

Animation of Boiling Phenomena

Abdullah Bulbul, Onur Kucuktunc, Bulent Ozguc
{bulbul, onurk}@cs.bilkent.edu.tr, ozguc@bilkent.edu.tr

Department of Computer Engineering, Bilkent University, Ankara, Turkey

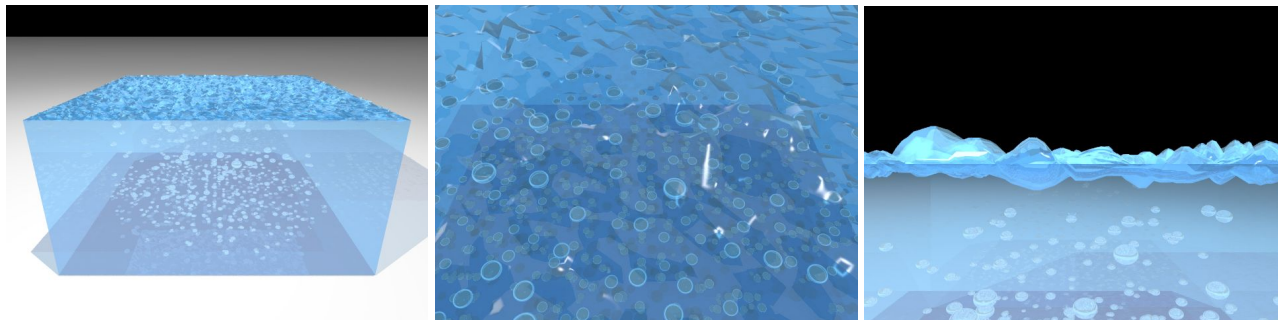


Fig. 1. Example images from our simulation (using raytracing)

ABSTRACT

Phenomenon of boiling is a challenging topic for computer graphics due to its complex hydrodynamics and formulation. Realistic fluid animations require very heavy three-dimensional fluid flow calculations, and surface estimations as well. However, realism and performance are the two important objectives of the boiling animation for a real-time application. We present an efficient method for the simulation of boiling water in this paper. The method is based on modeling the bubbles and waves as particles. Grid-based approach is used both for the heating and the fluid surface. Our technique makes it possible to produce the animation of boiling phenomena nearly in real-time.

Keywords— Animation, fluid simulation, implicit surfaces, boiling phenomena

1. INTRODUCTION

Fluid simulation has been a popular topic in the field of physics-based animation for over a decade. Obtaining visually satisfying animations that reflect real life physical events require modeling according to the laws of physics.

Recent research efforts on this topic focus on minimizing the complexity of the mechanisms about hydrodynamics. Similarly, the aim of this paper is to formulate the generation of the bubble and wave particles, and also interaction between them in such a way that no complex computations are needed for a full fluid model.

1.1. Background Information

Some methods are widely used in this area. Navier-Stokes equations are non-linear partial differential equations, which describe the motion of fluids. Shallow water (SW) equations simplify the complexity of Navier-Stokes equations, and form a 2D representation of a height field.

Smoothed particle hydrodynamics (SPH) is a method to solve Navier-Stokes equations with a Lagrangian approach. It is used for simulating fluid flows, and guarantees the conservation of mass without extra computations. It computes the attributes of the fluid particles from the neighboring particles rather than solving a linear equation system. Therefore, SPH can run in real-time when simulating fluid motion.

Water surface is generally represented as a 2D array of heights. Then these surface points are used as control points of a Spline algorithm, such as B-Spline, Bézier surface, etc.

1.2. Our Contributions

Our aim is to find a simplified method for the simulation of boiling water in real-time with a simple particle system. To achieve this goal, the contributions of this paper are:

- an efficient simulation model for boiling water that defines only the bubbles as particles,
- a simple wave model that has addition and reflection capabilities.

2. RELATED WORK

We briefly overview some of the related work on fluid-fluid interactions, simulations of bubbles, wave particles, and the animation of liquids in this section.

