



Tiled Characteristic Maps for Tracking Detailed Liquid Surfaces

F. Narita¹ [†] and R. Ando² [‡]

¹GAME FREAK Inc., Japan, ²Unaffiliated, Japan

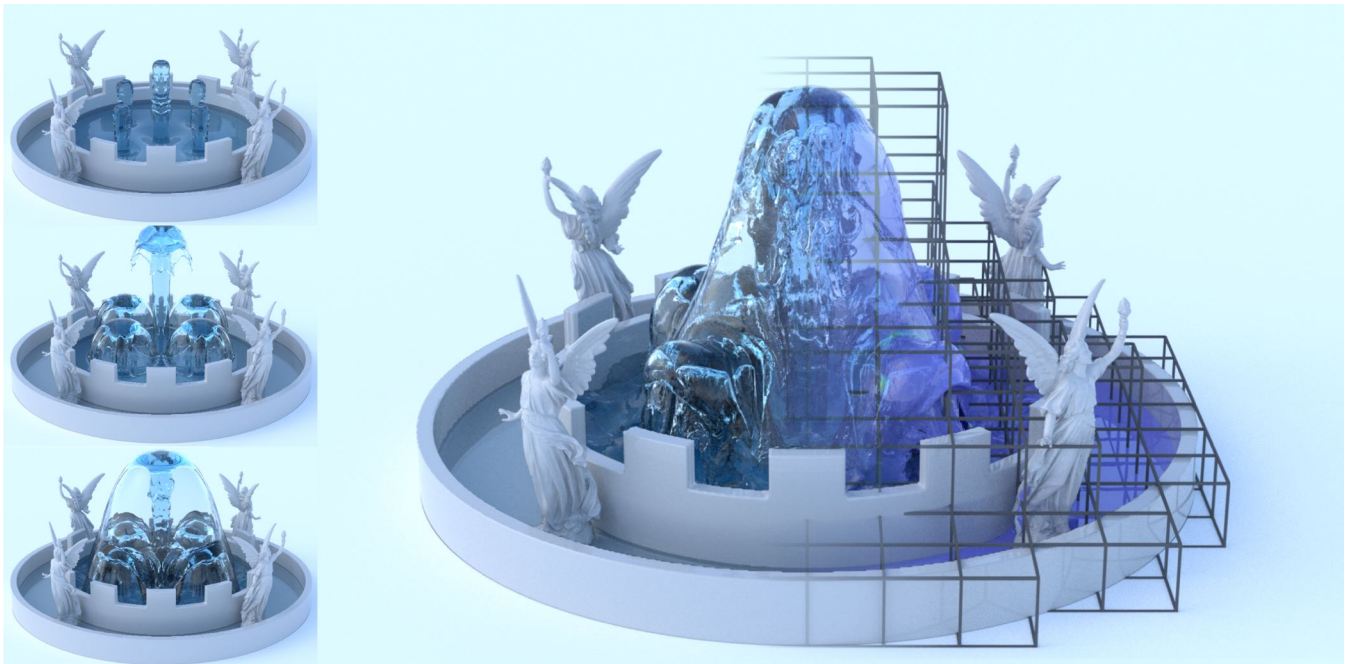


Figure 1: Fountain with lily statues. $300 \times 200 \times 300$ resolution, $12 \times 8 \times 12$ tiles. 54.9 seconds per video frame on average. Wired box and the color on the right visualize tiles and the map distortion, respectively.

Abstract

We introduce tiled characteristic maps for level set method that accurately preserves both thin sheets and sharp edges over a long period of time. Instead of resorting to high-order differential schemes, we utilize the characteristics mapping method to minimize numerical diffusion induced by advection. We find that although a single characteristic map could be used to better preserve detailed geometry, it suffers from frequent global re-initialization due to the strong distortions that are locally generated. We show that when multiple localized tiled characteristic maps are used, this limitation is constrained only within tiles; enabling long-term preservation of detailed structures where little distortion is observed. When applied to liquid simulation, we demonstrate that at a reasonably amount of added computational cost, our method retains small-scale high-fidelity (e.g., splashes and waves) that is quickly smeared out or deleted with purely grid-based or particle level set methods.

CCS Concepts

• *Computing methodologies* → *Physical simulation*;

[†] mail: fumiya.narita162@gmail.com

[‡] mail: ryich.ando@gmail.com

1. Introduction

Level set method has become a vital tool for tracking deforming surfaces in the visual effects industry [BLMB13; Mus13]. The notable advantages of level set methods are the robust handling of topological changes, ease of implementation, and the efficient fixed patterned memory access by the use of finite differential schemes.

