Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005)
K. Anjyo, P. Faloutsos (Editors)

Particle-Based Fluid-Fluid Interaction

Matthias Müller[1], Barbara Solenthaler[2], Richard Keiser[3], Markus Gross[3]

[1] NovodeX AG Zürich/AGEIA Inc. St. Louis
[2] Visualization and Multimedia Lab, University of Zürich
[3] Computer Graphics Lab, ETH Zürich

Abstract

The interesting and complex behavior of fluids emerges mainly from interaction processes. While interactions of fluids with static or dynamic solids has caught some attention in computer graphics lately, the mutual interaction of different types of fluids such as air and water or water and wax has received much less attention although these types of interaction are the basis for a variety of important phenomena.
In this paper we propose a new technique to model fluid-fluid interaction based on the Smoothed Particle Hydrodynamics (SPH) method. For the simulation of air-water interaction, air particles are generated on the fly only where needed. We also model dynamic phase changes and interface forces. Our technique makes possible the simulation of phenomena such as boiling water, trapped air and the dynamics of a lava lamp.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object ModelingPhysically Based Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and RealismAnimation and Virtual Reality

1. Introduction

Fluid simulation has become a hot topic in the field of physically-based animation in recent years. Fluids add substantially to the richness of a virtual world due to their ability to assume arbitrary shapes and to show complex behavior.

A fluid is defined as a substance that cannot support shear stress in static equilibrium or more intuitively, a substance that flows because it cannot resist deformation. In order to keep fluids in place, they have to be kept in tanks or other containers. This is the reason why the interaction of fluids with solid boundaries plays a major role in fluid simulations. Fluid-solid interaction has been addressed in many papers.

Another important phenomenon is the mutual interaction of different kinds of fluids. This type of interaction has not received as much attention in Computer Graphics so far although many interesting phenomena emerge from it. In boiling water for instance, a liquid interacts with a gas while water constantly changes its phase from liquid to gasiform. When water flows into a glass, air pockets get trapped in the fluid and form bubbles. In a lava lamp, two types of fluids with different polarity and density interact with each other and generate interesting motions.

In this paper, we propose a particle-based method to simulate the phenomena emerging from fluid-fluid interaction. With Eulerian, grid-based methods, the simulation of multiple fluids or multiple phases is a difficult problem. Particles seem to be the natural choice in this case. In a particle system, each individual particle can have its own attribute values and particles with different properties can be mixed arbitrarily. In addition, particles can easily be generated and deleted dynamically as needed.

1.1. Our Contributions

Our approach to simulating fluid-fluid interactions consists of a set of extensions to the regular SPH method targeting the following phenomena:

- *Multiple fluids*: To simulate multiple fluids with different particle types, we store parameters which are global in standard SPH-models on individual particles and extend the equations to cope with additional particle attributes.

Computer Animation 2005, 29–31 July 2005, Los Angeles CA

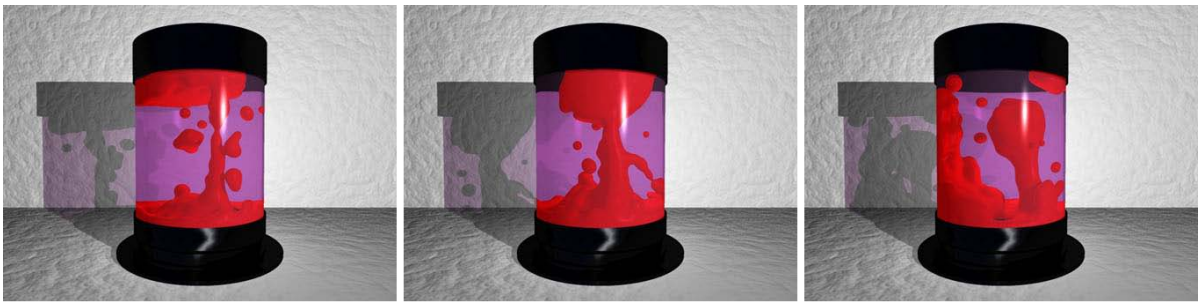


Figure 1: *The techniques presented in this paper allow the simulation of a lava lamp. The interesting effects inside the lamp are caused by interface tension forces between fluids of different polarity, temperature diffusion and buoyancy forces.*

- *Trapped Air*: We simulate trapped air by generating air particles dynamically where air pockets are likely to be formed. With this approach, we avoid a full multi-phase simulation. Air particles that are isolated from the liquid are deleted.
- *Phase transitions*: The phenomenon of boiling water is modeled by changing the types and densities of particles dynamically according to their temperatures.

We solve the diffusion equation on the particles to simulate the exchange of attribute values such as temperature or color as in [SAC*99].

