

Unified Spray, Foam and Bubbles for Particle-Based Fluids

Markus Ihmsen · Nadir Akinici · Gizem Akinici · Matthias Teschner

Abstract We present a new model for diffuse material, i. e. water-air mixtures, that can be combined with particle-based fluids. Diffuse material is uniformly represented with particles which are classified into spray, foam and air bubbles. Physically-motivated rules are employed to generate, advect and dissipate diffuse material. The approach is realized as a post-processing step which enables efficient processing and versatile handling. As interparticle forces and the influence of diffuse material onto the fluid are neglected, large numbers of diffuse particles are efficiently processed to realize highly detailed small-scale effects. The presented results show that our approach can significantly improve the visual realism of large-scale fluid simulations.

Keywords Animation · Fluids · Particles

1 Introduction

In the field of computational fluid dynamics, efficient methods have been developed for simulating large bodies of water. But still, the realistic animation of ocean scenes with breaking waves is challenging. High turbulent flows demand a high resolution to capture small scale details like splashes, while for the major part of the fluid a lower resolution is sufficient. This can be handled by employing multi-scale methods, e. g. for Eulerian approaches [19,18,16,6] and for Lagrangian approaches [1,29,27]. Furthermore, for realistic animations, the interplay of fluid and air has to be modeled. This interplay causes the formation of diffuse material,

perceived as foam, spray and tiny air bubbles, which is the focus of this paper.

In order to simulate fluid-air mixtures, multiphase approaches with an explicit air phase have been proposed [31]. However, the high density contrast of water and air is numerically challenging and restricts the time step. Therefore, some methods propose to model the air phase implicitly. In these approaches, air-water mixtures are determined based on hypothetical models. Implicit modeling is convenient to capture even small-scale phenomena. In the context of air bubble animations, this has been demonstrated for grid-based [12] and particle-based solvers [25,10,15], whereas implicit mist and spray models [33,20,6] have been coupled only to grid-based fluid solvers. To the best of our knowledge, we propose the first uniform model for capturing spray, foam and air-bubbles for Lagrangian fluids.

Contribution: We propose a technique for adding diffuse material to particle-based fluid simulations. Our approach does not employ a set of techniques for different types of diffuse material, but solely relies on particles.

We formulate physically-motivated criteria for determining regions where water mixes with air. In these regions, the amount of air concentration determines the number of generated and dissolved particles that represent diffuse material. It should be noted that appropriate criteria for the generation of diffuse particles are essential in order to obtain realistic flow patterns over time. The advection of diffuse particles is not based on interparticle forces, but mainly determined by the velocity field of the water phase. Thus, expensive neighborhood computations are avoided.

As the influence of diffuse material onto the water phase can be neglected for large-scale simulations, the proposed model is realized as a post-processing step

M. Ihmsen, N. Akinici, G. Akinici, M. Teschner
University of Freiburg
Georges-Koehler-Allee 052
79110 Freiburg
Tel.: +49-761-2038289
E-mail: ihmsen@informatik.uni-freiburg.de

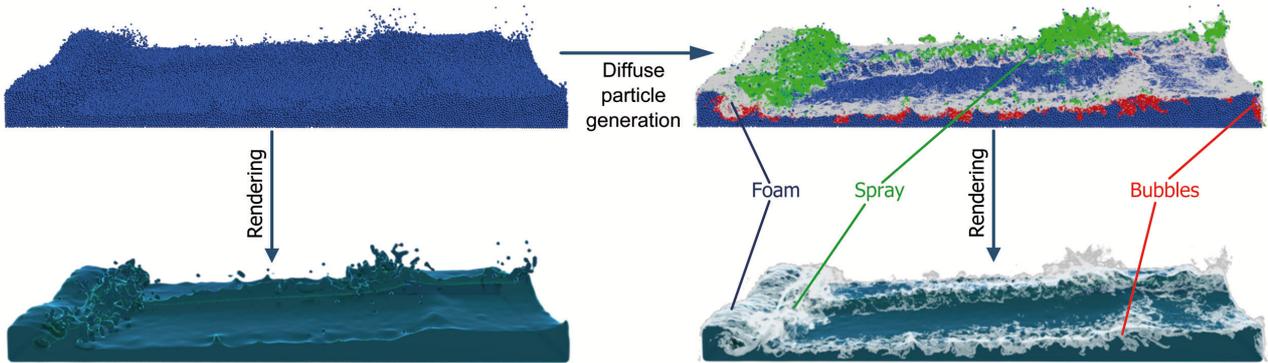


Fig. 1 Our diffuse model is applied as a post-process step for single-phase particle-based fluid (top left). Air-water mixtures like spray, foam and bubbles are generated, advected and dissolved (top right). For rendering, the fluid’s surface is triangulated neglecting isolated particles. A volumetric render technique is employed for diffuse material (bottom right).

on pre-computed fluid simulations. Furthermore, large time steps up to the frame rate of the animation can be handled.

The unified model for diffuse material requires a minimal set of input parameters which can be provided by any Lagrangian fluid simulation. Due to its computational efficiency and the intuitive parameter setting, the proposed model is an attractive tool to improve the visual realism of existing large-scale particle simulations.

