

A Practical Method for High-Resolution Embedded Liquid Surfaces

Ryan Goldade¹, Christopher Batty¹, and Chris Wojtan²

¹University of Waterloo

²IST Austria

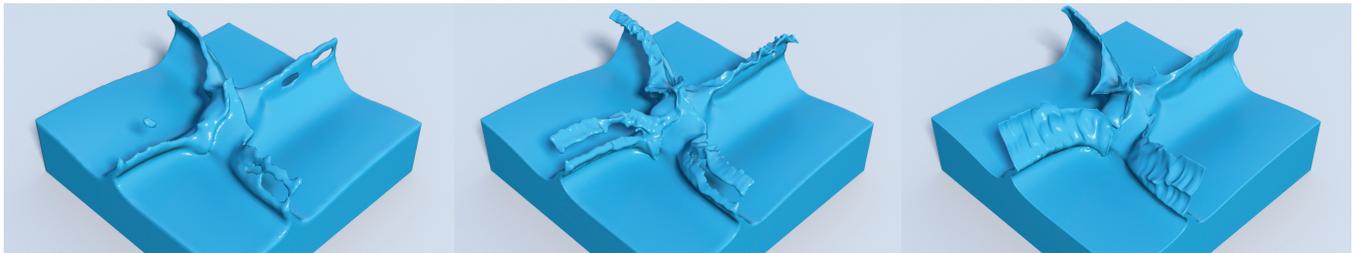


Figure 1: *Left:* A 4-Way Dam Break simulation with 128^3 grid resolution for both physics and level set surface exhibits limited detail in the splash (sim time of 41m44s). *Middle:* The same simulation at 256^3 resolution exhibits greater detail, but is 40 times more expensive (sim time of 1654m13s). *Right:* Our method is better able to preserve thin features and achieves much higher surface detail using a 128^3 simulation grid with a 1024^3 narrow band level set; at the same time, it is unconditionally stable, free from artifacts plaguing earlier embedded methods, and has lower computational cost (sim time of 1191m53s).

Abstract

Combining high-resolution level set surface tracking with lower resolution physics is an inexpensive method for achieving highly detailed liquid animations. Unfortunately, the inherent resolution mismatch introduces several types of disturbing visual artifacts. We identify the primary sources of these artifacts and present simple, efficient, and practical solutions to address them. First, we propose an unconditionally stable filtering method that selectively removes sub-grid surface artifacts not seen by the fluid physics, while preserving fine detail in dynamic splashing regions. It provides comparable results to recent error-correction techniques at lower cost, without substepping, and with better scaling behavior. Second, we show how a modified narrow-band scheme can ensure accurate free surface boundary conditions in the presence of large resolution mismatches. Our scheme preserves the efficiency of the narrow-band methodology, while eliminating objectionable stairstep artifacts observed in prior work. Third, we demonstrate that the use of linear interpolation of velocity during advection of the high-resolution level set surface is responsible for visible grid-aligned kinks; we therefore advocate higher-order velocity interpolation, and show that it dramatically reduces this artifact. While these three contributions are orthogonal, our results demonstrate that taken together they efficiently address the dominant sources of visual artifacts arising with high-resolution embedded liquid surfaces; the proposed approach offers improved visual quality, a straightforward implementation, and substantially greater scalability than competing methods.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Liquid simulations can be computationally costly, and they scale poorly as their spatial resolution increases. On the other hand, their visual appearance is dictated entirely by the motion of the surface, and the expense of surface tracking scales much more efficiently than the expense of solving the Navier-Stokes equations over a

volume. Many practitioners and researchers therefore choose to resolve the surface evolution at a significantly higher resolution than the underlying physics, in order to maximize the apparent detail at minimal cost. Unfortunately, this *embedded surface* approach gives rise to several artifacts; the most objectionable of these is that the surface tracker has more degrees of freedom than the physics is able

to control, resulting in unphysical behavior and lingering noise on the liquid surface. Previous work has identified and proposed one solution to this issue [BHW13], but leaves additional error sources unaddressed. Moreover, stability restrictions inherent in previous work make it inefficient for large resolution differences.

Our paper catalogues the dominant errors that occur when combining a low resolution grid-based fluid simulation with a high-resolution, narrow-band level set surface tracker. We provide explanations for why these errors occur in the first place, and we introduce efficient solutions for each of them. In particular, we investigate the following challenges:

- stable and scalable sub-grid error correction;
- efficient narrow band management for smooth free surfaces; and
- treatment of grid artifacts during level set advection.

Our approach is the first to tackle all three of these problems, creating an efficient and practical method for animating highly detailed liquid flows. We demonstrate the effectiveness of our approach on large-scale fluid animations with surface-to-simulation resolution ratios of up to 32:1 and surface resolutions up to 2048³.

2. Problem Statement

We identify three main types of errors that result from combining a standard grid-based Eulerian Navier-Stokes solver [EMF02] with a much higher resolution level set. Broadly speaking, these artifacts occur because surface tracking algorithms were not originally designed to be driven by low-resolution physics. Below, we single out each of these problems, discuss the mechanisms that cause them, and give an overview of our proposed solutions.

2.1. Unresolved High-Frequency Surface Variation

Liquid simulation involves a delicate two-way coupling between the fluid physics and the surface geometry: the physics dictate how the surface should move by providing a velocity field through which the surface is passively advected, and the surface provides boundary conditions during the pressure projection step of the fluid solver. (For brevity, we will typically refer to the level set surface tracker as the *surface*, and the grid-based Navier-Stokes solver as the *simulation*.) When the surface is discretized at a higher resolution than the simulation, surface advection becomes *under-constrained*.

In a fully-resolved fluid simulation, the pressure is fixed to a boundary condition at the surface [Bri08]. In effect, the fluid simulation can only extract as much information about the surface as there are pressure samples in the simulation grid. When surface details appear at a higher resolution than the fluid simulation, they are effectively invisible to the simulation and are not naturally eliminated or evolved (e.g., by gravity waves). Visually, because these high-frequency surface variations remain locally nearly rigid and do not obey the proper fluid physics, they can exhibit an overly viscous or viscoelastic appearance. This artifact is illustrated in Figure 2. The fluid simulation can also completely miss large gaps in the liquid surface, as illustrated in Figure 3.

Bojsen-Hansen and Wojtan [BHW13] selectively removed these unphysical artifacts through a variational surface smoothing routine

based on an error metric that measures the consistency of surface normals and fluid pressure gradients. Unfortunately, this smoothing routine has stability constraints imposed on it by the level set grid size and the vorticity strength. We observed experimentally that their variational smoothing requires a timestep restriction that is approximately linear in the grid resolution: $\Delta t < \mathcal{O}(\Delta x_{hi})$, for smoothing time step Δt and *level set* grid spacing Δx_{hi} . We also found that the timestep was restricted by $\Delta t < \mathcal{O}(\sqrt{\alpha})$, where α is their vorticity strength parameter controlling the degree of smoothing. As a result of these restrictions, finding the appropriate balance between stability and the desired smoothing behaviour is a tedious process, and the method becomes impractical for large surface resolutions.

