Two-way Coupled SPH and Particle Level Set Fluid Simulation

Frank Losasso, Jerry O. Talton, Nipun Kwatra, Ron Fedkiw

Abstract— **Grid-based methods have difficulty resolving features on or below the scale of the underlying grid. Although adaptive methods (e.g. RLE, octrees) can alleviate this to some degree, separate techniques are still required for simulating small-scale phenomena such as spray and foam, especially since these more diffuse materials typically behave quite differently than their denser counterparts. In this paper, we propose a two-way coupled simulation framework that uses the particle level set method to efficiently model dense liquid volumes and a smoothed particle hydrodynamics (SPH) method to simulate diffuse regions such as sprays. Our novel SPH method allows us to simulate both dense and diffuse water volumes, fully incorporates the particles that are automatically generated by the particle level set method in under-resolved regions, and allows for two way mixing between dense SPH volumes and grid-based liquid representations.**

Index Terms— **physically based modeling, fluid simulation, smoothed particle hydrodynamics, particle level set**

I. Introduction

The physics-based simulation of water has become prevalent in modern feature films, especially for scenes that include realistic secondary effects such as spray, foam, or bubbles. Such simulations have been used extensively in both photorealistic [1], [2] and animated [3]–[5] features. We differentiate between *dense* and *diffuse* water volumes noting that the incompressible Navier-Stokes equations are not appropriate for modeling diffuse regions such as spray/air mixtures which are clearly compressible. While dense volumes are adequately modeled with the incompressible Navier-Stokes equations as in [6], SPH methods are more appropriate for spray and foam because they can more accurately reflect the physical characteristics of the diffuse or aerated material. Thus, we propose a novel SPH method that is suitable for both dense *and* diffuse regions as well as the interactions between them. Moreover, since state-of-the-art grid-based solvers yield high fidelity results for dense regions (and especially for smooth surfaces), we also show how to two-way couple our SPH solver to the particle level set method.

Particle systems were first shown to be useful for computer graphics applications in [7]. Early works on using particles to model liquids include [8]–[10]. [11], [12] introduced the notion of smoothed particle hydrodynamics in which spatially smoothed particle representations can be used to solve the Navier-Stokes equations. [13] leveraged the smoothed particle representation to model gaseous phenomena, and [14] later introduced the full SPH methodology to the graphics community. [14] used an equation of state (EOS) to model pressure, thus imposing a severe time step restriction to resolve the sound waves present in compressible flow (similar to [15]). Notably, the computational cost of their method increases as the desired compressibility decreases, becoming most expensive in the incompressible flow limit. Contemporaneously, [16], [17] introduced another method for simulating particles with the Navier-Stokes equations based on [18]. Instead of integrating the Navier-Stokes equations on a smoothed particle basis, they carried out all calculations on a background grid, thus allowing for an efficient, fully incompressible simulation where the implicit handling of the acoustic waves removes the related time step restrictions (see also [19]). Later, [20] used level sets to represent the liquid interface downgrading particles from physical representations of fluid mass to auxiliary markers for interface tracking. Although this trend was continued by [6] adding interface trackers on the air side of the interface, [20] and subsequent papers [21]–[24] used these marker particles to represent spray and bubbles when they crossed over the interface.

Although various authors mixed grid-based solvers with particle methods for spray and foam using explicit rules for particle behavior [25], [26], one attractive aspect of the particle level set method is that it automatically produces particles in under-resolved regions. This is similar in spirit to the particle finite element method [27] where a standard finite element mesh is created for particles dense enough to form a continuum, and stray particles are simulated with methods more appropriate for spray. As researchers explored the strengths and weaknesses of different techniques, the distinction between grid-based methods and particle-based methods has blurred. For example, many vortex particle algorithms make use of some sort of background grid in order to decrease computational cost and algorithmic complexity (see e.g. [28]), and [29] use a background grid for all non-advection terms in their particle-based fluid solver. [22] attempted to use SPH for the removed particles in a standard particle level set implementation, but had difficulties using EOS-type methods to adequately enforce incompressibility. Thus, they handled the removed particles with a method similar to [29]. Our method is most similar in spirit to these works.

II. Previous Work

Building on the initial work of [14], [30] used EOS-based SPH for lava flows. This EOS SPH framework was also used in a series of papers to simulate water [31], melting solids [32]–[34], solid fluid coupling [35], [36], and multiphase flows [37]. [38] pointed out that the typical SPH EOS methods for simulating incompressible fluids lead to very stiff systems, making incompressible flow difficult to simulate. In fact, [39] states that SPH methods can only solve compressible fluid flows and proposes an SPH variant which does not use an EOS relationship for the pressure, but instead solves a global Poisson equation similar to grid-based methods. They obtain some rather impressive simulations of liquids.

III. Particle Level Set Method

Our fluid solver is predicated on previous grid-based Navier-Stokes implementations such as [40], which ignore viscous effects

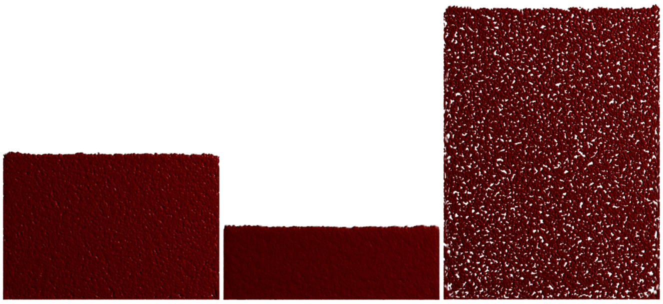

Fig. 1. (Left) We start an SPH simulation targeting a uniform particle number density. (Center) We then increase the target particle density causing the liquid to compress. (Right) Finally, we decrease the target particle density, and the fluid expands.

