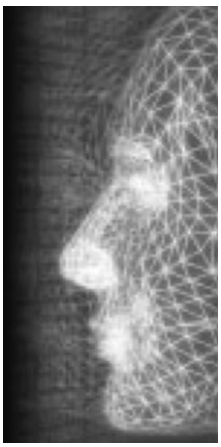
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2007; **18**: 69–82
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/cav.162

WILEY InterScience® DISCOVER SOMETHING GREAT

A unified particle model for fluid–solid interactions



By Barbara Solenthaler, Jürg Schläfli and Renato Pajarola*

We present a new method for the simulation of melting and solidification in a unified particle model. Our technique uses the Smoothed Particle Hydrodynamics (SPH) method for the simulation of liquids, deformable as well as rigid objects, which eliminates the need to define an interface for coupling different models. Using this approach, it is possible to simulate fluids and solids by only changing the attribute values of the underlying particles. We significantly changed a prior elastic particle model to achieve a flexible model for melting and solidification. By using an SPH approach and considering a new definition of a local reference shape, the simulation of merging and splitting of different objects, as may be caused by phase change processes, is made possible. In order to keep the system stable even in regions represented by a sparse set of particles we use a special kernel function for solidification processes. Additionally, we propose a surface reconstruction technique based on considering the movement of the center of mass to reduce rendering errors in concave regions. The results demonstrate new interaction effects concerning the melting and solidification of material, even while being surrounded by liquids. Copyright © 2007 John Wiley & Sons, Ltd.

Received: 12 July 2006; Revised: 3 December 2006; Accepted: 8 December 2006

KEY WORDS: physically based simulation; smoothed particle hydrodynamics; fluids; solids; melting; solidification

Introduction

The interaction between fluids, solids, and their surroundings is becoming increasingly important in computer graphics. These interaction processes are physically very complex and very difficult to simulate. It is highly desirable to have a single simulation method which can handle different types of materials and interactions between them as well. This eliminates the need to define an interface for coupling different fluid and solid models. Currently, coupled models are widely used in computer graphics, but the variety of the simulated materials and effects is often constrained by the interfaces between the models. Furthermore, they suffer from the effect that their methods, or the combination thereof, are not appropriate to simulate the whole variety of interaction processes, including phase changes between fluid and solid. Especially melting and solidification which are caused by a surrounding liquid did not receive enough attention yet, although they contribute a lot to a realistic simulation of the interaction of fluids with their environment.

In this paper, we present a unified particle model based on SPH[1] for the simulation of liquids and deformable as well as rigid objects, which eliminates the need to define an interface for coupling the different models. Using this approach, fluids and solids are both represented by particles, each of which knows its own attribute values describing its physical properties of matter. Since each particle interacts with its neighboring particles regardless of the state of matter, we achieve a two-way coupled fluid–solid interaction without any further treatment.

Phase change behavior is already addressed in previous work, but the proposed models are limited in the resulting interaction effects, as can be seen in Table 1. By using our technique the simulation of a wide range of effects is made possible which are not producible with any single of the previous methods alone. Our model can combine the following properties:

*Correspondence to: B. Solenthaler, Visualization and MultiMedia Lab, University of Zurich, Binzmühlestrasse 14, 8050 Zurich, Switzerland. E-mail: solenthaler@ifi.unizh.ch

Copyright © 2007 John Wiley & Sons, Ltd.

	Our model	[4]	[26]	[22]	[10]	[18]	[14]	[15]	[24]
Rigids	x		*			*			*
Elastics	x	+					*	*	
Rigid-elastics[1]	x								
Fluids	x	*		*	*	*	*	*	*
Melting	x	*		+			*	*	*
Solidification	x	+		+			+	+	
Distinction[2]	x		—			—		*	—
Merging[3]	x	+		*			*		
Splitting[3]	x	+		*			*		

Table 1. Overall comparison of the effects to related previous work

*** : Effect is covered in previous work.**

+ : Effect is covered in previous work but with limitations.

—: not applicable.

[1]Objects consisting of rigid as well as elastic parts.

[2]Between multiple close (touching) deformable objects and parts of the same deformable object which are close (touching).

[3]As illustrated in Figure 1.

- *Flexibility of materials*: Support for fluids, elastic and rigid objects, and even the combination of both, that is, elastic and rigid parts, in one single object.
- *Melting and solidification*: A solid body can turn into a fluid when heat is applied and vice versa for cooling. The simulation can handle partial and continuous melting and solidification, even while interacting with a surrounding liquid.
- *Distinction of objects*: The model supports distinction between multiple objects or parts of the same object which are close (touching), as long as no melting is involved in the process.
- *Merging and splitting*: Ability to merge multiple close (touching) objects into one when melting is involved (Figure 1, top) and split objects (Figure 1, bottom) as a result of phase changes.

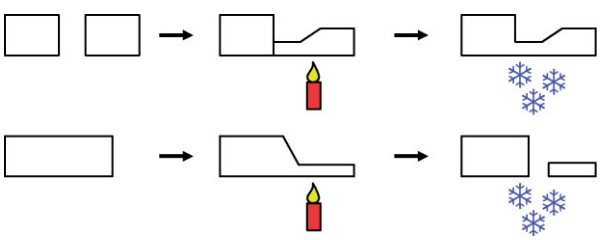


Figure 1. Top: Merging. A solid melts due to heating and touches the other solid in the process. After cooling, the two objects merge to a single one. Bottom: Splitting. A solid melts due to heating and some parts separate. Cooling down leads to two independent solids.

To achieve a smooth surface from particles, we propose a surface reconstruction similar to Reference [2], but with reduced reconstruction artifacts even for inhomogeneously distributed and sparse particles.

Related Work

Lagrangian mesh-based and mesh-free methods are widely used in computer graphics for the simulation of deformable objects as well as for fluids. With the preliminary work of Terzopoulos *et al.*[3], melting of solids into fluids through heat and heat transfer using particle dynamics was introduced in graphics. Their deformable objects are represented by mass-spring systems (MSS) and melting is achieved by varying the spring constants and finally removing the springs. The liquefied particles then interact with Lennard-Jones (L-J) potentials, corresponding to a fluid simulation on the microscopic level. This approach is extended in Reference [4], where the MSS is replaced by different L-J potential energy functions that vary the strength of the attractive and repulsive forces to produce fluid or solid behavior according to the particle temperature. Although both previous works succeeded in melting objects, the identification of the relevant parameters of the L-J interaction forces and the MSS remains a major problem. As discussed in Reference [5], spring constants of a MSS are often chosen arbitrarily since the model does

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

not allow the direct integration of physical parameters. This leads to problems when changing the model resolution as it is not clear how the parameters have to be modified to retain the same behavior. Similar problems exist for L-J potential functions. Another difficulty of MSS is that the behavior of the model is highly dependent on the topology, which is problematic during a solidification process where springs have to be added continuously. The use of different L-J potential functions causes problems during solidification as well, since the equilibrium between gravitational forces and inter-particle forces is shifted, leading to spurious particle expansion or contraction. Using a mesh-free continuum-mechanics-based framework for the animation of elastic objects, as we use in our unified SPH model, offers the advantage of not having to take care about topology at all and that the resolution has only an effect on the accuracy of the method and not on the parameters defining the material properties.