We describe a new method for judiciously eliminating unresolved features that is both easy to implement and unconditionally stable. This key contribution of our work is explained in Section 4.

2.2. Reversion to Low-Order Boundary Conditions

The use of narrow-band data structures (e.g., OpenVDB) and algorithms is crucial for efficient high-resolution level set-based surface tracking. However, if care is not taken at large surface-to-simulation ratios, the width of the narrow band can approach or fall below that of a single simulation grid cell. As a result, the centers of simulation grid cells on either side of a surface can lie entirely outside the valid narrow band region of the level set. Evaluating signed distance values at these locations will typically return highly inaccurate estimates with possibly only the sign being correct. Because these values are used to determine the sub-grid position of the surface when applying standard ghost fluid free surface boundary conditions, this can easily drop the accuracy of the boundary conditions back to first order and reintroduce objectionable stairstep artifacts. This artifact is illustrated in Figure 5 (top), and can be observed in prior work (e.g., [BHW13]). In Section 5, we describe a solution that nevertheless maintains the efficiency of the narrow band approach.

2.3. Non-Smooth Surface Patches

Even in the absence of the above two issues, we may still observe grid artifacts whose spacing corresponds to that of the low-resolution simulation grid lines, as illustrated in Figure 8. Because the velocity field produced by the fluid simulation contains far fewer degrees of freedom than the surface, the use of standard piecewise linear interpolation of the velocity field when advecting the surface will create such kinks or faceting along the grid lines, *even when using higher order level set advection schemes* (e.g. WENO5). We can address this remaining problem by providing the advection scheme with a velocity field generated using higher order interpolation, as described in section 6.

2.4. Summary

A summary of the entire simulation loop, including our contributions and associated sections, is presented in algorithm 1. We alternate steps of *Surface Tracker* and *Fluid Simulation*.

3. Related Work

Embedding Level Sets. We focus on the use of high-resolution level set methods [OF02] coupled to Eulerian liquid simulations on

Algorithm 1 Timestep loop

```

1: procedure SURFACE TRACKER
2:   Error correction (§4)
3:   Advect surface (§6)
4:   Sample surface for boundary conditions (§5)
5: end procedure
6: procedure FLUID SIMULATION
7:   Advect velocity
8:   Add external forces
9:   Pressure projection
10:  Extrapolate pressure gradient
11: end procedure

```

regular grids (e.g. [EMF02]). The first authors to discuss pairing a double-resolution level set with a regular grid liquid simulation were Goktekin et al. [GBO04] who argued that it better preserves details and volume. Bargteil et al. [BGOS06] combined an octree-based implicit surface with a regular grid fluid simulation to capture thin splashes, and discussed some artifacts arising from the resolution mismatch. Kim et al. [KSK09] proposed a “liquid-biased filtering” scheme to ensure that high-resolution details in the level set are always identified as active liquid regions even when they fall below the simulation resolution. In essence, this is done by thickening the thin liquid region by half of a simulation grid cell width. Although this changes the surface’s effective location for the fluid simulation, it largely preserves the attractive results achieved with accurate ghost-fluid-based liquid simulators [ENGF03]. Heo et al. [HK10] used a pseudo-spectral level set method with additional sub-grid quadrature points, in order to capture fine details on a relatively coarse grid. Most recently, Bojsen-Hansen et al. [BHW13] derived a technique to identify and resolve certain errors caused by resolution mismatches, by applying either a direct smoothing operation or by dynamically evolving surface details as waves. This method relies on explicit integration of a particular differential equation, which requires satisfying a fairly stringent, but generally unknown, stability restriction. We focus on the smoothing problem, and use this work as our primary point of comparison.

A contributing factor in the ongoing success of high-resolution level set schemes is the use of narrow-band data structures and algorithms, whether based on octrees [LGF04, BGOS06], run-length encoding [HNB*06], or nested block-grids [Bri03, Mus13]. Our work makes use of the OpenVDB library [MLJ*13] to support narrow-band level sets.

Alternatives to Level Sets. Mismatches between geometry and physics can also be an issue for Lagrangian surface tracking methods, whether using triangle meshes [WTGT10, BBB10] or particles [YWTY12]. Wojtan et al. adjust the surface mesh to match the fluid grid in complex regions [WTGT10], while Brochu et al. modify the simulation mesh to match the surface mesh [BBB10]. Yu et al. periodically project the surface mesh to match an implicit surface defined by the fluid particles [YWTY12].

Adding Sub-Grid Physics. Various authors have proposed to layer sub-grid physical models on top of the liquid surface. Thuerey et al. [TWGT10] proposed a wave model to capture high-resolution surface tension effects on explicit triangle mesh surfaces. Kim et

al. [KTT13] used a closest-point methodology for performing wave simulation on level sets. Mercier et al. [MBT*15] recently showed how a sub-grid wave model can be added atop particle-based liquid simulations. Since we aim to remove sub-grid errors in the *underlying surface*, we believe the tools we present could be beneficially applied alongside a variety of possible sub-grid physical models.

High-Resolution Physics A natural way to avoid the issues posed by high-resolution surfaces is to simply combine them with a matching high resolution or high order physics discretization, possibly exploiting adaptivity (e.g., [LGF04, BBB10, ATW13, EB14]). However, this typically entails substantial additional cost and implementation complexity, and in general it remains more expensive to scale the volume than the surface.

4. Stable Filtering of Sub-Grid Noise

The choice to couple a low resolution simulation and a high resolution surface implies that many discrete surface cells map onto a single cell in the fluid simulation. As a result of this many-to-one coupling, the simulation cannot correctly respond to surface details at a higher sampling rate than that of the pressure samples on the grid. This discrepancy in sampling rates allows high frequency details on the surface to persist as unphysical, noisy errors. Figure 2 (top right) exemplifies the persistent errors where the expected result is a flat surface. Additionally, Figure 3 (top right) demonstrates an

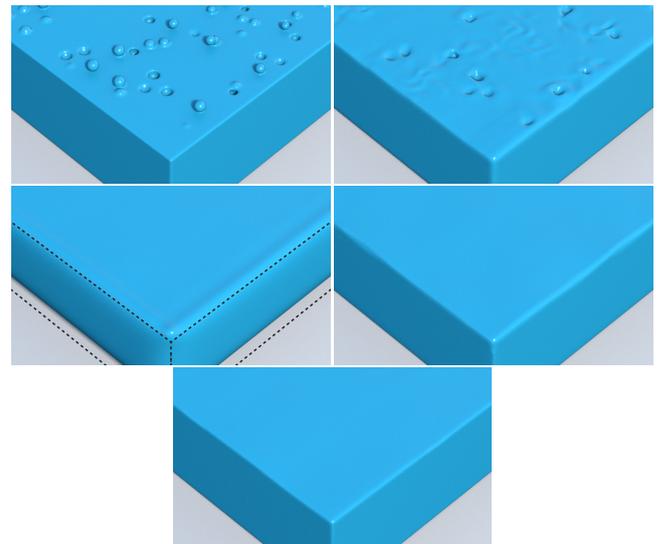


Figure 2: A Swiss Cheese setup, a high resolution surface with small droplets and divots (top left), creates persistent errors that the fluid simulation cannot correct (top right). Laplacian smoothing of the surface removes errors (middle left), but causes rounded corners and severe volume loss (indicated by the black dashed box). The error correction of Bojsen-Hansen and Wojtan [BHW13] (middle right) removes the errors and preserves sharp corners. Our solution (bottom) achieves similar results but with a much simpler implementation.