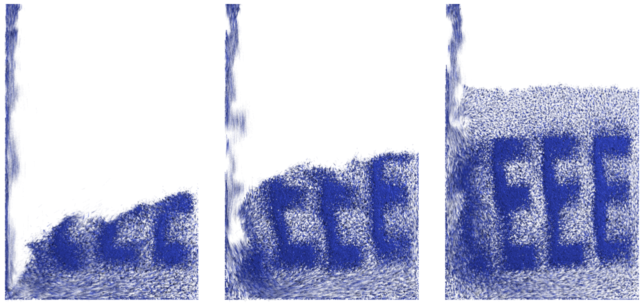

Fig. 2. SPH liquid flowing into a 120×240 box. The target particle number density is set to be low everywhere except in a region delineated by outlines of the letters "IEEE".

and use the inviscid form of the Navier-Stokes equations

$$\begin{aligned} \vec{u}_t + (\vec{u}\cdot\nabla)\vec{u} + \nabla p/\rho &= \vec{f} \\ \nabla\cdot\vec{u} &= 0 \end{aligned}$$

where $\vec{u} = (u, v, w)$ is the velocity, ρ is the density, and $\vec{f}$ accounts for body forces such as gravity and vorticity confinement. First, an intermediate velocity field $\vec{u}^\star$ is computed

$$(\vec{u}^\star - \vec{u}^k)/\Delta t + (\vec{u}^k\cdot\nabla)\vec{u}^k = \vec{f}$$

using a second-order unconditionally stable MacCormack method [41]. Since the modified MacCormack method can create spurious oscillations when sampling extrapolated velocities, we revert to the standard first-order accurate semi-Lagrangian method [42] near the liquid/air interface and object boundaries. These sorts of oscillations are acceptable (and may even be desirable) within a fluid volume to facilitate the simulation of complex flows, but they are detrimental to the visual quality of the surface when present near the interface. The transition point between the two methods can be controlled by the user of the system, but a small constant number of cells is usually sufficient. We employed a three-cell band in our simulations.

Next, we compute a scaled pressure $\hat{p} = p\Delta t$ via

$$\nabla\cdot(\nabla\hat{p}/\rho) = \nabla\cdot\vec{u}^\star \tag{1}$$

and use it to make the velocity field divergence free

$$(\vec{u}^{k+1} - \vec{u}^\star) + \nabla\hat{p}/\rho = 0. \tag{2}$$

We use the standard particle level set method to model the interface with particles on both sides as in [6]. The removed particles generated by the particle level set method are used to simulate secondary effects like spray and foam via our new SPH solver.

[14] proposed an SPH method predicated on an EOS of the form $p = k(\rho - \rho_0)$ where ρ_0 represents the target density of the fluid. However, [43] noted that equations of the form $p = k\rho$ are generally better behaved since attractive forces between particles are known to cause instabilities in SPH simulations. While this latter formulation is more stable, it no longer provides any mechanism for density targeting. Density targeting is important for adaptivity: being able to control the spatial density essentially allows one to create multiresolution SPH simulations in the same way that octrees and RLE grids allow for multiresolution simulations.

To avoid the time step restrictions induced by acoustic waves in EOS-based SPH models, one can instead solve a global Poisson equation for the pressure similar to the standard grid-based methods. [39] took this approach solving a Poisson equation that targeted the desired number density of particles based on the work of [44], [45]. These same authors ([46]) later realized that it is desirable to have an incompressible flow field regardless of the current particle density and proposed solving a Poisson equation for the pressure, targeting the removal of any divergence in the intermediate velocity field exactly as in grid-based methods. A similar projection procedure which uses pressure to make the velocity field divergence free was proposed for SPH simulation in [47] (see also [48]).

Later, [49] realized that it is desirable to both have a divergence free flow field and provide a mechanism for number density targeting. They first solve a Poisson equation for pressure to obtain a divergence free velocity field. Then, in order to target the desired particle number density, they solve a second Poisson equation to artificially alter the particle positions. However, since the velocities derived from this second solve are discarded, the method yields non-physical solutions. Consider, for example, a stationary flow field containing a subregion with particle density lower than the target density. The second Poisson solve will force particles from the higher density region to the lower density region as desired, but simply changing particle positions and ignoring the resulting velocities will leave the initially static velocity field unaltered. This is contrary to the true physical behavior where the subregion lower in density should have induced a flow of material into it.

As we show below, the Navier-Stokes equations allow for both the enforcement of incompressibility and the targeting of particle number density within a single Poisson solve, yielding the physically correct solution. This is clearly less expensive than a method that requires two Poisson solves. The equation for conservation of mass is

$$\frac{1}{\rho}\frac{D\rho}{Dt} + \nabla\cdot\vec{u} = 0$$

where $D/Dt = \partial/\partial t + \vec{u}\cdot\nabla$ is the material derivative, and ρ is the density which can be written in terms of the number of particles per region n_r, the mass per particle m_p, and the volume per region V_r as $\rho = n_r m_p / V_r$. This yields

