Interactive Flowing of Highly Viscous Volumes in Virtual Environments

Xiaoming Wei, Wei Li and Arie Kaufman
Center for Visual Computing (CVC) and Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794-4400
Email:{wxiaomin,liwei,ari}@cs.sunysb.edu

Abstract

We present a simple and linear 3D cellular automata approach for animating the behaviors of viscous flow volumes in virtual environments. An accurate modeling of fluid flow usually requires complicated physical simulations. In this paper, we concentrate on the behaviors of highly viscous fluids, such as wax, lava, plastic, metal, chocolate, etc. We animate the smooth behaviors of viscous fluid based on a local 3D cellular automata. The dynamic changing volume data is rendered directly on a VolumePro board or texture mapping hardware to achieve an interactive speed.

1 Introduction

The realistic computer animations of fluid flow [1, 2, 3] are important and challenging work in virtual environments. However, they are usually computationally expensive due to the complex 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 that leads to animation of highly viscous fluid, while guaranteeing both a low computational cost, stability and visual realism - making it ideal for real-time, interactive applications, such as virtual reality.

Our work is based on the following observations of highly viscous fluid. Its movement is mainly a result of gravity, viscosity damping and friction. There are not much turbulent and rotational behaviors in its movements. In this paper, we simplify the process of complex physically-based calculation and design a 3D cellular automata (CA) to model the movements of highly viscous fluid, such as wax, chocolate and mental, etc. The CA model we designed is inspired by the work of Fujishiro and Aoki [2]. Their method however deals only with the ice thawing problem. Since certain parts of the ice may evaporate just like dry ice, mass is not conserved in their method. While the conservation of mass and the effect of external forces are considered in our CA model.

2 CA Variables

For each cell in the volume, we define a few variables:

- Solid(i,j,k) = $\begin{cases} 1(boolean) & solid \\ 0(boolean) & otherwise \end{cases}$
- Liquid(i,j,k)= $\begin{cases} 1(boolean) & liquid \\ 0(boolean) & otherwise \end{cases}$
- Amount(i,j,k)= $\begin{cases} 1.0 & melt \\ 0.0 & otherwise \end{cases}$

  This variable is used to indicate the amount of liquid at each cell. The value can be transferred between neighboring cells.
- Energy(i,j,k)= $\begin{cases} 1.0 & melt \\ 0.0 & otherwise \end{cases}$

  The variable is used to indicate the expanding potential of a liquid cell. Setting a certain threshold for this variable enables us to control the expanding behavior of the liquid.

3 CA Gravity Rules

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 it does, 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 factor of mass to the cell below it, the energy will be transferred together.

We summarize these transition rules as follows:

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

where *factor* indicates the percentage of liquid moved down. It can be a value up to 1. $\Delta Amount()$ and $\Delta Energy()$ are used to record the change of *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 2 and 3, the conservation of mass is satisfied.

4 CA Spreading Rules

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

- If a liquid cell tends 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 the friction.
- For each of these spreading direction, different 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.

Proceedings of the IEEE Virtual Reality 2003 (VR'03)
1087-8270/03 $17.00 © 2003 **IEEE**

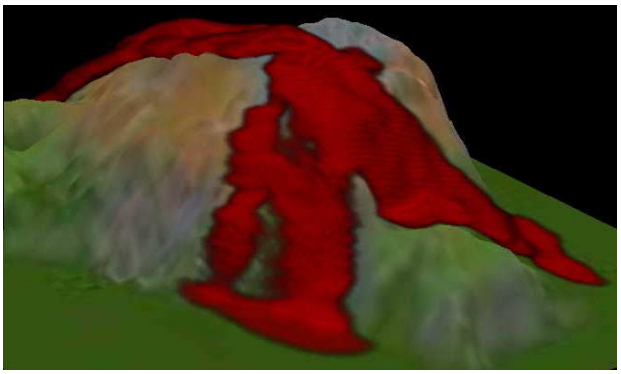

Figure 1: A view of the lava flow generated by our 3D cellular automata, shown on a textured volcano terrain.