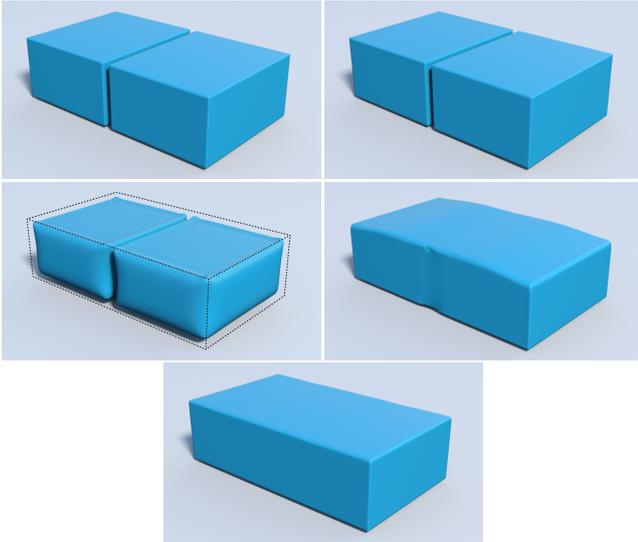


Figure 3: A Standing Dam setup contains a gap smaller than a fluid simulation grid cell (top left). The fluid simulation does not close the gap (top right). Laplacian smoothing causes a large degree of volume loss without actually closing the gap (middle left). The error correction of Bojsen-Hansen and Wojtan (middle right) removes the error without severe volume loss. Again our simpler solution (bottom) achieves comparable results.

extreme case where a long narrow gap between surfaces is not seen by the simulation and remains stationary.

A naïve approach to remove these errors is simple Laplacian surface smoothing. Unfortunately, this typically removes desirable surface features and thin splashes along with the errors. Figure 4 (middle) presents an example where surface smoothing destroys the scale features on a falling liquid dragon model after just a few simulation frames. Furthermore, regions where a free surface comes in contact with the solid boundary are erroneously rounded out due to smoothing of sharp corners (middle left of Figures 2 and 3).

Bojsen-Hansen and Wojtan [BHW13] recently presented a variational smoothing technique that minimizes the deviation between the pressure gradient and the surface normal, observing that alignment of these vectors implies that the surface is free from sub-grid errors. A key feature of this method is that the rate of smoothing applied at any given point depends on the magnitude of the pressure gradient, $\|\nabla p\|$. In nearly hydrostatic scenarios, such as a still pool, large vertical pressure gradients induce stronger smoothing and the surface settles to the desired flat configuration. In ballistic scenarios, such as falling liquid bodies and splashes, the pressure gradient is often near zero; as a result, smoothing is not activated and the surface details are preserved. This allows their method to selectively eliminate objectionable errors while maintaining thin sheets and splashes.

While our smoothing mechanism differs, we borrow the idea of using the magnitude of the pressure gradient as an indicator of the need for smoothing. Our error correction solution is able

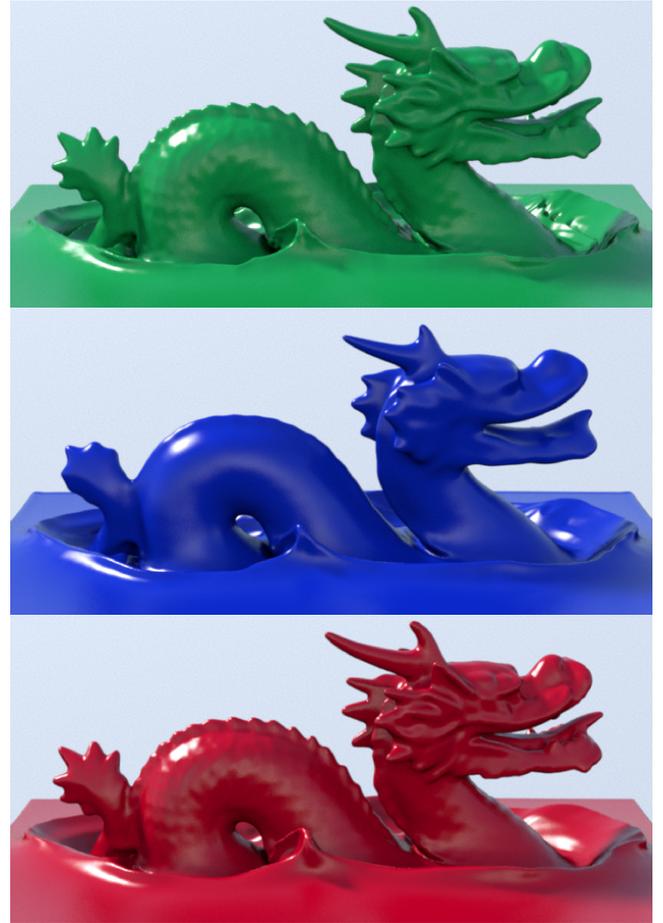


Figure 4: The high resolution details on a Falling Dragon (top) are quickly removed by uniform Laplacian smoothing (middle). Our selective-smoothing approach preserves such detailed features during ballistic motion (bottom).

to remove surface errors (see Figures 2 and 3 (both bottom)) and preserve details on ballistic volumes (see Figure 4 (bottom)). Figure 5 demonstrates that our strategic smoothing method maintains the same dynamic behavior as the original simulation while removing persistent errors on the surface.

4.1. Method

At each timestep we reduce high frequency errors that the fluid simulation cannot address. Our basic approach is to smooth the surface at a spatially-dependent rate proportional to the magnitude of the pressure gradient. To achieve this smoothing efficiently, we present an unconditionally stable multi-scale smoothing algorithm analogous to mip-mapping for anti-aliasing texture maps. We first generate a low-resolution smoothed version of the surface, and then we blend between this smooth surface and the original high resolution surface in a spatially-dependent manner, using the magnitude of the pressure gradient as our interpolation weight.

