ETH CS Technical Report #520, 15 June, 2006

Multiresolution Particle-Based Fluids

Richard Keiser[1], Bart Adams[2], Leonidas J. Guibas[3], Philip Dutré[2], and Mark Pauly[1]

[1]ETH Zurich, Switzerland [2]KU Leuven, Belgium [3]Stanford University, CA, USA

Abstract

We present a new multiresolution particle method for fluid simulation. The discretization of the fluid dynamically adapts to the characteristics of the flow to resolve fine-scale visual detail, while reducing the overall complexity of the computations. We introduce the concept of virtual particles to implement efficient refinement and coarsification operators, and to achieve a consistent coupling between particles at different resolution levels, leading to speedups of up to a factor of six as compared to single resolution simulations. Our system supports multiphase effects such as bubbles and foam, as well as rigid body interactions, based on a unified particle interaction metaphor. The water-air interface is tracked with a Lagrangian level set approach using a novel Delaunay-based surface contouring method that accurately resolves fine-scale surface detail while guaranteeing preservation of fluid volume.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling: Physically based modeling. I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism: Animation.

1. Introduction

Fluid flow is an inherently multi-scale problem. Small-scale turbulence, thin sheets of liquid, droplets, bubbles, or spray greatly enhance the perceptual realism of computer-generated animations. Yet the complexity of such phenomena is too overwhelming for direct manual control. Physics-based simulation provides the means to tame this complexity allowing the animator to focus on large-scale motion such as big eddies or laminar flow.

Capturing the multi-scale nature of fluid flow typically requires a high-resolution discretization of the computational domain. Since uniform methods, such as regular grids or constant-mass particle systems, suffer from cubic complexity and thus high computation and memory demands, spatially adaptive methods have been proposed to improve scalability. When designing such an adaptive scheme, two main questions need to be addressed: How to compute and dynamically update the adaptive discretization of the domain, and how to define the discrete differential operators on this non-uniform discretization? Clearly, an adaptive method can only be successful, if the savings in memory and processing time are not surpassed by the overhead for maintaining the adaptive spatial data structures. Similarly, an appropriate discretization of the underlying fluid flow equations is crucial to obtain a consistent and efficiently computable solution.

Losasso et al. [LGF04] recently introduced such a scheme based on an unrestricted octree decomposition. A careful design of the divergence and gradient operators yields a symmetric discretization of the Poisson equation, which can be efficiently solved even for large simulation domains (see also [SY04]). Adaptive mesh refinement [BO84, SAB*99] is a different Eulerian strategy that uses a set of uniform grids at different resolution, see [LFO05] for a comparison.

Spatial adaptivity in the Eulerian setting can be problematic, however. Fluid moving through space requires constant refinement and coarsening of the underlying mesh, which leads to frequent memory updates and thus high computational cost. Consider a simple drop falling down as illustrated in Figure 1. Since high spatial resolution is desired close to the interface, the mesh has to be continuously refined and coarsened as the drop moves through the spatially fixed grid, even when nothing interesting is happening in terms of fluid dynamics. Another difficulty in grid-based methods is that solid boundaries have to align with the voxel structure of the grid, which can lead to aliasing artifacts for irregularly shaped domains. To address this issue, Feldman et al. [FOK05] introduced a method for simulating gases on hybrid meshes that conform to irregular domain boundaries. By mixing regular hexagonal meshes with unstructured tetrahdedral meshes, their method leads to improved accuracy near irregular boundaries. Similarly, Elcott et al. [ETK*05] presented a method for simulating flow on arbitrary tetrahedral meshes using circulation preserving lo-

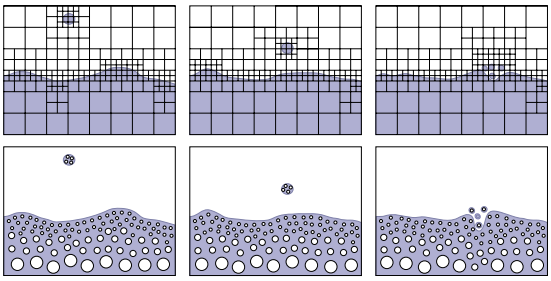


Figure 1: *Advection of spatial detail requires substantial re-organization of adaptive Eulerian grids (top row). In Lagrangian methods updates of the discretization are directly coupled to the dynamics of the flow (bottom row).*

cal operators to avoid numerical diffusion of vorticity. While ideally suited for static environments, these methods require complex dynamic re-meshing operations for moving object boundaries.

We propose a new multiresolution Lagrangian method that avoids grid-related aliasing artifacts as well as the complexity of maintaining conformity to dynamic object boundaries. At the same time, the spatial discretization is directly coupled to the dynamics of the flow. New particles are created near the interface to resolve visually important fine-scale surface detail. Thus simple advection of spatial detail requires no updates in the discretization (Figure 1, bottom), avoiding the numerous interpolation and averaging operations that adversely affect numerical accuracy.

A crucial ingredient in our approach is a new method for coupling the interaction between particles of different size. As noted in [Kou05] variable-sized particles lead to inconsistencies in the standard mollified kernel approximation of the differential operators. We avoid this problem by allowing only a discrete set of particle sizes, analogous to an octree discretization in the Eulerian setting. The coupling between particles at different levels is achieved using a new type of *virtual* particles that guarantee the consistency of the approximation. These virtual particles at the same time provide a natural way to dynamically refine and coarsen the discretization during the simulation. Our multiresolution scheme is based on Smoothed Particle Hydrodynamics [Mon05], but can be applied to other particle-based methods as well. We model two-phase coupling to simulate air-water interaction and extend this scheme to also implement solid-fluid and solid-solid interactions similar to [BYM05].

As a second contribution, we propose a new interface tracking method based on a 3D Delaunay triangulation of air and water particles close to the interface. We use the bichromatic edges of the resulting tetrahedra to approximate the zero-set of a potential function defined by the particles. This contouring method adapts automatically to the local discretization without requiring a background grid. We thus avoid discretization artifacts due to the fixed grid.