The transition rules related to spreading are organized as follows:

$$Liquid(i+1,j,k,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i+1,j,k,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (6)$$

$$Liquid(i-1,j,k,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i-1,j,k,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (7)$$

$$Liquid(i,j,k+1,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i,j,k+1,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (8)$$

$$Liquid(i,j,k-1,t+1) = Liquid(i,j,k,t) \wedge \neg Solid(i,j,k-1,t) \wedge IS(Energy(i,j,k) > Threshold) \quad (9)$$

$$\Delta Amount(i+1,j,k,t+1) += Amount(i,j,k,t) * factor1 \quad (10)$$

$$\Delta Amount(i-1,j,k,t+1) += Amount(i,j,k,t) * factor2 \quad (11)$$

$$\Delta Amount(i,j,k+1,t+1) += Amount(i,j,k,t) * factor3 \quad (12)$$

$$\Delta Amount(i,j,k-1,t+1) += Amount(i,j,k,t) * factor4 \quad (13)$$

$$\Delta Amount(i,j,k,t+1) -= (factor1 + factor2 + factor3 + factor4) * Amount(i,j,k,t) \quad (14)$$

$$\Delta Energy(i+1,j,k,t+1) += Energy(i,j,k,t) * \alpha * factor1 \quad (15)$$

$$\Delta Energy(i-1,j,k,t+1) += Energy(i,j,k,t) * \alpha * factor2 \quad (16)$$

$$\Delta Energy(i,j,k+1,t+1) += Energy(i,j,k,t) * \alpha * factor3 \quad (17)$$

$$\Delta Energy(i,j,k-1,t+1) += Energy(i,j,k,t) * \alpha * factor4 \quad (18)$$

$$\Delta Energy(i,j,k,t+1) -= Energy(i,j,k,t) * \alpha * (factor1 + factor2 + factor3 + factor4) \quad (19)$$

where α is the energy conservation coefficient, used to simulate the effect of friction. Different coefficient numbers can be used to describe the viscosity friction between liquid cells and the friction between liquid and solid, or floor. $factor1$, $factor2$, $factor3$ and $factor4$ are coefficients used to describe the amount of mass spread to the neighboring non-solid cells. As we add Equations 10 to 14, we see that the conservation of mass is also satisfied in this step.

5 Implementation and Results

To render the fluid volume, we update the density values of the voxels according to their current states in the cellular automata. For the solid voxels, they remain at their original position without changing the intensity values. For those 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 the sudden change of intensity value. Only the liquid voxels and their neighbors that are affected by the low-pass filtering are updated. We present a volcano terrain example to demonstrate our modeling method. The results have been generated on a PIV 2.53GHz PC. In this example, we consider the lava flow as a highly viscous fluid and it does not solidify and destroy the surrounding environments. Its movement is guided by the underlying terrain. The size of the volcano terrain volume is $128 \times 100 \times 128$. The Lava flow is generated from the top of the volcano mountain. User can define the location and amount of the new generated lava flow at each cell. The cellular automata model generates a time-varying lava flow volume. The average simulation time needed for this application is 3.2ms for each "gravity force" step and 21.6ms for each "spreading" step. The final images are rendered by combining the lava flow volume with the underlying textured volcano terrain based on the depth value. Figure 1 and 2 show the images of the lava flow on the textured terrain and the Responsive Workbench.

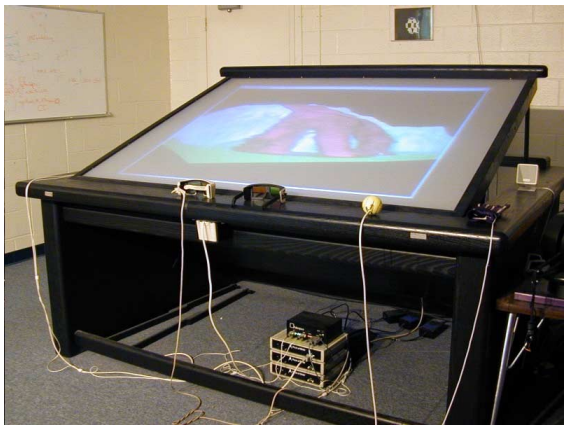

Figure 2: A volcano terrain with time-varying lava, shown on our Responsive Workbench.

6 Conclusions

We have presented a 3D CA approach to simulate the movements of highly viscous fluid in virtual environments. The method can be used in the volcano terrain simulation and the wax, chocolate, metal, ice objects melting simulations. Our model consists only of bit operation and simple addition and multiplication operations, which results in an interactive simulation speed, making it suitable for virtual reality applications. The model is also ideally suitable for hardware acceleration. Experiments have been carried out to demonstrate the feasibility of the current model.

Acknowledgments

This work has been supported by ONR grant N000140110034.

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

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