The SPH method was originally developed to model cosmological fluids[1,6] and was introduced to computer graphics in Reference [7]. Later, Desbrun and Cani[8] used SPH for the animation of highly deformable objects, and extended it in Reference [9] to animate lava by coupling the viscosity to temperature. Since then, SPH has been used for a wide range of applications in computer graphics. Reference [10] use SPH for the simulation of fluids at interactive rates. Their work has been extended later to simulate the interaction of fluids with deformable meshes by adding boundary particles to the surface of the mesh[11]. The interaction between multiple SPH fluids with different physical properties is introduced in Reference [12]. Melting and freezing using SPH particles is addressed in Reference [13], where particles are subjected to elastic restoring forces arranging them in a locally defined lattice. Müller *et al.*[14] proposed a technique to model elastic, plastic, and melting behavior of objects using particles, where a Moving Least Squares (MLS) approach is used to calculate the elastic forces. The elastic model is extended in Reference [15], where additionally a method for the handling of topological changes is proposed. Cani and Desbrun[16] presented a method that uses implicit surfaces for animating deformable models. Their elastic objects can collide under low pressure and merge to one object otherwise.

Recent work on the simulation of fluids with *Eulerian* approaches addressed the simulation of different materials and phase changes, as well as interaction processes between fluids and solids. Losasso *et al.*[17] presented the simulation of complex interactions between multiple fluids with different physical properties. Two-way interaction between fluid and solid was introduced in Reference [18], where the rigid objects are treated as a fluid constrained to rigid body motion. The coupling between an Eulerian fluid solver and deformable solids was demonstrated in References [19,20], and coupling water to thin deformable and rigid shells was shown in Reference [21]. A simulation of melting has been presented in Reference [22], where deformable bodies are represented as a very viscous fluid. Melting is made possible by adapting the viscosity depending on the temperature. By adding elasticity to an Eulerian fluid simulation instead of increasing the viscosity, Goktekin *et al.*[23] achieved animations of viscoleastic fluids. Recently, Losasso *et al.*[24] introduced a fluid model coupled with a solid simulator, where the solid objects are represented by meshes. Their simulation can handle the melting and burning of solid objects into liquids and gases, but the solidification process is still a major challenge. Using the lattice Boltzmann method (LBM) Zhao *et al.*[25] demonstrated melting and flowing in a multiphase environment.

We refer to an extensive survey on physically based deformable models in Reference [5]. Concerning the dynamics of rigid bodies a comprehensive introduction is given in the notes of Reference [26].

The model we propose in this work is a Lagrangian approach, which we find to be advantageous for the simulation of mixing processes between different fluids and solids. Our work has been motivated by the fact that previous particle models only fulfill a subset of the desired interaction processes as summarized in Table 1, which makes it difficult to combine them into a single model. Our approach borrows from many prior particle methods, and thus is not fundamentally different from these, but we have enhanced and altered many critical components as described mainly in the Sections 'Elastic Bodies' and 'Rigid Bodies'. Additionally, also an improved surface definition is presented in Section 'Surface Reconstruction'. However, the main contribution is the integration of all the presented modifications and effects into a single unified particle model.

Fluids

The basic Smoothed Particle Hydrodynamics (SPH) model we use for our fluid simulator is based on the work of Müller *et al.*[10] In particular, we use the extensions for the simulation of multiple fluids proposed

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

in Reference [12]. In this section, we give a short summary of the equations we use.

The SPH method belongs to the class of Lagrangian approaches to fluid simulation. In the Lagrangian or particle based approach, fluids are represented by a set of particles $i \in [1 \dots N]$ which carry positions $\mathbf{r}_i$, masses m_i, and additional attributes A_i. SPH defines how to compute the value of any attribute value A at an arbitrary position $\mathbf{r}$ in space by smooth interpolation over the set of all nearby particles j as

$$A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \tag{1}$$

where ρ_i is the density of particle i and $W(\mathbf{r}_{ij}, h)$ is a smoothing kernel with support radius h. The gradient and the Laplacian of this attribute function are calculated simply by using the gradient or the Laplacian of the kernel respectively.

$$\nabla A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \tag{2}$$

$$\nabla^2 A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \tag{3}$$

The density ρ_i can be computed as

$$\rho_i = \rho(\mathbf{r}_i) = \sum_j m_j W(\mathbf{r}_{ij}, h) \tag{4}$$

where $\mathbf{r}_{ij}$ is the distance vector $\mathbf{r}_i - \mathbf{r}_j$.

Using particles, the simplified version of the Navier-Stokes equation for incompressible fluids (Equation (5)) can be expressed as shown in Equation (6), where $\mathbf{a}$ corresponds to the acceleration and is integrated using the Leap-Frog scheme.[27]

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v} \tag{5}$$

$$\rho \mathbf{a} = \mathbf{f}^{\text{pressure}} + \mathbf{f}^{\text{external}} + \mathbf{f}^{\text{viscosity}} \tag{6}$$

Substituting the Equations (2) and (3) into the pressure and viscosity terms of the Navier-Stokes equation and symmetrizing them according to References [10,12] yields

$$\mathbf{f}_i^{\text{pressure}} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_{ij}, h) \tag{7}$$

$$\mathbf{f}_i^{\text{viscosity}} = \sum_j \frac{\mu_i + \mu_j}{2} m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_{ij}, h) \tag{8}$$

where p is the pressure, $\mathbf{v}$ the velocity, and μ the viscosity coefficient. The pressure p_i of particle i is computed via the modified ideal gas state equation suggested by Reference [8]

$$p_i = k(\rho_i - \hat{l}_i) \tag{9}$$

where k corresponds to the gas constant and $\hat{l}_i$) to the particle's rest density. For the calculation of the density and the body forces we use the kernels proposed in Reference [10].

Elastic Bodies

Model Extensions

Our method for modeling deformable bodies extends the work of Müller *et al.*[14] and Keiser *et al.*[15], where at every particle position the gradient of displacement from the undeformed (reference) shape of the body is used to compute the strain ϵ, stress σ, and elastic forces $\mathbf{F}^{\text{elastic}}$. However, the approach has been altered significantly:

1. In our model we use an SPH approach instead of MLS, which has the advantage that it can handle coarsely sampled and even coplanar particle configurations, as they often result from phase change processes (Figure 2). The use of SPH affects Equations (14) and (15) in the following section.
2. We have modified the elasticity model fundamentally with a new definition of the reference shape of a body. Instead of referring to an initial, global undeformed reference shape as in Reference [15], we consider a *locally undeformed object* condition. So, instead of storing the position of the reference shape, each particle stores a distance vector to each of its local neighbors. The neighborhood of a particle is defined by the support radius of the SPH smoothing kernel. The locally undeformed object condition is also critical for the merging of multiple bodies into one, and the splitting of bodies as a result of phase changes.

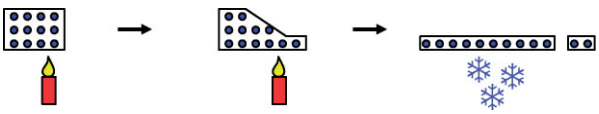


Figure 2. A phase change process may result in coarsely sampled regions and coplanar particle configurations which can be handled by SPH but not by MLS.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

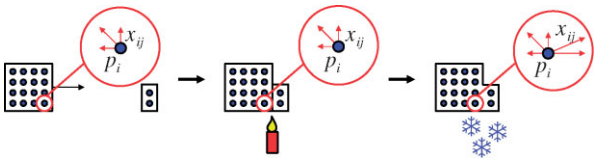


Figure 3. Our model uses a locally undeformed object condition, where each particle stores a distance vector to each of its local neighbors. If objects merge, the reference distances of the new neighbors are added to the particle.