1.2. Related Work

The proposed extensions draw from previous work in both, Eulerian and particle-based fluid simulation. With their paper on the realistic animation of liquids, Foster *et al.* [FM96] introduced fluid simulation to computer graphics. Their method was extended by stable semi-Lagrangian advection [Sta99] and level set methods [FF01, EMF02] to track the free surface of liquids. The problem of fluid solid interaction in the Eulerian setting has been addressed by Carlson *et al.* [CMT04]. However, the simulation of multiphase fluids and bubbles with the Eulerian approach is a difficult problem as [BT99] and [WGS00] show. Recently, Greenwood *et al.* [GH04] proposed a simpler method to simulate bubbles. They generate passive air-particles and advect them using the Eulerian velocity field. In contrast to our model, their bubbles have no effect on the fluid. Our model handles the full two-way coupling of water and air. Since bubbles are modeled as spheres in [GH04], they do not change their spherical shape and merging and splitting is not possible. Our technique can handle these effects as Fig. 2 shows. Hong *et al.* [HK03] use the volume of fluid method (VOF) on an Eulerian grid to simulate bubbles in liquids. Because the coarse grid only allows the simulation of relatively large bubbles, a hybrid approach is used. The smaller bubbles are simulated using a passive particle system like [GH04] with the same limitations.

As we show in this paper, Lagrangian, particle-based fluid models allow the seamless modeling of fine to large scale fluid-fluid interaction phenomena in a natural way. Most particle-based fluid models are based on the Smoothed Particle Hydrodynamics formulation [Mon92] with some exceptions [PTB*03]. SPH is used in [DC96] to animate highly deformable solid objects, or lava [SAC*99]. In the latter work the diffusion equation is solved to simulate heat transfer in a single fluid. The basic SPH-model we use is based on the work of Müller *et al.* [MCG03]. They also proposed a method for fluid-solid interaction for particle-based fluids [MST*04].

2. Fluid Model

In this section we shortly summarize the SPH fluid model proposed in [MCG03]. We then explain how we extend this model to handle multiple fluids.

2.1. SPH Model

A fluid is represented by a set of particles $i \in [1 \dots N]$ with positions $\mathbf{x}_i$, masses m_i and additional attributes A_i. SPH defines how to compute a smooth continuous field $A(\mathbf{x})$ from discrete attribute values A_i sampled at particle locations $\mathbf{x}_i$ as

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

where ρ_i is the density of particle i and $W(\mathbf{r}, h)$ is a smoothing kernel. The density ρ_i can be computed with Eq. (1) yielding

$$\rho_i = \rho(\mathbf{x}_i) = \sum_j m_j W(\mathbf{x}_i - \mathbf{x}_j, h). \tag{2}$$

The kernel function $W(\mathbf{r}, h)$ is typically a smooth, radially symmetric, normalized function with finite support, i.e. $\int W(\mathbf{r}, h) d\mathbf{r} = 1$ and $W(\mathbf{r}, h) = 0$ for $|\mathbf{r}| > h$. The gradient

and Laplacian of the smoothed attribute function $A(\mathbf{x})$ are respectively

$$\nabla A(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{x}-\mathbf{x}_j, h) \text{ and} \quad (3)$$

$$\nabla^2 A(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{x}-\mathbf{x}_j, h). \quad (4)$$

Substituting Eq. (1) into the Navier-Stokes equations and symmetrization yield the following particle body forces

$$\mathbf{f}_i^{\text{pressure}} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_{ij}, h) \quad (5)$$

$$\mathbf{f}_i^{\text{viscosity}} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_{ij}, h), \quad (6)$$

where $\mathbf{v}_i$ are the particles' velocities, μ the viscosity coefficient of the fluid and $\mathbf{r}_{ij}$ the distance vector $\mathbf{x}_i - \mathbf{x}_j$. We use the kernels proposed in [MCG03] to compute these body forces. The pressures p_i are computed via the constitutive equation

$$p_i = k(\rho_i - \rho_0), \quad (7)$$

where k is the stiffness of the fluid and ρ_0 its rest density. Finally, for the acceleration $\mathbf{a}_i$ of a particle i we have

$$\mathbf{a}_i = 1/\rho_i (\mathbf{f}_i^{\text{pressure}} + \mathbf{f}_i^{\text{viscosity}} + \mathbf{f}_i^{\text{external}}), \quad (8)$$

where $\mathbf{f}_i^{\text{external}}$ are external body forces such as gravity or forces due to surface tension.

2.2. Multiple Fluids

In the standard approach for a single fluid, many attributes are identical for all particles and can be stored globally, e.g. the particle mass m, the rest density ρ_0, the gas constant k and the viscosity coefficient μ. In our approach each particle carries all those attributes individually. We also introduce new attributes. For the simulation of interface and surface tension, we use color attributes c^i and c^s. In addition, particles store a temperature T. The attributes of a particle are summarized in Table 1.

The SPH fluid model given by Eq. (5) and Eq. (6) is already designed to handle individual particle masses m_i. However, because each particle now carries its own viscosity coefficient, we modify the computation of the viscosity force in Eq. (6) as follows

$$\mathbf{f}_i^{\text{viscosity}} = \sum_j \frac{\mu_i + \mu_j}{2} m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_{ij}, h), \quad (9)$$

meaning that the individual viscosity coefficients are averaged between interacting particles.

Attribute	Description	Unit
m	mass	kg
$\mathbf{x}$	position	m
$\mathbf{v}$	velocity	m/s
$\mathbf{f}$	accumulated body forces	N/m^3
ρ_0	rest density	kg/m^3
ρ	actual density	kg/m^3
μ	viscosity	Ns/m^2
k	gas constant (stiffness)	Nm/kg
c^i	color for interface tension	1
c^s	color for surface tension	1
T	Temperature	Celsius

Table 1: *Attributes stored with each particle.*

3. Interaction Methods

By storing fluid attributes with individual particles, we are now able to model a variety of fluid-fluid interaction phenomena.