$$\frac{1}{n_r}\frac{Dn_r}{Dt} + \nabla\cdot\vec{u} = 0. \tag{3}$$

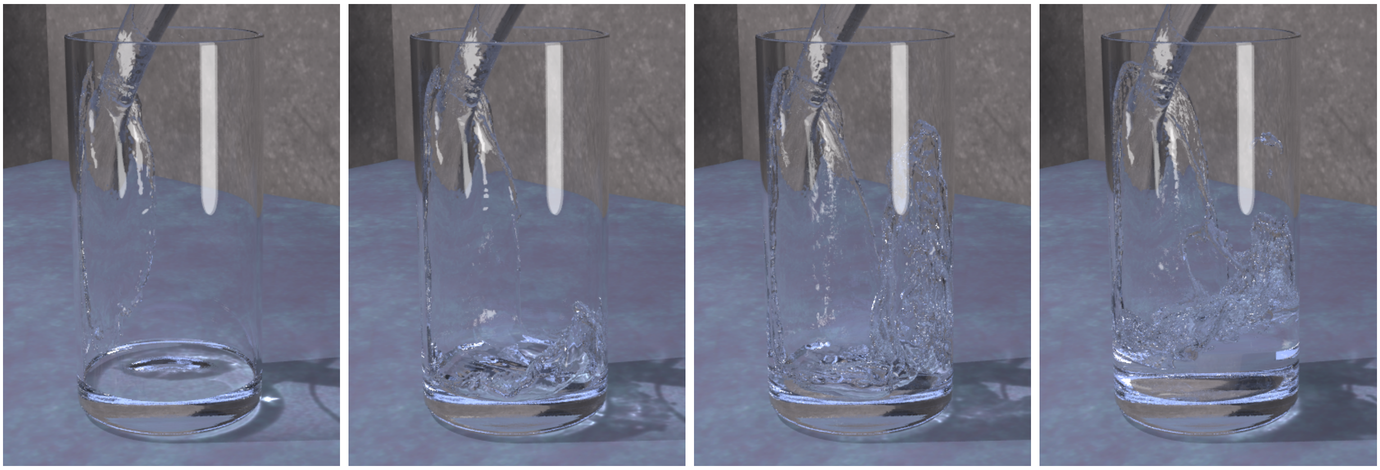

Fig. 3. Two-way coupling between SPH and the particle level set method to pour water into a glass. Note the small splashes provided by SPH that would not be resolvable by a traditional particle level set method.

Taking the divergence of both sides of equation (2) gives

$$\nabla \cdot (\nabla \hat{p}/\rho) = \nabla \cdot \vec{u}^\star - \nabla \cdot \vec{u}^{k+1}.$$

Substituting the value of $\nabla \cdot \vec{u}^{k+1}$ from equation (3) yields

$$\nabla \cdot (\nabla \hat{p}/\rho) = \nabla \cdot \vec{u}^\star + \frac{1}{n_r}\frac{Dn_r}{Dt}. \quad (4)$$

When the goal is to obtain a divergence free flow field as in equations (1) and (2), the divergence of $\vec{u}^{k+1}$ is set identically to zero. However, [50] noted that for certain physical phenomena (such as expansion caused by explosions) the ability to target a nonzero divergence can be useful. The above derivation illustrates that the same concept can be used to target a particle number density by replacing Dn_r/Dt in equation (4) with the desired change in particle number density. Since Dn_r/Dt is a material derivative it necessarily includes an advection term of the form $\vec{u}\cdot\nabla$, however our SPH-based solver is in the Lagrangian (rather than Eulerian) frame and therefore implicitly accounts for this term. Thus, we approximate

$$\frac{1}{n_r}\frac{Dn_r}{Dt} = \frac{1}{n_r^{k+1}}\left(\frac{n_r^{k+1} - n_r^k}{\Delta t}\right)$$

to obtain

$$\nabla \cdot (\nabla \hat{p}/\rho) = \nabla \cdot \vec{u}^\star + \frac{1}{n_r^{k+1}}\left(\frac{n_r^{k+1} - n_r^k}{\Delta t}\right) \quad (5)$$

Examples of our targeting can be seen in figures 1 and 2.

If the target number density is too far from the current density, our targeting scheme can introduce large velocities that result in substantial oscillations. In our implementation, we compensate by averaging the target divergence over a small time interval $\Delta\tau$. If we represent the last term in equation (5) as $T = \frac{1}{n_r}\frac{\Delta n_r}{\Delta t}$, we obtain the integral expression

$$\int_{n_r^k}^{n_r^{target}} \frac{dn_r}{n_r} = \int_{t_k}^{t_k+\Delta\tau} T dt$$

which can be solved to obtain $T = \frac{1}{\Delta\tau}\ln\left(\frac{n_r^{target}}{n_r^k}\right)$ replacing the last term in equation (5). In our simulations, we took $\Delta\tau$ between .25 and 1 seconds, which was sufficient to prevent any noticeable stability issues from arising.

IV. Diffuse SPH

The formulation described above is valid for dense fluid regions, but fails to adequately allow for diffuse behavior such as in spray/air mixtures. In diffuse fluid volumes, it is incorrect to assign a target density at each point in space, since diffuse regions are highly compressible and the distribution of particles is governed primarily by ballistic motion. Thus we modify our divergence formulation to better account for diffuse fluid by clamping the target divergence to be non-negative, which will force overly-dense regions to expand without causing diffuse regions to non-physically contract. However, this solution is not totally satisfactory since it still enforces incompressibility in regions below the target density threshold which is inappropriate for ballistic fluid features. The problem with equation (5) for diffuse regions is that one cannot ascertain the desired target density a priori without accounting for the ballistic motion of all the surrounding regions of flow.