Despite of its heavy use, level set methods inherently induce numerical diffusion at both advection and the re-initialization steps. This limitation has been ameliorated in a number of ways, such as high-order schemes [MG07; HK10] second-order accurate boundary conditions [RS00], semi-Lagrangian contouring [BGOS06] and the particle level set method [ENGF03]. These aforementioned techniques achieve higher accuracy; however, the added computational cost can be considerably large due to the extended stencil size or an increased number of variables (e.g., particles or quadrature points) [HK10; ENGF03].

Alternative approaches to complement the shortcomings of the level set method is to combine different surface representations, such as particles [CFL*07; FAW*16; SWT*18] or explicit meshes [BB09; WMB11]. This works to some extent; however, these strategies mainly focus on specific types of features (e.g., particulate splashes, thin sheets or wrinkles) and do not equally resolve a wide range of geometric fidelity.

In this paper, we leverage characteristic mapping method; we store multiple characteristic maps representing the state of the level set at a time, and advect them instead of advecting the signed distance field. Such an application has been hinted in some work [Hac05; QZG*19]; however, to the best of our knowledge, no previous work demonstrated the practical use for liquid simulation.

In particular, we propose tiled characteristic maps with the aim of distributing the re-initialization of mapping. The distributed re-initialization avoids globally re-setting the maps, which we find important for liquid simulation since dynamically moving surfaces generate strong local distortions. Altogether, our contributions are listed as follows

- A first practical framework for the level set method that uses characteristic maps
- Tiled characteristic maps avoiding the global re-initialization
- A memory-saving tiling using an upwind tracer predictor
- Estimated topological changes to detect necessity for local re-distancing

2. Related Work

In the following subsections we will discuss various surface tracking methods in the context of liquid simulation for graphics.

2.1. Level Set Methods

We refer readers to a thorough review by Osher and Fedkiw [OF03] and Gibou et al. [GFO18] for a vast body of literature of the level set methods. In graphics, level set methods are used not only to track moving surfaces but also to simulate fracture [HJST13; WFL*19] and to detect collisions for cloth [WWYW20], solid [JST*16] and rigid bodies [MEM*20].

2.1.1. PDE-based Methods

Advancing the state of the level set is given by the solution to a partial differential equation (PDE) of the form [OF03]

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \quad (1)$$

where ϕ , \mathbf{u} denote a scalar-valued signed distance field and velocity, respectively. Level set re-distancing may be solved as a solution to a PDE of the form [SSO94]

$$\frac{\partial \phi}{\partial \tau} + \text{sgn}(\phi_0) (|\nabla \phi| - 1) = 0, \quad (2)$$

where τ , ϕ_0 denote fictitious time and the initial state of ϕ (beginning of the time step, see [FF01]), respectively. In solving (1) and (2), the gradient evaluation plays an important role. Foster and Fedkiw [FF01] and Museth [Mus13] suggested using Essentially Non-Oscillatory (ENO) or Weighted ENO (WENO) schemes. Heo and Ko [HK10] proposed to use a high order polynomial approximation using a pseudo-spectral representation. These high-order schemes are quite effective at preserving the original shape; but the added computational cost can be also reportedly expensive [HK10].

Despite its accuracy, high order schemes have at least two limitations. First, many high order schemes require a large stencil support, making boundary treatments difficult. Constrained interpolation profile (CIP) [SKK07] exceptionally handles this issue. Second, PDE-based methods come with severe CourantFriedrichsLewy (CFL) constraint (i.e., $\text{CFL} < 1$). This limitation could be alleviated by a multi-step temporal integration, which adds extra computational overhead.

Note that our contributions are orthogonal to those high-order schemes; for example, we have used a fifth-order WENO scheme [GFO18] for the gradient evaluation.

2.1.2. Other Approaches

We used a semi-Lagrangian scheme for advecting level set because of its simplicity and stability. Several researchers also have explored the use of semi-Lagrangian schemes instead of solving the PDE [IGLF06; EQYF13; FWD14]. However, some semi-Lagrangian schemes are highly diffusive [FF01]; therefore, special treatment is needed if a high accuracy is desired, such as employing a WENO interpolation [KTT13].