3.1. Buoyancy

Buoyancy is a phenomenon which simply emerges from the fact that we use individual rest densities ρ_0 per particle. The parameter ρ_0 in the constitutive law defined in Eq. (7) has the effect that the pressure term not only pushes close particles apart but also contracts particles in order to reach a non-zero rest density. When two fluids with different rest densities are mixed, a density gradient and, thus, a pressure gradient will emerge at their common interface. This pressure gradient will cause the less dense fluid to rise inside the denser fluid.

3.2. Immiscible Liquids

The solubility of two liquids depends mainly on the interactions of the different types of molecules. Water and oil are immiscible because the water molecules are polar while the oil molecules are not. In the polar water molecule there are excess electrons on the oxygen side and a lack of electrons or excess of positive charges on the hydrogen side. This asymmetry causes the water molecules to form hydrogen bonds. The energy of bonded water molecules in a cluster is lower than the energy of single water molecules dispersed in a homogeneous mixture of oil and water. Therefore, water droplets spontaneously form inside the oil.

We model this behavior by considering the interface tension between polar and non-polar liquids. We introduce an interface body force which acts perpendicular to the interface of

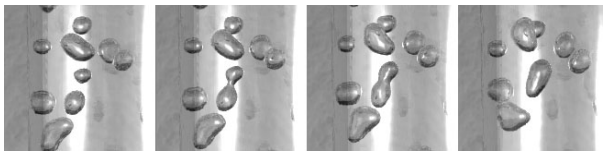


Figure 2: *Small bubbles merge to larger bubbles due to surface tension forces.*

Particle Type	c^i	c^s	c^p
air particle	0	0	1
polar liquid particle	-0.5	1	1
non-polar liquid particle	$+0.5$	1	1

Table 2: *The color values used for different particles types.*

two liquids trying to minimize the curvature κ of the interface. The force is proportional to κ and the interface tension coefficient σ^i which, in turn, depends on the two interacting fluids.

In [MCG03], the authors describe a surface tension term which models an analogous effect between water and air. We adapt their technique to model the interface tension between immiscible liquids. The color attribute c^i of polar particles is set to -0.5 while non-polar particles get a value of $+0.5$.

The gradient ∇c^i of the smoothed color field c^i can be computed using Eq. (3). We use it as an estimation of the normal on the interface $\mathbf{n} = \nabla c^i$. With this normal, the interface body force becomes

$$\mathbf{f}^{\text{interface}} = |\mathbf{n}|\sigma^i \kappa \frac{\mathbf{n}}{|\mathbf{n}|}, \tag{10}$$

where the leading factor $|\mathbf{n}|$ amplifies the force at the interface itself, σ^i is the interface tension coefficient and the curvature κ can be estimated from the divergence of the normal $\mathbf{n}$ as $\kappa = -\nabla^2 c^i / |\mathbf{n}|$. We then get the final equation for the body force

$$\mathbf{f}^{\text{interface}} = -\sigma^i \nabla^2 c^i \frac{\mathbf{n}}{|\mathbf{n}|}. \tag{11}$$

To simultaneously model surface tension, the color attribute c^s of any type of liquid particle is set to 1 while the surrounding air implicitely gets $c^s = 0$. Later, when we introduce air particles in Section 3.4.1, we explicitely set $c^s = 0$ on those particles. Surface tension forces are computed as

$$\mathbf{f}^{\text{surface}} = -\sigma^s \nabla^2 c^s \frac{\mathbf{n}}{|\mathbf{n}|}, \tag{12}$$

where the normal is $\mathbf{n} = \nabla c^s$ in this case. Fig. 2 shows the merging of bubbles due to surface tension forces.

3.3. Diffusion

Diffusion effects play an important role in connection with fluids. The diffusion equation describes how heat gets distributed in a fluid or how cream dissolves in a cup of coffee. The evolution of an attribute $A(\mathbf{x},t)$ due to diffusion is given by

$$\frac{\partial A}{\partial t} = c\nabla^2 A. \tag{13}$$

Using the SPH formalism, the evolution of discrete attributes A_i sampled at the particles can be computed as

$$\frac{\partial A_i}{\partial t} = c\sum_j m_j \frac{A_j - A_i}{\rho_j} \nabla^2 W(\mathbf{r}_{ij}, h). \tag{14}$$

We integrate the attributes in time using a simple Euler-scheme:

$$A_i \leftarrow A_i + \Delta t \frac{\partial A_i}{\partial t}. \tag{15}$$

To simulate a lava lamp (Fig. 1), we model the evolution of the temperature T_i of individual particles with the diffusion equation and let the temperature influence the rest density ρ_0. According to the ideal gas law, temperature and density are inversely proportional

$$\rho_0 \sim \frac{1}{V} \sim \frac{1}{T}, \tag{16}$$

where V is the volume of the gas. We, thus, set $\rho_0 = \alpha/T$, where α is a user defined constant.