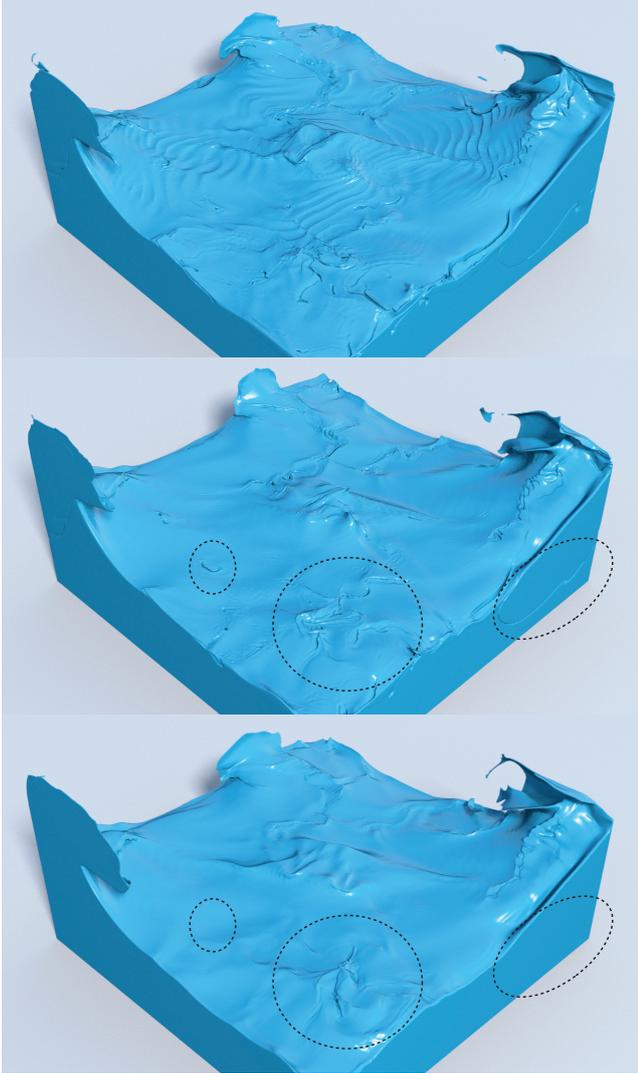


Figure 5: *Top: Stair step artifacts occur when erroneous values outside the level set’s narrow band are used to determine ghost fluid boundary conditions. Middle: Our narrow-band method (§5) efficiently ensures a proper ghost fluid treatment. Bottom: We also remove surface artifacts (highlighted by dashed ellipses) with our filtering method (§4) while preserving the overall dynamics of the simulation.*

We obtain a low-resolution representation of the surface by low-pass filtering and downsampling. The filtered surface has the same effective resolution as the fluid simulation, so features on this surface can always be handled by the underlying physics. By using the pressure gradient to blend between this low-resolution surface and the original one, unphysical surface errors will be eliminated gradually over several frames, and desirable high-resolution features will be preserved. The user can adjust how rapidly this process occurs by scaling the blend weight (α in §4.4), which provides a degree of artistic control.

4.2. Gaussian Pyramid Filtering

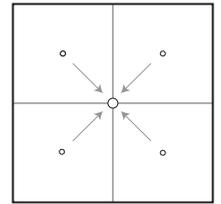
We filter the high resolution surface progressively through a Gaussian pyramid approach. We apply a Gaussian filter to the level set surface with a 27-point stencil that is convolved over the surface to remove high frequency information. We define the weights at each stencil point by the 3D Gaussian function

$$f(x_{i,j,k}) = e^{-\frac{(i-i_0)^2 + (j-j_0)^2 + (k-k_0)^2}{2\sigma^2}} \quad (1)$$

where i_0, j_0, k_0 components are the integer indices of the center grid cell and i, j, k are the integer indices of the remaining 26 points on the stencil. Integer offsets allow us to filter the level set relative to the sampling rate of the current grid. We found that using $\sigma = \frac{2}{\pi}$ as our variance for (1) effectively attenuates high frequencies that could cause aliasing. Finally, we normalize our stencil weights so they sum to 1 and convolve the filter over the level set grid.

After the filter pass, a new surface is created by downsampling onto a level set grid one octave lower (i.e., smaller by a factor of two in each dimension). Similar to decimation in signal processing, we alternate between filtering and downsampling to progressively remove aliasing artifacts that would appear from directly downsampling to the lowest resolution.

Since level set data is stored at grid cell centers, these sample points do not naturally coincide across adjacent levels of the hierarchy. The downsampling step therefore interpolates from the surrounding eight higher resolution cell centers onto the lower resolution cell center (as illustrated in 2D in the inset figure). We build this pyramid only up to one resolution octave higher than the fluid simulation, and perform a final filter pass on this surface. (After this final filtering step the appropriate higher frequencies in the surface have already been treated; halting at this level simply avoids unnecessary downsampling/interpolation and memory allocation for the last level grid.)



The resulting low resolution surface no longer contains the high resolution surface errors; any remaining surface features can be handled directly by the simulation. However, a naive application of this filtering method will yield artifacts similar to Laplacian smoothing: rounding of corners where the free surface meets the solid boundary and excessive loss of detail in dynamic splashes. We describe below how to circumvent these issues.

4.3. Boundary Extension

Applying the low-pass filter at fluid-solid-air triple-points will round out the sharp features present in the high resolution level set, as illustrated in Figure 6 (top). To remove errors on the high resolution surface without artificial rounding near solids, we extrude the level set surface outwards into the solid boundary at the beginning of each level in the Gaussian pyramid filtering step. After the extrusion, we apply the Gaussian filter but only on regions of the fluid surface that are outside the solid boundary, as shown in Figure 6 (bottom). This method addresses the rounding issue by preventing the filter from drawing data from the problematic corner features.

We perform the surface level set extrusion along the solid boundary normal, using a technique loosely analogous to semi-Lagrangian level set advection. Starting from the position of a given cell center inside the solid, we find the closest point on the solid boundary by iteratively stepping in the normal direction until it is a small tolerance β just outside of the solid (e.g., as in [LFO05] §2.2.2). We then interpolate the *liquid* level set value at that point, and assign it to the liquid level set sample at our starting point inside the solid. We used $\beta = \Delta x_{hi}$ for the grid-dependent β parameter in our examples (where Δx_{hi} is the high resolution grid spacing). This operation can be easily performed in parallel.

Once the surface is extended in this fashion, applying the filter to the surface near the solid boundary will no longer induce rounding artifacts. The extended information *inside* the solid boundary is only created temporarily and then discarded each time step, so we do not bother filtering this region. Figure 6 demonstrates the difference between directly applying the filter to the surface and extruding outwards first.