Level set re-distancing can be done either by the fast marching [Set99], a fast sweeping technique [Zha04] or the semi-Lagrangian contouring [BGOS06]. In our implementation, we have used the PDE-based approach because the resulting surfaces are heuristically smoother this way [KMA20].

2.2. Mesh-based Methods

Mesh-based surface trackers have been studied as an alternative to the the level set method [WMB11]. In contrast to level set methods, mesh-based surface trackers are not diffusive by nature since mesh components are explicitly moved forward by velocity. On the other hand, unlike level set methods, handling topology changes (e.g., splitting and merging) is not trivial. With this difficulty in mind, many researchers focused on how to gracefully deal with such topological events.

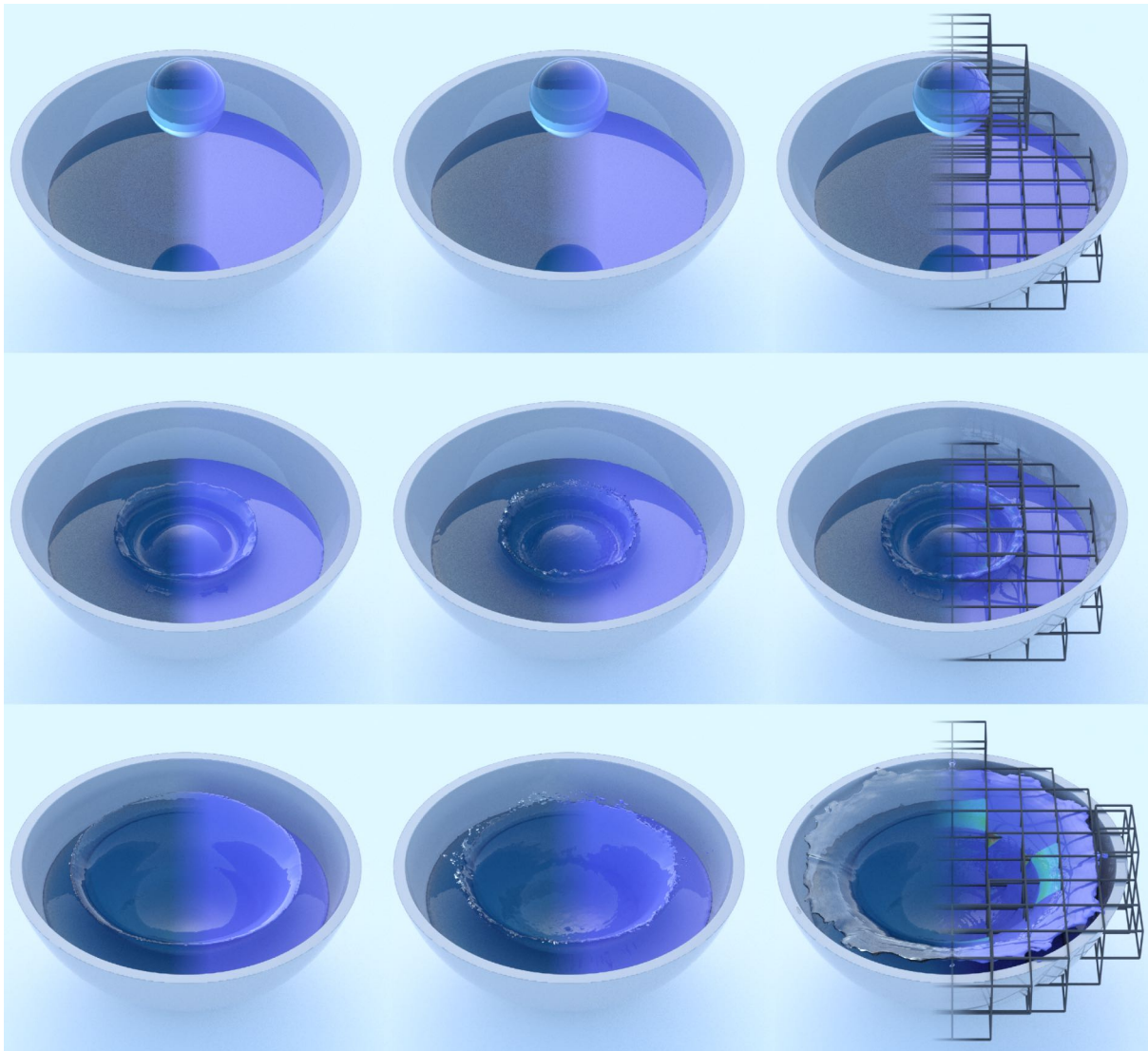


Figure 2: Waterdrop. $250 \times 200 \times 250$ resolution, $10 \times 8 \times 10$ tiles. 21.9 seconds per video frame on average. Level set simulation and the particle level set method are shown on the left and middle columns, respectively. Our method is shown on the right column. Notice that our method better preserves thin sheets of the crown.