3.4. Trapped Air

With the surface term in Eq. (12) and the surface tension coefficient σ^s we can control the free surface of the fluid. Splashing particles will cluster into drops and droplets. However, since air is not explicitly modeled in the standard SPH approach, trapped air inside the fluid will immediately disappear because the fluid particles will close the vacuity without resistance.

One way to solve this problem would be to explicitly simulate air as a separate fluid. However, one advantage of a particle-based fluid simulation over an Eulerian model is that the fluid can move freely in space without a bounding grid. Thus, in order to model the surrounding air, a large number of air particles would have to be introduced. The solution to this problem is to generate air particles on the fly whenever bubbles are about to be formed and to delete the particles when they don't contribute to the simulation anymore.

3.4.1. Air Particle Generation

In general, air particles need to be generated near the surface of a liquid, i.e. where the gradient ∇c^s of the surface color field is large. However, the generation needs to stop when

there are enough air particles near the surface. To this end, we introduce a third but implicit color attribute c^p. It is 1 on air particles as well as on all other particles and, therefore, does not need to be stored explicitely. Table 2 summarizes all color fields and their values on the different particle types.

While a large ∇c^s indicates the liquid-air interface, a large ∇c^p indicates the boundary of any type of particles. Thus, air particles need to be generated at locations where both, ∇c^s and ∇c^p are large (see Fig. 3). Because only liquid particles generate air particles, it is enough to test whether ∇c^p is large. Liquid particles that fulfill this criterion generate air particles. The location of the air particle is chosen to be the location of the liquid particle shifted by the vector $-d\nabla c^p$, where d is a user defined distance. The velocity of the air particle is initialized with the velocity of the liquid particle.

The procedure just described produces particles everywhere near the liquid-air interface. However, an air particle is only a good candidate for being trapped if it is located below the liquid front. Therefore, we only generate air particles if the vertical component (y-component) of ∇c^p is positive. In summary, the condition for a liquid particle with location $\mathbf{x}$ and velocity $\mathbf{v}$ to generate an air particle is

```
if |∇c^p|_x > t_p and ∇c^p · e_y|_x > 0.0
    generateAirParticle(x − d ∇c^p|_x, v);
```

where $\mathbf{e}_y$ is the vertical unit vector and t_p a scalar threshold parameter for air particle generation.

3.4.2. Air Particle Deletion

A simple criterion for the deletion of air particles would be to test whether the vertical component of ∇c^p is negative near the particle. This criterion can cause the deletion of single air particles surrounded by water. To avoid this, we only delete air particles whose ∇c^s is sufficiently small, i.e. air particles who are sufficiently far from liquid particles. With this criterion, air particles inside large trapped bubbles get deleted. Their deletion can be avoided by testing whether ∇c^p is larger than a threshold, because inside a bubble filled with air particles ∇c^p is close to zero while on the boundary of the fluid, it is large. One last problem are isolated strayed air particles. A simple test to identify isolated air particles is to check whether their actual density ρ got below a threshold. In summary, the condition to delete an air particle with location $\mathbf{x}$ and actual density ρ is

```
if ( |∇c^s|_x < t_s and |∇c^p|_x > t_p ) or ρ < t_ρ
    deleteAirParticle();
```

3.4.3. Artificial Buoyancy

We already mentioned how the locations and velocities of air particles are initialized. The mass m of air particles is chosen such that the spacing of air particles is about the same as the spacing of the liquid particles in the rest state. Another important quantity to be specified is the rest density ρ_0 of air

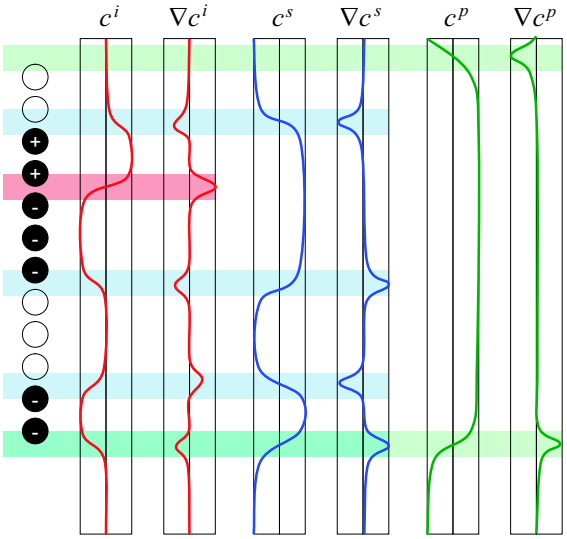


Figure 3: *The three color fields used and their gradients in a 1D example. Air particles are shown in white, polar liquid particles in black with a minus and non-polar liquid particles with a plus. Liquid-liquid interfaces are indicated by the larger peaks in ∇c^i. Peaks in ∇c^s are found at liquid-air interfaces and peaks in ∇c^p show the boundary of particle clusters. At the bottom, new air particles are generated since both, ∇c^s and ∇c^p show a peak and are both positive. At the top air-liquid interface, ∇c^p is zero, indicating that air particles are already present and don't need to be generated.*

paticles. The density of water is about a thousand times the density of air. This large ratio can cause stability problems in a SPH simulation. In our demos, we used a rest density of 1000 kg/m^3 for water and 100 kg/m^3 for air particles. However, this smaller ratio of 10 causes bubbles to rise more slowly in water. To compensate for this effect, we introduce an artificial buoyancy force.

