Melting and Flowing of Viscous Volumes

Xiaoming Wei[1], Wei Li[1] and Arie Kaufman[1]
Center for Visual Computing (CVC)
and Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794-4400

Abstract

We present a simple, linear 3D cellular automata approach for animating the melting process of solid volumetric models. Accurate modelling of object melting usually requires complicated physical simulations of heat transfer and phase transition from solid to liquid. Instead, we propose a simplified model to describe the melting behaviors of highly viscous objects, such as wax, lava, plastic, metal and chocolate. We simulate the process by which a volumetric solid transforms into a viscous liquid as the amount of heat it accumulates on its surface reaches a certain temperature. We then animate smooth fluid behavior using a cellular automata. The dynamic volume data is rendered directly on texture mapping hardware to achieve an interactive speed.

KEYWORDS Object Melting, Cellular Automata, Natural Phenomena Modelling

1 Introduction

Natural phenomena, from clouds to fire and smoke to water, have always fascinated and awed human kind. Over the last few years, many efforts have been made to bring realism into graphics animations. Realistic computer animations of phase transitions, such as melting, freezing, thawing and evaporation, are important and challenging in many applications where heat and temperature play a role. However, such simulations are usually computationally expensive due to the complexity of the underlying physical equations. Moreover, animation in interactive environments usually requires that the model be robust against user actions, while also ensuring a high frame rate. This paper proposes a simple method for the animation of highly viscous objects melted by a heat source. Our method guarantees both stability and low computational cost - making it ideal for real-time, interactive applications.

To describe the problem better, let us consider what happens when a solid object melts. When an object is in a solid form, it has a rigid structure on microscopic scales. The strong attraction between microscopic particles holds them tightly together. When a solid object is heated, these microscopic particles gain energy and start to vibrate faster and faster, which has the effect of expanding the solid. Further heating provides more energy until these particles start to break free from the structure. Although still loosely connected, they are now able to move around. At this point, the object is melting from a solid to form a liquid. The liquid flows down the unmelted solid due to the gravity and gradually stops because of friction on the ground.

To have a faithful simulation of object melting, two parts are needed: One is the heat energy required by the solid to undergo the phase transition to a liquid. The other is the modelling of the fluid flow behavior. In order to be suitable for interactive computer animation, the method we have adopted can achieve both real-time rendering speed and visual realism. In this paper, we simplify the process of heat transfer based on the conduction of heat between neighboring voxels and design a 3D cellular automata to model the flow of the highly viscous fluid as the solid object melts into thick liquid.

The remainder of the paper is organized as follows. In the next section, we provide an overview of the previous work in the modelling of fluid behavior. The simulation of heat conduction is discussed in Section 3. In Section 4, we present our 3D CA models. Our rendering method is described in Section 5. In Section 6, we demonstrate a few example applications.

2 Previous Work

Physically-accurate modelling based on partial differential equations is a broadly used method both in simulating fluid behavior and in modelling heat convection and diffusion. However, it usually requires a large amount of computation time and memory, especially in 3D. In 1991, Wejchert and Haumann [19] gave an analytic solution to the Navier Stokes (NS) equations, which govern the fluid flow by using simple flow primitives. Chen and Lobo [2] solved a simplified NS equation in 2D using a finite difference approach. Later, Foster and Metaxas [8] presented a full 3D solution to simulate the turbulent rotational motion of gas. They [7] also demonstrated how to animate water and control its behavior. Because of the inherent instability of the finite difference method for larger time steps, the speed of this approach is limited. Stam [14] proposed an unconditionally stable implicit solution for the NS equations. Fedkiw et al. [5] further improved the method by introducing the concept of vortex confinement to the graphics field and generated realistic animations of liquid [6]. To get an accurate velocity field with visible turbulence detail, their methods require a large 3D grid, which makes the computation slow. Carlson et al. [1] simulated the phenomena of melting wax by solving the physical equations for highly viscous fluid and allowed the material property of objects to change from rigid solid to sloshing liquid according to the variation in temperature.