This behavior is illustrated in Figure 3. Without our extension, if two separated and undeformed bodies move and merge during the simulation, large strains will erroneously be measured. The definition of this local reference shape of a body has effects on the computation of the body volume of a particle (see Equation (11)) and on Equation (15).

3. In contrast to Reference [14], the reference neighborhood of a particle does not change during elastoplastic processes. This property allows for the distinction between multiple close (touching) objects or parts of the same object.

Resulting Elasticity Model

For calculating the elastic force of particle i, we need to determine the strain energy U_i of the particle. This is usually measured in terms of an energy density and is given as

$$U_i = \bar{v}_i \frac{1}{2}(\epsilon_i \cdot \sigma_i) \tag{10}$$

where $\bar{v}_i$ is the body volume (reference volume) of particle i. $\bar{v}_i$ is computed as if all of its body neighbors j_{body} were located at a relative position $\mathbf{x}_{ij}$, which is equal to the distance vector between the reference positions of particle i and j in our *locally* undeformed object definition. As the volume is calculated by dividing mass by density the body volume of particle i is defined as

$$\bar{v}_i = \bar{v}(\mathbf{x}_i) = \frac{m_i}{\sum_{j_{\text{body}}} m_j W(\mathbf{x}_{ij}, h)} \tag{11}$$

Since we use a linear stress–strain relationship it holds that $\sigma = \mathbf{C}\epsilon$, which is known as Hooke's law. In addition, we only use isotropic materials in our simulations, which means that $\mathbf{C}$ depends only on the Young's Modulus E and the Poisson's Ratio ν. There are different formulas for calculating the strain, the one employed here is called the Green-Saint-Venant strain tensor, but it can be easily replaced by a different one.

The elastic force $\mathbf{F}^{\text{elastic}}$ can then be defined as the negative gradient of strain energy U with respect to displacement. The force that particle i exerts on its jth neighbor is given by

$$\mathbf{F}^{\text{elastic}}_{ji} = -\nabla_{\mathbf{u}_j} U_i = -2\bar{v}_i(\mathbf{I} + \nabla \mathbf{u}_i^T)\sigma_i \mathbf{d}_{ij} \tag{12}$$

where $\mathbf{I}$ is the identity matrix, $\nabla \mathbf{u}_i$ is the gradient of the displacement from the reference shape of the body and $\mathbf{d}_{ij}$ is defined by

$$\mathbf{d}_{ij} = \frac{\partial \nabla \mathbf{u}_i}{\partial \mathbf{u}_j} \tag{13}$$

For more detailed derivations of (12) and (13), refer to Reference [14]. We calculate (13) using the SPH method, thus $\nabla \mathbf{u}$ is defined as

$$\nabla \mathbf{u}_i = \sum_{j_{\text{body}}} \bar{v}_j \nabla W(\mathbf{x}_{ij}, h)(\mathbf{u}_{ji})^T \tag{14}$$

where the displacement difference vector $\mathbf{u}_{ji}$ is a function of the current position $\mathbf{r}$ and $\mathbf{x}_{ij}$:

$$\mathbf{u}_{ji} = \mathbf{u}_j - \mathbf{u}_i = \mathbf{r}_j - \mathbf{r}_i + \mathbf{x}_{ij} \tag{15}$$

The derivative of (14) with respect to $\mathbf{u}_j$ is computed by the resulting SPH equation for $\mathbf{d}_{ij}$

$$j \neq i \rightarrow \mathbf{d}_{ij} = \bar{v}_j \nabla W(\mathbf{x}_{ij}, h) \tag{16}$$

Elasticity Kernel

The kernel function used to calculate a certain attribute or force has a great influence on the behavior of an SPH simulation. The most obvious effects of the smoothing kernel are those on stability and speed. When melting and solidification is introduced, arbitrary sets of particles can solidify into deformable bodies. While a fluid cools, there may be parts where already very few cold particles (possibly only two) form small objects. The 'spiky kernel' presented in Reference [8] turned out to be unsuitable for those situations resulting in an unstable simulation. To cope with this problem, we designed the following new kernel function (Figure 4):

$$W(\mathbf{r}, h) = \begin{cases} c\frac{2h}{\pi}\cos\left(\frac{(r+h)\pi}{2h}\right) + c\frac{2h}{\pi} & 0 \leq r \leq h \\ 0 & \text{otherwise} \end{cases}$$

Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav

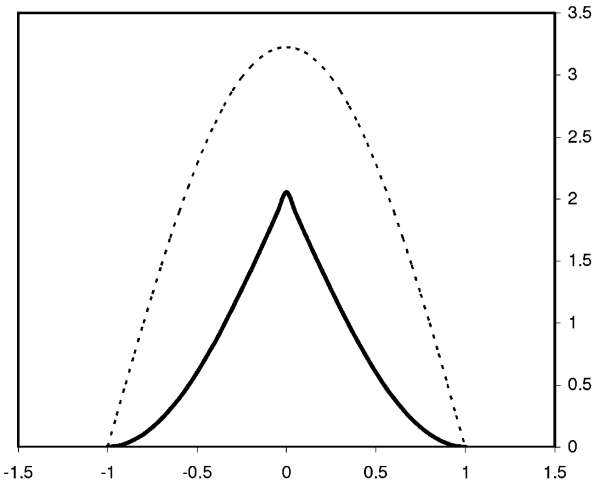


Figure 4. The smoothing kernel along one axis used for the elastic forces, for smoothing length $h = 1$. The thick line shows the kernel and the dotted line its gradient.

As we use W in a normalized form, c is determined by

$$c = \frac{\pi}{8h^4\left(\frac{\pi}{3} - \frac{8}{\pi} + \frac{16}{\pi^2}\right)} \tag{17}$$

Given that c is a constant, it can be precomputed at the beginning of the simulation.

Plasticity and Fracture

Objects in the real world are not perfectly elastic. Depending on the amount of experienced strain, materials often do not fully return to their original shape. This effect is called plasticity, which we capture by integrating the model proposed in Reference [28]. If plastic flow occurs in an object, a part of the deformation is absorbed by the material, and its shape is permanently changed. However, this occurs only after the object has been deformed sufficiently, which can be defined by an elastic limit that we test with the von Mise's yield criterion for each particle. If this criterion is met, plastic deformation will occur. The base change of plastic deformation as well as the plastic strain is computed as described in Reference [28]. The current elastic strain of each particle is then represented by the difference between the particle's plastic strain and the particle's total strain. Moreover, the plastic deformation will not go beyond some plastic limit. Every particle knows its own elastic and plastic limits, which makes it possible to simulate different materials at the same time.

Additionally, a simplified fracture rule is added to the deformable model, considering the distance between two body neighbors. If this distance gets too large and exceeds a limit, they discard each other as object neighbors. As a result, they do not perform any forces on each other anymore and the object breaks at this location.

Rigid Bodies

For modeling rigid bodies the basic SPH fluid model is extended to enforce rigid body motion. For that, the total forces acting on the particles belonging to a rigid body are accumulated in the body, then its movement is restricted to translation and rotation[26]. The rigidity method described here builds upon the body neighbor object representation introduced for elastic objects. Our model keeps track of particles that belong to the same rigid body to allow for merging and splitting during phase change processes.