2 Related Work

Foam and splash effects are important features of realistic fluid simulations [32, 5, 6], movies [7, 9, 4] and commercial software like Naiad (Exotic Matter) and RealFlow (Next Limit) [26]. These works, however, only briefly describe the generation, advection and dissolution of diffuse material. Furthermore, they combine different techniques, e. g. particles and textures.

In Computer Graphics, only a few models have been presented for the unified simulation of diffuse materials. In [31], a model for simulating splashes and foam is proposed, which couples the particle-based diffuse material with a grid-based fluid solver. Splash and foam particles are generated when the surface curvature of the fluid exceeds a threshold. State change rules are employed to advect and dissolve splashes and foam. Our model shares the motivation of this work. However, we propose alternative criteria for Lagrangian fluids that create diffuse material not only at the crest of a wave, but also capture the entrainment of air.

A common strategy for grid-based solvers is to generate secondary particles in under-resolved regions [8, 10]. In [17], massless marker particles that are escaping

from the main body of water are transformed into water particles to represent subcell-level features like splashes and droplets. This idea is adopted in [20] to simulate splash and spray. In this method, dense fluid regions are solved on a grid using the combination of FLIP and PIC described in [34], while the interface is modeled by the particle level set method [8]. Instead of removing particles that cross the interface, they are further simulated as diffuse material using the Smoothed Particle Hydrodynamics (SPH) method. Air-water mixtures are naturally handled without employing explicit criteria for the generation of diffuse material. Impressive results could be achieved by the two-way coupling of dense and diffuse materials. However, this approach is restricted to the level set method, whereas our model can be applied to any particle-based fluid. Moreover, [20] primarily focuses on capturing small-scale details which would be lost otherwise. In contrast, our model considers the advection and dissolution of different types of diffuse materials. Furthermore, forces between diffuse particles are not computed which avoids expensive neighborhood computations.

Various approaches use purely geometric criteria like curvature to detect air-water mixtures, e. g. [11, 25]. However, as discussed in [22], fluid dynamics also plays an important role. Mihalef et al. determine the generation of small-scale splashes according to the relation of the kinetic energy and the surface energy which is referred to as the Weber number [28]. In this approach, the large-scale flow is simulated with the Marker Level Set method [21], while small-scale details are modeled with a particle system. Splashes and air bubbles could be realistically captured with this model, however, the simulation and rendering of floating foam is not discussed.

Bagar et al. [3] use the Weber number to classify fluid regions into water and foam for real-time rendering of SPH fluids. In this method, fluid particles are rendered as foam particles, when the Weber number exceeds a threshold. Thereby, the realism of real-time simulations is significantly improved. The simulation, however, does not distinguish between foam and fluid particles and does not generate additional particles. Thus, the level of detail is restricted to the resolution of the underlying fluid. In contrast, our model generates diffuse particles with an initial momentum based on the kinetic energy of the fluid. Furthermore, the level of detail of the diffuse material is scalable.

3 Diffuse Material

In the proposed model, the air phase is simulated implicitly in order to avoid numerical instabilities, small time steps and expensive computations. Our model computes the potential of each fluid particle to mix with air. If there is some potential, a fluid particle generates diffuse particles. Based on their location, particles are classified as foam, spray or bubbles (see Fig. 1). The type determines how particles are advected. In Sec. 3.1, we describe how the potential for diffuse material is computed. Furthermore, we explain how the position and velocity of generated diffuse particles are initialized. Subsequently, we state how diffuse particles are advected and dissolved in Sec. 3.2.

3.1 Formation of Diffuse Material

In nature, diffuse material develops when the surface tension of water molecules is reduced and water mixes with air. E. g., if the crest of a wave gets unstable, water becomes aerated, resulting in turbulent whitewater which either spills down as foam or sprays away from the wave as mist. The activity of mixing might be caused by high wind and wave speeds, but also by trapped air due to high velocity differences at the fluid surface. In both cases, the amount of generated diffuse material increases with the kinetic energy of the fluid.

In our model, the potential of a fluid particle to mix with air is determined by its potential to trap air, its likelihood to be at the crest of a wave and its kinetic energy. A fluid particle generates diffuse particles, if the combined potential is larger than zero. In order to scale the amount of newly generated diffuse material and, thus, the level of detail, the user can provide minimum τ^{min} and maximum τ^{max} thresholds for each criterion. According to these thresholds, the potential I for the generation of diffuse material is mapped to the

range between zero and one using a clamping function Φ which is defined as

$$\Phi(I, \tau^{min}, \tau^{max}) = \frac{\min(I, \tau^{max}) - \min(I, \tau^{min})}{\tau^{max} - \tau^{min}}. \quad (1)$$

We now describe the proposed potential followed by the explanation, how the position and velocities of generated particles are sampled.