Another reason to introduce a buoyancy force is the fact that the SPH formalism is not well suited for small air bubbles consisting of only a few air particles. SPH only works well when objects are reasonably well sampled. Due to the overlap of kernels from surrounding dense water particles the density of isolated air particles computed via Eq. (2) gets too high. Such particles only rise slowly or not at all. We choose the buoyant force to be proportional to the difference of the measured density ρ of the air particle and the rest density of air ρ_0

$$\mathbf{f}^{\text{buoyant}} = b(\rho - \rho_0)\mathbf{g}, \tag{17}$$

where $\mathbf{g}$ is gravity and b a user parameter to control the buoyancy effect.

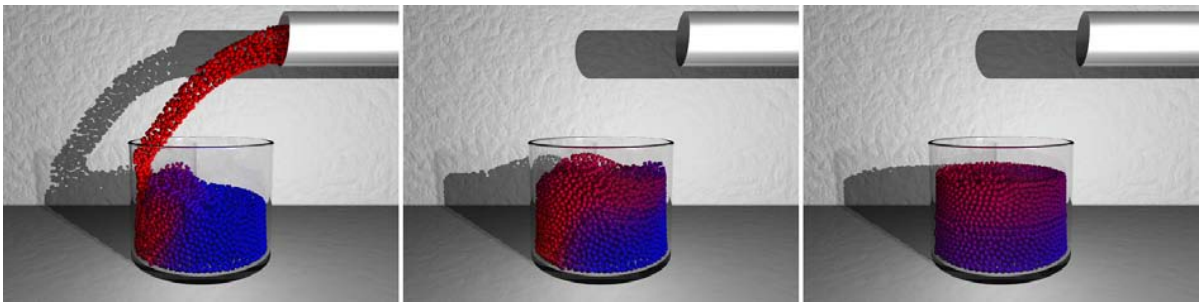


Figure 4: *Two types of fluid are mixed. The less dense fluid (red particles) rises to the surface while color differences are smoothed out at the interface due to diffusion effects.*

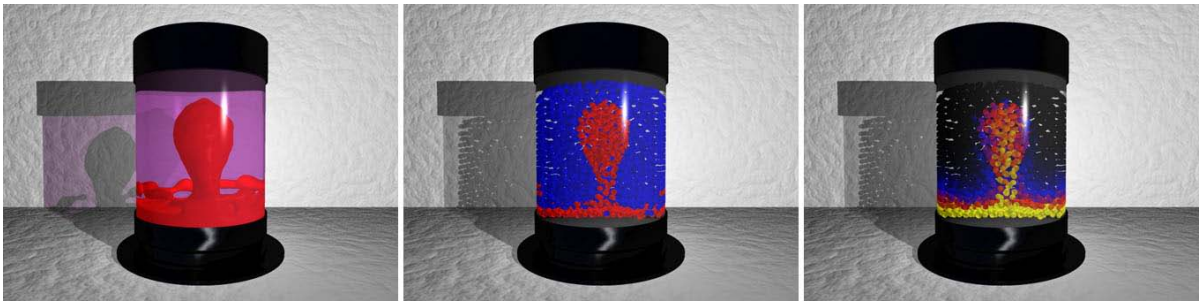


Figure 5: *The final rendering of a lava lamp is shown on the left. The other two images depict a cut through the two interacting fluids. The middle image shows the color attribute and the right image the temperatures of the individual particles.*

4. Results

The results we present in this section were computed on a Pentium 4 processor at 3.2 GHz. To generate fluid surfaces and interfaces we extract the iso surfaces of the color fields using the Marching Cubes algorithm [LC87]. The resulting triangles were rendered with POV-Ray†.

In our examples, the particles were generated randomly in a pipe with radius 0.06 m (see Figure 4). We integrate the forces using the leap-frog scheme with a time step Δt of 6 ms. The particles have a support radius of 0.045 m. We set the viscosity constant μ to 50 Ns/m^2, the gas constant k to 20 Nm/kg, and both, the interface tension coefficient σ^i between water and air, and the surface tension coefficient σ^s, to 0.6 (see Section 3.2).

4.1. Diffusion Effects

The first example depicted in Fig. 4 shows the interaction of two fluids with different densities and colors. These attribute differences are smoothed out at the interface due to diffusion effects. The less dense substance correctly ends up on top of the denser one. We used an interface tension coefficient σ^i of 0.6 between the two fluids.

4.2. Lava Lamp

Fig. 1 and Fig. 5 show the simulation of a lava lamp inside which two fluids with opposite polarity interact. The fluids are represented by 4800 blue and 1200 red particles. The blue fluid has density 500 kg/m^3, particle mass 0.006 kg while the red fluid has density 1000 kg/m^3 and particle mass 0.012 kg. The artificial buoyancy introduced in Section 3.4.3 is turned off in this example because the particle sampling is high enough for the coarse fluid formations.

Both fluids start with a temperature of 10 degrees. Then we start heating the ground and cooling the ceiling of the lamp. The temperature T influences the density ρ_0 of the red fluid via $\rho_0 = 10,000/T$. For the temperature diffusion constant c we chose 0.0001 and set the interface tension coefficient σ^i between the red and blue fluid to 2. The simulation ran at 11 frames per second. The reconstruction of the surface via Marching Cubes took about 3 minutes per frame.