An early work of purely mesh-based surface tracker in graphics is the work of Brochu and Bridson [BB09]. This framework is extended to multi-material [DBG14] and fluid simulation [BBB10; DHB*16]. Other types of Lagrangian approaches were also proposed such as tetrahedra [CWSO13; WJL*20] and simplicial complexes [MEB*14]. These purely mesh-approach handle collision detection by explicitly detecting collisions between meshes. This task can be facilitated by the aid of grids. For example, Müller [Mül09] proposed an extended marching cube for robustly handling topological changes that preserved inner sheets within cells. Some state-of-the-art methods even eliminated the need for grids [CMMK15].

2.3. Particle-based Methods

Particles also have been used to track the evolution of liquid surfaces. For an overview of particle-based reconstruction methods, we refer readers to a survey by Lefebvre et al. [IOS*14]. Generally, surfaces of particles are represented as an iso-contour of a distance function computed from a set of particles. Usually, naive computation of particle surfaces this way introduces undesired bumps. Some researchers smoothed these bumps. For example, Yu and Turk [YT10] introduced an anisotropic particle-based mesh reconstruction method. Bhattacharya et al. [BGB11] proposed a new bi-harmonic energy function to reconstruct minimize bumps on the particle surfaces.

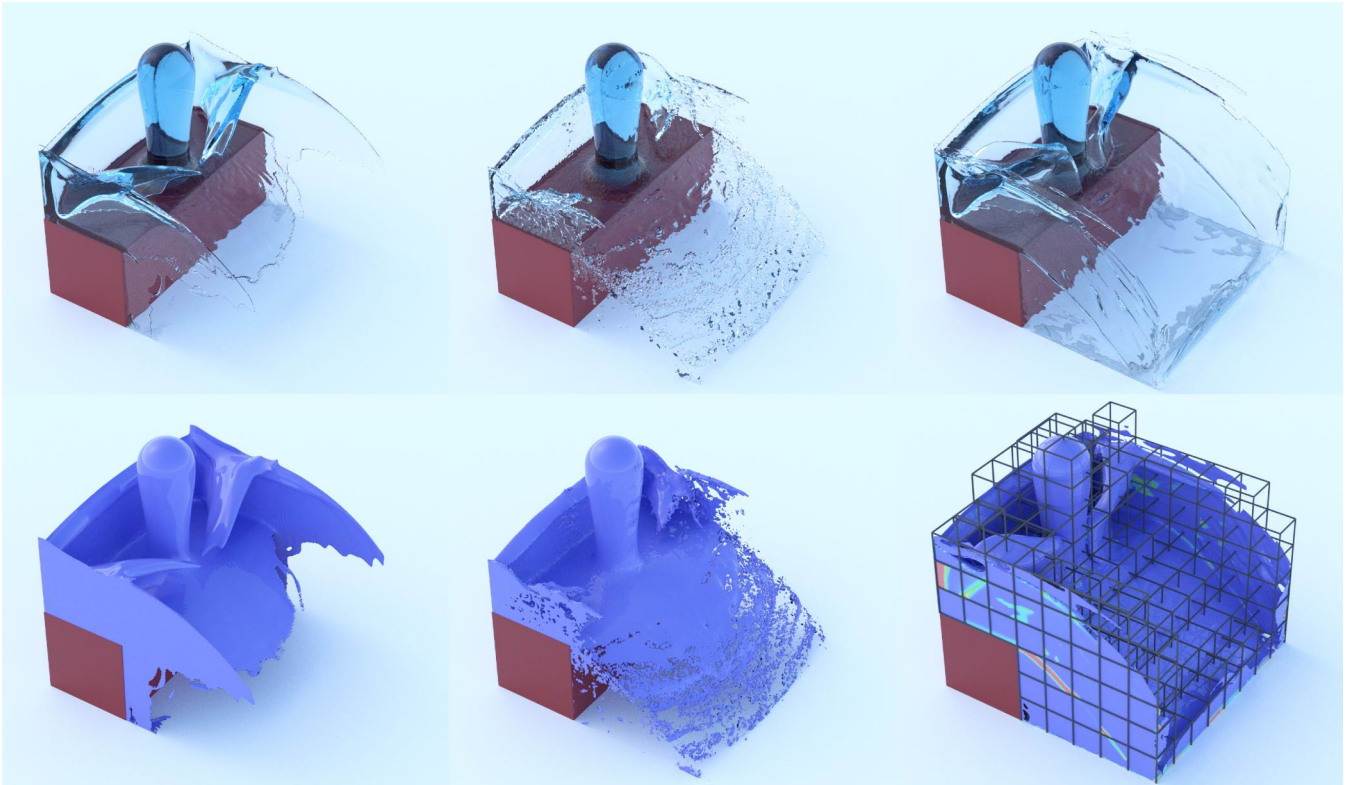


Figure 3: *Pouring.* $200 \times 200 \times 200$ resolution, $8 \times 8 \times 8$ tiles. 37.3 seconds per video frame on average. Our method is shown on the right column while the level set method and the particle level set method are shown on the left and the middle columns, respectively. Notice that consistent thin sheets are only preserved with our method.

2.4. Hybrid Methods

Some surface trackers combine best of the multiple (mostly two) methods. Wojtan et al. [WTGT09; WTGT10] proposed mesh-based surface trackers that handle topology changes aided by grid-based marching cubes. Yu et al. [YWTY12] presented a mesh-based surface tracker applied for SPH fluids. Sato et al. [SWT*18] seamlessly stitched level set and particle-based surfaces.