Thürey *et al.* present a technique to perform simulations of bubbles and foam in real-time [1]. They use SPH with surface tension for formation and clustering of foam. Bubble approaching the fluid surface is handled like our method. The usage of shallow water (SW) reduced the complexity of Navier-Stokes equation.

In [2], Yüksel, House and Keyser discuss an approach for the real-time simulation of fluid surface waves. Waves are efficiently generated by the water-object interactions. By the help of newly developed graphics hardware, they offer very fast and stable wave simulation. Wave generation is explained in detail, and the superposition of all the wave particles forms the water surface.

Holmberg and Wünsche propose a method for modeling and rendering turbulent water over natural terrain in [5]. They use hydrostatic theory for 2D height field, and a particle system to model the turbulent water. The results have shown that a smooth representation of water surface by fitting a uniform B-Spline surface to the height field provides realistic simulations for the given purpose. Fitting height field to a Spline surface is also used in our approach.

Particle based fluid-to-fluid interaction is discussed by Müller *et al.* in [4]. A particle system is preferred since particles with different properties are mixed in order to simulate boiling water or water flowing into a glass. Their approach is mainly based on SPH method, and handles phase changes like liquid to gas in boiling.

One other important work about boiling water simulation is done by Mihalef *et al.* [3]. They propose a new Eulerian method that uses physics based models to obtain realistic results. Circulation of the water, very energetic boiling called roiling-boil, and interactions with heated object are the points that this paper contributes for the topic.

There is also some notable work done by Foster and Fedkiw [6] about particle-based animation of fluids. They used semi-Lagrangian method to efficiently solve the fluid flow equations and combined the strengths of level set method with particle system to form a hybrid surface model.

3. WATER AND BUBBLE MODELS

We begin by explaining our method with the water and bubble models, introducing the detailed information about bubble generation and merging in the next section.

3.1. Heating and Diffusion

We defined the floor of the water container with a grid-based model in order to manage bubble generation easily according to heating. In the rest of the paper, $FT_{x,y}$ will stand for the temperature of grid element at location (x,y) .

Also, to avoid an ordered bubble generation and get more realistic results we add a random value that affect floor points. The initiation equation of the grid cells is

$$FT_{x,y}(0) = T_{initial} + rand[-5,5]$$

Heating of the floor grids is performed by diffusing temperatures of some grids that are directly affected by the energy source. The diffusion is done according to following formula:

$$FT_{x,y}(t) = \frac{\sum_{i=x-a}^{x+a} \sum_{j=y-a}^{y+a} FT_{i,j}(t-1)}{(2a+1)^2}$$

where a defines the speed of diffusion.

3.2. Bubble Particles

Bubble is the most important particle in our system since we do not model the water elements as particles.

Important attributes of a bubble particle are size, position, moving direction, velocity, temperature, and the phase.

3.2.1. Bubble Generation

The temperatures of the floor grids determine when and where the bubbles are generated. When the temperature of a floor grid reaches to 100 degrees Celsius, a bubble is generated at a random place within this grid. This randomness avoids bubble generation to be grid-like.

When a bubble is generated in a floor grid, a portion of grid's energy is transferred into the bubble, so temperature of this grid decreases respectively. This adjustment modifies the heating formula as

$$FT_{x,y}(t) = \begin{cases} \frac{\sum_{i=x-a}^{x+a} \sum_{j=y-a}^{y+a} FT_{i,j}(t-1)}{(2a+1)^2} - T_{gen} & FT_{x,y}(t-1) \geq 100 \\ \frac{\sum_{i=x-a}^{x+a} \sum_{j=y-a}^{y+a} FT_{i,j}(t-1)}{(2a+1)^2} & \text{otherwise} \end{cases}$$

Naturally, bubbles stick to the container when they are generated. Size of a newly generated bubble is 1, and they become larger when they merge with others. In our system, a size threshold is defined for bubbles to leave the surface. When the size of a bubble particle exceeds this threshold, it changes phase and becomes free to move in the water.