In order to constrain the motion of an object to rigid body motion, we have to handle rotation explicitly. To do so, we compute a torque vector τ according to Reference [26]:

$$\tau_i = (\mathbf{r}_i - \mathbf{r}^{\text{cm}}) \times \mathbf{F}_i \tag{18}$$

where $\mathbf{r}^{\text{cm}}$ is the center of mass of a body and $\mathbf{F}_i$ denotes the total force exerted on the ith particle. $\mathbf{F}_i$ is the sum of all forces calculated with SPH (the force densities $\mathbf{f}_i$ are multiplied by the volume of i to get forces) and all external forces present in the simulation. The total force acting on a body is given by $\mathbf{F}_{\text{body}} = \sum_{i_{\text{body}}} \mathbf{F}_i$ and the total torque is defined by $\tau_{\text{body}} = \sum_{i_{\text{body}}} \tau_i$. Note that for efficiency, the computation of force densities between pairs of particles that belong to the *same* rigid body are skipped.

After the total force and torque of a body are determined, time integration is performed by first iterating over a rigid object to calculate the effect of the forces and torques, that is, the effect on the position and the linear and angular velocity of the body, then the particles belonging to the body are updated to reflect the changes of their parent. The angular velocity of a rigid body is defined as

$$\omega = \mathbf{I}^{-1}\mathbf{L} \tag{19}$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

where $\mathbf{I}$ is the inertia tensor and $\mathbf{L}$ is the angular momentum, which is updated in every time step by calculating

$$\mathbf{L} \leftarrow \mathbf{L} + \tau \Delta t \tag{20}$$

Phase Changes

Temperature Effects

A temperature is attributed to every particle. It can change either because of heat diffusion among neighbor particles or because of outside influences. Using the SPH formalism, the evolution of the temperature T_i due to diffusion sampled at the particles can be computed analogously to Reference [9] as

$$\frac{\partial T_i}{\partial t} = c \sum_j m_j \frac{T_j - T_i}{\rho_j} \nabla^2 W(\mathbf{r}_{ij}, h) \tag{21}$$

where T is the temperature and c is a diffusion constant. We integrate the attributes in time using a simple Euler-scheme. Additionally, every particle stores a melting point t_{melt} and a solidification point t_{solid} according to the material. If t is above t_{melt}, the particle is liquid, and if t is below t_{solid}, it is solid belonging either to a deformable body with maximal Young's Modulus E (maximal elastic stiffness) or to a rigid body. In between, the particle belongs to a deformable body with E and the viscosity μ interpolated linearly, whereas the Poisson's Ratio ν stays constant (Figure 5). By choosing $t_{\text{solid}} = t_{\text{melt}}$ the intermediate state is left out and it is possible for a liquid to solidify directly to a rigid body or melt a rigid body directly into a liquid.

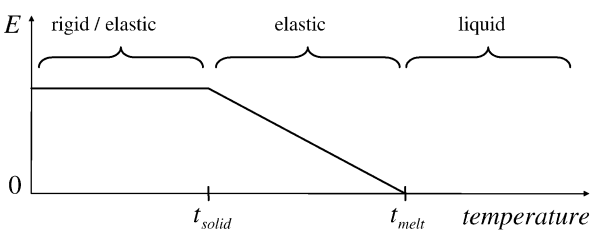


Figure 5. Stages of the phase change process. A particle belongs either to a liquid, an elastic or a rigid object according to its type, its current temperature, and its melting and solidification temperatures stored.

Phase Changes of Elastic and Rigid Bodies

During melting, a particle must be able to separate from its parent object as soon as it is liquified, and it must be able to merge with a touching solid object during solidification as well. To model this behavior for *elastic* particles the following needs to be updated: the set of body neighbors j_{body}, the set of reference distance vectors $\mathbf{x}_{ij}$, and the body volume $\bar{v}_i$. Note, this has to be done for both the particle that solidifies or melts, and the body neighbors that are added or discarded respectively. For merging and splitting behavior of *rigid* objects, the model must keep track of particles that belong to the same rigid body. In all these cases we have to update the following object quantities: the mass m, the center of mass $\mathbf{r}^{\text{cm}}$, the inertia tensor $\mathbf{I}$, the velocity $\mathbf{v}$, and the angular momentum $\mathbf{L}$.

Surface Reconstruction

A challenge with particle methods is to generate a smooth renderable surface from the resulting set of points. There are many different approaches for solving this problem, one was presented in Reference [29] where a level-set simulation guided by particles is used. Another method is presented in Reference [10] where the surface reconstruction is based on a color function, and Wald and Seidel[30] presented a method which leads to smooth surfaces, but which depends on high quality normals at every point. Each of the above approaches has advantages and disadvantages concerning quality and speed. Recently, Zhu and Bridson[2] presented an approach, where the center of mass is taken into account without the need of normals. They achieve very smooth surfaces, but their technique leads to significant artifacts in concave regions and between isolated particles and splashes. It is proposed to remove these artifacts in a postprocessing step. In our work we propose a modification which uses a detector for errors located in concave regions and corrects them on the fly.

The implicit function proposed by Zhu and Bridson[2] is defined as

$$\phi(\mathbf{r}) = |\mathbf{r} - \bar{\mathbf{r}}(\mathbf{r})| - R \tag{22}$$

where $\bar{\mathbf{r}}(\mathbf{r})$ is the center of mass of a query point's neighborhood and R can be interpreted as a desired

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

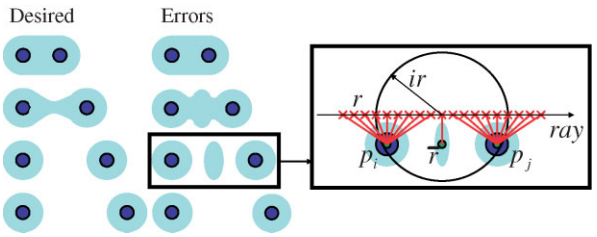


Figure 6. Surface reconstruction errors in the method presented by[2] before postprocessing: when particles are separated by distances comparable to the influence radius, the center of mass may move very quickly leading to surface artifacts.

distance of the surface from the particles. The problem with the use of $\bar{\mathbf{r}}(\mathbf{r})$ is that it can happen that we get a center of mass which erroneously ends up outside of the surface to be reconstructed in concave regions or between near but separated particles. Examining the changes of $\bar{\mathbf{r}}(\mathbf{r})$ when moving the query point $\mathbf{r}$, one can observe that in problematic situations $\bar{\mathbf{r}}(\mathbf{r})$ changes substantially faster than the corresponding $\mathbf{r}$ (Figure 6). To determine how $\bar{\mathbf{r}}(\mathbf{r})$ changes we investigate $\nabla_{\mathbf{r}}(\bar{\mathbf{r}}(\mathbf{r}))$. This 3×3 matrix specifies how small changes in $\mathbf{r}$ translate into a change of $\bar{\mathbf{r}}(\mathbf{r})$. Since we are interested in detecting fast movements we check the largest Eigenvalue $\mathrm{EV}_{\max}$.