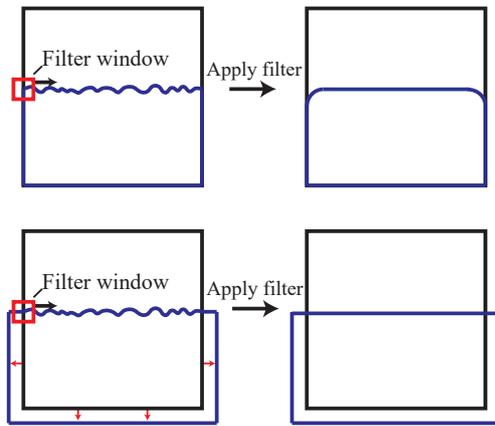


Figure 6: Filtering without extrusion artificially rounds the interface near solids (top). With surface extension, the filtered result is noise-free without rounded corners (bottom).

An advantage of this approach is that we can uniformly apply a single filtering method that is particularly robust near solid boundaries. By contrast, the method of Bojsen-Hansen et al. [BHW13] exhibits discontinuous behavior near solids; they apply separate error-minimizations near the free surface and the solid boundary, and they use a (noise-sensitive) distance-based threshold to switch between them. Furthermore, while their approach drives the surface normals to conform to the solid boundary normals, the error-correction slows down as the alignment improves, causing unphysical artifacts near boundaries to persist longer than desired.

4.4. Level Set Interpolation

The above process filters out all features above the Nyquist frequency of the simulation grid, leaving us with a smooth low-resolution surface. Unfortunately, uniform filtering will also quickly remove desirable small details and splashes. In order to remove the errors but preserve details in interesting dynamic regions, we

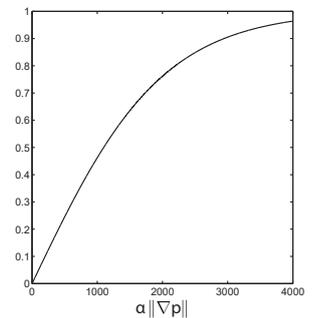
strategically blend regions of the filtered and high resolution surface together in a spatially-dependent manner. We build a final level set surface at the high resolution by interpolating the signed distance value of each voxel between the original high resolution surface and the filtered low resolution surface.

As noted earlier, we will use the magnitude of the pressure gradient to dictate the degree of interpolation, inspired by the energy minimization method of Bojsen-Hansen et al. [BHW13]. Our task is to map the domain of the pressure gradient’s magnitude, $[0, \infty]$, to the range $[0, 1]$ for linear blending. Many functions can perform this mapping; we chose the modified sigmoid function

$$s(\|\nabla p\|) = \frac{2}{1 + e^{-\alpha\|\nabla p\|}} - 1 \quad (2)$$

where α is a tuning parameter to artistically control the smoothing strength.

The inset figure visualizes this function. A very large α value would give emphasis to the filtered surface and remove the surface errors immediately. Since our filter is not truly ideal (in the sense of attenuating strictly those frequencies above a given threshold), a pathologically large α value can create noticeable volume loss. A very small α value would favour the original surface, such that surface errors would remain mostly present. We found $\alpha = 0.001$ to give visually pleasing results; all our examples use this parameter choice.



The new error-reduced surface is constructed using linear interpolation

$$\phi_{new} = (1 - s)\phi_{original} + s\phi_{filtered}$$

where $\phi_{original}$, $\phi_{filtered}$, and ϕ_{new} , are the original, filtered, and updated level set functions respectively, and s is the interpolation weight defined in (2).

Because the level set narrow band contains voxels outside of the fluid, we have to extrapolate the current simulation’s pressure gradient into the air (as indicated by the last *fluid simulation* step in Algorithm 1). We extrapolate the pressure gradient in the low-resolution simulation with a fast marching method. This is analogous to extrapolating the velocity field from the fluid into the air. We use linear interpolation from the newly extrapolated, low-resolution pressure gradient to provide values for (2) at the high resolution grid positions. As with the filtering step, the extended surface region inside the solid boundary is unimportant, so for any ϕ_{new} voxels located inside the boundary we simply assign $\phi_{new} = \phi_{original}$. Finally, we redistance the new level set.

4.5. Surface-Boundary Intersection

After blending is performed the resulting level set, ϕ_{new} , will often extend into the solid boundary. We trim off this excess liquid surface

region by applying a CSG difference operation [MBWB02] against the solid level set, before redistancing one last time. This yields our final surface with sub-grid errors appropriately reduced.

5. Narrow Band Free Surfaces

5.1. Ghost Fluid

High resolution level sets require an efficient implementation to reduce memory and computational costs. A narrow band data structure ensures that only voxels near the surface are stored in memory, and dramatically reduces the computational cost of filtering, redistancing, and advecting the level set. However, a subtle issue that can occur in this setting is a failure to properly apply free surface boundary conditions during the pressure projection, using the *ghost fluid* method.

The ghost fluid method enforces the Dirichlet boundary condition such that pressure is zero on the free surface [ENGF03, CS70], by accounting for the sub-grid position of the interface. Figure 7a illustrates the free surface between the two pressure samples on the grid, where p_L is the pressure sample inside the liquid and p_G is the *ghost pressure* in the ambient air. The zero pressure condition at the free surface is set implicitly by

$$p_G = \frac{\phi_G}{\phi_L} p_L, \quad (3)$$

where ϕ_G and ϕ_L are the level set signed distance values at the positions of p_G and p_L respectively. This amounts to a linear extrapolation for p_G based on p_L inside and $p = 0$ on the surface. Use of the simpler first order condition, $p_G = 0$, is well-known to lead to objectionable stairstep artifacts.

5.2. A Modified Narrow Band Approach

A narrow band level set initializes only voxels that are within a small distance of the implicit surface. A surface that is at the *same* resolution as the fluid simulation will contain enough voxels in the narrow band to properly assign ϕ_L and ϕ_G values to the fluid simulation (note that ϕ_L and ϕ_G are only used near the free surface). However, a higher resolution surface might not contain a wide enough narrow band to ensure that active voxels are present at the required low resolution ϕ_L and ϕ_G sample locations. Figure 7b visualizes this case. The narrow band (small solid gray cells) is not wide enough to cover both sample locations so (3) relies on erroneous values. Failure to properly apply this boundary condition creates dramatic visual artifacts on the surface, as observed in Figure 5 and prior work [BHW13].

An obvious solution is to always use a wide-enough narrow band to guarantee valid signed distance values for (3). However, this means that the width of the narrow band is no longer a small fixed constant, but instead depends on the *ratio* between surface and simulation resolutions. For large ratios this results in a major performance penalty during level set operations (i.e., advection, redistancing, filtering) which would otherwise be relatively cheap.