To correct this problem, we observe that particles in dense regions should be incompressible, and thus move according to the velocities generated by the Poisson solve. On the other hand, solitary particles that are far from any other fluid feature should follow ballistic trajectories independent of incompressibility. Therefore, our SPH method introduces the notion of *particle slip* in which the smoothed number density at each particle position determines the degree to which the particle is affected by the Poisson solve. This slipping is highly desirable for scenes with both dense and diffuse particle regions as it allows particles to smoothly transition between incompressible and ballistic behavior in a physical manner.

As in most SPH algorithms, each particle represents a smoothed, radially-symmetric attribute field that distributes its associated quantities in a local neighborhood of influence. For a single particle p with radius r_p and position $\mathbf{x}_p$, we define the influence at a point $\mathbf{x}$ to be

$$\omega_p(\mathbf{x}) = \begin{cases} c(1 - \|\mathbf{x}-\mathbf{x}_p\|^2/r_p^2) & \text{when } \|\mathbf{x}-\mathbf{x}_p\|^2 \le r_p^2 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where c is a normalization constant. Since our method enforces incompressibility in an efficient fashion using a background MAC grid, we compute the influence of each particle at both cell centers

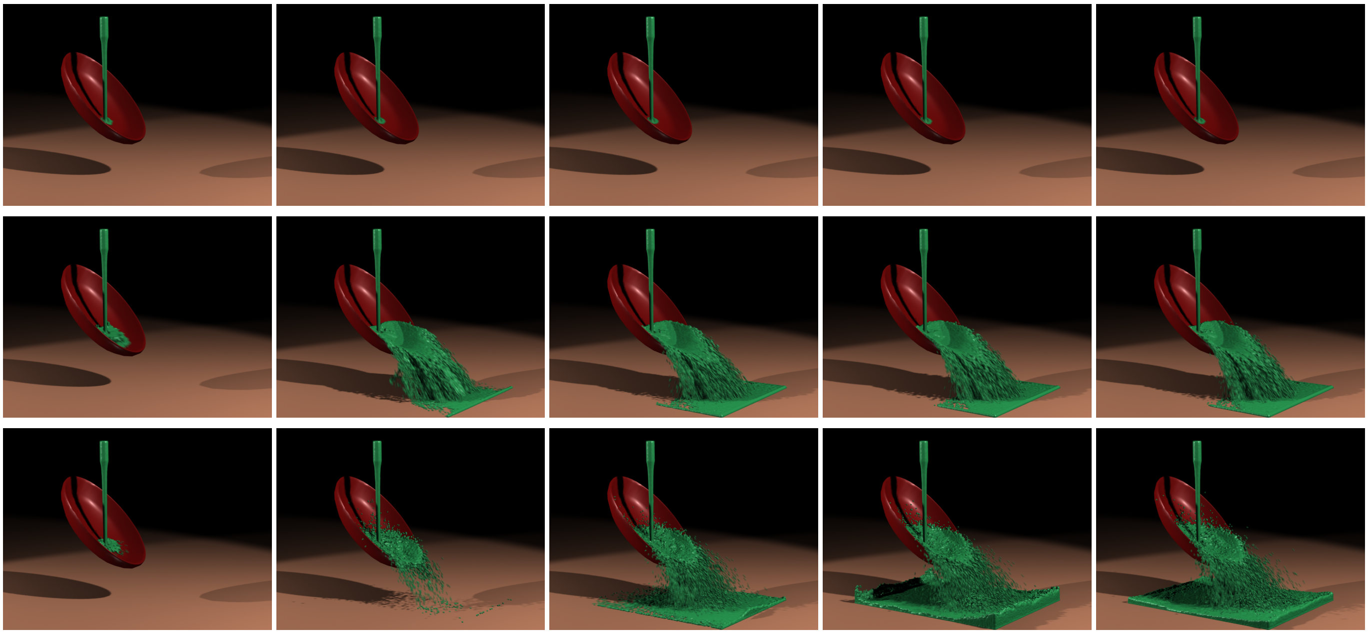

Fig. 4. (Top) A thin particle level set source pouring onto an upturned bowl on a $150 \times 200 \times 200$ grid. The grid cannot resolve the thin film at the point of impact and catastrophically loses mass. (Middle) The simulation with the removed negative particles visualized. The particles have no notion of volume and thus end up compressed on the bottom right edge of the domain. (Bottom) Our SPH solver treats the removed negative particles as an integral part of the liquid representation, and convincingly conserves volume.

and faces. A different normalization constant is used for each, and cumulative number densities are also calculated separately for cells and faces.

Each time step of our algorithm proceeds as follows. First, we apply gravity (and other body forces) to the particles. Next, we compute cell and face weights (i.e. particle number densities) using the blending kernel given in equation (6). Then we rasterize weighted particle velocities onto the faces of the grid, and store a copy of this velocity field for later use. Next we calculate the target divergence for each cell, solve the variable-density Poisson equation (5), and update the grid-based velocity field using equation (2). We calculate the change in the grid-based velocities and use the FLIP method [29] to compute a candidate change in velocity $\Delta\mathbf{v}$ for each particle by taking a linear combination of nearby face velocity differences weighted by the particle's influence at each face center. One disadvantage of FLIP is that a cell may contain particles with widely varying velocities and merely mapping $\Delta\mathbf{v}$ back to each particle adds no viscous behavior. If the particle velocities in a cell have high variance, we introduce a weighted average between FLIP and PIC on a per particle basis, since the PIC method will substantially damp outlying particle velocities by forcing them to more closely conform to those computed on the background grid.