3.2.2. Merging Bubbles

Two bubbles are merged when they touch each other. Bubble merging may happen in two cases: when they are moving up, and after they are generated.

The position of the merged bubble will be adjusted as the weighted average of positions of the bubbles according to their sizes. In the following formulas, properties of the merged bubble (B_m) are shown after bubbles B_1 and B_2 merges. p , s , and v stand for position, size and velocity.

$$p(B_m) = \frac{p(B_1)s(B_1) + p(B_2)s(B_2)}{s(B_1) + s(B_2)}$$

$$s(B_m) = s(B_1) + s(B_2)$$

$$v(B_m) = \frac{v(B_1)s(B_1) + v(B_2)s(B_2) + p(B_1)s(B_2) + p(B_2)s(B_1)}{s(B_1) + s(B_2)}$$

The equations above are valid in both merging cases. An example scene of merging event is shown in Fig. 2.

3.2.3. Bubble Movement

In our system, movements of bubbles are affected by two attributes: size and the height of the water above it. The following equation gives the y component of a bubble's velocity.

$$r_{B_x} = c_2 \left(1 + \frac{p(B_x)_y}{h_c}\right) \sqrt[3]{s(B_x)}$$

$$\Delta v = c_1 (h_c - p(B_x)_y) r_{B_x}^3$$

$$v_y(B_x)_{t+1} = v_y(B_x)_t + \Delta v$$

In the equations, c_1 and c_2 are constants, Δv is the velocity change between two time steps in y axis, h_c is the height of the container. Another factor that affects velocity of a bubble is the merging event. When two bubbles are merged, the new bubble gets a velocity towards the smaller (see $v(B_m)$ equation), which is decayed in time. The y component of a bubble is not affected by this decay. We do not use external forces that affect movements of bubbles but they can be added simply by acting on velocities of bubbles.

4. FLUID SURFACE SIMULATION

Fluid surface should be modeled and rendered very carefully, because it is as important as the bubble modeling in order to get a realistic animation of boiling water. We have used the height-fields method for constructing the surface. Like the grid-based floor model, there is also a grid structure for the water surface. Each point on this grid is initially zero and affected by wave particles and bubbles that approach to the surface. $SH_{i,j}$ denotes surface height of a grid point at location (i,j) , and calculated as follows:



Fig. 2. Merging bubbles on the floor

$$SH(i, j) = Effect_{waves}(i, j) + Effect_{bubbles}(i, j)$$

4.1. Wave Particles

Wave particles are used to simulate waves. A wave particle has the following properties: time, center, height, width and size. Size and center attributes are initiated with the size and position of the bubble that generates the wave. Time is used to indicate how much time has passed after the wave is generated. Height and width are indicated as functions of size and time. They are proportional to $\sqrt[3]{size/time}$.

Using these properties, each wave particle generates a ring-like wave and their effects are summed up to get the overall wave effect.

$$Effect_{waves}(i, j) = \sum_{k \in W} EW_k(i, j)$$

$$EW_k(i, j) = h(W_k) \sin \frac{(r - h(W_k) - d_c)\pi}{w(W_k)}$$

where $EW_k(i,j)$ stands for effect of wave k for surface grid (i,j) ; $h(W_k)$ and $w(W_k)$ for height and width of wave k; d_c for distance of the surface grid to center of the wave. Lastly, radius of the wave (r) is a function of time attribute of the wave particle. The above formula is calculated only for the waves that affect the specified grid.

A wave particle has also reflection capability. When a wave reaches to an edge of the container, we calculate the wave effect as if there are grid cells outside the container. Those effects are then transferred to the symmetric grid cells inside.

4.2. Effects of Bubbles

Each bubble particle increases the height of the fluid surface grids above it. A threshold value indicates effective distance, and the effect of a bubble linearly increases when it approaches the surface. It is equal to the height when a bubble reaches to surface.

$$Effect_{bubbles}(i,j) = \sum_{k \in B} EB_k(i,j)$$

$$EB_k(i,j) = \frac{d_b(p(B_k)_y - (h_c - th))}{th}$$

In the above equation, effect of bubble k to the surface grid (i,j) is formulated. d_b indicates the distance between the grid and projection of center of the bubble to the surface, and th is the threshold value we mentioned earlier.