Instead of carrying an expensive wide band throughout, we temporarily dilate the narrow band region outwards (i.e., add extra layers of voxels) before the pressure projection step and (conceptually) erode it back down afterwards. Voxel dilation provides the

fluid simulation with the active voxels necessary to properly sample the signed distance values in (3). The subsequent erosion removes the extra cells before any further expensive level set operations are performed. This is illustrated with the dashed squares in Figure 7b.

While comparably simple to the naïve approach, dilation and erosion can be performed very efficiently [MLJ*13]; this allows us to restrict the use of a widened narrow band to just a handful of operations, and in particular avoid the expense of *advection* on a widened band.

To ensure that the level set covers all the necessary sample locations in the simulation, we widen the narrow band width by $\lceil \Delta x_{lo} / \Delta x_{hi} \rceil$ voxels, where Δx_{hi} and Δx_{lo} are the grid spacings of the level set and fluid simulation, respectively.

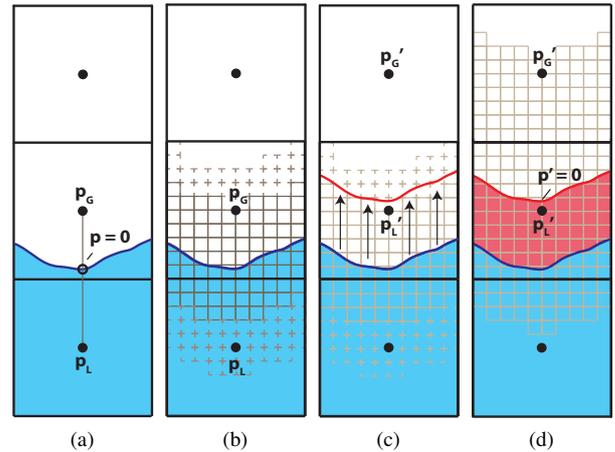


Figure 7: The ghost fluid method assigns zero pressure at the free surface (a) by (3), based on the signed distance values extracted from the liquid level set. If the narrow band is not wide enough (solid grid in (b)), it must be dilated outwards to provide accurate signed distance values at the pressure samples (dashed grid). Liquid-biased filtering shifts the free surface outwards (red offset in (c)), however the dashed grid does not cover p'_G . A second adjustment of the narrow band ensures accurate signed distance values for both p'_L and p'_G sample locations (d).

5.3. Liquid-biased Filtering

Kim et al. [KSK09] noticed that high resolution surfaces might contain thin regions that lie between pressure sample locations in the simulation. In this case, both samples are erroneously labelled as *air* and the liquid region does not contribute to the physical simulation. The resulting velocity field can lead to substantial volume loss and unrealistic motion. The authors handle this problem by (temporarily) shifting the level set surface outwards into the *air* region (presented in Figure 7c), so that all liquid elements are sufficiently large as to be visible on the coarse grid.

We adopt this same strategy. In our examples, we shift the surface by half the grid size of the simulation by subtracting $\Delta x_{lo}/2$ from the signed distance values, ϕ . However, moving the surface in this way introduces a further issue in the narrow band setting. Referring again to Figure 7c, the original *ghost pressure* sample point, p_G , is now inside the liquid (relabelled as p'_L) and the new *ghost pressure*

sample point, p'_G , is outside of the level set's narrow band. To correct this we must adjust the narrow band once again, in the same manner as §5.2. Figure 7d visualizes this final step, with valid signed distance values at the new pressure sample locations, p'_L and p'_G .

To summarize, we first make a copy of the surface, dilate the narrow band outwards as in Figure 7b, and assign distance values to any new cells via redistancing. This ensures that liquid-biased filtering doesn't cause the surface to shift outside the active narrow band. We then offset the surface via liquid-biased filtering, and apply voxel dilation/erosion to balance the narrow band around the new surface position (again assigning any new cells by redistancing) while ensuring that the necessary pressure locations are properly contained within the widened band. At this point, we sample the level set values to apply ghost fluid. Because we have no need for this shifted surface after establishing boundary conditions, we release this level set from memory thus avoiding the need to actually erode it back down in practice.

6. High Order Velocity Interpolation

Another common artifact induced by resolution mismatches between surface and simulation grids is the creation of grid-aligned kinks on the surface, shown in Figure 8. Perhaps counterintuitively, this can arise even when using certain apparently higher order schemes for both velocity advection and level set advection steps (e.g. MacCormack advection [SFK*08], WENO5 advection [OF02], etc.) The culprit turns out to be the use of piecewise trilinear interpolation on the low-resolution velocity field to provide sub-grid velocities to the high-resolution level set advection scheme.

The inset figure gives an illustrative 1D example of a high resolution surface (black) advected to a new position (red) under linear (top) and cubic (bottom) interpolation of velocity. Although the surface has many more degrees of freedom than the velocity field in both cases, the lack of smoothness in the piecewise linear interpolant manifests in the surface as disturbing kinks. Furthermore, simply using a more advanced level set advection scheme (while relying on linear velocity interpolation) will only serve to exacerbate the artifacts by better preserving the kinks. By instead applying higher order interpolation of velocity, the surface can fully benefit from its additional resolution while remaining smooth. (Naturally, this issue is imperceptible near 1:1 simulation-to-surface ratios, because the level set resolution is insufficient to exhibit the induced kinks.)

We alleviate this effect in our simulation using a cubic interpolation scheme [FSJ01] whenever we sample the velocity field. In our implementation, level set advection was performed using OpenVDB's WENO5-based advection scheme (which is orthogonal to the choice of velocity interpolant, as noted above). Figure 8 demonstrates a simple case with linear and cubic velocity interpolation

during level set advection, where simulation-grid scale artifacts are very noticeable in the linear case.

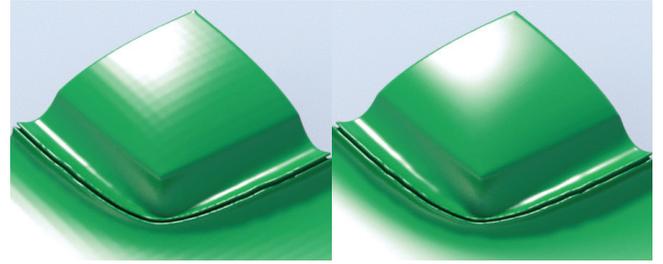


Figure 8: A Corner Box setup with linear velocity interpolation (left) produces “hatching”-like grid artifacts on the surface, which are especially apparent in the specular highlights. Simulation with cubic velocity interpolation (right) gives much smoother surfaces.

This issue is also independent of the surface representation, and therefore cubic interpolation of velocity should be considered for explicit mesh schemes as well.

7. Results

We ran a range of simulations to examine the effectiveness of our proposed techniques; many of these animations are included in our accompanying video. Our simulation is built on top of *mantaflow* [PT13] and rendered with *Mitsuba* [Jak10]. Unless otherwise noted, all of our examples were computed on a 16-core server with 64 GB of RAM. Table 1 provides a breakdown of the computation time for each of the above components. Below we discuss several of these experiments in detail.