2.5. Characteristic Map

Characteristic map (a.k.a the method of characteristics) is shown effective for advection since the method is non-diffusive by nature. It was first introduced in graphics by Hachisuka [Hac05] followed by the works of Tessendorf and Pelfrey [TP11], Sato et al. [SWT*18] and Qu et al. [QZG*19]. An overview of the characteristic map is described in the next section. We note that these methods primarily focus on smoke, and the applications to the level set are only experimental. To the best of our knowledge, our work is the first to show that characteristic mapping can be used for liquid simulation.

3. Recapitulations

Our work builds upon the substantial body of the level set method and the characteristic map. For this reason, we outline important fundamentals of both methods before detailing our extension.

3.1. Level Set Method Recapitulation

Level set method is a grid-based surface tracker that facilitates topological changes. In level set methods, a signed distance function ϕ is defined near the surfaces (e.g., 3 cells bandwidth), and the surfaces are defined as a set of points where $\phi = 0$. Advection is performed by either solving (1) or via a semi-Lagrangian (or its extended) advection. Topology changes are automatically handled during advection.

Re-distancing must be performed at some intervals by (mostly) either solving (2) or the fast marching [Set99] since the slope of a signed distance field may be gradually flattened in the course of simulation [Bri08]. Our method directly extends the level set method. In doing so, we perform both advection and re-distancing using those accuracy/efficacy proven algorithms.

3.1.1. Characteristic Map Recapitulation

Characteristic map defines a mapping X (where X itself is also a function of time) such that $X(\mathbf{x}) = \mathbf{y}$, where both $\mathbf{x} \in \mathcal{R}^3$ and