Before mapping this change in velocity to the particles we introduce our notion of particle slip. To determine how much of the calculated velocity change to apply for a given particle, we compute a particle slip ratio s as the particle number density at the particle's position divided by the global incompressibility target density. Then, we update each particle's velocity $\mathbf{v}_p^{k+1} = \mathbf{v}_p^k + s\Delta\mathbf{v}$ and subsequently update the particle position using this velocity.

One of the more attractive features of this particle slip method is that we can control the degree to which the SPH solver influences our particles simply by adjusting the slip coefficient s. Also, we typically maintain a ballistic particle threshold of around 15% of the incompressible target density and remove cells with weights below this level from the Poisson solve altogether. Because of this level of control, our method can be trivially adapted to interact with arbitrary particle systems simply by disabling or scaling back the influence of our solver in regions that are subject to other dominant forces. In this manner, we can take into account spring-based attractive forces, elasticity, or even completely non-physical particle rules.

V. Two-way Coupling

In the standard particle level set algorithm, passive marker particles are seeded on both sides of the fluid interface and advected along the fluid flow. In areas where the grid is unable to fully resolve the level set's behavior, these marker particles will pass from one side of the interface to the other indicating error in the level set representation and prompting a local rebuilding of the level set function with the characteristic information present in these particles. When a particle strays too far across the interface, it can be removed from the set of interface tracking markers and instead used to represent spray or bubbles depending on whether it is a removed water or air particle, respectively. We use the removed negative particles that were originally on the interior of the fluid volume to seed our SPH algorithm (although other particles can be introduced as well).

For simulations with sparse particles or those in which the scale of the negative removed particles is small, it suffices for the fluid to exert force on the particles without the particles affecting the behavior of the fluid. In these cases, we one-way couple the grid-based solver to the SPH solver by first carrying out all the steps of a normal grid-based solve, and then using the result to generate boundary conditions for the SPH solver. In particular, each face that lies along the level set interface is set to a Neumann condition with the velocity provided by the grid-based solver.

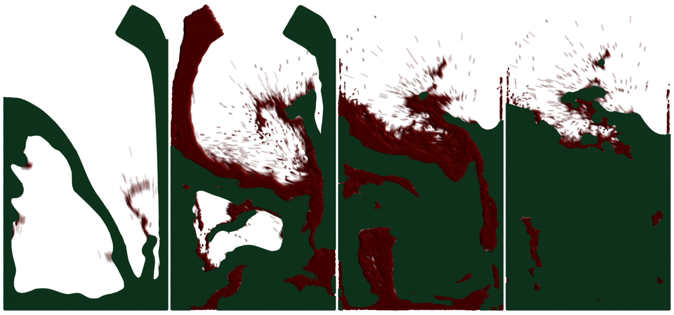

Fig. 5. Two-way coupled SPH and particle level set fluid simulation. The level set is depicted in green. The far left image shows a few negative removed particles generated from the level set that are subsequently simulated with SPH. In the second figure, we have added a source of further SPH particles. After turning off the source in the third figure, we turn on reincorporation of particles in dense regions so that they transition to a level set representation when possible providing for a smoother interface representation.

This coupling strategy is particularly convenient for adding detail to existing simulations, since it can be run entirely as a post-process. When employed simultaneously with the fluid solver, however, we can improve the visual quality of the simulation by reincorporating particles that penetrate the level set surface and applying a local momentum conservation force to slightly alter the level set velocities in reincorporation regions. Figure 4 shows a thin particle level set source impacting a tilted bowl. At the point of contact, the grid resolution is insufficient to resolve the thin liquid sheet resulting in complete mass loss (top). In the middle row, we see the same simulation visualizing the removed negative particles generated by the level set following their initial ballistic trajectories. These particles provide better visual cues, but fail to exhibit any fluid-like behavior. In the bottom row, we use the removed negative particles in a one-way coupled SPH simulation with improved results.

For simulations with dense particle regions, full two-way coupling is desirable. Thus, we use a single Poisson solve to compute the divergence free flow field simultaneously for the grid-based level set fluid volume and the SPH-governed regions. First we carry out all the non-projection steps of the grid-based particle level set solver. Then, for faces interior to the level set that have both valid SPH particle and valid grid-based level set velocities, we discard the particle velocity and instead use only the grid-based velocity for the Poisson solve. Cells inside the level set volume have their divergence set to zero, whereas SPH cells outside the fluid volume have their divergence set as described above (section III). After projection and the subsequent updating of the velocity field according to equation (2), the velocity is mapped back to the particles incorporating any desired slip. As in [6] the last step in the particle level set algorithm is to extrapolate velocities from the valid fluid volume to a band of surrounding air cells. However if particles occupy part of this region, their velocities should not be overwritten. Therefore we mark the cells in the grid-based air volumes where the particles provide an adequate external velocity field and only extrapolate to cells that do not contain a sufficient number of particles. Figure 5 shows a two-way coupled simulation of a particle-only SPH source interacting with a particle level set source. In this simulation, the particles are seeded at the target density and thus are given roughly equal weight to the fluid.

For added efficiency and surface smoothness, one can optionally convert SPH particles back to a grid-based level set representation in areas with sufficient particle density. This is accomplished by defining a level set around each particle in exactly the same manner as is done for the marker particles in the particle level set method. This computes new values for the level set function whereas new velocity values are defined directly from the smooth particle kernel (see rightmost image in figure 5).