We define the implicit surface function as

$$\phi(\mathbf{r}) = |\mathbf{r} - \bar{\mathbf{r}}(\mathbf{r})| - Rf \tag{23}$$

where f is a factor $\in [0..1]$. $\bar{\mathbf{r}}(\mathbf{r})$ is given by

$$\bar{\mathbf{r}}(\mathbf{r}) = \frac{\sum_j \mathbf{r} W(|\mathbf{r} - \mathbf{r_j}|, ir)}{\sum_j W(|\mathbf{r} - \mathbf{r_j}|, ir)} \tag{24}$$

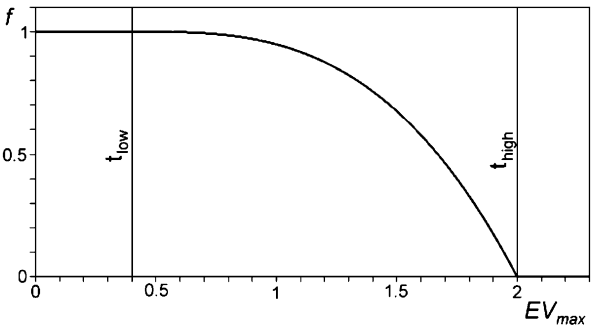


Figure 7. Plot of the factor f (Equation (25)) using the thresholds $t_{\mathrm{low}} = 0.4$ and $t_{\mathrm{high}} = 2.0$. The closer $\mathrm{EV}_{\max}$ is to t_{high}, the smaller is the resulting distance of the surface to the particles.

where ir is the influence radius used in the visualization, defining the smoothness of the surface, and W is the density kernel function.

We compare the largest Eigenvalue $\mathrm{EV}_{\max}$ to two previously defined thresholds t_{low} and t_{high}. We adjust f and with this the resulting distance of the surface from the particles according to the following rule, which makes sure that the first and the second order derivatives are smooth, in order to avoid hard transitions on the surface (Figure 7):

$$f = \begin{cases} 1 & \mathrm{EV}_{\max} < t_{\mathrm{low}} \\ \gamma^3 - 3\gamma^2 + 3\gamma & \text{otherwise} \end{cases} \tag{25}$$

$$\gamma = \frac{t_{\mathrm{high}} - \mathrm{EV}_{\max}}{t_{\mathrm{high}} - t_{\mathrm{low}}}. \tag{26}$$

Note that vanishing derivatives are not required around $\mathrm{EV}_{\max} = t_{\mathrm{high}}$ since in these situations the surface is contracted to one point anyway as the resulting f will then be zero. With this modification

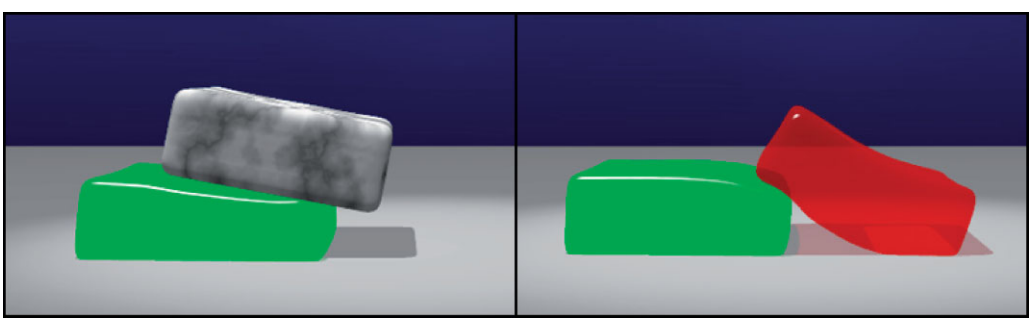


Figure 8. Rigid–elastic interaction (left) and elastic–elastic interaction (right). No special collision handling is necessary to avoid penetration.

Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav

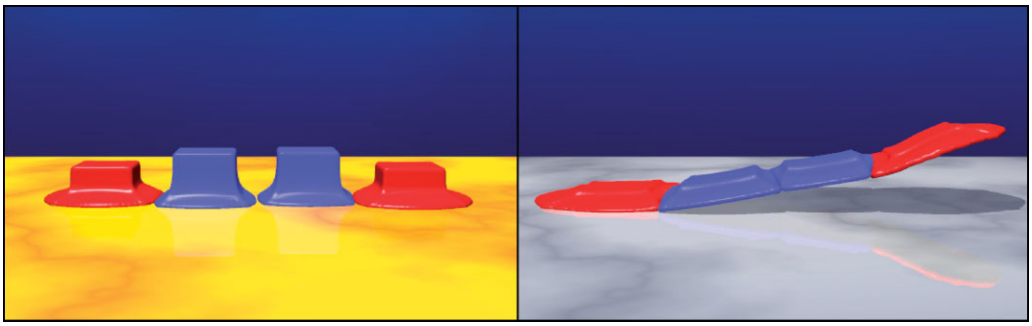


Figure 9. Merging of rigid (blue) and elastic objects (red) after melting and solidification.

the reconstructed surface avoids most of the errors without sacrificing the smoothness which is particularly important if a simulation consists only of a sparse set of particles.

We use our surface reconstruction on the fly during the rendering process. For this purpose, we have adapted the raytracer Povray (http://www.povray.org) in such a way that it can directly raytrace our particles.

Results

We have tested our method with several example simulations. The aim of these examples is not to compete for highest visual quality but to demonstrate the flexibility of the unified model, and to show the wide range of interaction effects not producible with any single of the previous methods alone. In Table 1 the effects covered by our model are compared to previous work

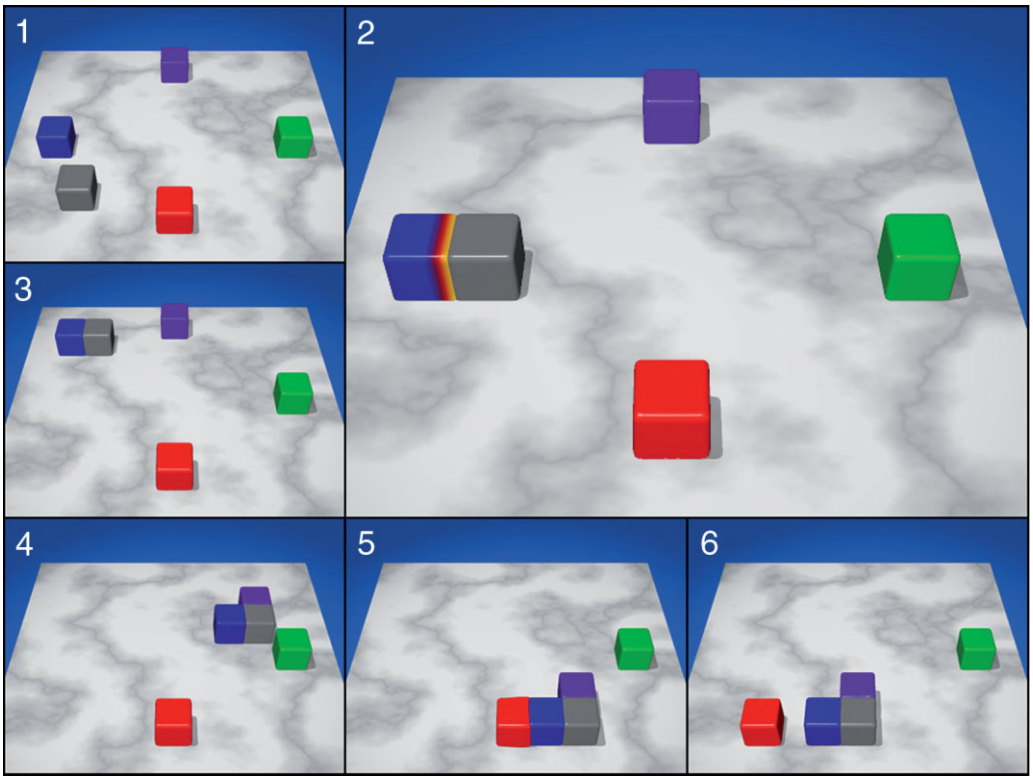


Figure 10. A couple of rigid blocks and one elastic block (red) on a plate. The greedy block is moved, passing by the other blocks. If heating is on while passing, the objects merge (blue and purple), otherwise distinction can be observed.

Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav

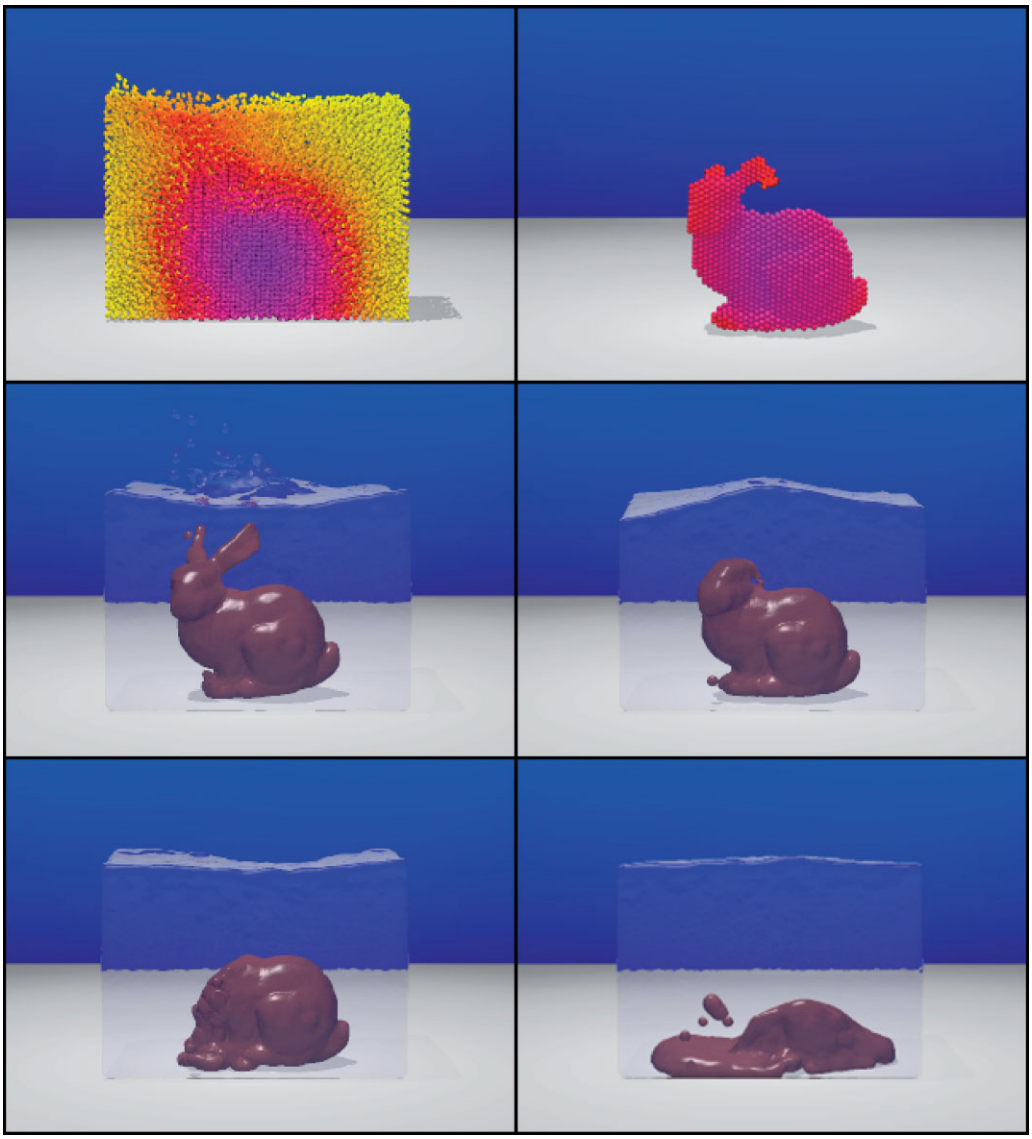


Figure 11. A solid bunny melts inside hot liquid. Top: particles with a temperature coded color, where the left image shows a cut through the particles. Middle, Bottom: raytraced particles.

which is related to or integrated in our method. The full animation sequences can be found in the accompanying supplemental video which is available on the Journal's archive site.

All simulation scenes are performed with about 40 000 particles (except the scene in Figure 8 which consists of 3000 particles), where each particle has 40 neighbors on average. On a Intel iMac 2 GHz, the calculation of the physics takes around 0.5 second per frame in the slowest case, whereas high quality raytracing using our surface reconstruction including shadows and antialiasing take together on average 1 minute per frame in Povray for a resolution of 640 × 480.

Figure 8 shows the rigid–elastic interaction on the left and the elastic–elastic interaction on the right. Apart from choosing a high gas constant k, no special collision handling is necessary to avoid penetration of the objects.

After heating the ground, the rigid (blue) and elastic (red) blocks melt until cooling is turned on in Figure 9. This leads to solidification and merging of all objects since they touch each other. The

Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav

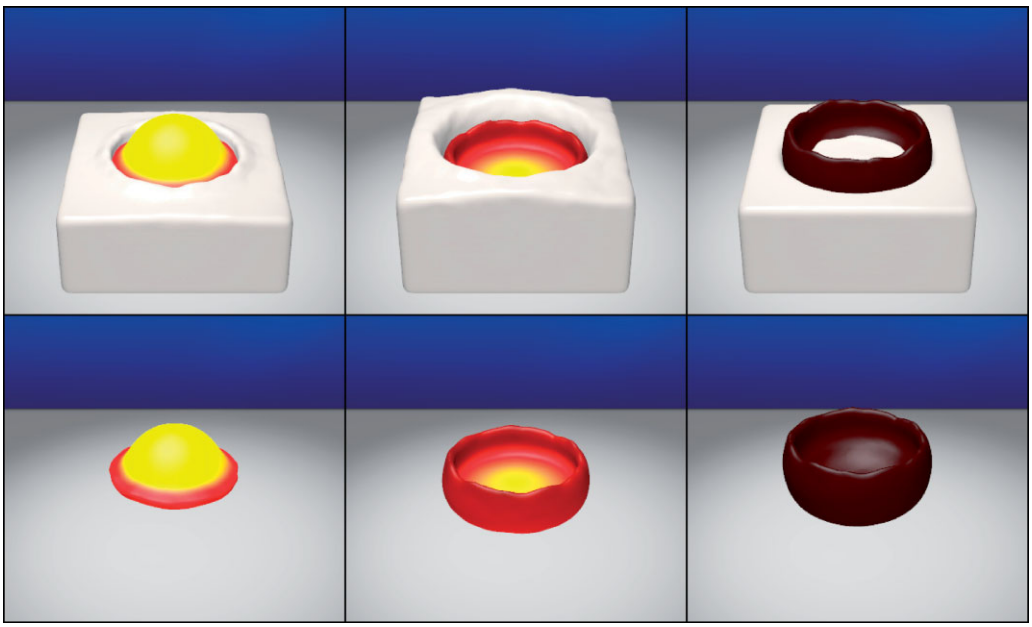


Figure 12. A shell is formed as hot fluid solidifies when it drops into a cold viscous fluid.

