°, entering the thermoelastic regime of the goop ($\theta^s = 6°$ in (10)). We see the goop softening and sagging deeper into the funnel under its own weight.

Figures 5(h)–(l) show what happens after we have set the temperature of the surfaces to 10°, exceeding the goop's melting point ($\theta^m = 8°$ in equation (10)). First the goop collapses (Figure 5(h)), then melts into glop as the thermoelastic units connecting nodes near hot surfaces begin to fuse (Figure 5(i)). As more and more of the goop melts, 'gloplets' dribble through the funnel opening onto the hot floor below. In our first simulation attempt using the above temperatures, a semi-liquid mixture of goop and glop fell through the funnel opening and accumulated on the floor. For demonstration purposes, however, we wanted only liquid glop to fall from the funnel. We therefore incorporated into the second simulation an additional step which automatically melts all the springs connected to nodes that have flowed through the funnel opening. Figures 5(i)–(l) show the result.

We used $a = 2$, $b = 4$, $\alpha = 1{\cdot}0$ and $\beta = 1{\cdot}0 \times 10^4$ in (11) (and $p = 0$ in (12)), which makes the gloplets spread viscously on the hot floor (Figure 5(l)). Increasing α would thicken the consistency of the fluid, whereas increasing β would increase its incompressibility.

10. DISCUSSION AND EXTENSIONS

This paper developed deformable models that conduct heat, exhibit thermoelastic phenomena and melt into molecular fluids. We conclude by placing our approach into perspective and suggesting possible extensions and variations.

Greenspan[13] suggests discrete solid models which are based on molecular dynamics that are conceptually similar to his fluid models. Instead of incorporating the heat equation, a macroscopic law involving the thermodynamic quantity temperature, his models regress to the microscopic level, treating heat as the kinetic energy of random molecular vibration of particles and temperature as the time-average of this kinetic energy. A different class of discrete models are the cellular automaton fluids proposed by Wolfram.[23] These are discrete analogues of molecular dynamics, in which ensembles of particles with discrete velocities populate the links of a fixed array of sites that subdivide the space occupied by the fluid. Greenspan's approach is a discrete version of the Lagrange formulation of fluid dynamics, whereas Wolfram's is a discrete version of the Euler formulation.[16]

Our model is a convenient blend of elasticity and heat transfer in solids and the molecular dynamics of fluids. Because of the lattice infrastructure, the elastic forces in a solid model are computable in $O(N)$ time, where N is the number of nodes. However, computing the fluid forces brute-force takes $O(N^2)$ time. It is fairly easy to reduce this to $O(N \log N)$ by clustering particles hierarchically.[24] The problem of further reducing the complexity of force computations in N-body systems has recently attracted attention with the development of $O(N)$ algorithms for Coulombic field interactions.[25,26] It remains to be seen whether this linear-time approach generalizes to non-Coulombic fields of the type used in our fluid model.

The work in this paper can be extended in various interesting directions. By incorporating the heat equation into the inelastic models described in References 1, 8 and 11, we may straightforwardly generalize our techniques to include inelastic behaviour, such as thermoplasticity. The incorporation of these ideas within an interactive modelling environment would allow users to heat or cool simulated modelling materials during a sculpting session in order to modify their malleabilities.

Another straightforward extension to our models would be to simulate heat generation through deformation, a phenomenon evident in many real-world materials (e.g. a quickly stretched rubber band becomes warm). In the heat equation (5), $q(\mathbf{u},t)$ represents the

*Goop: a soft, sticky solid. Glop: a thick, gluey liquid.—With apologies to *Webster's New World Dictionary*.