VI. Examples

Our three-dimensional examples were carried out on a number of 4 processor Opteron machines and averaged between 30 seconds and 3 minutes per frame. We employed 32 particles per cell for all our 3D examples and 16 for the 2D ones. In figure 6, we apply our SPH method to an ocean scene with crashing waves on a $560 \times 120 \times 320$ grid. A number of authors have considered using the three-dimensional Navier-Stokes equations to simulate large ocean views, but it has proven quite challenging to convincingly convey the appropriate sense of scale without resorting to non-physical post-processes. We use full two-way coupling with the negative removed particles generating convincing SPH-simulated spray. We also run a secondary one-way coupled simulation of the air to generate fine-detail mist and foam that is sourced from the spray particles along the lines of [51]. Positive removed particles are also passively advected and used to represent bubbles. To generate good initial conditions for our waves, we use the same wave formulation as [52]. We rendered this scene in Pixar's RenderMan with a deep-water texture applied to the surface. We also consider the simulated pouring of water into a glass as in [6]. Figure 3 shows the results obtained using our two-way coupled variable-density SPH solver with particle to fluid conversion enabled on a $120 \times 240 \times 120$ grid. The particles provide convincing splashes and add physically-based turbulence to the fluid surface, resulting in a more realistic simulation that appears noticeably less viscous than those predicated on the particle level set method alone.

VII. Conclusions

We proposed a novel SPH solver which allows us to enforce incompressibility in an efficient fashion similar to standard grid-based methods as well as target arbitrary particle number densities with a single Poisson solve. We introduced the notion of particle slip in order to extend this SPH solver to simulate diffuse phenomena such as mixtures of spray and air. Finally we showed how to two-way couple our new SPH solver with a standard particle level set method and illustrated the efficacy of our approach with a number of examples including the crashing of ocean waves against a lighthouse and beach.

One of the main limitations of our approach is that the FLIP method we employ can introduce unwanted noise, due to the fact that it encourages particles to have wildly varying velocities. This problem can be ameliorated to some extent by averaging with PIC, at the cost of undesirable numerical viscosity. A more principled approach would be to apply PIC averaging only in areas of high particle velocity variance, relying entirely on FLIP in regions where particle velocities are relatively constrained. Another limitation is that particle density computation is unreliable near the air/liquid interface, where SPH particles do not have neighbors on all sides. We experimented with several strategies to reduce noise in these areas by reflecting dummy particles across the interface before computing particle densities, but met with