Some graphics researchers also used physically-based particle models to describe fluid behaviors. Particle systems were first introduced by Reeves [13] as a technique for modelling fuzzy objects, such as fire, clouds, smoke and water. Concepts from molecular dynamics have been combined with particle systems to simulate thermal phenomena. Tonnesen [17] used a discrete model for the heat transfer equations to describe the interaction of particles due to the thermal energy. Terzopoulos et al. [16] implemented a similar approach. Particles and springs have been utilized to render a series of particle *"blobbies"* using implicit surfaces. Desbrun and Gascue [3] developed a paradigm extended from the Smoothed Particle Hydrodynamics approach used by physicists for cosmological fluid simulation. This technique defines a type of particle system which uses smoothed particles as samples of mass smeared out in space. A highly inelastic fluid is simulated by computing the variations of continuous functions such as mass density, speed, pressure, or tem-

[1]Email:{wxiaomin,liwei,ari}@cs.sunysb.edu

perature over space and time. The density of each particle indicates how mass is distributed in space. Later, iso-values of these densities are used to define the implicit surfaces for rendering. Stora et al. [15] also used smooth particles to simulate lava flow. Particles are coated with implicit surfaces to get the final rendering result.

Based on simple, local calculations, CA have been used in several graphics applications. Dobashi et al. [4] implemented a realistic animation of clouds based on a CA model proposed by Nagel and Raschke [11]. Fujishiro and Aoki [9] implemented mathematical morphology to model the ice thawing effect based on volume models. A separate CA model was used to represent the feature of regelation. A finite difference solver with time step 1 can also be understood as a CA model. However, our work is not a direct result from finite difference solutions of partial differential equations.

Our work is based on the following observations of object melting. A rigid object can start melting from its surface as the amount of heat energy it accumulates reaching a certain level. Heat is transferred through the volume by the means of conduction, convection and radiation. As the object's sub-volume transforms into a liquid, it behaves like a highly viscous fluid. Moreover, its movement is mainly a result of the force of gravity, viscosity damping, and friction. There is not much turbulent and rotational motion in its movement. Our system consists of two parts: One is the modelling of heat transfer. The other is a specific CA for the modelling of viscous liquid.

The CA model we designed is inspired by the work of Fujishiro and Aoki [9]. Their method however deals only with the problem of ice thawing. Since parts of the ice may sublimate, mass is not conserved in their method. In contrast, the conservation of mass and the effect of external forces are considered in our CA model. The recent work of Carlson et al. [1] also deals with the problem of solid melting and flowing. Their method is a physical simulation of viscous fluid based on the incompressible NS equations. The solid and liquid parts of the object are incorporated into a whole system based on different viscosity values. Different types of fluid can be simulated using their system. For a $35 \times 28 \times 38$ bunny model, their approach requires 550ms simulation time, which is still not fast enough for interactive computer animation. Our purpose in this work is to design a simple, local CA to achieve a fast and faithful visual simulation of object melting and flowing that is suitable for real time or interactive applications. The CA we used is specifically designed for highly viscous liquid.

3 Heat

In this paper, we consider heat sources as the main driver of the transformation of objects from solid to liquid. The discipline of heat transfer is concerned with two things: temperature and the flow of heat. Temperature is a measure of the amount of thermal energy, whereas heat flow represents the movement of thermal energy from one place to another.

Several material properties serve to modulate the heat transfer between two regions at differing temperatures. Examples include thermal conductivities, specific heats, material densities, fluid velocities, fluid viscosities, surface emissivities, and more. Taken together, these properties serve to make the solution of many heat transfer problems a complicated process. Heat transfer mechanisms can be grouped into three broad categories:

- Conduction: Regions with greater temperature will pass their thermal energy to regions with lower temperature through direct microscopic molecular collisions. It takes place mainly in solids. For example, metal has a better heat conduction ability than wood.
- Convection: In liquids and gases (fluids) heat energy is transferred mainly by a process called *Convection*. The heat source causes the fluid to expand and rise upwards. The hot fluid gives up its heat energy to cooler parts of the fluid.
- Radiation: All materials radiate thermal energy in amounts determined by their temperature. When temperatures are uniform, the radiative flux between objects is in equilibrium and no net thermal energy is exchanged. The balance is upset when temperatures are not uniform and thermal energy is transported through space.