4.3. Surface Estimation

After getting all the height values for the surface grids, we can use these points to generate the surface. If the surface is divided into enough cells, the results will be more realistic. But using too many surface grids will increase the execution time remarkably. A better way is to use the surface grids as control points for a spline-based interpolation technique. We have used B-Spline algorithm for surface estimation. Between each 2 consecutive height values, 4 points are calculated and added, which results to 21 additional points in a surface cell. In our implementation the grid-based surface is an 80x80 matrix, after surface estimation we have 400x400 height values. A comparison of using B-Spline surface interpolation is shown in Figure 3.

5. RESULTS

Boiling water simulations based on our method using OpenGL were run on a PC with Centrino¹ 1.6 GHz CPU in real-time. Fluid surface was drawn in wireframe, and the bubble limit was 2,000. When we add surface interpolation and texture mapping to the surface, the frame rate of the simulation severely decreases. Yet, bubble and wave generations and calculations can be done in real-time.

The simulations shown in Figure 1, 2, and 3 are run on a Core 2 Duo¹ CPU (2.2 GHz) and without any GPU optimization. Rendering the frames in ray-tracer with 640x480 resolution and anti-aliasing enabled takes 5 to 50 seconds depending on the complexity of the scene. These results are generated with Pov-Ray v3.6 for more realistic rendering. Sample image and video results can be seen at (<http://cs.bilkent.edu.tr/~bulbul/boiling08/>).

6. CONCLUSIONS

We have proposed an efficient technique for the simulation of boiling water in this paper. Instead of using a full fluid model, we only model bubbles and waves as particles to simplify the complexity of calculating hydrodynamic formulations. Grid-based approach is used for modeling the heating and the fluid surface easily. Results have shown that

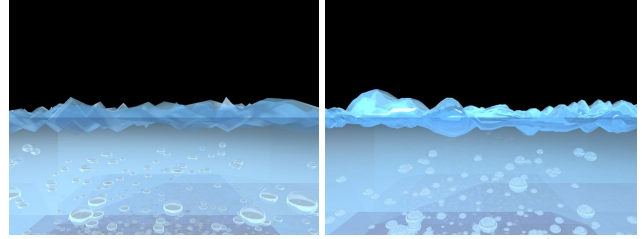


Fig. 3. Fluid surface before (left) and after (right) B-Spline interpolation.

with our technique, we can produce the animation of boiling water nearly in real-time. Moreover, it is also possible to get better rendering results of this simulation by using offline shading techniques, such as ray tracing.

7. ACKNOWLEDGEMENTS

This work is supported by European Union 6th Framework Program under Grant No. FP6-511568 (3DTV NoE Project).

REFERENCES

- [1] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross, “Real-time simulations of bubbles and foam within a shallow water framework,” in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 191-198, 2007.
- [2] C. Yuksel, D. H. House, and J. Keyser, “Wave particles,” in *Proceedings of ACM SIGGRAPH*, pp. 99, 2007.
- [3] V. Mihalef, B. Unlusu, D. Metaxas, M. Sussman, and M. Y. Hussaini, “Physics based boiling simulation,” in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 317-324, 2006.
- [4] M. Müller, B. Solenthaler, R. Keiser, and M. Gross “Particle-based fluid-fluid interaction,” in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 237-244. 2005.
- [5] N. Holmberg, and B. C. Wünsche, “Efficient modeling and rendering of turbulent water over natural terrain,” in *Proceedings of the 2nd international Conference on Computer Graphics and interactive Techniques in Australasia and South East Asia*, pp. 15-22, 2004.
- [6] N. Foster, and R. Fedkiw, “Practical Animation of Liquids,” in *Proceedings of ACM SIGGRAPH*, pp. 15-22, 2001.

¹ Centrino® and Core™ 2 Duo are registered trademarks of Intel Corporation