Trapped air. Air is trapped by impacts, e. g. when the lip of a wave hits shallow water. In this case, air is dragged under water. Furthermore, high turbulences cause aerated splashes. In order to determine those regions, the curl operator $\nabla \times \mathbf{v}$ might be a good choice. However, while the curl is large in turbulent regions, it might be small for impacts. Therefore, we propose to use relative velocities to determine regions where air is potentially trapped since these are large for impacts and vortices. We further assume that the amount of trapped air is larger, if the fluid particles move towards each other which can be measured by $(1 - \hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{x}}_{ij})$ with $\hat{\mathbf{v}}_{ij} = \frac{\mathbf{v}_i - \mathbf{v}_j}{\|\mathbf{v}_i - \mathbf{v}_j\|}$ denoting the normalized relative velocity between two particles and $\hat{\mathbf{x}}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|}$ their normalized distance vector. This term is zero for particles that move away from each other while it is two for particles that move towards each other. The scaled velocity difference

$$v_i^{diff} = \sum_j \|\mathbf{v}_{ij}\| (1 - \hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{x}}_{ij}) W(\mathbf{x}_{ij}, h) \quad (2)$$

is then used to compute the trapped-air potential as $I_{ta} = \Phi(v_i^{diff}, \tau_{ta}^{min}, \tau_{ta}^{max})$. $W(\mathbf{x}_{ij}, h)$ is a radially symmetric weighting function defined as

$$W(\mathbf{x}_{ij}, h) = \begin{cases} 1 - \|\mathbf{x}_{ij}\|/h & \|\mathbf{x}_{ij}\| \leq h \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

where h is the influence radius of the fluid simulation. Compared to commonly employed kernels, e. g. cubic spline [23], we found this weighting function to give better estimates near free surfaces where the neighborhood of fluid particles is not sufficiently sampled.

Wave crest. At the crest of a wave, whitewater is created either by strong winds or if the wave gets unstable, i. e. if the wave base can no longer support its top. In order to identify these areas, we assume that the surface curvature is high and the surface is locally convex. For a set of points, the surface curvature κ can be approximated with

$$\kappa_i = \sum_j \kappa_{ij} = \sum_j (1 - \hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j) W(\mathbf{x}_{ij}, h), \quad (4)$$

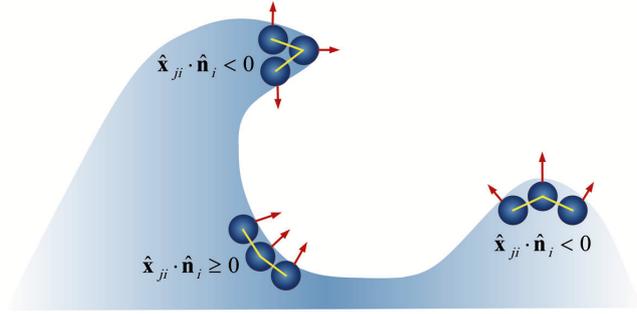


Fig. 2 Convex and concave regions are determined using the angle between the surface normal of a fluid particle and the relative position vectors to its neighbors. Diffuse particles are created in convex regions only if the velocity of the fluid is in normal direction.

where $\hat{\mathbf{n}}$ is a normalized surface normal. In order to distinguish convex from concave regions, the angles between $\hat{\mathbf{n}}$ and $\hat{\mathbf{x}}_{ji}$ are considered (see Fig. 2). Accordingly, wave crests are identified using

$$\tilde{\kappa}_i = \sum_j \tilde{\kappa}_{ij} \quad (5)$$

with

$$\tilde{\kappa}_{ij} = \begin{cases} 0 & \hat{\mathbf{x}}_{ji} \cdot \hat{\mathbf{n}}_i \geq 0 \\ \kappa_{ij} & \hat{\mathbf{x}}_{ji} \cdot \hat{\mathbf{n}}_i < 0 \end{cases} \quad (6)$$

Experiments show that (5) is a good estimate to identify wave crests. However, (5) would also identify all edges of, e. g., a cube of water as wave crests. Therefore, we propose to additionally check, if the fluid particle moves in normal direction using

$$\delta_i^{vn} = \begin{cases} 0 & \hat{\mathbf{v}}_i \cdot \hat{\mathbf{n}}_i < 0.6 \\ 1 & \hat{\mathbf{v}}_i \cdot \hat{\mathbf{n}}_i \geq 0.6 \end{cases} \quad (7)$$

Finally, the likelihood of a particle to be at the crest of a wave is computed as $I_{wc} = \Phi(\tilde{\kappa}_i \cdot \delta_i^{vn}, \tau_k^{min}, \tau_k^{max})$.

Energy. In fluid dynamics, the Weber number is a useful quantity for analyzing the air entrainment and the formation of droplets. However, its exact computation requires to correctly model the change in surface tension, for example caused by strong winds or the influence of solute concentrations. Since this is challenging to model, [22] and [3] assume a constant surface tension. As changes in the surface tension are not considered, only the kinetic energy $E_{k,i} = 0.5m_i\mathbf{v}_i^2$ is used as a measurement for air entrainment. Therefore, we relate the amount of diffuse material generated by a fluid particle to its kinetic energy. Accordingly, the potential to generate diffuse particles due to kinetic energy is computed as $I_k = \Phi(E_{k,i}, \tau_k^{min}, \tau_k^{max})$, with user-defined τ_k^{min} and τ_k^{max} .