1.1. Previous Work

Fluid simulation for computer animation has become popular due to a series of papers by Foster and Metaxas [FM96, FM97a, FM97b], who solved the Navier-Stokes equations using finite differences on a Eulerian grid. The seminal paper of Stam [Sta99] improved on this method by introducing an unconditionally stable semi-Lagrangian technique and implicit solvers. Fedkiw et al. [FSJ01] added a vorticity confinement term to reduce dissipation, a method that was recently extended with passively advected vorticity particles [SRF05]. Kim et al. [KLLR05] presented a semi-Lagrangian technique with back and forth error compensation and correction. Hong and Kim [HK03, HK05b] simulated effects such as bubbles and surface tension by modeling the discontinuity at the interface between air and water. Two-way coupling between rigid bodies and fluid was achieved in [CMT04] by projecting the rigid body on the Eulerian grid. Guendelman et al. [GSLF05] presented a technique for coupling water and smoke with shells. Losasso et al. [LIGF06] introduced a method for animating melting and burning of solids.

Most state-of-the-art Eulerian fluid simulation algorithms use level set methods [OS88, Set99, OF03] to track the free surface. The semi-Lagrangian particle level set method [Str99, EFFM02] was introduced to reduce volume loss. Adaptivity in the Eulerian setting is achieved by using an octree data structure as proposed by Losasso et al. [LGF04] and Shi and Yu [SY04]. An alternative adaptive contouring method based on octrees and backwards advection to obtain distance values was recently proposed by Bargteil et al. [BGOS06].

As an alternative to Eulerian schemes, particle-based methods for fluid simulation have recently become popular. Premoze et al. [PTB*03] used the Moving-Particle Semi-Implicit (MPS) method for solving the Navier-Stokes equations. An interactive system for water simulation using Smoothed Particle Hydrodynamics (SPH) has been presented by Müller et al. [MCG03, MSHG04, MSRG05]. A variation of SPH in combination with springs between particles has been applied for simulating viscoelastic fluids [CBP05]. Desbrun and Cani [DC96] used SPH for animating elastically and plastically deformable solids. For computational efficiency, they adapted the particle system over space and time [DC99] by enabling particles to split and merge while adapting the integration time steps automatically. However, because there is no transition between refined or coarsened particles, splitting and merging introduces a discontinuity in the force computation, which is a potential source of instabilities. To speed-up particle systems, O'Brien et al. [OFL01] propose a subdivision scheme for clustering particles and compute the dynamics on these clusters.

The free surface is most often defined as the isolevel of an implicit function defined by a color field [Mor00]. Hieber

and Koumoutsakos [HK05a] denote this as Lagrangian particle level sets and show how both color fields and distance fields can be used to define the level set function. Müller et al. [MCG03] propose to convert SPH particles near the isolevel to surface particles for interactive visualization using surface splatting.

Recent work on SPH in computational fluid dynamics focuses on improving the accuracy by remeshing the particles onto a grid [CPK02] or regularizing the distribution of the particles [BOT01, BOT05]. Colagrossi and Landrini [CL03] showed how interfacial flows such as splashing and sloshing can be accurately simulated using SPH.

2. Fluid Model

Our fluid model is derived from the Navier-Stokes equations using Smoothed Particle Hydrodynamics (SPH). Below we provide a brief summary and derive the relevant equations for the following sections. For a detailed discussion we refer to the excellent SPH reviews by Monaghan [Mon92, Mon05].

2.1. Smoothed Particle Hydrodynamics

The motion of fluid particles is based on the Lagrangian formulation of the Navier-Stokes equations

$$\frac{D\rho}{Dt} = -\rho\nabla\mathbf{v}, \tag{1}$$

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla P + \mu\nabla^2\mathbf{v} + \rho\mathbf{g}, \tag{2}$$

where ($\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla$) denotes the material derivative, ρ the density, $\mathbf{v}$ the velocity, P the pressure, $\mathbf{g}$ gravity acceleration, and μ viscosity. The continuous field properties $A(\mathbf{x})$ are approximated using SPH from the discrete function values A_j at particles p_j

$$A(\mathbf{x}) = \sum_j A_j V_j W(\mathbf{r}, h), \tag{3}$$

where V_j is the volume of the discrete particle elements, $\mathbf{r} = \mathbf{x} - \mathbf{x}_j$ is the distance vector from the particle position $\mathbf{x}_j$ to $\mathbf{x}$, and $W(\mathbf{r}, h)$ is typically a smooth, even, normalized kernel function with finite support h that indicates the smoothing scale length of the kernel. The gradient and Laplacian of a field property $A(\mathbf{x})$ are, respectively

$$\nabla A(\mathbf{x}) = \sum_j A_j V_j \nabla W(\mathbf{r}, h) \quad \text{and} \tag{4}$$

$$\nabla^2 A(\mathbf{x}) = \sum_j A_j V_j \nabla^2 W(\mathbf{r}, h). \tag{5}$$

Applying Equation (3) to the density of a particle p_i yields the so-called summation density

$$\rho_i = \sum_j m_j W(\mathbf{r}_{ij}, h), \tag{6}$$

where $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ is the distance vector of a particle pair. Initially, the mass m_i is chosen such that the average density

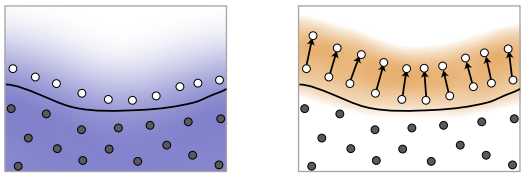


Figure 2: *Automatic air generation. Left: color field computed by assigning the value +1 to water (gray) and air (white) particles. Right: color coding of magnitude of color field gradient (scaled and clamped so that white corresponds to the threshold c). The air particles near the interface generate new particles in gradient direction.*

corresponds to the physical density ρ^0 of the fluid. The volume V_i is computed as $V_i = m_i/\rho_i$. By applying symmetrized versions of Equations (4) and (5) to the momentum equation (2), we get the following forces acting on a particle p_i

$$\mathbf{f}_i^{\text{pressure}} = -V_i \sum_j V_j \frac{P_i + P_j}{2} \nabla W(\mathbf{r}_{ij}, h), \tag{7}$$

$$\mathbf{f}_i^{\text{viscosity}} = \mu V_i \sum_j V_j (\mathbf{v}_j - \mathbf{v}_i) \nabla^2 W(\mathbf{r}_{ij}, h). \tag{8}$$

We use the kernels $W(\mathbf{r}, h)$ described in [MCG03]. The pressure P_i is computed via the constitutive equation

$$P_i = k\left(\frac{\rho_i}{\rho^0} - 1\right), \tag{9}$$

where k is the stiffness of the fluid.