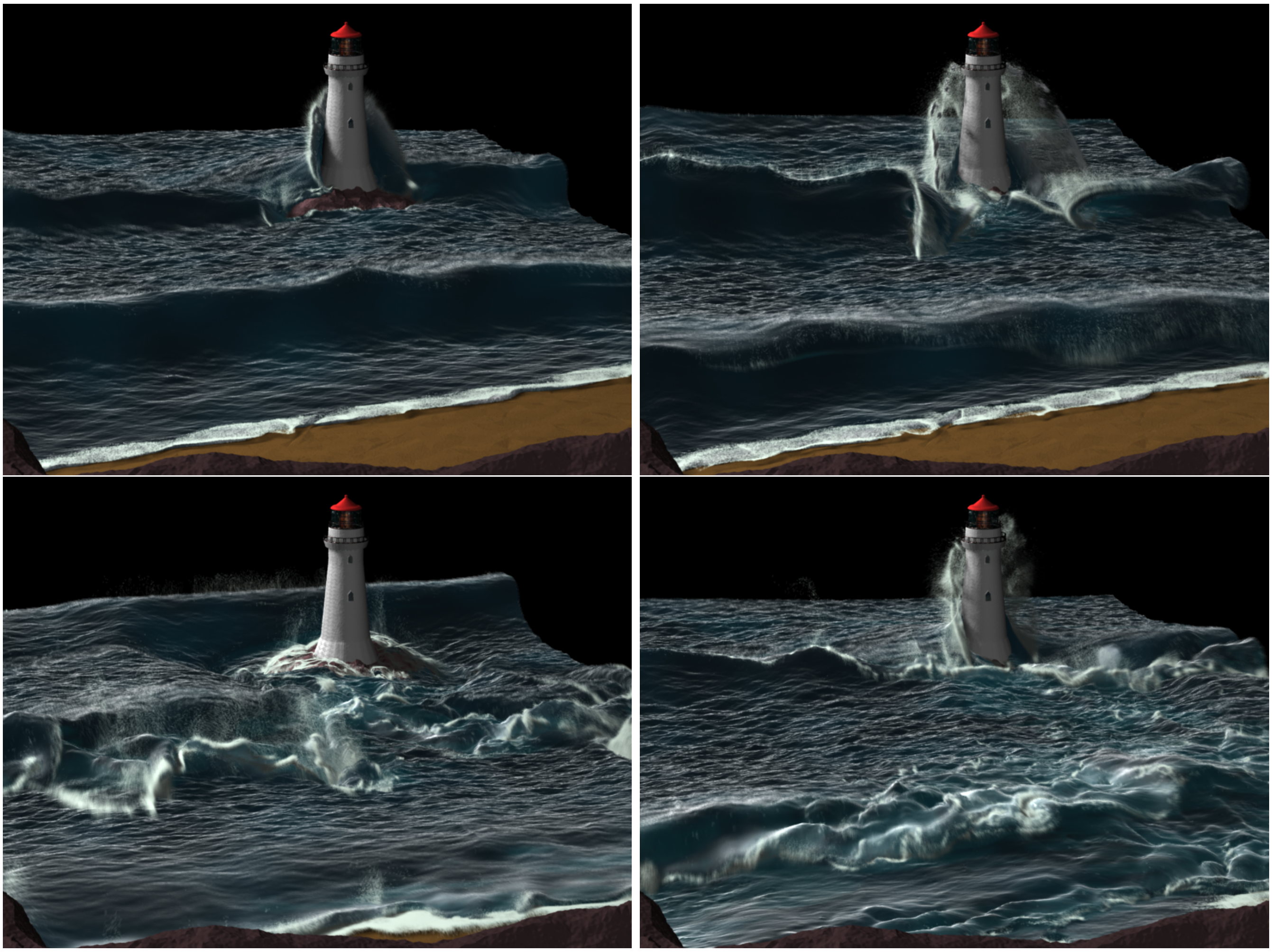

Fig. 6. A large ocean scene simulated with two-way coupling between our SPH method and the particle level set method. Besides the full two-way coupling, a secondary air simulation is used to generate a second layer of fine-detail mist and foam sourced from the SPH particles.

only limited success. Future work in these areas would be broadly applicable to the simulation community at large, since neither of these shortcomings is specific to our method.

Acknowledgment

Research supported in part by a Packard Foundation Fellowship, an Okawa Foundation Research Grant, ONR N0014-06-1-0393, ONR N00014-06-1-0505, ONR N00014-02-1-0720, ONR N00014-05-1-0479 for a computing cluster, NIH U54-GM072970, NSF ACI-0323866 and NSF IIS-0326388. We'd also like to thank Ranjitha Kumar for modeling the lighthouse and Pixar Animation Studios for providing RenderMan licenses.

References

[1] W. Geiger, M. Leo, N. Rasmussen, F. Losasso, and R. Fedkiw, "So real it'll make you wet," in *SIGGRAPH 2006 Sketches & Applications*. ACM Press, 2006.

[2] J. Iversen and R. Sakaguchi, "Growing up with fluid simulation on "The Day After Tomorrow"," in *SIGGRAPH 2004 Sketches & Applications*. ACM Press, 2004.

[3] D. Baraff, A. Witkin, M. Kass, and J. Anderson, "Physically based modeling (a little fluid dynamics for graphics)," in *SIGGRAPH Course Notes*. ACM, 2003.

[4] S. Brown and R. Collier, "Flooding "ice age: The meltdown" using wavesynth and point-based froth," in *SIGGRAPH 2006 Sketches & Applications*. ACM Press, 2006.

[5] J. Thornton, "Directable simulation of stylized water-splash effects in 3D space," in *SIGGRAPH 2006 Sketches & Applications*. ACM Press, 2006.

[6] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surfaces," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 21, no. 3, pp. 736–744, 2002.

[7] W. Reeves, "Particle systems - a technique for modeling a class of fuzzy objects," in *Comput. Graph. (Proc. of SIGGRAPH 83)*, vol. 17, 1983, pp. 359–376.

[8] D. Terzopoulos, J. Platt, and K. Fleischer, "Heating and melting deformable models (from goop to glop)," in *Graph. Interface*, 1989, pp. 219–226.

[9] G. Miller and A. Pearce, "Globular dynamics: A connected particle system for animating viscous fluids," *Comput. and Graph.*, vol. 13, no. 3, pp. 305–309, 1989.

[10] D. Tonnesen, "Modeling liquids and solids using thermal particles," in *Graph. Interface*, 1991, pp. 255–262.

[11] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics-theory and application to nonspherical stars," *Mon. Not. R. Astron. Soc.*, vol. 181, p. 375, 1977.

[12] L. Lucy, "A numerical approach to the testing of the fission hypothesis," *Astronomical J.*, vol. 82, pp. 1013–1024, 1977.

[13] J. Stam and E. Fiume, "Depicting fire and other gaseous phenomena using diffusion process," in *Proc. of SIGGRAPH 1995*, 1995, pp. 129–136.

[14] M. Desbrun and M.-P. Cani, "Smoothed particles: A new paradigm for animating highly deformable bodies," in *Comput. Anim. and Sim. '96 (Proc. of EG Wrkshp. on Anim. and Sim.)*, R. Boulic and G. Hegron, Eds. Springer-Verlag, Aug 1996, pp. 61–76, published under the name