Sampling. The proposed criteria are finally composed in order to compute the amount of generated diffuse

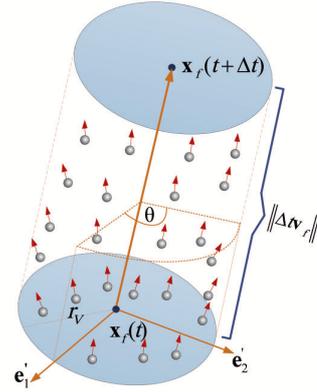


Fig. 3 Diffuse particles are generated by a fluid particle f . They are uniformly distributed in a cylinder spanned by the volume radius r_V , the current position $\mathbf{x}_f(t)$ and the position $\mathbf{x}_f(t + \Delta t)$ of the fluid particle. The velocities are initialized according to the positions and the fluid velocity \mathbf{v}_f .

material for each fluid particle. Here, we assume that air either mixes with water at the crest of a wave or when air is trapped. In both cases, the amount of mixed air scales with the kinetic energy. Due to the mapping function Φ , all three potentials are in the range between zero and one. The user controls the amount of generated diffuse material per particle and time step by providing a maximum number of samples that should be generated on wave crests k_{wc} and for trapped air k_{ta} per second. Consequently, the number of diffuse particles generated by a fluid particle is computed as

$$n_d = I_k (k_{ta}I_{ta} + k_{wc}I_{wc}) \Delta t, \quad (8)$$

where Δt denotes the time step.

We propose to sample the generated particles in a cylinder spanned by the volume radius r_V of a fluid particle and its velocity \mathbf{v}_f . Therefore, we compute a reference plane orthogonal to \mathbf{v}_f , spanned by \mathbf{e}_1 and \mathbf{e}_2 with the fluid particle position \mathbf{x}_f as the origin. The position of a diffuse particle \mathbf{x}_d is computed according to three uniformly distributed random variables $X_r, X_\theta, X_h \in [0, 1]$. They determine the distance to the cylinder axis $r = r_V \sqrt{X_r}$, the azimuth $\theta = X_\theta 2\pi$ and the distance $h = X_h \cdot \|\Delta t \mathbf{v}_f\|$ from the reference plane (see Fig. 3). Accordingly, \mathbf{x}_d is computed as $\mathbf{x}_d = \mathbf{x} + r \cos \theta \mathbf{e}_1 + r \sin \theta \mathbf{e}_2 + h \hat{\mathbf{v}}_f$. Thereby, particles are uniformly sampled in the cylinder.

The proposed model does not compute internal forces for diffuse particles. They are mainly advected by the velocity field of the fluid as explained in the subsequent section. In order to simulate splashes into different directions, the velocities of newly created diffuse particles are initialized as $\mathbf{v}_d = r \cos \theta \mathbf{e}_1 + r \sin \theta \mathbf{e}_2 + \mathbf{v}_f$ where \mathbf{e}_1 and \mathbf{e}_2 act as velocities.

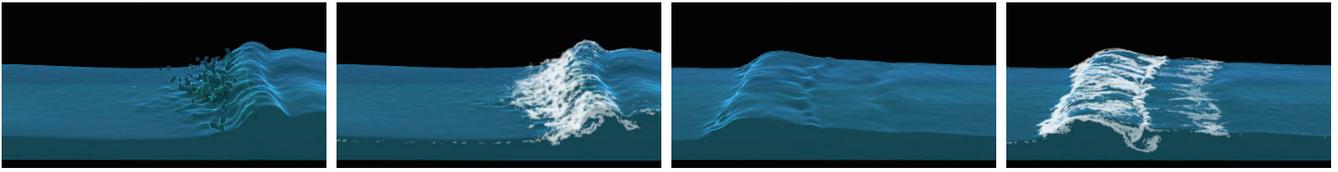


Fig. 4 Post-processing of a single-phase wave simulation. Water-air mixtures are generated at the crest of the wave and in the impact zone of the wave. The realism of the coarse fluid simulation is significantly improved by adding multiphase effects with small-scale features.

It should be noted that an appropriate sampling is important to obtain a volumetric appearance of diffuse material and also to avoid regular patterns.

3.2 Advection and Dissolution

As long as diffuse material is not influenced by water, its motion can be assumed to be ballistic, whereas under water, diffuse material is highly influenced by the water phase. Water exerts high pressures on the buoyant air bubbles which are under strong velocity diffusion. When rising bubbles reach the surface, they form floating foam which is transported by the liquid.

In accordance with these observations, we do not compute internal forces for diffuse particles, but determine their motion using the fluid velocities and external forces. Since the influence of the fluid depends on the location, we classify the diffuse material into spray, foam and bubbles (see Fig. 1). A diffuse particle d is classified as spray, foam or air bubble according to its fluid particle density $\tilde{\rho} = \sum_f K(\mathbf{x}_d - \mathbf{x}_f, h)$, where f denotes fluid particles and K should be a normalized, symmetric kernel, e. g. the cubic spline kernel [23]. At the surface, a diffuse particle has less fluid neighbors than inside the fluid volume. Consequently, inside the fluid volume, the particle density is always higher than on the surface, at least for weakly or incompressible fluids. Therefore, surface particles can be either determined by the gradient of the density field [24] or simply by the number of neighbors. In all our experiments, diffuse particles with less than 6 fluid neighbors are considered as spray particle. Particles with more than 20 fluid neighbors are classified as air bubbles. In all other cases, particles are considered to be foam.