2.2. Multiphase SPH

Air influences the behavior of water both at the free surface and as air pockets enclosed by water. By simulating both fluids at the same time, two-way coupling between water and air can be achieved. Similar to [MSRG05], we exploit the adaptivity of particle-based methods by generating air particles dynamically during the simulation. However, while Müller et al. generated air particles only where air pockets are likely to appear, we maintain a narrow band of air particles everywhere around the interface to accurately model two-way coupling at the free surface. Furthermore, we extended this coupling also to rigid bodies, such that fluid-fluid and fluid-rigid body interaction are simulated using a unified particle-based approach.

2.2.1. Air Generation

For two-way coupling between water and air as well as for an accurate reconstruction of the free surface (see Section 4), we require that a narrow band within a distance $d^{\min}$ from the interface is adequately sampled with air particles. Air particles further away than a distance $d^{\max}$ are deleted dynamically. We define a color field similar to [MSRG05] by assigning the constant value +1 to all particles p_i. Using Equation (4), we compute the gradient of this color field for all air particles p_i with a distance smaller than $d^{\min}$ to the interface as $\nabla c_i = -\sum_j V_j \nabla W(\mathbf{r}_{ij}, h)$. At the boundary of the

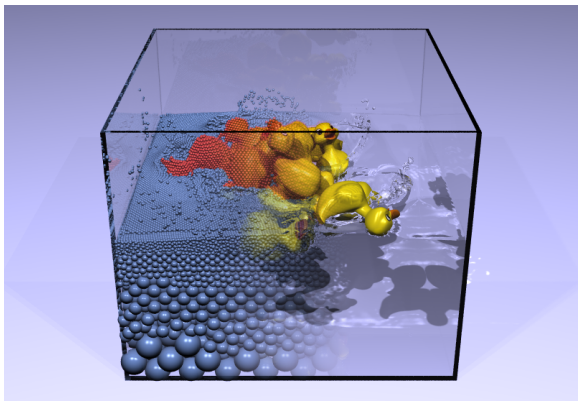


Figure 3: *Multiresolution fluid simulation and rigid bodies-fluid interaction using a unified particle-based approach.*

narrow band ∇c_i is orthogonal to this boundary. A new air particle is inserted if $\|\nabla c_i\|$ is larger than a threshold c and if the vector ∇c_i points in opposite direction to the interface, see Figure 2. The position $\mathbf{x}_{\text{new}}$ of the new air particle is computed by displacing it from the position $\mathbf{x}_i$ of p_i in direction of the color field gradient: $\mathbf{x}_{\text{new}} = \mathbf{x}_i + s\nabla c_i/\|\nabla c_i\|$, where s is the average particle spacing. In our case $s = h/2$, where h is the smoothing length of particles at the interface. For all our simulations we use $d^{\min} = h$, $d^{\max} = 2h$, and $c = 0.8s$.

2.2.2. Water-Air Interaction

Between two immiscible fluids, surface tension due to molecular cohesion gives rise to a sharp pressure jump at the interface, which prevents fluids from mixing freely [HK05b]. By computing the density of the fluids using Equation (6) independently of each other, we get the desired discontinuity at the interface. To model the pressure jump at the interface, we add a cohesion term $-k^{\text{cohesion}}\rho_i^2$ to the pressure equation (9) similar to [NP00] and [CL03] and obtain

$$P_i' = k\left(\frac{\rho_i}{\rho^0} - 1\right) - k^{\text{cohesion}}\rho_i^2. \qquad (10)$$

The sharpness of the interface is controlled with the constant $k^{cohesion}$. The pressure force is then computed using Equation (7). When we apply this equation only for the cohesion term we get the following force

$$\mathbf{f}_i^{\text{cohesion}} = k^{\text{cohesion}} V_i \sum_j V_j \frac{\rho_i^2 + \rho_j^2}{2} \nabla W(\mathbf{r}_{ij}, h). \qquad (11)$$

This force points in normal direction to the interface, therefore giving rise to surface tension.

2.2.3. Rigid Bodies-Fluid Interaction

We exploit the idea of [CMT04] and treat rigid bodies as a rigid fluid. A rigid body is sampled with particles the same way as a fluid (see Figure 3), where the density and mass of the particles depend on the rigid body material properties. During the simulation, the forces acting between rigid body and fluid particles are computed exactly as for water-air interaction described above. The net force and torque acting on the rigid body are computed from the sum of its rigid body particle forces. Integration in time then yields the linear and angular velocity of the rigid body, from which we compute the new particle positions. To simulate rigid body collisions we use the approach of Bell et al. [BYM05]. They propose to sample an offset surface of a rigid body with particles and derive the contact forces between overlapping particles from molecular dynamics. This showed to be efficient and yields very good results even for the case of rigid body stacking. Instead of sampling the surface with another set of particles, we use the rigid body volume particles for computing the contact forces.

3. Multiresolution Particle System

The multiphase SPH fluid model defined above allows complex simulations of water and its interaction with the surrounding air. The visual quality and numerical accuracy of these simulations directly depend on the number of particles, which, when sampling uniformly, increases cubically with respect to the inverse of the smallest resolved scale. Since the computational complexity is superlinear in the number of particles, high-resolution simulations quickly become intractable. Spatially adaptive sampling, that concentrates more particles in regions close to the interface, can thus lead to significant savings in memory and computation time. Such a sampling scheme should also be temporally adaptive, i.e., dynamically adjust the discretization, if the flow is non-stationary and detail is generated or destroyed during the simulation.

In the fluid model introduced above we assume that two interacting particles have the same smoothing length to guarantee bi-directional consistency in the particle force computation (Section 2.1). If this condition is violated, Newton's third law is no longer satisfied: Particle p_i can exert a force on particle p_j even though p_j has no influence on p_i. Thus any spatially adaptive scheme requires an adaptation of the way particles interact to ensure conservation of momentum. A straightforward solution is to apply some averaging of the smoothing lengths or the interpolation kernels [LL03]. However, as shown in [BOT01], this yields errors in the order of 10% already for a factor two difference in the kernel width and leads to severe instabilities with larger kernel variations. A different approach has been coined Regularized SPH (RSPH) by Borve et al. [BOT01, BOT05]. Since SPH is derived from Monte-Carlo interpolation with the particles as interpolation points, they propose the use of additional interpolation points, either defined on a background lattice or using auxiliary particles. While substantial improvements in numerical accuracy were reported for stationary flows, dynamically changing flows are difficult to model using this method.