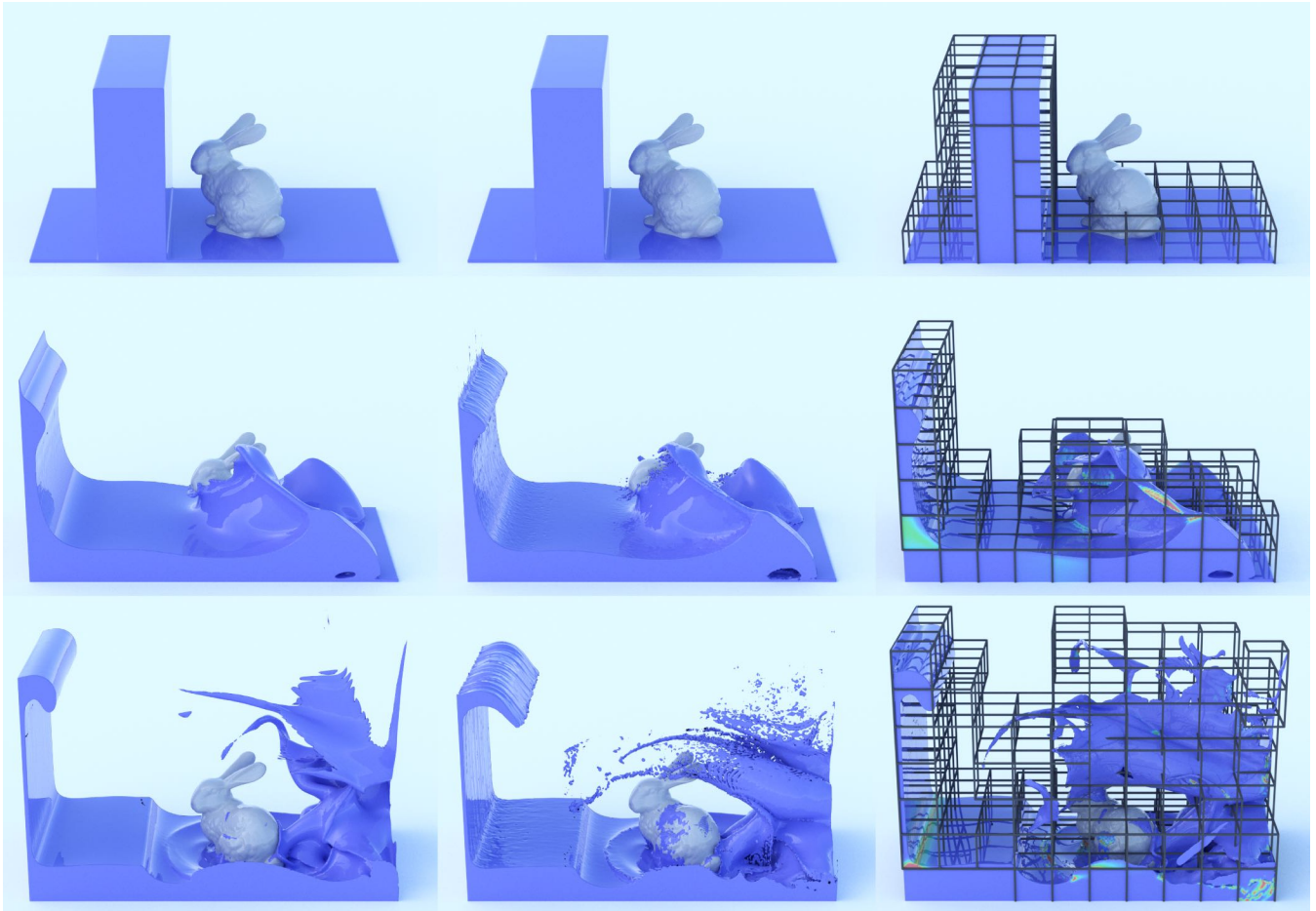


Figure 4: Breaking dam. $250 \times 250 \times 125$ resolution, $10 \times 10 \times 5$ tiles. 25.8 seconds per video frame on average. Our method is shown at the right column. Level set method and the particle level set method are shown on the left and the middle columns, respectively. Notice that our method captures more vibrant splashes reflected by the wall.

$\mathbf{y} \in \mathcal{R}^3$ denote spatial positions. With this setting, a solution to the advection equation

$$\frac{\partial q(\mathbf{x})}{\partial t} + \mathbf{u}(\mathbf{x}) \cdot \nabla q(\mathbf{x}) = 0 \quad (3)$$

can be alternatively solved by

$$q(\mathbf{x}) = q_0(X(\mathbf{x})), \quad (4)$$

$$\frac{\partial X(\mathbf{x})}{\partial t} + \mathbf{u}(\mathbf{x}) \cdot \nabla X(\mathbf{x}) = 0, \quad (5)$$

$$X_0(\mathbf{x}) = \mathbf{x}, \quad (6)$$

where $q(\mathbf{x})$, q_0 , and X_0 denote an arbitrary scalar field to be advected, the initial fields of q and the initial map, respectively. This way, we can solve for the time evolution of X instead of q to track the transport of q without numerical diffusion [TP11]. Usually, the mapping is defined discretely on grids, which is passively advected together with velocity.

When applied to fluids (level sets), map distortion consistently develops, making locally high compression. Eventually, the map

is no longer resolvable. Therefore, it is a common practice to re-initialize mapping when such a high distortion is detected [Hac05; TP11; QZG*19]. We follow the same strategies except that we use localized tiles as we explain in the next section.