The motion of spray is computed solely using the momentum, external forces \mathbf{F}_{ext} and gravity \mathbf{g} . Using the Euler-Cromer method, the velocity of a spray particle is updated as $\mathbf{v}_{spray}(t + \Delta t) = \mathbf{v}_{spray}(t) + \Delta t(\frac{\mathbf{F}_{ext}(t)}{m} + \mathbf{g})$, where m is the mass of the spray particle. The position is then updated as

$$\mathbf{x}_{spray}(t + \Delta t) = \mathbf{x}_{spray}(t) + \Delta t\mathbf{v}_{spray}(t + \Delta t).$$

In contrast, foam and bubble particles are highly influenced by the fluid. Foam is purely advected according

to the averaged local fluid velocity at the position of the diffuse particle d which is computed as

$$\tilde{\mathbf{v}}_f(\mathbf{x}_d, t + \Delta t) = \frac{\sum_f \mathbf{v}_f(t + \Delta t)K(\mathbf{x}_d(t) - \mathbf{x}_f(t), h)}{\sum_f K(\mathbf{x}_d(t) - \mathbf{x}_f(t), h)},$$

where $\mathbf{v}_f(t + \Delta t) = \frac{\mathbf{x}_f(t + \Delta t) - \mathbf{x}_f(t)}{\Delta t}$. In order to constrain foam particles to the fluid surface, we only update their positions as

$$\mathbf{x}_{foam}(t + \Delta t) = \mathbf{x}_{foam}(t) + \Delta t\tilde{\mathbf{v}}_f(\mathbf{x}_d, t + \Delta t),$$

but not their velocities.

Due to the high density contrast of water and air, the motion of air bubbles is additionally governed by buoyancy which counteracts gravity. Therefore, the velocity of an air bubble is computed as

$$\mathbf{v}_{bub}(t + \Delta t) = \mathbf{v}_{bub}(t) + \Delta t \left(-k_b\mathbf{g} + k_d \frac{\tilde{\mathbf{v}}_f(\mathbf{x}_d, t + \Delta t) - \mathbf{v}_{bub}(t)}{\Delta t} \right),$$

where k_b and k_d are user-defined constants which control the buoyancy and drag effects. It should be noted that if k_d is chosen as 1, air bubbles are immediately dragged into the flow direction of the fluid. The position of an air bubble is then updated as

$$\mathbf{x}_{bub}(t + \Delta t) = \mathbf{x}_{bub}(t) + \Delta t\mathbf{v}_{bub}.$$

Dissolution. When diffuse particles are created, their lifetime is initialized with a predetermined value. In each simulation step, the time step is subtracted from the lifetime of foam particles, whereas the lifetime of air bubbles and spray particles is not reduced. Foam particles are finally deleted when the lifetime is smaller or equal to zero. In nature, large clusters of foam are more stable than smaller ones. In order to capture this effect without determining the foam area, we set the lifetime in relation to the generation potentials. Thus, the lifetime is not constant, but in the range between a user-defined minimum and maximum value.

Table 1 Timings and particle counts for the presented show-cases.

	# particles		computation times		frames
	fluid	diffuse	fluid	diffuse	
Wave	220k	2.7m	53 min	28 min	1000
Tower	3.5m	7m	995 min	81 min	700
Ship	2m	2m	185 min	25 min	400
Lighthouse	640k	3.3m	420 min	35 min	1000
Lighthouse	640k	15m	420 min	121 min	1000

3.3 Implementation and Rendering

In the proposed model, interparticle forces for the diffuse material are not computed. Therefore, the diffuse neighbors of a diffuse particle are not required. However, for determining the formation potentials and for the integration of particles, neighborhood relations between fluid particles, as well as for diffuse and fluid particles need to be computed. This can be efficiently computed with, e. g., compact hashing [13]. We simulated the fluid with PCISPH [30] using adaptive time-stepping [14] and for rigid-fluid coupling, we employed [2].

The sequences are rendered with mental ray 3.9, for which we have implemented a volume shader based on ray casting which accounts for absorption and emission of radiance, but neglects scattering effects. Accordingly, for each pixel of the final image, an eye ray is cast through the diffuse volume which is bounded by an axis aligned bounding box with entry point \mathbf{x}_s and exit point \mathbf{x}_e for a considered ray. The ray is sampled using equally spaced intervals $\Delta\mathbf{x}$. For each sample point \mathbf{x} on a ray with direction ω , the local volume density $\rho(\mathbf{x})$ is computed. Based on the density at each sample point \mathbf{x} , the radiance L along ω is attenuated according to

$$L(\mathbf{x}, \omega) = \prod_{i=0}^s e^{-\rho(\mathbf{x}_s + i\Delta\mathbf{x})\tau\Delta\mathbf{x}}, \quad (9)$$

where $s = \lfloor \frac{\mathbf{x} - \mathbf{x}_s}{\Delta\mathbf{x}} \rfloor$ and τ is a user-defined scaling factor. L is in the range between zero and one, where zero means that the light is fully absorbed by the diffuse volume. Let C_d be the user-defined color of the diffuse material and C_b the background color. The final color C_p for the corresponding pixel is then determined as $C_p = (1 - L(\mathbf{x}_e, \omega))C_d + L(\mathbf{x}_e, \omega)C_b$.

The volume density $\rho(\mathbf{x})$ is computed using constant thresholding. It is zero if there is no diffuse particle in the volume radius r_d around \mathbf{x} and otherwise it is one. In order to make spray look sharper than foam, we use different volume radii for spray and foam. Please note that alternative weighting functions could be employed, e. g. a Gaussian function. However, we found constant thresholding to give the best results.