To simulate the physically correct heat conduction is too time-consuming for an animation system. Instead of solving complex partial differential equations, we adopt a simple approach to model the conduction and convection of heat inside the solid/liquid volume and the radiation of heat from a nearby heat source. Based on the volume structure, to simulate the effect of conduction, the temperature field is defined in each voxel. Heat is conducted between neighboring cells due to the difference in the temperature field, as indicated by the following equation:

$$T(i,j,k,t+1) = T(i,j,k,t) + \sum \gamma (T(s,m,n,t) - T(i,j,k,t))/\pi l^2 \quad (1)$$

where (s,m,n) is the neighboring cell of (i,j,k). $T(s,m,n,t)$ is the temperature of the cell (s,m,n) at time t, l is the distance between these two cells, and γ is the heat conservation coefficient.

We first determine all those voxels that are located on the surface of the solid model. For each surface voxel, we calculate its distance to the heat source, located at a specific position. If nothing blocks the ray, certain amount of energy is accumulated at that cell. As its temperature reaches the "melting point", the cell changes its form from solid to liquid whereupon it is free to move.

4 3D Cellular Automata

Cellular automata (CA) were first proposed by John Von Neumann as formal models of self-reproducing organisms. They are discrete dynamic systems, where the space, time, and the state of the cells are all discrete. The states of cells are synchronously updated according to local rules. The CA can produce extremely complex structures from the evolution of rather simple and local rules.

4.1 Variables

The behavior of highly viscous fluid consists mainly of smooth movements as a result of gravity force, viscosity damping force, and friction force. Our model is designed based on these forces. For each cell in the volume, we define a few variables:

- Solid(i,j,k) = $\begin{cases} 1 & solid \\ 0 & otherwise \end{cases}$

  This is a boolean variable.

- Liquid(i,j,k)= $\begin{cases} 1 & liquid \\ 0 & otherwise \end{cases}$

  This is a boolean variable.

- Amount(i,j,k)= $\begin{cases} 1.0 & melt \\ 0.0 & otherwise \end{cases}$

  This variable is a real number between 0.0 and 1.0, used to indicate the amount of liquid at each cell. The value can be transferred between neighboring cells. During the design of transition rules in the next section, we guarantee the conservation of mass locally and globally.

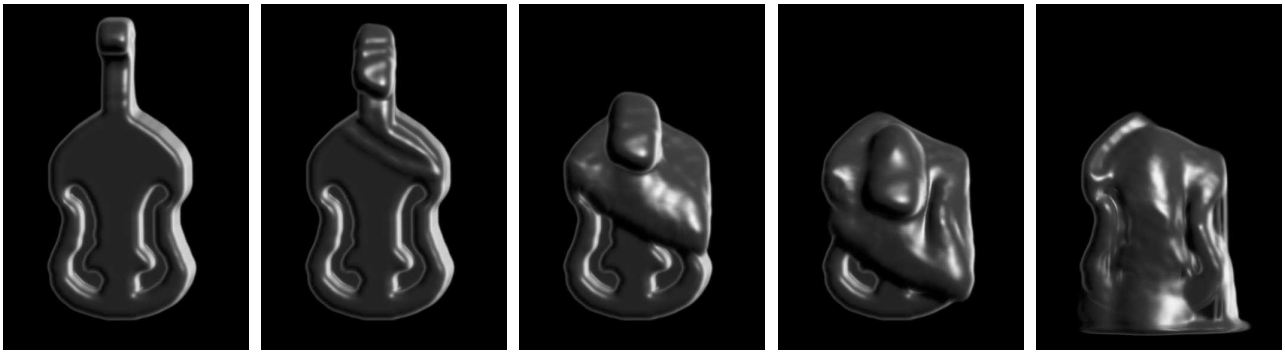


Figure 1: A tin cello heated by a point heat source. (Please see Color Plate 1 in color section.)

- Energy(i,j,k)= $\begin{cases} 1.0 & melt \\ 0.0 & otherwise \end{cases}$

  This variable is a real number between 0.0 and 1.0, used to indicate the expansion potential of a liquid cell. As we observe objects changing states from solid to liquid, and even transformed into gas, we see their physical space increases due to the decrease of the bond between those microscopic particles. Setting a certain threshold for this variable enables us to control the expansion characteristic of the liquid.