We propose a new, general approach for temporally and spa-

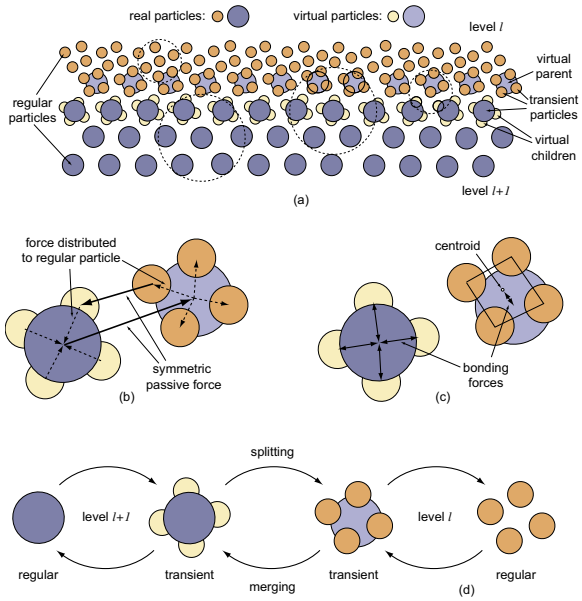


Figure 4: *Multiresolution coupling with virtual particles. (a) two levels of particles with neighborhoods (dashed circles), (b) distribution of SPH forces, (c) bonding forces for virtual particles, (d) splitting and merging.*

tially adaptive particle simulations that provides a consistent approximation, while minimizing computational overhead. Similar to [BOT01], we only allow kernel widths of size $2^l h$, where $l = 0, 1, \ldots$ denotes the particle resolution level and h is the smoothing length at the highest resolution. To dynamically adapt the discretization during the simulation, particles can transition from one level to another by splitting or merging (Section 3.2). The key idea of our approach is the concept of *virtual* particles that serve two main purposes: They allow a consistent coupling of particles at different resolution levels and they provide a mechanism for modeling particle splitting and merging without introducing temporal discontinuities.

3.1. Virtual Particles

As will be described below, a splitting criterion ensures that neighboring particles at most differ by one level, similar to a balanced octree decomposition in the Eulerian setting. To simplify the exposition we thus only consider two resolution levels l and $l+1$, as shown in Figure 4 (a). We distinguish three types of *real* (i.e. non-virtual) particles: Particles of level $l+1$ close to level l particles carry virtual level l children particles (4 in 2D, 8 in 3D). Particles of level l close to level $l+1$ particles are assigned to a virtual level $l+1$ parent particle. These two types are called *transient* particles. Other real particles of either level are *regular* particles that are not associated with any virtual particles.

SPH forces are computed only for real particles. When evaluating Equations (7) and (8), only particles of the same level, be they virtual, transient, or regular, are considered as neighbors. This means that if a particle finds a transient particle in its neighborhood, it either selects this particle or the associated virtual particle(s) when computing the interaction forces, depending on their level. If a force is computed between a real and a virtual particle, this force is symmetrically applied to the virtual particle to ensure preservation of momentum and re-distributed uniformly to the associated real particle(s) (see Figure 4 (b)). This coupling between real and virtual particles ensures the consistency of the force computations. In effect, transient particles behave like intermediates between two levels.

Virtual particles are passively advected with the flow using the SPH approximation of the velocity field. In order to keep virtual and associated real particles close together we introduce additional bonding forces that act only on virtual particles. As illustrated in Figure 4 (c), virtual child particles are attached to the associated transient parent particle using a linear spring force with a rest length equal to half the sample spacing of the corresponding level. Virtual parent particles are pulled towards the centroid of the associated transient child particles using a zero rest length spring.

3.2. Splitting and Merging

The transition of particles from one level to another is guided by two counter-acting principles: On the one hand, the particle resolution should be high enough to resolve visually important fine-scale surface detail. On the other hand, we want as few particles as possible to minimize computational overhead.

The adaptation to the characteristics of the flow is achieved by enforcing that particles at the air-water interface always have smoothing length h, i.e., are at the highest resolution. This ensures that we capture all surface detail up to scale h, since our surface model is directly coupled to the particle resolution (see Section 4). The hierarchy is determined by ensuring that only same-sized particles interact, i.e., guarantee that no real particle has regular particles of a different level within its neighborhood.

The transition from a regular level $l+1$ to a regular level l particle consists of three steps as shown in Figure 4 (d), from left to right: A regular level $l+1$ particle becomes a transient level $l+1$ particle, if it finds a real (regular or transient) level l particle in its neighborhood. Virtual children are created using a pre-computed uniform sampling on the sphere. They are assigned half the support radius of their parent particle and 2^{-d} of its mass and volume with d the number of dimensions. If a transient $l+1$ particle has a regular level l particle in its neighborhood, its virtual children become transient real particles and itself turns into a virtual parent for its children. Transient level l particles are released from their relation with their virtual parent particle and turn

into regular level l particles, if they have separated beyond their kernel smoothing length, i.e., if the maximum distance between two child particles exceeds $2^l h$. A split operation is also initiated, if all transient child particles of a common virtual parent have no virtual level l particles in their neighborhoods. Note that splitting always proceeds from coarse to fine, i.e., the splitting criterion is first evaluated for the coarse level particles. This ensures that smaller particles will not find higher level regular particles in their neighborhoods, as these would have been split already.

Merging operations proceed in the opposite direction (see Figure 4 (d), from right to left). In general, particles are merged whenever the new (type of) particle potentially created by the merging operation does not fulfill the corresponding splitting criterion. The transition from regular to transient child particles requires some special treatment. To force the total number of particles to be as small as possible, we allow even single regular particles to become transient child particles with a single virtual parent. If two such virtual parents are neighbors, we merge their children to a common parent if their total number does not exceed the allowed maximum (4 in 2D, 8 in 3D). This approach has the additional advantage that we can perform the merging incrementally and do not need a full set of eight children before a merging operation can be applied. Note that this approach does not lead to undesirable fragmentation, since merging is always performed at the transition layer between two resolutions (see also the accompanying video).

4. Lagrangian Particle Level Sets

A crucial component of fluid animation systems is the method for reconstructing the interface between water and air. We propose a new particle-based approach using an adaptive contouring algorithm of a Lagrangian level set function. Level set information is stored and advected with the simulated water and air particles and is used to construct a smooth triangle mesh representing the interface between the two fluids. Our new method has several advantages: The construction of the interface implicitly adapts to the resolution of the physical particles without requiring a background grid. The obtained surface is guaranteed to separate water from the air particles and avoids mass loss through numerical dissipation. No single visually important feature is missed, even the smallest drop or bubble consisting of a single particle is faithfully reconstructed. In addition, the method is exact in the case of rigid transformations.