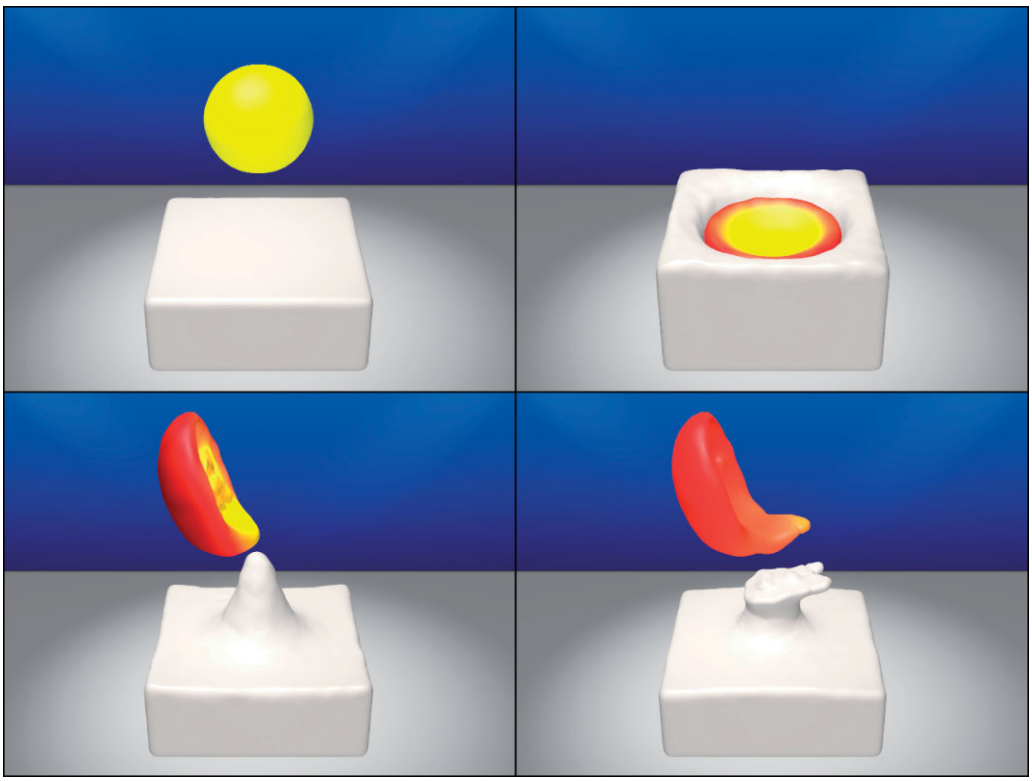


Figure 13. Hot liquid matter partly solidifies inside a cold viscous liquid. After lifting and turning the shell, the liquid inner part flows out and solidifies after colliding with the cold splash.

Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav

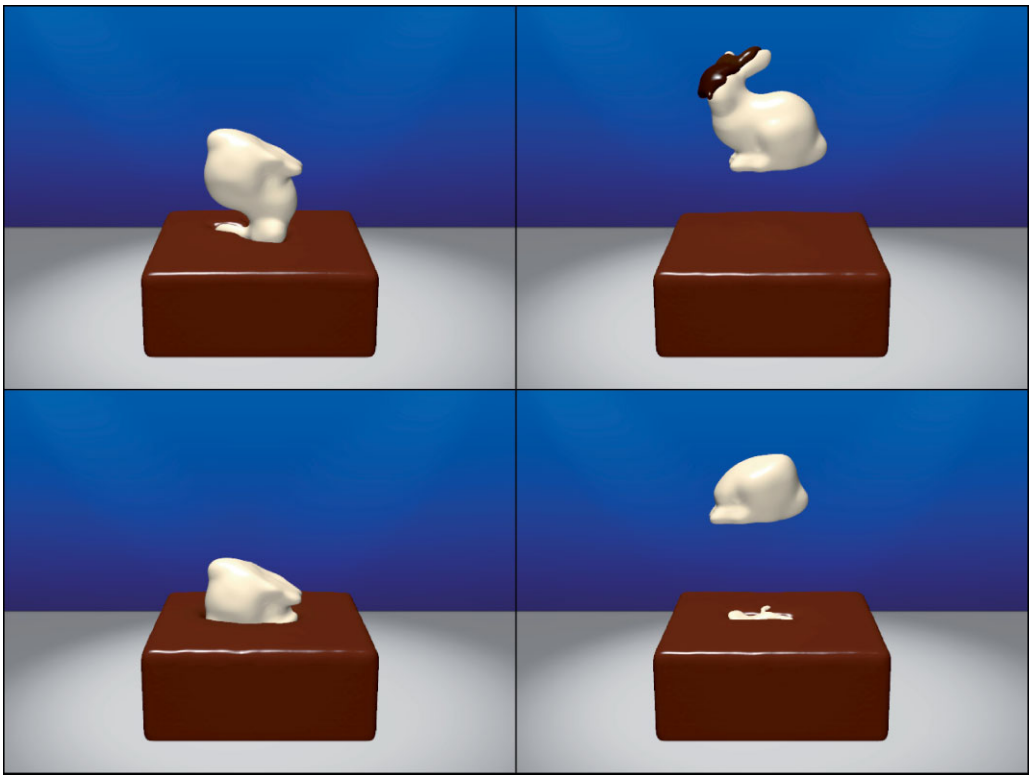


Figure 14. A solid cold white chocolate bunny is dipped into hot brown chocolate. Depending on the time in the hot chocolate the bunny becomes coated with chocolate or melts.

resulting object consists of rigid as well as elastic parts.

Figure 10 shows a gray block which is moved on a board. While passing the other blocks, heating is in some cases turned on. Due to heating and cooling, the first two blocks melt at one side and merge to a single solid afterwards. If there is no heating, involved as with the last two blocks they do not stick to the moving block although they touch each other as it moves by.

In Figure 11 a rigid bunny is dropped into a liquid and starts to melt as soon as the temperature exceeds its melting point. Slower melting can be realized by choosing a higher melting point or by reducing the temperature diffusion rate. Different buoyancy behavior can be achieved by varying the density of the bunny particles.

In Figures 12 and 13 hot liquid matter is dropped into a cold viscous liquid, that is, cream. Due to temperature diffusion, the poured liquid cools down and either fully (Figure 12) or partially (Figure 13) solidifies to a shell. The partially solidified shell is lifted up and rotated while the simulation of the cream is temporarily stopped. Due to gravity, the still hot and liquid part of the shell flows out and solidifies after colliding with the cold splash of the cream. The color shows the object temperature, where a light color corresponds to a high temperature and a dark to a low one.

A solid cold white chocolate bunny is dipped into hot brown chocolate in Figure 14. Depending on the time in the chocolate, the bunny's head gets coated with it (dark brown color indicates that the chocolate is cold) or melts.

Conclusion and Future Work

We have presented a unified SPH model for simulating a wide variety of fluid–solid interaction processes and effects. To achieve this, we modified various previous fluid and solid simulation models and integrated the different methods into a single one. The use of a unified method renders an interface between a fluid and solid model unnecessary and simplifies the interaction between them. As the physical experiments have shown,

Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav

our method is flexible and able to simulate a variety of different effects and interactions. New effects like solidification of hot fluid matter inside cold liquid or solidification of liquid on a cold object are demonstrated as well. Additionally, we have presented a surface reconstruction technique leading to smooth surfaces and less rendering errors in concave regions.

The scenes in the screenshots do not compete for highest visual quality, since only a few thousand particles are used. Although it is our intention in the future to simulate larger scenes to improve the visual quality, it was the focus to create a model capable of running at interactive rates. Our current physics implementation runs at roughly 2 frames per second using 40 000 particles even without applying elaborated optimization techniques. We believe that an acceleration by a factor of 10 is feasible using a combination of fast incremental neighbor search, parallelism, and hardware accelerated techniques. An additional increase in efficiency can be achieved by using adaptive particle sizes to have more details where necessary and less computational costs in unchanging regions.