4.2 Rules

Now let us consider the transition rules for the 3D CA. We group them into two parts: gravity force and spreading.

4.2.1 Gravity Force

As a solid cell transforms to a liquid cell, gravity is the driving force that makes it flow down to the ground. For each liquid cell, we test the cell below it:

- A solid cell: the liquid cell stays where it is;
- A liquid cell: test if it still has space for more liquid, if so, move certain amount of liquid to it, the amount of energy will be transferred accordingly; else the liquid cell remains where it is;
- An empty cell: move a portion of the mass to the cell below it, the energy will also be transferred.

We summarize these transition rules as follows (here we assume Y axis is aligned with the gravity direction):

$$Liquid(i,j+1,k,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i,j+1,k,t) \quad (2)$$
$$\Delta Amount(i,j+1,k,t+1) \mathrel{+}= Amount(i,j,k,t) * factor \quad (3)$$
$$\Delta Amount(i,j,k,t+1) \mathrel{-}= Amount(i,j,k,t) * factor \quad (4)$$
$$\Delta Energy(i,j+1,k,t+1) \mathrel{+}= Energy(i,j,k,t) * \beta * factor \quad (5)$$
$$\Delta Energy(i,j,k,t+1) \mathrel{-}= Energy(i,j,k,t) * \beta * factor \quad (6)$$

where *factor* indicates the percentage of liquid to move. It can have a value up to 1. $\Delta Amount()$ and $\Delta Energy()$ are used to record the change of the *Amount* and *Energy* variables at each voxel. β is the energy acceleration coefficient, simulating the increasing energy for the liquid cell as it flows down. As we add Equations 3 and 4, the conservation of mass is indeed satisfied.

4.2.2 Spreading

For each liquid cell, if its energy is higher than a specified threshold, it has the potential to spread along its horizontal neighboring cells. For cell (i,j,k), there are four nearest neighbors: $(i+1,j,k)$, $(i-1,j,k)$, $(i,j,k+1)$, $(i,j,k-1)$ and four second nearest neighbors: $(i+1,j,k+1)$, $(i+1,j,k-1)$, $(i-1,j,k+1)$ and $(i-1,j,k-1)$. In our current model, we consider only the nearest neighbors.

- If a liquid cell has a potential to spread, we test all its nearest neighboring cells to decide its spreading pattern. There are 2^4 possibilities all together. As the liquid cell spreads, certain amount of its energy gets lost due to friction.
- For each spreading direction, random numbers can be used to control the choice of spreading patterns. In our current model, all the spreading patterns have the same priorities. If there are several spreading patterns available, we choose one randomly.

The transition rules related to spreading are as follows:

$$Liquid(i+1,j,k,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i+1,j,k,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (7)$$
$$Liquid(i-1,j,k,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i-1,j,k,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (8)$$
$$Liquid(i,j,k+1,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i,j,k+1,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (9)$$
$$Liquid(i,j,k-1,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i,j,k-1,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (10)$$
$$\Delta Amount(i+1,j,k,t+1) \mathrel{+}= Amount(i,j,k,t) * factor1 \quad (11)$$
$$\Delta Amount(i-1,j,k,t+1) \mathrel{+}= Amount(i,j,k,t) * factor2 \quad (12)$$
$$\Delta Amount(i,j,k+1,t+1) \mathrel{+}= Amount(i,j,k,t) * factor3 \quad (13)$$
$$\Delta Amount(i,j,k-1,t+1) \mathrel{+}= Amount(i,j,k,t) * factor4 \quad (14)$$
$$\Delta Amount(i,j,k,t+1) \mathrel{-}= (factor1 + factor2 + factor3 + factor4) * Amount(i,j,k,t) \quad (15)$$
$$\Delta Energy(i+1,j,k,t+1) \mathrel{+}= Energy(i,j,k,t) * \alpha * factor1 \quad (16)$$
$$\Delta Energy(i-1,j,k,t+1) \mathrel{+}= Energy(i,j,k,t) * \alpha * factor2 \quad (17)$$
$$\Delta Energy(i,j,k+1,t+1) \mathrel{+}= Energy(i,j,k,t) * \alpha * factor3 \quad (18)$$
$$\Delta Energy(i,j,k-1,t+1) \mathrel{+}= Energy(i,j,k,t) * \alpha * factor4 \quad (19)$$
$$\Delta Energy(i,j,k,t+1) \mathrel{-}= Energy(i,j,k,t) * \alpha * (factor1 + factor2 + factor3 + factor4) \quad (20)$$

where α is the energy conservation coefficient, used to simulate the effect of friction. Different coefficient values are used to describe the viscosity friction between liquid cells and the friction between