4.1. Lagragian Level Set Representation

The idea of level set methods (see [OS88], [EFFM02] and references therein) is to define an interface Γ, which bounds an open region Ω as the zero level set of a function $\phi(\mathbf{x},t)$. This function is strictly positive for points $\mathbf{x} \in \Omega$ and negative for all other points. The interface is then defined as $\phi^{-1}(0)$. In our setting, we define the open region Ω as the water volume. The level set function is evolved over time using the velocity field according to the level set equation $\phi_t + \mathbf{v} \cdot \nabla\phi = 0$. In the Eulerian setting, the evolution of the level set function is computed on a grid or octree using finite differences to approximate the derivatives. In the Lagrangian setting, we obtain the simple expression $D\phi/Dt = 0$, i.e., the level set information is simply advected with the particles.

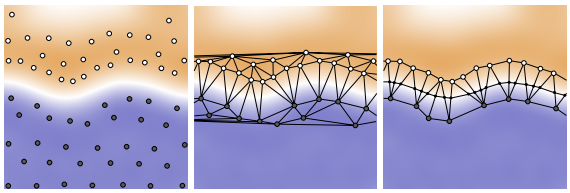


Figure 5: *Surface construction using Lagrangian particle level sets and Delaunay triangulations. Left: water particles (gray) and air particles (white) and corresponding color field. Middle: triangulation of narrow band particles. Right: surface construction from bichromatic triangles and color field.*

Typically, ϕ is chosen to be a signed distance function, such that $|\nabla\phi(\mathbf{x},t)| = 1$. After the advection of the interface, this property is often lost and needs to be reinforced using redistancing. However, in case this distance property is not required, significant computational savings can be achieved using an alternative description based on a color function [HK05a]. In our setting, the color function is defined by assigning the color value $\phi_j = +1$ to water particles and $\phi_j = -1$ to air particles. Using SPH, we easily obtain a smooth level set function (see also Figure 5) as

$$\phi(\mathbf{x}) = \sum_j \phi_j V_j W(\mathbf{r},h), \tag{12}$$

where we use the spline kernel described in [MCG03]. The smoothness of the surface and the size of connected components (e.g. thin sheets) increases with larger support radius h. For efficiency reasons we use the same support as for the particles, as this allows us to reuse the computed particle neighbors.

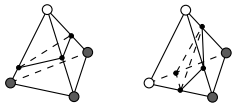
One advantage of the Lagrangian formulation is that solving the level set equation is trivial: Level set values are simply advected along with the physical particles, costly redistancing operations are completely avoided, and no additional memory is required to store the level set values.

4.2. Adaptive Triangulation of the Zero Level Set

Previous particle-based fluid methods compute an explicit representation of the interface using some variant of marching cubes, see, e.g., [MCG03], [MSRG05] and [CBP05]. Apart from the difficulty of choosing an appropriate grid resolution, these approaches have the drawback that the background grid is fixed to the ambient space, i.e., the discretization of the interface changes constantly even when fluid is

moving rigidly. Hieber and Koumoutsakos [HK05a] remesh the particles onto a grid. This allows them to use the fast marching method for redistancing, but yields similar grid-based artifacts.

To overcome this problem, we propose an alternative meshing strategy based on a 3D Delaunay triangulation of particles in a narrow band around the interface. We compute the triangle mesh of the zero level set using the following three-step procedure (see Figure 5): First, we collect a narrow band of particles p_j, which have a differently colored particle within their support radius. Since the color function can only change sign in the neighborhood of water and air particles, we can safely ignore all other particles. In a second step, we construct a Delaunay triangulation of these narrow band particles and extract the bichromatic simplices (triangles in 2D, tetrahedrons in 3D), i.e., those which have vertices in both fluids. To separate water from air particles, the surface has to pass through the bichromatic simplices. Therefore, in a final step, we compute the interface surface by cutting the bichromatic simplices (see also [RCPM05]). For a tetrahedron with three bichromatic edges one triangle is created, while two triangles are required to cut four bichromatic edges. The exact position of the triangle vertices is obtained by finding the zero value of the color function along the corresponding bichromatic edge using a simple bisection method. In case the color function does not change sign on the bichromatic edge (which is very unlikely, but can happen, e.g., for small bubbles), we position the vertex at a fixed distance from the particle with the smallest absolute color function value. This guarantees that we recover all small-scale features which would otherwise be smoothed out by the level set function.



5. Results

We implemented the multiphase fluid model of Section 2.2 based on the presented multiresolution particle system, and integrated our new adaptive contouring method to extract the water-air interface.

Figure 7 shows the break-up of a water crown and illustrates the capabilities of our multiresolution approach for resolving fine-scale detail. Preserving mass at small scales is crucial in this example as it avoids visually disturbing loss of fluid, e.g., for the drops splashing against the walls. Also note the tiny bubbles at the front of the sphere created by air particles being dragged under the water surface. Figure 8 shows multiple rigid bodies, fluid, and air interacting with each other. This example demonstrates the effectiveness of our particle-based coupling method that requires only minimal adaptations to incorporate rigid or even deformable solids, although we have not implemented the latter yet. Mixing of two viscous fluids is shown in Figure 9. Since the simulation nodes are not fixed to the ambient space, fluid can be distributed arbitrarily around the scene without requiring constant updates to an underlying simulation grid. Note that in the final frame all 60k water particles are at the highest resolution level, hence multiresolution does not yield any performance gains for this example. Figure 10 shows a highly turbulent flow created by a swirling water jet. Multiphase effects greatly enhance the realism of this animation. This example also highlights the robustness of our surface reconstruction algorithm for complex water-air interactions.

5.1. Discussion

The crucial argument for multiresolution is scalability. While single resolution simulations require $O(n^3)$ with $n = 1/h$ computation nodes, multiresolution reduces the complexity to $O(n^2)$. As can be seen from the performance data of Table 1, our approach leads to substantial savings in both memory and computation time. Clearly, these efficiency gains depend on the characteristics of the flow. Speedups of a factor of six are obtained for the scenes of Figures 7 and 8, where large regions of fluid can be sampled at lower resolutions without loss of accuracy. The highly turbulent flows of Figures 9 and 10 show smaller performance gains as most particles are on the highest resolution level. Note, however, that the computations for splitting and merging are very efficient, i.e., even if all particles are at the highest resolution, performance is comparable to a pure single resolution simulation. In addition, the overhead for virtual particles is minimal, since they are not considered in the SPH force computation and hence require no costly spatial queries.