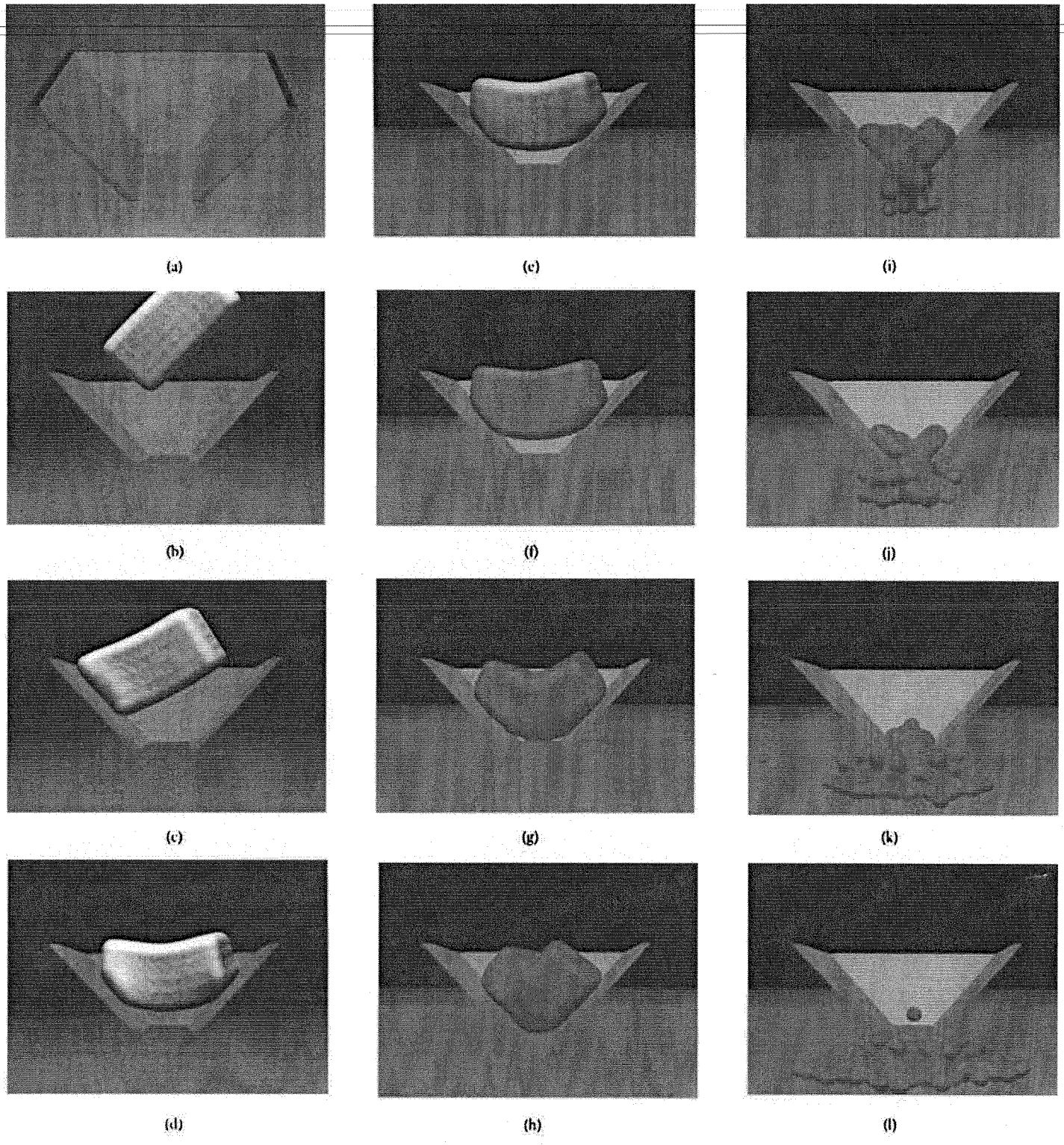

Figure 5. Selected frames from a goop-to-glop animation:
(a) funnel geometry;
(b) goop falls in gravity;
(c) collision with left funnel wall;
(d) at rest in cold funnel;
(e) funnel hot; goop conducts heat through its interior;
(f) temperature distribution at equilibrium;
(g) funnel temperature increases; goop softens and sags;
(h) funnel temperature at melting point; goop collapses;
(i) goop begins melting into glop;
(j), (k) gloplets dribble to hot floor;
(l) glop spread out on hot floor (note gloplet sticking to back funnel wall)

rate of internal heat generation. In our discrete model, we assign to each node a heat-generation nodal variable $q_i(t)$ whose value depends on the average deformation rate of the thermoelastic units connected to that node. The heat equation will diffuse the deformation-induced heat through the model, along with any heat transferred from the outside world through boundary conditions. This paper treated contact with hot objects, but another obvious extension is to transfer into a thermoelastic model the heat generated by friction as it slides against other objects.

If we introduce a boiling point, a mechanism for modelling evaporation into a gaseous state would be virtually in place. When the specified boiling point is exceeded by a fluid particle, we can alter the parameters of its interaction force to model a gas particle, i.e. in (11) we make $\alpha = 0$ and increase β, so that the particles will tend to fill the available space like a gas. Such a molecular gas may be used directly to model the convection of heat from the surfaces of hot models.

The modelling of radiative heat transfer would be another natural extension to the work presented in this paper. We can apply the Newton boundary condition given in Section 3.2 and treat the emitted heat as infrared radiation. The amount of heat which would be transmitted to nearby heat-conducting models is specified by the rendering equation.[27] Efficient radiosity algorithms[28] would come in handy for such computations.