Open problems exist in the handling of collisions with solids. Particle penetrations may occur if strong forces are involved. In our experience, most collision problems can be handled very well by choosing a high gas constant (Figure 8, 10), but an explicit collision handling or additional boundary forces would be desired to guarantee no penetrations despite strong forces involved.

Still to be investigated is how the numerical accuracy of the method is affected by the number of particles discretizing a certain mass and volume as well as by the time steps used in the Leap-Frog integration scheme. It is difficult to give an estimate for the accuracy of our method. Clear is, that the method gets more accurate as particle sizes and time steps get smaller. However, specific investigations and comparisons to the real world are in order here to get more insight.

References

1. Monaghan JJ. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics* 1992; **30**: 543.
2. Zhu Y, Bridson R. Animating sand as a fluid. *ACM Transactions on Graphics* 2005; **24**(3): 965–972.
3. Terzopoulos D, Platt J, Fleischer K. Heating and melting deformable models (from goop to glop). In *Proceedings of Graphics Interface*, 1989; pp. 219–226.
4. Tonnesen D. Modeling liquids and solids using thermal particles. In *Proceedings of Graphics Interface*, 1991; pp. 255–262.
5. Nealen A, Müller M, Keiser R, Boxerman E, Carlson M. Physically based deformable models in computer graphics. In *Eurographics State of the Art Report*, 2005; pp. 71–94.
6. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**: 375–389.
7. Stam J, Fiume E. Depicting fire and other gaseous phenomena using diffusion processes. In *Proceedings of SIGGRAPH'95*, 1995; pp. 129–136.
8. Desbrun M, Cani M-P. Smoothed particles: a new paradigm for animating highly deformable bodies. In *6th Eurographics Workshop on Computer Animation and Simulation*, 1996; pp. 61–76.
9. Stora D, Agliati P, Cani MP, Neyret F, Gascuel J. Animating lava flows. In *Proceedings of Graphics Interface*, 1999; pp. 203–210.
10. Müller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003; pp. 154–159.
11. Müller M, Schirm S, Teschner M, Heidelberger B, Gross M. Interaction of fluids with deformable solids. *Journal of Computer Animation and Virtual Worlds* 2004; **15**(3–4): 159–171.
12. Müller M, Solenthaler B, Keiser R, Gross M. Particle-based fluid-fluid interaction. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005; pp. 237–244.
13. Wicke M, Hatt P, Pauly M, Müller M, Gross M. Versatile virtual materials using implicit connectivity. In *Proceedings of Eurographics Symposium on Point-Based Graphics*, 2006; pp. 137–144.
14. Müller M, Keiser R, Nealen A, Pauly M, Gross M, Alexa M. Point based animation of elastic, plastic and melting objects. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2004; pp. 141–151.
15. Keiser R, Adams B, Gasser D, Bazzi P, Dutre P, Gross M. A unified lagrangian approach to solid-fluid animation. In *Proceedings of Eurographics Symposium on Point-Based Graphics*, 2005; pp. 125–133.
16. Cani M-P, Desbrun M. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 1997; **3**(1): 39–50.
17. Losasso F, Shinar T, Selle A, Fedkiw R. Multiple interacting liquids. *ACM Transactions on Graphics* 2006; **25**(3): 812–819.
18. Carlson M, Mucha PJ, Turk G. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics* 2004; **23**(3): 377–384.
19. Génevaux O, Habibi A, Dischler J-M. Simulating fluid-solid interaction. In *Proceedings of Graphics Interface*, 2003; pp. 31–38.
20. Chentanez N, Goktekin TG, Feldman BE, O'Brien JF. Simultaneous coupling of fluids and deformable bodies. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006; pp. 83–89.
21. Guendelman E, Selle A, Losasso F, Fedkiw R. Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics* 2005; **24**(3): 973–981.

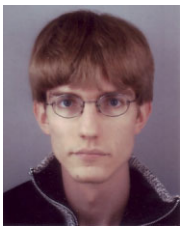
Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav

22. Carlson M, Mucha PJ, R. Brooks Van Horn, Turk G. Melting and flowing. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002; pp. 167–174.
23. Goktekin TG, Bargteil AW, O'Brien JF. A method for animating viscoelastic fluids. *ACM Transactions on Graphics* 2004; **23**(3): 463–468.
24. Losasso F, Irving G, Guendelman E, Fedkiw R. Melting and burning solids into liquids and gases. *IEEE Transactions on Visualization and Computer Graphics (2005)* 2006; **12**(3): 343–352.
25. Zhao Y, Wang L, Qiu F, Kaufman A, Mueller K. Melting and flowing in multiphase environment. *Computers & Graphics* 2006; **30**(4): 519–528.
26. Baraff D. An introduction to physically based modeling: rigid body simulation 1—unconstrained rigid body dynamics. SIGGRAPH Course Notes, 1997.
27. Pozrikidis C. *Numerical Computation in Science and Engineering*. Oxford Univ. Press: NY, 1998.
28. O'Brien JF, Bargteil AW, Hodgins JK. Graphical modeling and animation of ductile fracture. *ACM Transactions on Graphics*, 2002; **21**(3): 291–294.
29. Premoze S, Tasdizen T, Bigler J, Lefohn A, Whitaker RT. Particle-based simulation of fluids. In *Proceedings of Eurographics*, 2003; pp. 401–410.
30. Wald I, Seidel H-P. Interactive ray tracing of point based models. In *Proceedings of Eurographics Symposium on Point Based Graphics*, 2005; pp. 9–16.

Authors' biographies:

Barbara Solenthaler is a Research Assistant and doctoral student in the Visualization and MultiMedia Lab (VMML) in the Department of Informatics at the University of Zurich. She received the degree of Dipl Inf-Ing ETH in Computer Science from the Swiss Federal Institute of Technology (ETH) Zurich in 2004. Her research interests include physically based simulations, particle animations, geometric modeling and virtual reality.



Jürg Schläfli received the degree of Dipl Inform in Informatics from the University of Zurich in 2006. Since then he has been working as a Software Engineer at Avaloq in Switzerland, where he is developing banking systems.

Renato Pajarola received Dipl Inf-Ing ETH and Dr Sc Techn degrees in Computer Science from the Swiss Federal Institute of Technology (ETH) Zurich in 1994 and 1998, respectively. Following his dissertation, he was a post-doctoral researcher and lecturer in the Graphics, Visualization & Usability (GVU) Center at Georgia Tech. In 1999, he joined the University of California Irvine as an Assistant Professor and founded the Computer Graphics Lab. Since 2005 he has been leading the Visualization and MultiMedia Lab (VMML) at the University of Zürich as an Associate Professor in the Department of Informatics. He is also a member of the ACM, ACM SIGGRAPH and IEEE Computer Society. His research interests include real-time 3D graphics, multiresolution modeling, point based graphics, interactive scientific visualization, remote and parallel rendering, compression and interactive 3D multimedia. He has published a wide range of over 40 peer-reviewed research articles in top journals and conferences. Dr Pajarola frequently serves on program committees, such as the IEEE Visualization Conference (2004–06), Pacific Graphics (2002–03) or EuroVis (2001,2006–07) and is co-chairing the 2007 IEEE/Eurographics PBG Symposium papers program. He also received a Eurographics Second Best Paper Award (as co-author) in 2005.

Copyright © 2007 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2007; **18**: 69–82
DOI: 10.1002/cav