4. Method Overview

At the heart of our method is a localized characteristic map populated per tile. To avoid confusion, we consistently use the terminology "tile", which corresponds to a box in 3D. To begin with, we first lay tiles with each tile having a fixed resolution of width N_x , height N_y and depth N_z , where N_x, N_y, N_z are all integers. For clarity, we illustrate procedures in two dimensions.

4.1. Simulation Loop

We base our method on a level set based liquid simulator [FF01] and the second-order accurate pressure solver on free surfaces [ENGF03]. Since the only difference from these works is how we advect the level set field; we focus our discussion only around

Algorithm 1 Level Set Step (Level Set Advection)

```

1: for all tile do
2:   Allocate / Delete / Retain Characteristic Map {§4.2}
3:   Advect Map  $X$  {§4.3}
4:   if High Distortion Detected {§4.4} or
      Tracer Out Of Bounds {§4.5} or
      Collision Expected {§4.6} then
5:     Reset Reference Level Set  $\phi_{i,j,k}^{\text{ref}} = \phi_{i,j,k}$ 
6:     Reset Characteristic Map s.t.  $X(\mathbf{x}) = \mathbf{x}$ 
7:   end if
8: end for
9: for all cell  $i, j, k$  do
10:   $\phi_{i,j,k} \leftarrow 0$ 
11:   $\theta_{\text{sum}} \leftarrow 0$ 
12:  for all tile do
13:     $\theta \leftarrow$  Compute Interpolation Weight 4.7
14:     $\phi_{i,j,k} \leftarrow \phi_{i,j,k} + \theta \phi_{i,j,k}^{\text{ref}}(X(\mathbf{x}))$  {§4.7}
15:     $\theta_{\text{sum}} \leftarrow \theta_{\text{sum}} + \theta$ 
16:  end for
17:   $\phi_{i,j,k} \leftarrow \phi_{i,j,k} / \theta_{\text{sum}}$ 
18: end for
19: Re-distance  $\phi$  {§4.8}
20: Post Process  $\phi$  {§4.9}

```

the level set calculations. The overview of our algorithm is illustrated in Algorithm 1.

4.2. Allocating Characteristic Maps

At the beginning of each time step, we find tiles that contains at least one pair of adjacent cells of which level set signs at two cells are different; indicating air and liquid cells. For convenience, we call these tiles as "liquid/air" tiles. Next, we allocate a characteristic map for each tile with each map having $3N_x \times 3N_y \times 3N_z$ resolution. Such a redundant space is needed as it allows longer back-tracking of characteristic maps. Note that we allocate maps only for liquid/air tiles to reduce memory consumption. If a map is already allocated, we skip this allocation (and keep the original intact). We delete the map if a tile is no longer classified as liquid/air.

4.3. Map Advection

Our next step is to advect characteristic maps. We do this by solving for (5) using a trilinearly interpolated semi-Lagrangian method [Sta99]. If needed, some source term (e.g., injected liquids) may be integrated [TP11].

4.4. Distortion Computation

We adopt the Cauchy-Green deformation tensor (conventionally denoted by $E \in \mathbb{R}^{3 \times 3}$) as a deformation estimator. Finally, the distortion is given as the Frobenius norm of E . If the squared distortion average within a tile exceeds a pre-defined tolerance ϵ_{tol} (we use 0.1 throughout our examples), we re-initialize the map associated with the tile.

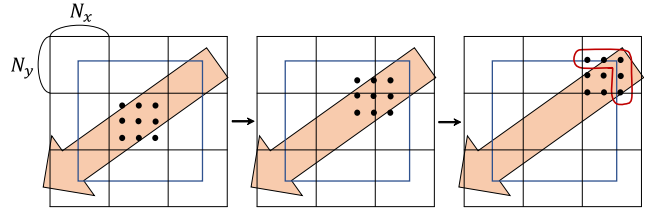


Figure 5: Minding the edge. If virtual tracers (re-mapped particles) shown in black dots approach an edge within the range of half a tile size (rightmost), we re-initialize the map. Orange arrow indicates the velocity field.

4.5. Minding the Edge

Our characteristic maps are bounded by tiles; therefore, we also re-initialize tiles in which at least one referencing position (which we reinterpret as a backward tracer position) is approaching a boundary of the map (that is, the edges of $3N_x \times 3N_y \times 3N_z$ domain). Proximity criterion we use is to check if the shortest distance to the edges is less than half a tile width (Figure 5).