Marie-Paule Gascuel.
[15] G. Yngve, J. O'Brien, and J. Hodgins, "Animating explosions," in *Proc. of ACM SIGGRAPH 2000*, 2000, pp. 29–36.
[16] N. Foster and D. Metaxas, "Realistic animation of liquids," *Graph. Models and Image Processing*, vol. 58, pp. 471–483, 1996.
[17] ——, "Controlling fluid animation," in *Comput. Graph. Int.*, 1997, pp. 178–188.
[18] F. Harlow and J. Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Phys. Fluids*, vol. 8, pp. 2182–2189, 1965.
[19] N. Foster and D. Metaxas, "Modeling the motion of a hot, turbulent gas," in *Proc. of SIGGRAPH 97*, 1997, pp. 181–188.
[20] N. Foster and R. Fedkiw, "Practical animation of liquids," in *Proc. of ACM SIGGRAPH 2001*, 2001, pp. 23–30.
[21] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw, "Coupling water and smoke to thin deformable and rigid shells," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 24, no. 3, pp. 973–981, 2005.
[22] J. Kim, D. Cha, B. Chang, B. Koo, and I. Ihm, "Practical animation of turbulent splashing water," in *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2006, pp. 335–344.
[23] F. Losasso, G. Irving, E. Guendelman, and R. Fedkiw, "Melting and burning solids into liquids and gases," *IEEE Trans. on Vis. and Comput. Graph.*, vol. 12, no. 3, pp. 343–352, 2006.
[24] S. T. Greenwood and D. H. House, "Better with bubbles: enhancing the visual realism of simulated fluid," in *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 2004, pp. 287–296.
[25] J. F. O'Brien and J. K. Hodgins, "Dynamic simulation of splashing fluids," in *Comput. Anim. '95*, apr 1995, pp. 198–205.
[26] T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki, "Realistic animation of fluid with splash and foam," *Comp. Graph. Forum (Eurographics Proc.)*, vol. 22, no. 3, pp. 391–400, 2003.
[27] E. Oñate, J. Garcia, S. Idelsohn, and F. D. Pin, "Finite calculus formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches," *Comput. Meth. Appl. Mech. Eng.*, vol. 195, pp. 3001–3037, 2006.
[28] A. Selle, N. Rasmussen, and R. Fedkiw, "A vortex particle method for smoke, water and explosions," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 24, no. 3, pp. 910–914, 2005.
[29] Y. Zhu and R. Bridson, "Animating sand as a fluid," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 24, no. 3, pp. 965–972, 2005.
[30] D. Stora, P.-O. Agliati, M.-P. Cani, F. Neyret, and J.-D. Gascuel, "Animating lava flows," in *Graph. Interface*, 1999, pp. 203–210.
[31] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 2003, pp. 154–159.
[32] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa, "Point based animation of elastic, plastic and melting objects," in *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 2004, pp. 141–151.
[33] M. Wicke, P. Hatt, M. Pauly, M. Mueller, and M. Gross, "Versatile virtual materials using implicit connectivity," in *Proceedings of the Eurographics Symp. on Point-Based Graph.* Eurographics Association, 2006, p. 8.
[34] B. Solenthaler, J. Schlfli, and R. Pajarola, "From fluid to solid and back in a unified particle model," in *ACM SIGGRAPH / Eurographics Symp. on Computer Animation - Posters and Demos*. ACM Press, 2006, pp. 13–14.
[35] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross, "Interaction of fluids with deformable solids," *J. Comput. Anim. and Virt. Worlds*, vol. 15, no. 3–4, pp. 159–171, July 2004.
[36] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutré, and M. Gross, "A unified Lagrangian approach to solid-fluid animation," in *Eurographics Symp. on Point-Based Graph.*, 2005.
[37] M. Müller, B. Solenthaler, R. Keiser, and M. Gross, "Particle-based fluid-fluid interaction," in *Proc. of the 2005 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 2005, pp. 237–244.
[38] F. Pighin, J. Cohen, and M. Shah, "Modeling and editing flows using advected radial basis functions," in *Proc. of 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 2004.
[39] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. Whitaker, "Particle–based simulation of fluids," in *Comp. Graph. Forum (Eurographics Proc.)*, vol. 22, no. 3, 2003, pp. 401–410.
[40] R. Fedkiw, J. Stam, and H. Jensen, "Visual simulation of smoke," in *Proc. of ACM SIGGRAPH 2001*, 2001, pp. 15–22.
[41] A. Selle, R. Fedkiw, B.-M. Kim, Y. Liu, and J. Rossignac, "An unconditionally stable maccormack method," *J. Sci. Comput.*, vol. In review. http://graphics.stanford.edu/∼fedkiw/, 2007.
[42] J. Stam, "Stable fluids," in *Proc. of SIGGRAPH 99*, 1999, pp. 121–128.
[43] J. Morris, P. Fox, and Y. Zhu, "Modeling low reynolds number incompressible flows using SPH," *J. Comput. Phys.*, vol. 136, pp. 214–226, 1997.
[44] S. Koshizuka, H. Tamako, and Y. Oka, "A particle method for incompressible viscous flows with fluid fragmentation," *Comput. Fluid Dyn. J*, 1995.
[45] S. Koshizuka, A. Nobe, and Y. Oka, "Numerical analysis of breaking waves using the moving particle semi-implicit method," *Int. J. Num. Meth. in Fluids*, vol. 26, pp. 751–769, 1998.
[46] H. Yoon, S. Koshizuka, and Y. Oka, "A particle-gridless hybrid method for incompressible flows," *International Journal for Numerical Methods in Fluids*, vol. 30, pp. 407–424, 1999.
[47] S. Cummins and M. Rudman, "An SPH projection method," *J. Comput. Phys.*, vol. 152, no. 2, pp. 584–607, 1999.
[48] F. Colina, R. Egli, and F. Lin, "Computing a null divergence velocity field using smoothed particle hydrodynamics," *J. Comput. Phys.*, vol. 217, pp. 680–692, 2006.
[49] J. Liu, S. Koshizuka, and Y. Oka, "A hybrid particle-mesh method for viscous, incompressible, multiphase flows," *J. Comput. Phys.*, vol. 202, no. 1, pp. 65–93, 2005.
[50] B. Feldman, J. O'Brien, and O. Arikan, "Animating suspended particle explosions," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 22, no. 3, pp. 708–715, 2003.
[51] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw, "Multiple interacting liquids," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 25, no. 3, pp. 812–819, 2006.
[52] V. Mihalef, D. Metaxas, and M. Sussman, "Animation and control of breaking waves," in *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 2004, pp. 315–324.