Figure 2: A chocolate goblet in a hot oven. (Please see Color Plate 2 in color section.)

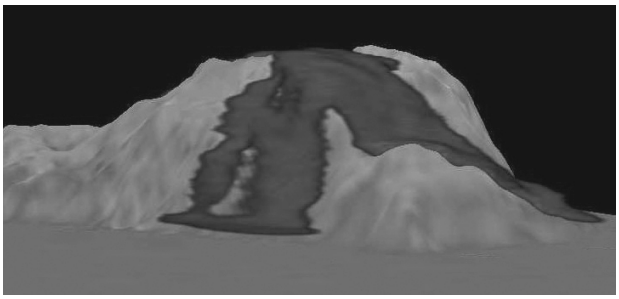


Figure 3: A view of lava flow generated by our 3D CA. (Please see Color Plate 3 in color section.)

liquid and solid, or the floor. They can have values between 0 and 1. $factor1$, $factor2$, $factor3$ and $factor4$ are coefficients used to describe the amount of mass spread to the neighboring non-solid cells. These values are predefined for each spreading pattern. As we add Equations 11-15, we get $\Delta Amount(i+1,j,k,t+1)+\Delta Amount(i-1,j,k,t+1)+\Delta Amount(i,j,k+1,t+1)+\Delta Amount(i,j,k-1,t+1)+\Delta Amount(i,j,k,t+1)=0$. The conservation of mass is also satisfied in this step.

After the first two steps, some liquid cells may have too much liquid. In such a case, we redistribute the excess to its upward neighboring cells. By doing this, the piling effect of highly viscous fluids is manifested.

5 Rendering

To render the melting volume, we update the density values of the voxels according to their current state in the CA. Solid voxels remain at their original position without changing intensity values. For liquid voxels, the *Amount* variable is linearly mapped to an intensity value between 0 to 255. When all the voxels have been processed, we low-pass filter the volumetric data to smooth away any sudden change of intensity value. Only the liquid voxels and their neighbors that are affected by the low-pass filtering are updated.

The modified volume can then be visualized with any volume rendering method. To achieve interactive speed, we exploit hardware accelerated volume rendering, either with dedicated hardware, such as VolumePro [12], or general texture mapping hardware [20]. In either case, we need to transfer the modified volume from host memory to graphics memory. To reduce the transfer time, the original volume is divided into bricks (for VolumePro and 3D texture-based volume rendering) or sub-slices. Only the bricks or the sub-slices containing modified voxels are transferred. In addition, for texture-based volume rendering, if lighting is enabled, the gradient values in the modified regions are recomputed and updated in texture memory as well.

6 Implementation and Results

We present three examples to demonstrate our method. All results have been generated on a Pentium IV 2.53GHz PC. Table 1 summarizes the results for each of the three example including the grid size, the average simulation time required by each gravity force step, the average time for each spreading step, and the time for updating the volume model, low-pass filtering and transfer of the modified portion of the volume from host memory to graphics memory. The frame rate for the three examples ranged between 8-12 frames/sec. Figure 1 shows a sequence of a cello, made of tin alloy, heated by a heat source located at the front upper right. Users can interactively modify the location and intensity of the heat source. Figure 2 simulates a chocolate goblet in a hot oven. In this example, conduction and convection are considered as the main heat transfer method. The top portion of the goblet is melted first. The highly viscous liquid drips down along the remaining solid parts. Eventually, the whole goblet is melts.