7.1. Filtering

High-Resolution Error Removal. We compare the effectiveness of our filter-based sub-grid error correction solution under two simple test conditions used by Bojsen-Hansen and Wojtan [BHW13]: the Swiss Cheese surface in Figure 2 and the Standing Dam in 3. We are able to remove persistent surface errors and achieve similar results to Bojsen-Hansen in both test cases, without the artifacts exhibited by Laplacian smoothing. Both examples are computed with 32^3 simulation grid and a 256^3 level set grid. At such low resolutions, the two approaches are comparable in computational cost, averaging just a few seconds per frame for both test cases. However, as we discuss below, at larger scales our solution is substantially more computationally efficient.

Detail Preservation. We demonstrate in Figures 4 and 5 that our strategic smoothing approach preserves details and thin splashes for surfaces undergoing ballistic motion. The overall dynamic behaviour of the original simulation is maintained while our solution removes persistent surface errors. Figure 4 was computed on a 64^3 simulation grid and a 512^3 level set with the error correction component averaging 8.3s per frame, amounting to about 15% of the total simulation cost. Figure 5 was computed on a 128^3 simulation grid and a 1024^3 level set with the error correction component averaging 10% of the total simulation time. We also applied the error correction method

by Bojsen-Hansen and Wojtan [BHW13] to the initial condition in Figure 5. Our experiments indicated that, given the timestep restriction of the method, we would need to cycle through 10 substeps per frame to maintain stability (compared to a single step for our unconditionally stable method). They did not present results of their smoothing solution at resolutions this high, presumably because of this severe stability restriction. Table 1 demonstrates that our method remains efficient and effective in the highly energetic Stanford Drop scenario, averaging 27.3s for filtering and 328s overall per frame.

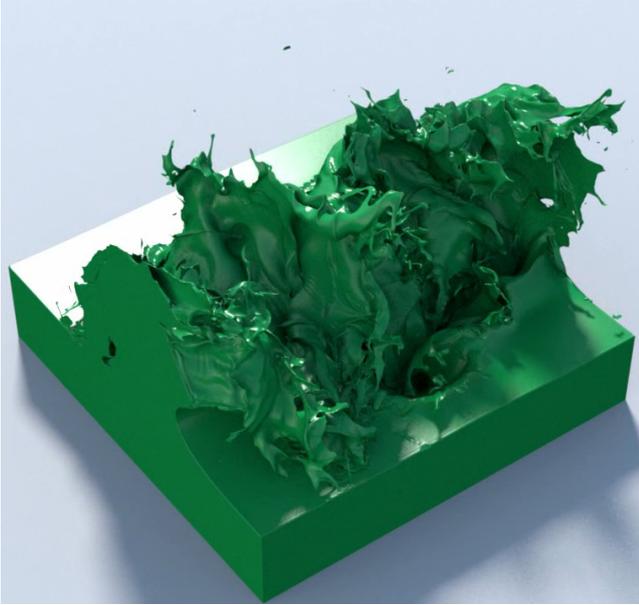


Figure 9: A Stanford Drop simulation that applies our method using a 256^3 simulation grid with a 2048^3 narrow band level set.

Scalability and Stability. We also tested our method at an even higher resolution with a 256^3 simulation grid and a 2048^3 level set, using a more powerful 44-core server (see Figure 9). We observed that the error correction component averaged just 1m25s per frame, compared to 6m57s for the pressure projection and 30m2s for WENO5 level set advection. This illustrates that even as the resolution increases, the overhead of our filtering approach remains relatively small. (Note that the high cost of level set advection here is due to the tremendous resolution of the embedded level set.)

Most of our tests are performed at an 8:1 surface-to-simulation ratio, though this is not a limitation. We ran two additional 4-way dam break simulations with a 1024^3 level set, using 64^3 and 32^3 simulation grids. These represent ratios of 16:1 and 32:1, respectively. While the motion quality naturally decreases at lower simulation resolutions, our proposed techniques remain effective.

Qualitatively, we found that the unconditional stability of our scheme simplified artistic parameter tuning, since we were able to try different smoothing strengths without also having to empirically determine a stable time step for each setting.

7.2. Narrow Band Ghost Fluid

Figure 5 (top) visualizes the stair step artifacts that arise when high resolution surfaces are used without properly accounting for the information needed by the ghost fluid method. The naïve solution of consistently using a wide-enough narrow band introduces a dramatic cost compared to the original simulation, in the range of an additional 130% computation time for a 128^3 simulation with a 1024^3 level set. This is primarily due to the cost of applying advection on many more cells of the high-resolution grid than would otherwise be required. Our approach also correctly applies ghost fluid and eliminates stair steps, as indicated in Figure 5 (middle). However, it introduces only a very minor overhead of 7% computation time for the same 128^3 simulation with a 1024^3 level set.

7.3. High Order Velocity Interpolation

As demonstrated in Figure 8 the use of cubic rather than linear interpolation when advecting embedded level sets largely alleviates problems with visible grid lines. The cubic interpolant entails a relatively small overhead: we observed an approximate 10% increase in computation time for both a 32^3 simulation with a 256^3 level set and a 128^3 simulation with a 1024^3 level set.

8. Discussion and Conclusions

We have presented a set of *practical* solutions for problems that occur when pairing a high resolution level set with a low resolution fluid simulation. Our filter-based error correction offers similar results to previous work but with the added benefits of being more efficient, unconditionally stable, and easier to implement and tune. Our narrow band ghost fluid and smooth velocity interpolation solutions address more subtle problems that were not identified by previous work, but are nevertheless critical for high-quality results.

Because we are using a standard Gaussian filter approach, our low pass filter lacks a *perfect* frequency cutoff. In theory this will mildly attenuate some low frequencies while preserving some high frequencies, after a single filter pass. We did not observe this to be a noticeable issue in our simulations, and therefore defer investigation of a theoretically ideal 3D level set filter to future work. Our filtering approach also gives up the guarantee of an analytically divergence-free velocity field provided by Bojsen-Hansen’s vortex sheet model; however, we believe the added stability and scalability of our approach makes it far more viable in practical scenarios.

Our method of extruding the liquid surface into the solid boundary does not work well with thinner obstacles, because it introduces the possibility of “crosstalk” in which liquid level set surfaces on opposing sides of the solid erroneously interact. Disabling the filtering near thin solids would prevent this problem, but it would also allow some surface errors near the solid to remain. In the future, we would like to investigate a method to implicitly extrude the surface in a “ghost fluid” sense to prevent crosstalk.