† www.povray.org

4.3. Pouring Water into a Glass

Fig. 6 illustrates the pouring of water into a glass. The consideration of trapped air increases the realism of the simulation substantially. We used up to 3000 water particles and 400 air particles. The frame rate varied between 18 and 40 frames per second. Rendering of the images took about 8 minutes per frame. Water particles have a density of 1000 kg/m^3 and a mass of 0.012 kg. Air particles have a density of 100 kg/m^3 and a mass of 0.0012 kg. We used a buoyancy factor b of 5 (see Section 3.4.3). For particle generation and deletion we chose the thresholds $t_p = 22$ and $t_s = 0.0001$, respectively (see Section 3.4.1 and 3.4.2).

4.4. Boiling Water

In Fig. 7 we show the interaction of three types of fluid. Via temperature diffusion ($c = 0.1$), the flame with a temperature of 1000 degrees Celsius heats cold water of at about 0 degrees. Bubbles always form first on solid surfaces in contact with the liquid at so-called cavitation sites. A mineral deposit, a bit of suspended dust, an irregularity, or a scratch on any solid surface can serve as a site for bubble formation. We randomly chose 5 cavitation sites at the bottom of the glass. Water particles close to a cavitation site which reach a temperature of 100 degrees are transformed into air particles with a probability of 0.005 per time step and particle. Due to the smaller density of the air particles, they start to rise inside the water. In the sequence, we start with 5500 water particles and 3000 flame particles. We used the same parameters as in the previous example for water and air particles. The physics simulation runs at about 8 frames per second while rendering takes about 5 minutes per frame.

5. Conclusions and Future Work

We have presented a variety of techniques to enhance particle based fluid simulations by considering fluid-fluid interaction effects. These techniques can be added in a straight forward way to any existing particle simulator. We have found that - in contrast to Eulerian fluid simulations, particles are particularly well suited for modeling the interaction of different types of fluids and phase transitions. Particles can also be generated and deleted dynamically which allows to represent the surrounding and trapped air only where needed.

A problem we encountered was the limitation of the SPH-approach for single particles or badly sampled droplets. We proposed a technique to circumvent the problem but there might be different ways such as bilateral filtering [DD02] of the densities - an idea we want to investigate in the future.

References

[BT99] BUNNER B., TRYGGVASON G.: Direct numerical simulations of three-dimensional bubbly flows. *Physics of Fluids 11*, 8 (1999), 1967–1969.

[CMT04] CARLSON M., MUCHA P. J., TURK G.: Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Trans. Graph. 23*, 3 (2004), 377–384.

[DC96] DESBRUN M., CANI M.-P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In *6th Eurographics Workshop on Computer Animation and Simulation '96* (1996), pp. 61–76.

[DD02] DURAND F., DORSEY J.: Fast bilateral filtering for the display of high-dynamic-range images. In *Computer Graphics Proceedings* (Aug. 2002), Annual Conference Series, ACM SIGGRAPH 2002, pp. 257–266.

[EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 736–744.

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 23–30.

[FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. *Graphical Models and Image Processing 58*, 5 (1996), 471–483.

[GH04] GREENWOOD S. T., HOUSE D. H.: Better with bubbles: enhancing the visual realism of simulated fluid. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 287–296.

[HK03] HONG J.-M., KIM C.-H.: Animation of bubbles in liquid. *Eurographics 22*, 3 (2003), 253–262.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), ACM Press, pp. 163–169.

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. *Proceedings of 2003 ACM SIGGRAPH Symposium on Computer Animation* (2003), 154–159.

[Mon92] MONAGHAN J.: Smoothed particle hydrodynamics. *Annu. Rev. Astron. Physics 30* (1992), 543.

[MST*04] MÜLLER M., SCHIRM S., TESCHNER M., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids. *Journal of Computer Animation and Virtual Worlds (CAVW) 15*, 3-4 (2004), 159–171.

[PTB*03] PREMOZE S., TASDIZEN T., BIGLER J., LEFOHN A., WHITAKER R. T.: Particle-based simulation of fluids. *Eurographics 22*, 3 (2003), 401–410.

[SAC*99] STORA D., AGLIATI P., CANI M. P., NEYRET F., GASCUEL J.: Animating lava flows. In *Graphics Interface* (1999), pp. 203–210.

[Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 121–128.

[WGS00] WAGNER A., GIRAUD L., SCOTT C.: Simulation of a cusped bubble rising in a viscoelastic fluid with a new numerical method. *Comp. Phys. Comm. 129* (2000), 227.