Another application of our 3D CA is the modelling of lava flow from a volcano. A physically based simulation of lava flow can be a complicated process (see the work of Stora et al. [15] for details). In this example, we consider the lava flow as a highly viscous fluid and it does not solidify or destroy the surrounding environment. Its movement is guided by the underlying terrain. Figure 3 is rendered by combining the lava flow volume with the underlying textured volcano terrain based on the depth value. The user can also interactively control when and where the lava is generated at the top of the mountain. Figure 4 shows a different sequence of lava flowing down the mountain generated using a 3D texture-based volume rendering method. A better rendering result including lava glow can be achieved by generating texture maps based on the temperature of the lava.

Table 1: Calculation times (in *ms*) for three melting examples

	Resolution	Gravity	Spreading	Update
Cello	$107\times127\times107$	3.2	15.6	65.8
Chocolate	$141\times94\times141$	3.2	18.4	83.8
Lava	$128\times100\times128$	3.2	21.6	90.2

7 Conclusion and Future Work

We have presented a 3D CA approach to simulate the melting and flowing process of highly viscous objects in different applications, such as the volcano terrain simulation and the wax, chocolate, metal and ice objects melting animations. Our model works directly on volume datasets. It consists only of bit operations and simple addition and multiplication operations, which result in an interactive

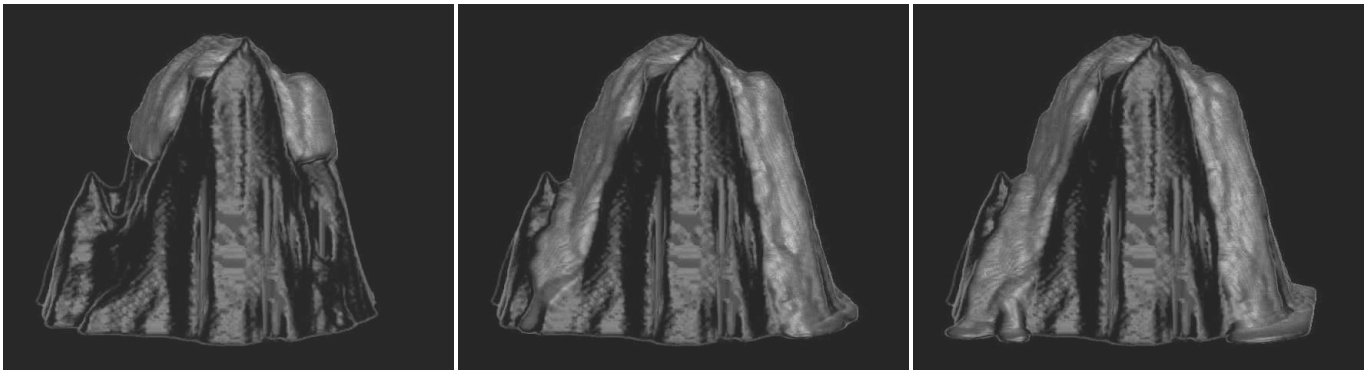


Figure 4: Lava flowing down a volcano mountain (mountain elevation scaled up by a factor of 2).

animation speed, making it suitable for animation and interactive applications. The model is also ideally suited for hardware acceleration. Different types of heat transfer methods were simulated based on the volume data structure.

In our future work, we will implement an adaptive or multi-resolution CA grid to supress artifacts that might be generated by the regular grid. We will also deal with the application of melting a solid object in the middle. The top portion of the solid object will be separated from the rest, and as a result, it will fall down or flow with the liquid part. We believe that combining our current CA model with the work of Desbrun et al. [3] can be a promising direction. We will also implement the computation of the CA and the low-pass filtering completely in commodity graphics hardware to further improve the speed, similar to the fluid simulation on graphics hardware [10, 18].

Acknowledgments

This work is partially supported by ONR grant N000140110034. We would like to thank Suzanne Yoakum-Stover for helpful discussions and proofreading.

References

[1] M. Carlson, P. J. Mucha, R. Brooks Van Horn III, and G. Turk. Melting and flowing. *ACM SIGGRAPH Symposium on Computer Animation*, pages 167–174, July 2002.

[2] J. X. Chen, N. Da, and V. Lobo. Toward interactive-rate simulation of fluids with moving using Navier-Stokes equationsn. *Graphical Models and Image Processing*, 57(2):107–116, March 1995.