Our new interface tracking approach can be understood as a hybrid between explicit and implicit surface reconstruction. While the level set function is implicit and the space decomposition of the 3D Delaunay triangulation guarantees a topologically consistent manifold surface, the extracted mesh is essentially advected with the flow (see Figure 6). As long as the topology of the Delaunay triangulation remains stationary, e.g., when the fluid is moving rigidly through space, the connectivity of the surface mesh is maintained. Thus our approach leads to a *persistent* and temporally coherent surface discretization, in contrast to marching cubes, where the interface is reconstructed from scratch in each frame. For instance, the discretization of the surface of a small drop does not change while it moves through space (cf. Figure 1). In addition, our method does not suffer from volume loss caused by numerical dissipation, since particles preserve mass and the interface is guaranteed to pass between water and air particles. Note also that the entire surface reconstruction computations including building the Delaunay triangulation (we use CGAL [SVY]) and extracting the surface mesh typically requires only between $10-20\%$ of the total simulation time.

An inherent limitation of particle-based approaches is the computational overhead caused by the particle neighborhood computations. As shown in Table 1, spatial queries amount

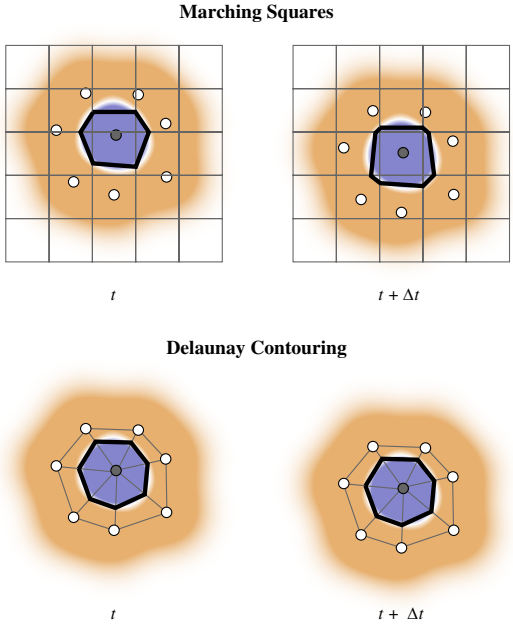


Figure 6: *Comparison of surface extraction using marching cubes (marching squares in 2D) and our novel surface extraction algorithm. One water particle (gray) and seven air particles (white) are shown at time step t and $t+\Delta t$. Top row: the iso-level (white region between blue and brown regions) is extracted by interpolating level set values on the fixed grid. The discretization can change significantly as the particles move through the grid. Bottom row: the surface is defined as the connection of zero crossings on the bichromatic edges of the Delaunay triangles. This leads to a temporally persistent surface extraction.*

to up to 80% of the total simulation time. All neighborhood computations are performed using static kd-trees [AM93], which are re-initialized in each time step. Rebuilding a kd-tree is fast (in the range of 0.1-0.5 seconds in our examples), and in our examples kd-trees showed to yield better performance than hash grids [THM*03]. Furthermore, kd-trees are more suitable than hash grids in a multiresolution setting where choosing an appropriate cell size is difficult. However, we believe that substantial performance gains can be achieved using more efficient caching schemes that exploit temporal coherence. Nevertheless, the implicit neighborhood relations of regular grids are clearly a performance advantage of Eulerian methods. Another strength of grid-based approaches is incompressibility. While methods for SPH have been proposed to make the velocity field (approximately) divergence free [CR99], the computational overhead is substantial and typically not justified for most applications in computer animation. However, the choice of the stiffness constant k (we use $k = 400'000$) in Equation 9 is non-trivial as it depends on the time step (1ms in our simulations). With larger k, the fluid is less compressible but the time step needs to be reduced to guarantee a stable simulation. Another limitation is that due to the smoothing inherent in the SPH method, fine scale turbulences are often damped out. However, both our multiresolution and the surface reconstruction method are not bound to SPH, but can be adapted to most particle systems.

Our method can be extended in various ways. A view-dependent splitting criterion would allow to dynamically reduce the resolution in regions that are not visible from a certain viewpoint and could potentially lead to additional performance gains. Another possible application of our multiresolution framework is for hybrid molecular-continuum simulations (see e.g. [Kou05] for an overview). Our interface tracking method allows the advection of surface parameter information, which could be used to advect surface textures. Given a bichromatic edge, we can compute the intersection of this edge with the surface of the previous time step and transfer the parameter coordinates to the current time step, similar to the approach of [BGOS06]. This can be done efficiently by exploiting that most bichromatic edges in one time step also exist in the next and therefore only a small amount of edges needs to be tested for intersection.

6. Conclusion

In this paper we show how spatially and temporally adaptive sampling of the fluid volume leads to substantial savings in memory and computation time thus enabling significantly more complex flow simulations with the same processing resources. The concept of virtual particles achieves a consistent coupling of particles at different levels and provides an efficient mechanism for dynamic re-sampling. It is lightweight, simple to implement, and can easily be incorporated into other particle-based simulation environments. Similarly, our novel Delaunay-based interface tracking method enables advection of the surface together with the particles and efficiently generates high-quality surfaces that automatically adapt to the resolution of the simulation.

References

[AM93] ARYA S., MOUNT D. M.: Algorithms for fast vector quantization. In *Data Compression Conference* (1993), pp. 381–390.

[BGOS06] BARGTEIL A. W., GOKTEKIN T. G., O'BRIEN J. F., STRAIN J. A.: A semi-lagrangian contouring method for fluid simulation. *ACM Trans. Graph. 25*, 1 (2006).

[BO84] BERGER M., OLIGER J.: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys. 53* (1984).

[BOT01] BORVE S., OMANG M., TRULSEN J.: Regularized smoothed particle hydrodynamics: A new approach to simulating magnetohydrodynamic shocks. *The Astrophysical Journal 561*, 1 (2001), 82–93.

[BOT05] BORVE S., OMANG M., TRULSEN J.: Regularized smoothed particle hydrodynamics with improved multi-resolution handling. *J. Comput. Phys. 208*, 1 (2005), 345–367.

[BYM05] BELL N., YU Y., MUCHA P. J.: Particle-based simulation of granular materials. In *SCA '05: Proc. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 77–86.