REFERENCES

1. D. Terzopoulos and K. Fleischer, 'Deformable models', *The Visual Computer*, **4**, (6), 306–331 (1988).
2. J. Weil, 'The synthesis of cloth objects', *Computer Graphics (Proc. SIGGRAPH)*, **20**, (4), 49–54 (1986).
3. C. R. Feynman, 'Modeling the appearance of cloth', *MSc thesis*, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1986.
4. D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, 'Elastically deformable models', *Computer Graphics (Proc. SIGGRAPH)*, **21**, (4), 205–214 (1987).
5. D. Lundin, 'Ruminations of a model maker', *IEEE Computer Graphics and Applications*, **7**, (5), 3–5 (1987).
6. D. Haumann, 'Modeling the physical behavior of flexible objects', in A. Barr *et al.* (eds), *Topics in Physically-based Modeling*, ACM SIGGRAPH '87 Course Notes, Vol. 17, Anaheim, CA, 1987.
7. D. Terzopoulos and A. Witkin, 'Physically-based models with rigid and deformable components', *IEEE Computer Graphics and Applications*, **8**, (6), 41–51 (1988).
8. J. Platt and A. Barr, 'Constraint methods for flexible models', *Computer Graphics (Proc. SIGGRAPH)*, **22**, (4), 279–288 (1988).
9. J. Platt, 'Constraint methods for neural networks and computer graphics', *Ph.D. Thesis*, Department of Computer Science, California Institute of Technology, Pasadena, CA, 1989.
10. G. Miller, 'The motion dynamics of snakes and worms', *Computer Graphics (Proc. SIGGRAPH)*, **22**, (4), 169–178 (1988).
11. D. Terzopoulos and K. Fleischer, 'Modeling inelastic deformation: viscoelasticity, plasticity, fracture', *Computer Graphics (Proc. SIGGRAPH)*, **22**, (4), 269–278 (1988).
12. D. Terzopoulos and K. Waters, 'Physically-based facial modelling, analysis, and animation', *The Journal of Visualization and Computer Animation*, **1**, (2), 73–80 (1990).
13. D. Greenspan, *Discrete Models*, Addison-Wesley, Reading, MA, 1973.
14. H. J. C. Berendsen and W. F. van Gunsteren, 'Molecular dynamics simulations: techniques and approaches', in A. J. Barnes, W. J. Orville-Thomas and J. Yarwood (eds), *Molecular Liquids—Dynamics and Interactions*, D. Reidel, Dordrecht, Holland, 1984, 475–500.
15. J. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture*, Halstead Press, Horwood, NY, 1981.
16. M. B. Allen, I. Herrera and G. F. Pinder, *Numerical Modeling in Science and Engineering*, Wiley, New York, NY, 1988.
17. S. Timoshenko and J. N. Goodier, *Theory of Elasticity*, McGraw-Hill, New York, NY, 1951.
18. H. N. V. Temperley and D. H. Trevena, *Liquids and Their Properties: A Molecular and Macroscopic Treatise with Applications*, Ellis Horwood, Chichester, U.K., 1978.
19. G. Miller, 'Natural phenomena', *Film and Video Show*, ACM SIGGRAPH '88, Atlanta, GA, 1988.
20. R. Barzel and A. Barr, 'A modeling system based on dynamic constraints', *Computer Graphics (Proc. SIGGRAPH)*, **22**, (4), 179–188 (1988).
21. M. Moore and J. Wilhelms, 'Collision detection and response for computer animation', *Computer Graphics (Proc. SIGGRAPH)*, **22**, (4), 289–298 (1988).
22. J. F. Blinn, 'A generalization of algebraic surface drawing', *ACM Trans. Graphics*, **1**, 235–256 (1982).
23. S. Wolfram, 'Cellular automaton fluids 1: basic theory', Thinking Machines Corp., Cambridge, MA, *CA86-2*, 1986.
24. A. Appel, 'An efficient algorithm for many-body simulation', *SIAM J. Sci. Stat. Comput.*, **6**, (1), 85–103 (1985).
25. L. F. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, MA, 1988.
26. F. Zhao, 'An $O(N)$ algorithm for three-dimensional N-body simulations', MIT Artificial Intelligence Lab., Cambridge, MA, *AI-TR-995*, 1987.
27. J. T. Kajiya, 'The rendering equation', *Computer Graphics (Proc. SIGGRAPH)*, **20**, (4), 143–150 (1986).
28. M. F. Cohen, S. E. Chen, J. R. Wallace and D. P. Greenberg, 'A progressive refinement approach to fast radiosity image generation', *Comuter Graphics (Proc. SIGGRAPH)*, **22**, (4), 75–84 (1988).