[3] M. Desbrun and Marie-Paule Gascue. Animating soft substances with implicit surfaces. *Proceedings of SIGGRAPH*, pages 287–290, August 1995.

[4] Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, and T. Nishita. A simple, efficient method for realistic animation of clouds. *Proceedings of SIGGRAPH*, pages 121–128, August 2000.

[5] R. Fedkiw, J. Stam, and H. Jensen. Visual simulation of smoke. *Proceedings of SIGGRAPH*, pages 129–136, August 2001.

[6] N. Foster and R. Fedkiw. Pratical animation of liquids. *Proceedings of SIGGRAPH*, pages 15–22, August 2001.

[7] N. Foster and D. Metaxas. Controlling fluid animation. *Computer Graphics International*, pages 178–188, june 1997.

[8] N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. *Proceedings of SIGGRAPH*, pages 181–188, August 1997.

[9] I. Fujishiro and E. Aoki. Volume graphics modeling of ice thawing. *Proceedings of Volume Graphics*, pages 45–50, June 2001.

[10] M. Harris, G. Coombe, T. Scheuermann, and A. Lastra. Physically-based visual simulation on graphics hardware. *SIGGRAPH / Eurographics Workshop on Graphics Hardware*, pages 1–10, Sep 2002.

[11] K. Nagel and E. Raschke. Self-organizing criticality in cloud formation. *Phisica A*, 182:519–531, 1992.

[12] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The volumepro real-time ray-casting system. *Proceedings of SIGGRAPH 99*, pages 251–260, August 1999.

[13] W. T. Reeves. Particle system-a technique for modeling a class of fuzzy objects. *Proceedings of SIGGRAPH*, 17(3):359–376, July 1983.

[14] J. Stam. Stable fluids. *Proceedings of SIGGRAPH*, pages 121–128, August 1999.

[15] D. Stora, P. O. Agliati, M. P. Cani, F. Neyret, and J. D. Gascuel. Animating lava flows. *Graphics Interface*, pages 203–210, June 1999.

[16] D. Terzopoulos, J. Platt, and K. Fleischer. Heating and melting deformable models (from goop to glop). *Graphics Interface*, pages 219–226, June 1989.

[17] D. Tonnesen. Modeling liquids and solids using thermal particles. *Proceedings of Graphics Interface*, pages 255–262, 1991.

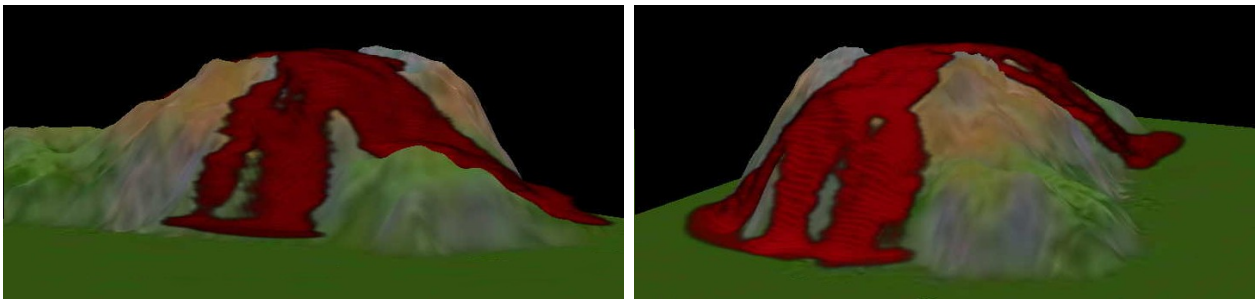
[18] X. Wei, W. Li, K. Mueller, and A. Kaufman. Simulating fire with texture splats. *IEEE Visualization*, pages 227–237, October 2002.

[19] J. Wejchert and D. Haumann. Animation aerodynamics. *Proceedings of SIGGRAPH*, 25(4):19–22, July 1991.

[20] R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. *Proceedings of SIGGRAPH 98*, pages 169–178, July 1998.

Color Plate 1: A tin cello heated by a point heat source.

Color Plate 2: A chocolate goblet in a hot oven.



Color Plate 3: Two views of lava flow generated by our 3D CA, shown on a textured volcano terrain.