[CBP05] CLAVET S., BEAUDOIN P., POULIN P.: Particle-based viscoelastic fluid simulation. In *Symposium on Computer Animation 2005* (2005), pp. 219–228.

[CL03] COLAGROSSI A., LANDRINI M.: Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *J. Comput. Phys. 191*, 2 (2003), 448–475.

	#real particles	#virtual particles	#air particles	#rig. body particles	spatial queries	time integration	air handling	splitting, merging	surface reconstruction	total time per frame
Sphere	170k	110k	85k	3k	64	6	9	1.1	10	91
	641k	0k			472	21		0		513
Ducks	166k	107k	73k	34k	54	9	9	1.4	8	81
	641k	0k			460	25		0		504
Teapot	98k	29k	87k	0k	56	4	12	1.6	14	87
	153k	0k			68	5		0		99
Armadillo	43k	3k	235k	0k	62	9	29	0.4	22	122
	43k	0k			62	9		0		122

Table 1: *Statistics for the animations shown in the paper. The upper row shows the averaged data per frame for a multiresolution simulation, the equivalent single resolution results are shown in the lower row. All timings are in seconds, measured on a 2.8Ghz Pentium IV with 2GB memory.*

[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.

[CPK02] CHANIOTIS A. K., POULIKAKOS D., KOUMOUTSAKOS P.: Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows. *J. Comput. Phys. 182*, 1 (2002), 67–90.

[CR99] CUMMINS S. J., RUDMAN M.: An sph projection method. *Journal of Computational Physics 152* (1999).

[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.

[DC99] DESBRUN M., CANI M.-P.: *Space-Time Adaptive Simulation of Highly Deformable Substances*. Tech. rep., INRIA Nr. 3829, 1999.

[EFFM02] ENRIGHT D., FEDKIW R., FERZIGER J., MITCHELL I.: A hybrid particle level set method for improved interface capturing. *J. Comput. Phys. 183*, 1 (2002), 83–116.

[ETK*05] ELCOTT S., TONG Y., KANSO E., SCHRÖDER P., DESBRUN M.: *Discrete, circulation-preserving, and stable simplicial fluids*. Tech. rep., Caltech, 2005.

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

[FM97a] FOSTER N., METAXAS D.: Controlling fluid animation. In *Proc. CGI '97* (1997), pp. 178–188.

[FM97b] FOSTER N., METAXAS D.: Modeling the motion of a hot, turbulent gas. In *SIGGRAPH '97* (1997), pp. 181–188.

[FOK05] FELDMAN B. E., O'BRIEN J. F., KLINGNER B. M.: Animating gases with hybrid meshes. *ACM Trans. Graph. 24*, 3 (2005).

[FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *SIGGRAPH '01* (2001), pp. 15–22.

[GSLF05] GUENDELMAN E., SELLE A., LOSASSO F., FEDKIW R.: Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph. 24*, 3 (2005), 973–981.

[HK03] HONG J.-M., KIM C.-H.: Animation of bubbles in liquid. *Computer Graphics Forum 22*, 3 (2003).

[HK05a] HIEBER S. E., KOUMOUTSAKOS P.: A lagrangian particle level set method. *Journal of Computational Physics 210*, 1 (2005), 342–367.

[HK05b] HONG J.-M., KIM C.-H.: Discontinuous fluids. *ACM Trans. Graph. 24*, 3 (2005), 915–920.

[KLLR05] KIM B., LIU Y., LLAMAS I., ROSSIGNAC J.: Flowfixer: Using bfecc for fluid simulation. *Eurographics Workshop on Natural Phenomena* (2005), 51–56.

[Kou05] KOUMOUTSAKOS P.: Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech. 37* (2005).

[LFO05] LOSASSO F., FEDKIW R., OSHER S.: Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids* (2005).

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Trans. Graph. 23*, 3 (2004), 457–462.

[LIGF06] LOSASSO F., IRVING G., GUENDELMAN E., FEDKIW R.: Melting and burning solids into liquids and gases. *IEEE TVCG 12*, 3 (2006), 343–352.

[LL03] LIU G. R., LIU M. B.: *Smoothed Particle Hydrodynamics, A Meshfree Particle Method*. World Scientific Publishing, 2003.

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

[Mon92] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Annu. Rev. Astron. and Astrophysics 30* (1992), 543–574.

[Mon05] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Rep. Prog. Phys. 68* (2005), 1703–1758.

[Mor00] MORRIS J.: Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids 33*, 3 (2000), 333–353.

[MSHG04] MÜLLER M., SCHIRM S., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids. In *Computer Animation and Virtual Worlds (CAVW)* (2004), pp. 159–171.

[MSRG05] MÜLLER M., SOLENTHALER B., RICHARD, GROSS M.: Particle-based fluid-fluid interaction. In *SCA '05: Proc. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 237–244.

[NP00] NUGENT S., POSCH H. A.: Liquid drops and surface tension with smoothed particle applied mechanics. *Phys. Rev. E 62*, 4 (2000), 4968–4975.

[OF03] OSHER S., FEDKIW R.: *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag, New York, 2003.

[OFL01] O'BRIEN D., FISHER S., LIN M. C.: Automatic simplification of particle system dynamics. In *Computer Animation 2001* (2001).

[OS88] OSHER S., SETHIAN J. A.: Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys. 79*, 1 (1988), 12–49.

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

[RCPM05] RAY N., CAVIN X., PAUL J.-C., MAIGRET B.: Intersurf : dynamic interface between proteins. *Journal of Molecular Graphics and Modelling 23*, 4 (2005), 347–354.

[SAB*99] SUSSMAN M., ALMGREN A., BELL J., COLELLA P., HOWELL L., WELCOME M.: An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys. 148* (1999).

[Set99] SETHIAN J. A.: *Level Set Methods and Fast Marching Methods*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.

[SRF05] SELLE A., RASMUSSEN N., FEDKIW R.: A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. 24*, 3 (2005), 910–914.

[Sta99] STAM J.: Stable fluids. In *SIGGRAPH '99* (1999), pp. 121–128.

[Str99] STRAIN J.: Semi-lagrangian methods for level set equations. *J. Comput. Phys. 151*, 2 (1999), 498–533.

[SVY] SCHIRRA S., VELTKAMP R., YVINEC M.: The cgal reference manual. Release 2.0 (www.cgal.org).

[SY04] SHI L., YU Y.: Visual smoke simulation with adaptive octree refinement. In *Computer Graphics and Imaging* (2004).