4 Results

We discuss four scenarios which show different aspects of our approach and we evaluate the performance. The timings presented in Tab. 1 are measured on a 3.33 GHz Six-Core Intel machine. In all scenarios, diffuse material is simulated with respect to the frame rate of the fluid simulation which is 50 frames per second, resulting in a time step of 0.02 seconds. The volume radius of diffuse particles is ten times smaller than the radius of fluid particles. For all scenarios, we found the following thresholds to give good results: $\tau_{wc}^{min} = 2$, $\tau_{wc}^{max} = 8$, $\tau_{ta}^{min} = 5$, $\tau_{ta}^{max} = 20$, $\tau_k^{min} = 5$ and $\tau_k^{max} = 50$.

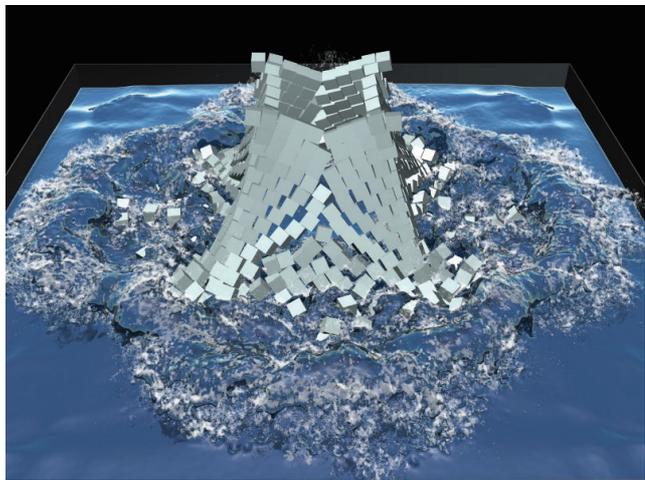


Fig. 5 A tower with 1000 cubes collapses. Diffuse material is generated at the crest of waves, for splashes and where air is trapped.

4.1 Generation of Diffuse Material

The wave scene (see Fig. 4) is a basic test for the generation criteria. It features a cuboid of low-viscous water that is dropped into a box. Due to the initialization, regions with high surface-curvature and velocity differences occur. Additionally, waves are generated by a moving plane. Due to the low resolution of the fluid, small scale details are not captured and isolated fluid splashes are too coarse.

Diffuse material is plausibly generated at wave crests and in regions with high velocity differences, while it is not generated at sharp features at the boundary. This scene captures many prominent effects like the formation of foam at the lip of the wave and the creation of whitewater in the so-called surf zone. The amount of generated diffuse material depends on the kinetic energy of the wave. Furthermore, the foam patterns and

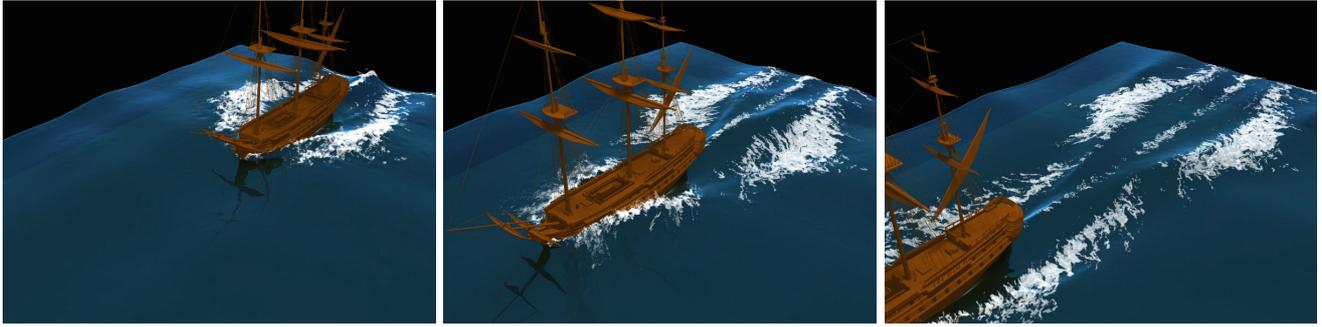


Fig. 6 Ship scene. Air entrainment of moving objects can be realistically captured as the foam pattern illustrates.

the air bubbles emphasize the dynamics of the fluid. The criteria are also working in more complex settings as demonstrated by the tower scene (see Fig. 5).

4.2 Ship: Air Entrainment by Moving Objects

Air entrainment caused by moving objects is a prominent effect that can be observed in nature. In order to demonstrate that our model is able to capture this effect, we employ an animated ship within a fluid simulation. Spray and foam are plausibly generated at the bow and the stern of the ship, resulting in realistic foam patterns (see Fig. 6). The visual appearance of the base simulation is significantly improved as also demonstrated in the accompanying video.

4.3 Lighthouse: Scaling of Quality and Performance

The animator can control the level of detail by defining the maximum amount of generated diffuse material on wave crests and in regions of high velocity differences (see Sec. 3.1). As a showcase for the scaling of quality and performance, a large-scale domain with a lighthouse is flooded. Large velocity differences and turbulences occur around the lighthouse, while larger waves form at the wall in the back.