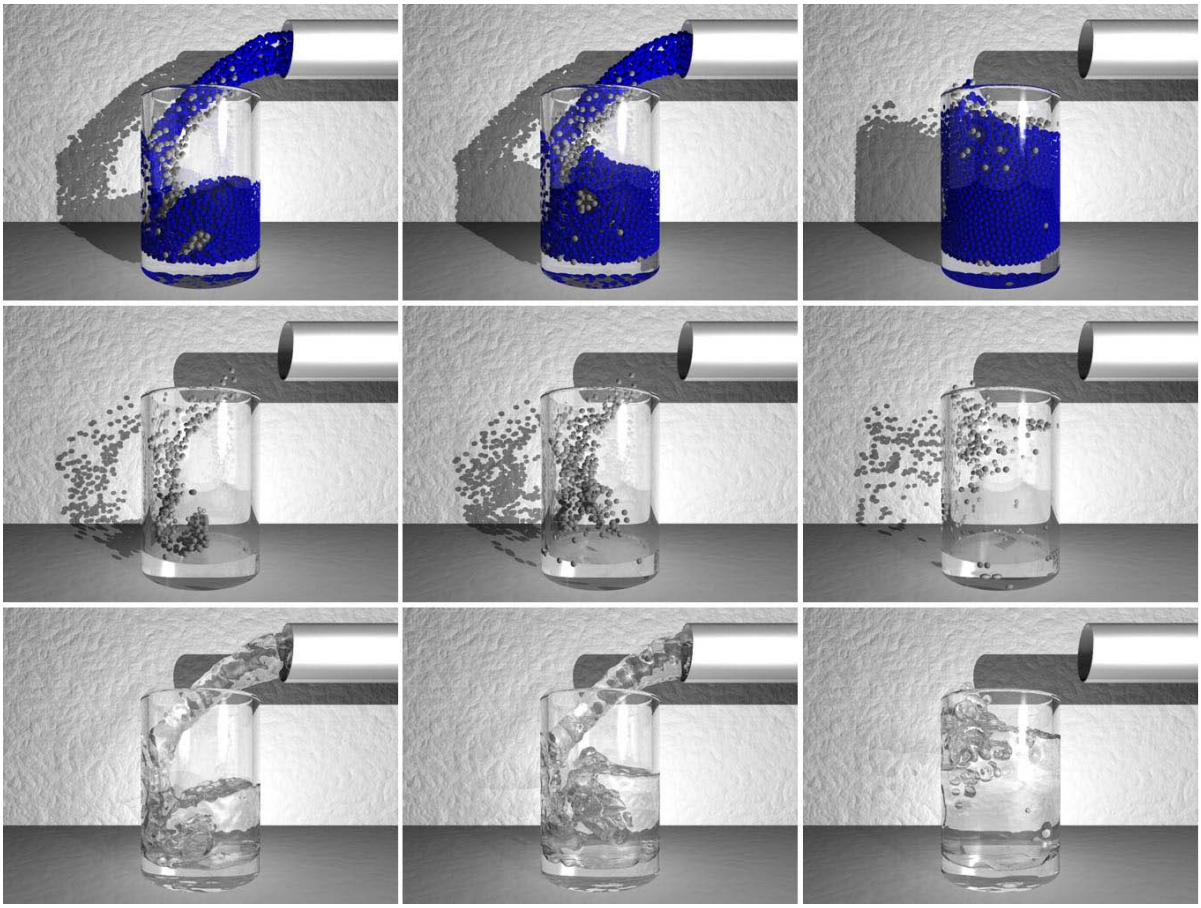


Figure 6: *Pouring water into a glass: The top row shows the water particles in blue. The gray air particles are generated where air is about to be trapped. The second row of images show only the air particles. They are responsible for the air bubbles in the final rendering shown in the bottom row.*

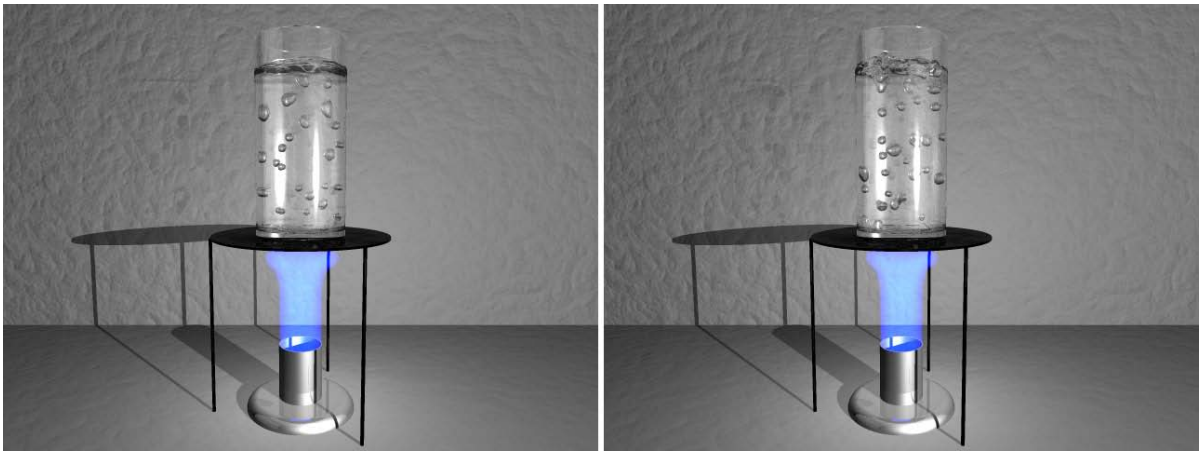
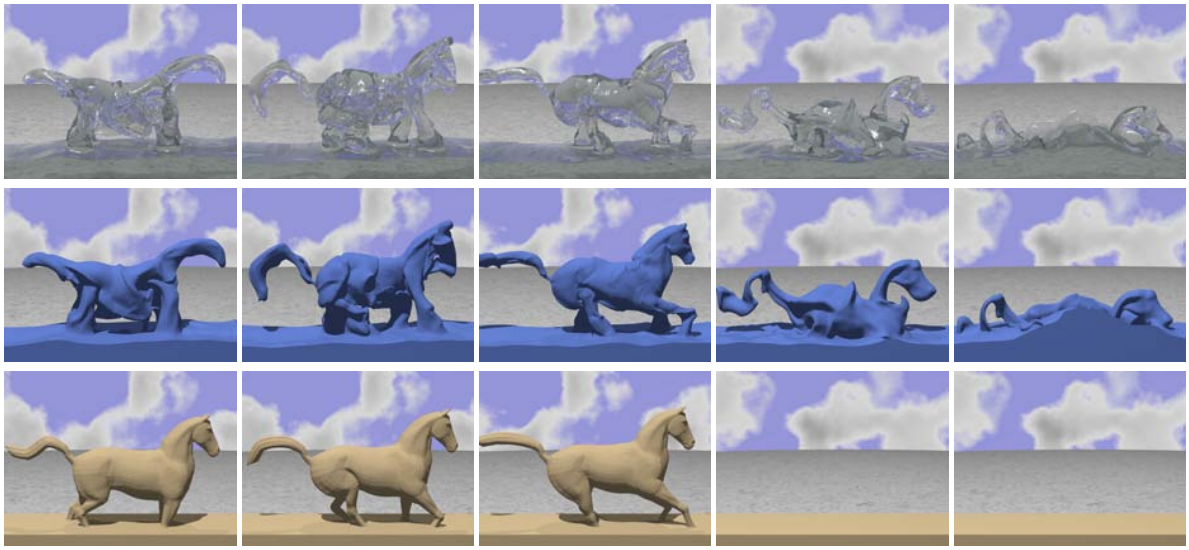