[THM*03] TESCHNER M., HEIDELBERGER B., MÜLLER M., POMERANERTS D., GROSS M.: Optimized spatial hashing for collision detection of deformable objects. In *Proc. Vision, Modeling, Visualization VMV* (2003), pp. 47–54.

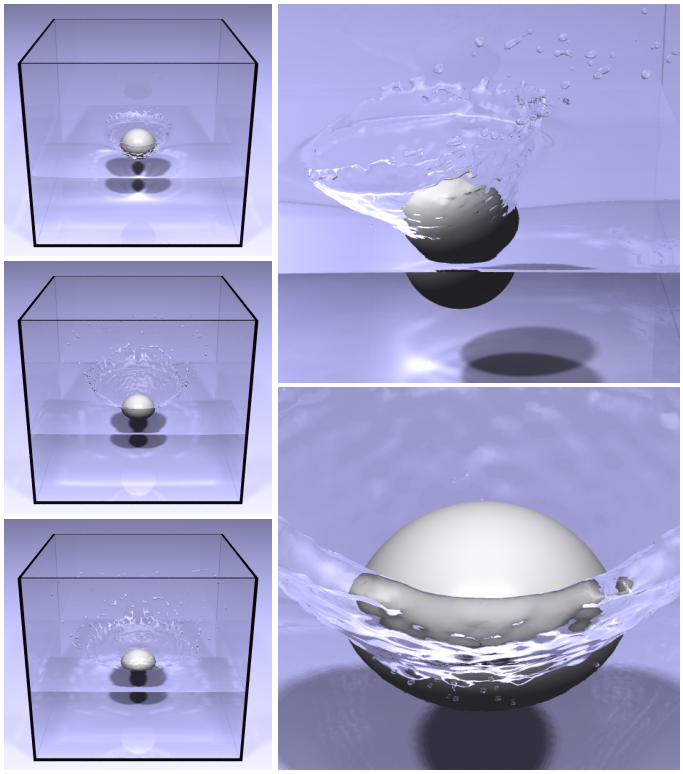


Figure 7: *Multiphase fluid simulation of a splashing sphere. Note the small splashes and bubbles.*

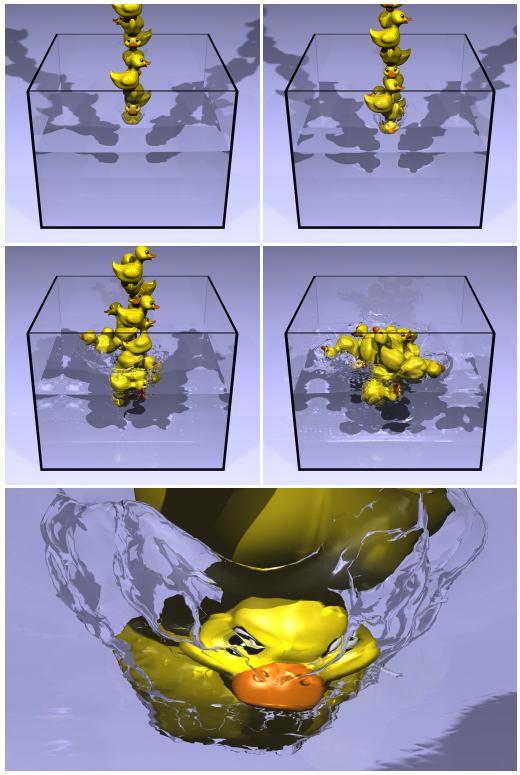


Figure 8: *Particle-based rigid body collisions and interaction with fluid.*

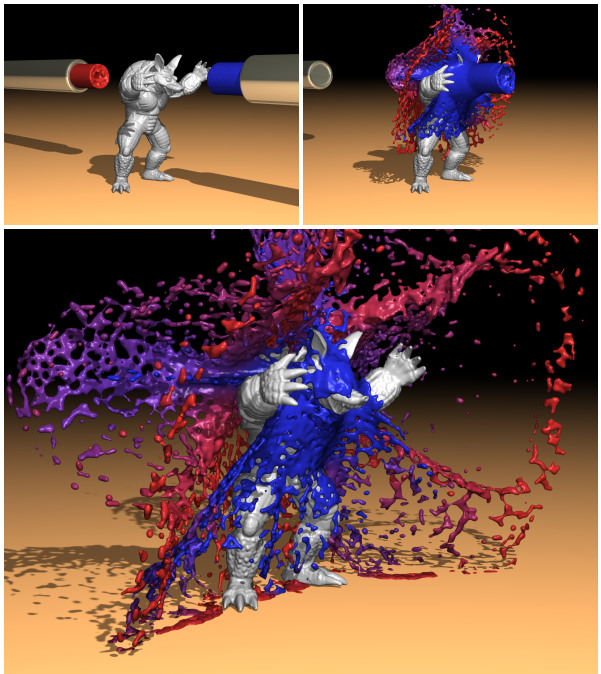


Figure 9: *Mixing of two viscous fluids.*

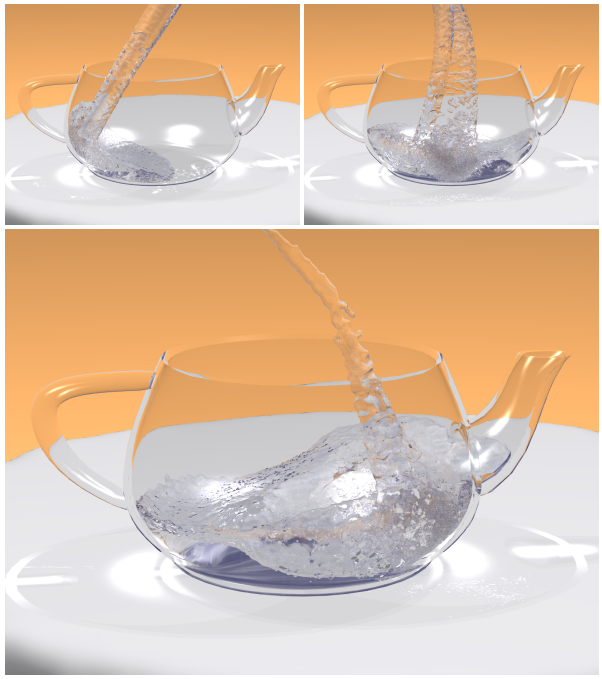


Figure 10: *Complex multiphase effects on a turbulent flow.*