A current limitation of our surface embedding approach is the appearance of occasional locking-like effects from topologically disjoint high-resolution surfaces being driven by the same simulation cell [WTGT10]. A possible solution might be the application of topology-aware virtual-node schemes common in embedded FEM

Initial Condition	Sim	Surface	Base	Ghost Fluid Fix	Cubic Interpolation	Filter
Corner Box	32 ³	256 ³	2.2s	0.46s	0.61s	2.0s
Falling Dragon	64 ³	512 ³	26.3s	2.1s	2.4s	8.3s
4-Way Dam Break	128 ³	1024 ³	155s	6.3s	8.0s	21.6s
Stanford Drop	128 ³	1024 ³	293s	7.7s	36.0s	27.3s

Table 1: Average computation cost per timestep for several simulations presented above. Base refers to the basic embedded simulation; the remaining three columns give the incremental cost of each of our three proposed features. We use a fixed time-step (and use one time-step per frame) for our simulation, although OpenVDB's implementation of level set advection may perform substepping internally.

methods, as recently explored by Edwards and Bridson [EB14] for their discontinuous Galerkin cut-cell fluid scheme.

We have sought to extend the limits of what can be achieved in practice with high-resolution embedded level set schemes by identifying and treating a trio of issues arising in this setting. A question this poses is whether these ideas may be extended to similarly improve triangle mesh-based and particle-based liquid simulators.

Acknowledgments

This research was supported by NSERC (RGPIN-04360-2014), ERC (638176), and IST Austria. We are thankful to Morten Bojsen-Hansen for providing us with his code and answering questions, and Side Effects Software for their generous software donation.

References

- [ATW13] ANDO R., THUERNEY N., WOJTAN C.: Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph. (SIGGRAPH)* 32, 4 (2013), 103. 3
- [BBB10] BROCHU T., BATTY C., BRIDSON R.: Matching fluid simulation elements to surface geometry and topology. *ACM Trans. Graph. (SIGGRAPH)* 29, 4 (2010), 47. 3
- [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 (Jan. 2006), 19–38. 3
- [BHW13] BOJSEN-HANSEN M., WOJTAN C.: Liquid surface tracking with error compensation. *ACM Trans. Graph.* 32, 4 (2013), 68. 2, 3, 4, 6, 7, 8, 9
- [Bri03] BRIDSON R.: *Computational aspects of dynamic surfaces*. PhD thesis, Stanford University, 2003. 3
- [Bri08] BRIDSON R.: *Fluid Simulation for Computer Graphics*. AK Peters/CRC Press, 2008. 2
- [CS70] CHAN R. K.-C., STREET R. L.: A computer study of finite-amplitude water waves. *Journal of Computational Physics* 6, 1 (1970), 68–94. 7
- [EB14] EDWARDS E., BRIDSON R.: Detailed water with coarse grids: Combining surface meshes and adaptive discontinuous Galerkin. *ACM Trans. Graph. (SIGGRAPH)* 33, 4 (2014), 136:1–136:9. 3, 10
- [EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (2002), 736–744. 2, 3
- [ENG03] ENRIGHT D., NGUYEN D., GIBOU F., FEDKIW R.: Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *Proceedings of the 4th ASME-JSME Joint Fluids Engineering Conference* (2003), ASME, pp. 337–342. 3, 7
- [FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *Proceedings of SIGGRAPH 2001* (2001), Fiume E., (Ed.), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 15–22. 8
- [GBO04] GOKTEKIN T. G., BARGTEIL A. W., O'BRIEN J. F.: A method for animating viscoelastic fluids. *ACM Trans. Graph. (SIGGRAPH)* 23, 3 (Aug. 2004), 463–468. 3
- [HK10] HEO N., KO H.-S.: Detail-preserving fully-Eulerian interface tracking framework. *ACM Trans. Graph. (SIGGRAPH Asia)* 29, 6 (2010), 176:1—176:8. 3
- [HNB*06] HOUSTON B., NIELSEN M. B., BATTY C., NILSSON O., MUSETH K.: Hierarchical RLE level set: A compact and versatile deformable surface representation. *ACM Trans. Graph.* 25, 1 (Jan. 2006), 151–175. 3
- [Jak10] JAKOB W.: Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>. 8
- [KSK09] KIM D., SONG O.-Y., KO H.-S.: Stretching and wiggling liquids. *ACM Transactions on Graphics* 28, 5 (2009), 120. 3, 7
- [KTT13] KIM T., TESSENDORF J., THUERNEY N.: Closest point turbulence for liquid surfaces. *ACM Trans. Graph.* 32, 2 (2013), 15. 3
- [LFO05] LOSASSO F., FEDKIW R., OSHER S.: Spatially adaptive techniques for level set methods and incompressible flow. *Computers & Fluids* 35, 10 (2005), 995–1010. 6
- [LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH)* 23, 3 (Aug. 2004), 457–462. 3
- [MBT*15] MERCIER O., BEAUCHEMIN C., THUERNEY N., KIM T., NOWROUZSAHRAI D.: Surface turbulence for particle-based liquid simulations. *ACM Trans. Graph. (SIGGRAPH Asia)* 34, 6 (2015). 3
- [MBWB02] MUSETH K., BREEN D. E., WHITAKER R. T., BARR A. H.: Level set surface editing operators. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (2002), 330–338. 7
- [MLJ*13] MUSETH K., LAIT J., JOHANSON J., BUDSBERG J., HENDERSON R., ALDEN M., CUCKA P., HILL D., PEARCE A.: Openvdb: An open-source data structure and toolkit for high-resolution volumes. In *ACM SIGGRAPH 2013 Courses* (New York, NY, USA, 2013), SIGGRAPH '13, ACM, ACM, pp. 19:1–19:1. 3, 7
- [Mus13] MUSETH K.: VDB: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.* 32, 3 (2013), 27. 3
- [OF02] OSHER S., FEDKIW R.: *Level set methods and dynamic implicit surfaces*. Springer, New York, 2002. 2, 8
- [PT13] PFAFF T., THUERNEY N.: mantaflow, 2013. <http://mantaflow.com/>. 8
- [SFK*08] SELLE A., FEDKIW R., KIM B., LIU Y., ROSSIGNAC J.: An unconditionally stable maccormack method. *J. Sci. Comput.* 35, 2-3 (June 2008), 350–371. 8
- [TWGT10] THUERNEY N., WOJTAN C., GROSS M., TURK G.: A multi-scale approach to mesh-based surface tension flows. *ACM Trans. Graph. (SIGGRAPH)* 29, 3 (2010). 3
- [WTGT10] WOJTAN C., THUERNEY N., GROSS M., TURK G.: Physically-inspired topology changes for thin fluid features. *ACM Trans. Graph. (SIGGRAPH)* 29, 3 (2010), 50. 3, 9
- [YWTY12] YU J., WOJTAN C., TURK G., YAP C.: Explicit mesh surfaces for particle based fluids. *Computer Graphics Forum (Eurographics)* 31, 2 (2012), 815–824. 3