Figure 7: *A Bunsen burner modeled with a particle emitter heats a beaker filled with water. Heat transfer increases the temperature of the water at the bottom of the glass. Air bubbles form and rise to the surface.*



(a) (b) (c) (d) (e)

Figure 8: *First row: a liquid simulation representing a water horse emerging (a & b), running (c) and collapsing (d & e). Second row: the simulated liquid surfaces rendered as meshes. Third row: the underlying target animation. The resolution is 275x250x75, and the computational time is 4.4 minutes per frame.*

Müller et al. / Particle-Based Fluid-Fluid Interaction

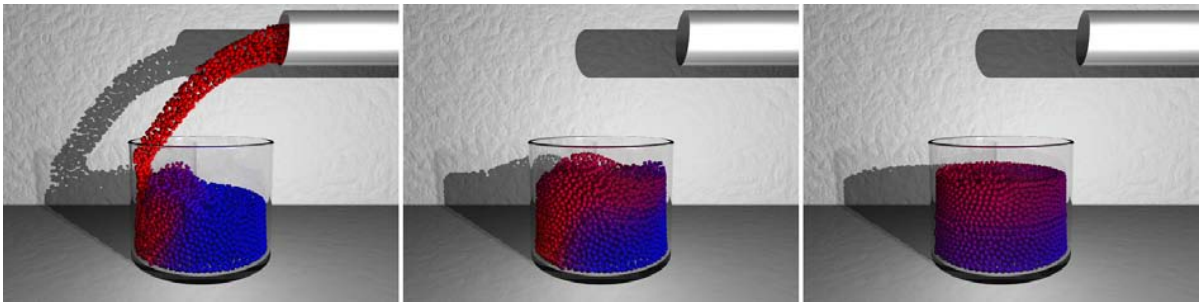


Figure 4: *Two types of fluid are mixed. The less dense fluid (red particles) rises to the surface while color differences are smoothed out at the interface due to diffusion effects.*

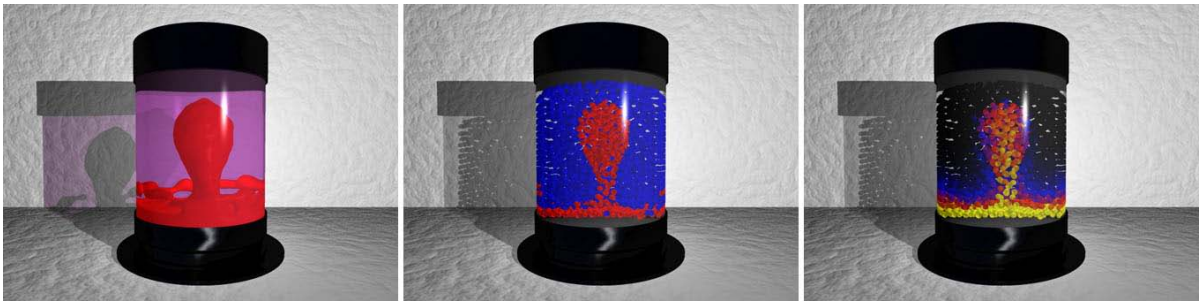


Figure 5: *The final rendering of a lava lamp is shown on the left. The other two images depict a cut through the two interacting fluids. The middle image shows the color attribute and the right image the temperatures of the individual particles.*