In the top row of Fig. 7, an air-trapping fluid particle generated up to 30 diffuse particles per frame, while for wave crests, the maximum number of samples has been set to 50. Thereby, up to 3.3 million diffuse particles have been simulated per frame.

In the bottom row of Fig. 7, the number of generated samples has been increased by a factor of 4.5 for both, trapped air and wave crests. Thereby, up to 15 million diffuse particles have been simulated per frame. As also illustrated in the accompanying video, the additional diffuse particles significantly improve the level of detail. This is due to the proposed sampling which distributes positions and velocities of diffuse particles uniformly

in a cylinder spanned by the current and next frame position of the generating fluid particle.

For each scenario, the performance and the level of detail scale with the user-defined maximum thresholds for the number of generated samples. Since increasing these thresholds does not influence where diffuse material is generated, the animator can first adjust the minimum and maximum thresholds of the generation potential and then increase the resolution.

5 Conclusion

We have presented a unified method for adding diffuse material to single-phase particle fluids in a post-processing step. Diffuse particles like foam, spray and air bubbles are either generated at the crest of a wave or in regions where air is likely to be trapped. The amount of generated diffuse particles scales with the estimated air concentration and the kinetic energy. The proposed criteria together with the initialization of position and momentum, and the velocity-based one-way coupling are appropriate to realistically simulate the dynamics of diffuse material, resulting in visually plausible, highly detailed flow patterns. Expensive neighborhood computations are avoided and large time steps can be handled. The presented showcases demonstrate that the realism of fluid simulations is significantly improved especially for low resolution fluids.

In this work, we add air-water mixtures in a secondary simulation. Accordingly, we ignore the effect of diffuse material on the fluid simulation. Furthermore, we do not delete fluid particles when they become aerated. This does not conserve mass. In future work, we plan to address these issues by employing a two-scale approach in the sense of [29].

Acknowledgements

This project is supported by the German Research Foundation (DFG) under contract numbers SFB/TR-8 and

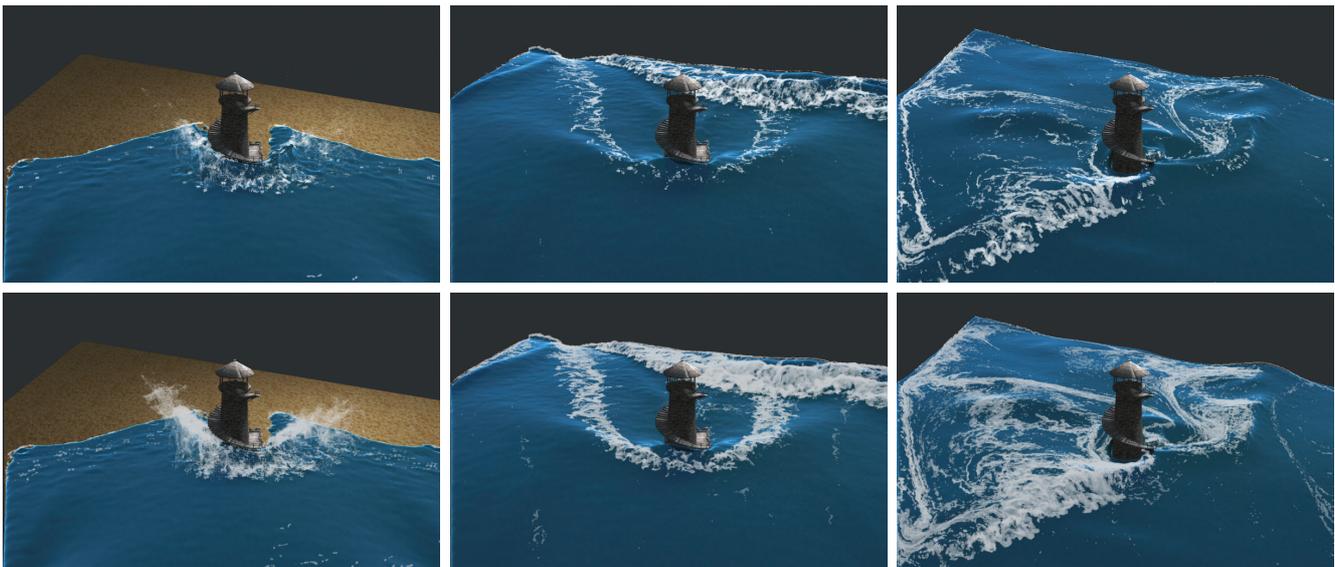


Fig. 7 Lighthouse scene. Foam and spray effects are added in a post-process using up to 3.3 million (top) and up to 15 million (bottom) diffuse particles. The level of detail can be controlled without changing the overall behavior.

TE 632/1-1. We also thank NVIDIA ARC GmbH for supporting this work. We are very grateful to Philipp Vath for his valuable ideas and his contribution to this project.

References

- Adams, B., Lenaerts, T., Dutre, P.: Particle Splatting: Interactive Rendering of Particle-Based Simulation Data. Tech. Rep. CW 453, Katholieke Uni. Leuven (2006)
- Anonymous: Versatile Rigid-Fluid Coupling for Incompressible SPH. submitted
- Bagar, F., Scherzer, D., Wimmer, M.: A Layered Particle-Based Fluid Model for Real-Time Rendering of Water. *Computer Graphics Forum (Proceedings EGSR 2010)* **29**(4), 1383–1389 (2010)
- Bredow, R., Schaub, D., Kramer, D., Hausman, M., Dimian, D., Duguid, R.S.: Surf’s up: the making of an animated documentary. In: *SIGGRAPH 2007 courses*, pp. 1–123 (2007)
- Chentanez, N., Müller, M.: Real-time simulation of large bodies of water with small scale details. In: *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 197–206 (2010)
- Chentanez, N., Müller, M.: Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid. *ACM Trans. on Graphics* **30**, 82:1–82:10 (2011)
- Fangmeier, S., Anderson, J., Zargarpour, H., Smythe, D., Alexander, T.: *Industrial Light + Magic : The Making of "The Perfect Storm"*. SIGGRAPH Panel (2000)
- Foster, N., Fedkiw, R.: Practical animation of liquids. In: *Proc. SIGGRAPH 2001*, pp. 23–30 (2001)
- Geiger, W., Leo, M., Rasmussen, N., Losasso, F., Fedkiw, R.: So real it’ll make you wet. In: *SIGGRAPH 2006 sketches* (2006)
- Greenwood, S.T., House, D.H.: Better with bubbles: enhancing the visual realism of simulated fluid. In: *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 287–296 (2004)
- Guendelman, E., Selle, A., Losasso, F., Fedkiw, R.: Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graphics* **24**, 973–981 (2005)
- Hong, J.M., Lee, H.Y., Yoon, J.C., Kim, C.H.: Bubbles alive. *ACM Trans. on Graphics* **27**, 48:1–48:4 (2008)
- Ihmsen, M., Akinci, N., Becker, M., Teschner, M.: A Parallel SPH Implementation on Multi-core CPUs. *Computer Graphics Forum* **30**(1), 99–112 (2011)
- Ihmsen, M., Akinci, N., Gissler, M., Teschner, M.: Boundary Handling and Adaptive Time-stepping for PCISPH. In: *Proc. VRIPHYS*, pp. 79–88 (2010)
- Ihmsen, M., Bader, J., Akinci, G., Teschner, M.: Animation of Air Bubbles with SPH. In: *Computer Graphics Theory and Applications GRAPP*, pp. 225–234 (2011)
- Irving, G., Guendelman, E., Losasso, F., Fedkiw, R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Trans. Graphics* **25**, 805–811 (2006)
- Kim, J., Cha, D., Chang, B., Koo, B., Ihm, I.: Practical Animation of Turbulent Splashing Water. In: *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 335–344 (2006)
- Klingner, B., Feldman, B., Chentanez, N., O’Brien, J.: Fluid animation with dynamic meshes. *ACM Trans. on Graphics* **25**(3), 820–825 (2006)
- Losasso, F., Gibou, F., Fedkiw, R.: Simulating water and smoke with an octree data structure. *ACM Trans. on Graphics* **23**, 457–462 (2004)
- Losasso, F., Talton, J., Kwatra, N., Fedkiw, R.: Two-Way Coupled SPH and Particle Level Set Fluid Simulation. *IEEE Trans. on Visualization and Computer Graphics* **14**(4), 797–804 (2008)
- Mihalef, V., Metaxas, D., Sussman, M.: Textured Liquids based on the Marker Level Set. *Computer Graphics Forum* **26**, 457–466 (2007)
- Mihalef, V., Metaxas, D.N., Sussman, M.: Simulation of two-phase flow with sub-scale droplet and bubble effects. *Computer Graphics Forum* **28**(2), 229–238 (2009)
- Monaghan, J.: Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.* **30**, 543–574 (1992)

24. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 154–159 (2003)
25. Müller, M., Solenthaler, B., Keiser, R., Gross, M.: Particle-based fluid-fluid interaction. In: Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 237–244 (2005)
26. Next Limit Technologies: Realflow 2012, Hybrido. White Paper (2011)
27. Raveendran, K., Wojtan, C., Turk, G.: Hybrid Smoothed Particle Hydrodynamics. In: Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 33–42 (2011)
28. Sirignano, W.A.: Fluid dynamics and transport of droplets and sprays. Cambridge University Press (1999)
29. Solenthaler, B., Gross, M.: Two-Scale Particle Simulation. *ACM Trans. on Graphics* **30**(4), 72:1–72:8 (2011)
30. Solenthaler, B., Pajarola, R.: Predictive-corrective incompressible SPH. *ACM Trans. on Graphics* **28**, 40:1–40:6 (2009)
31. Takahashi, T., Fujii, H., Kunimatsu, A., Hiwada, K., Saito, T., Tanaka, K., Ueki, H.: Realistic Animation of Fluid with Splash and Foam. *Computer Graphics Forum* **22**(3), 391–400 (2003)
32. Thürey, N., Müller-Fischer, M., Schirm, S., Gross, M.: Real-time Breaking Waves for Shallow Water Simulations. *Proc. of the Pacific Conference on Computer Graphics and Applications* pp. 39–46 (2007)
33. Thürey, N., Rüdte, U., Stamminger, M.: Animation of Open Water Phenomena with Coupled Shallow Water and Free Surface Simulations. *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* pp. 157–164 (2006)
34. Zhu, Y., Bridson, R.: Animating sand as a fluid. *ACM Trans. on Graphics* **24**, 965–972 (2005)