4.6. Handling Collision

We find that special care is needed where liquid is expected to merge within the next time steps to prevent undesired air gaps to be generated. Note that a similar issue is also reported [BBB10]. We address this issue as follows. First, we detect possible future collision events similarly to the method of Koike et al. [KMA20]; that is, we trace positions forward in time and check if the tracer path bridges liquid-air-liquid cells and satisfy the following two conditions

- Relative velocity of two liquid cells residing in the opposite sides across the air cells is greater than or equal to the maximal velocity of these liquid cells.
- Dot product of two level set gradients on the liquid cells residing in the opposite sides across the air cells is less than -0.5 .

Finally, if such tracers are found, we re-initialize tiles that overlap with the trajectories (Figure 7).

4.7. Seamless Level Set Stitching

In the above exposition we illustrated how our per-tile level set is computed, but we still have not detailed how to define the global level set, which we will need to compute pressure and surface visualization. Hence, our next task is to flatten these localized level sets onto a single global grid by stitching these tiles without artifacts at its seams.

To obtain a global level set value at an arbitrary position \mathbf{x} , we first locate nearby four tiles and find the minimal L_1 distances to the edges and corners from these tiles. If all the distances are larger than a given threshold (which we define as transition bandwidth, denoted by D_{band}), we simply copy a value from the underlying tile. If any of distances falls below $D_{\text{band}}/2$, we interpolate value from the tiles. Overall, actual interpolation occurs only when query

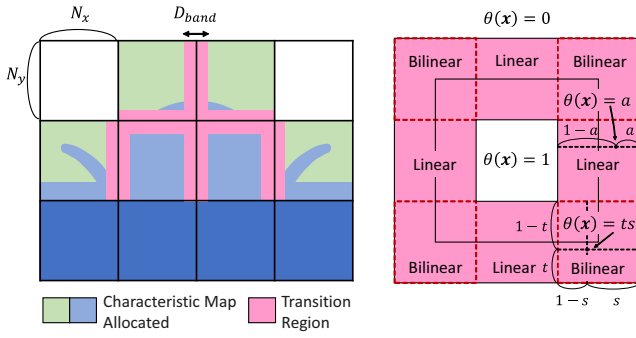


Figure 6: Level set interpolation at tile boundaries. When computing a globally defined level set value at a query position, we interpolate tile level sets among four neighbor tiles if necessary. More specifically, interpolation takes place only within red bands shown on the left where associated interpolation weights are shown on the right.

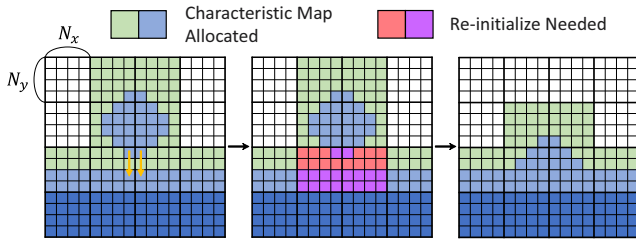


Figure 7: Predicted topological events. We seed virtual particles starting from liquid cells and let them move by the underlying velocity to check if a tracer bridges liquid-air-liquid cells. If found, we re-initialize maps associated with tiles crossing with the trajectory.

points hit red regions in Figure 6 left. We set $D_{band} = 6\Delta x$ throughout our examples.

Our interpolant kernel with respect to a tile is designed to be the combination of linear and bi-linear interpolants. For example, consider a case where the nearest point from a query point lie on an edge. In this case, the weight is given as a linear function such that $\theta_i = 1 - d_i/D_{band}$ where d_i , D_{band} denote the distance to an edge and the bandwidth of transition. For a case where the nearest position refers to a corner, the bi-linear interpolant is used as illustrated in Figure 6 right.

4.8. Re-distancing

Once the preliminary global level set is computed, we perform level set re-distancing to ensure that $|\nabla\phi| = 1$ near liquid surfaces. This is particularly important for second-order accurate boundary conditions for pressure as an actual distance is needed to calculate an appropriate liquid fraction [ENGF03]. Note that since the global level set will not be carried to the next time step, this re-distancing should be performed (preferably) every time step.

4.9. Post Processing

In principle, all the above steps are sufficient to get a valid globally defined level set. However, in our implementation we find that flickering slits are observed around the tips of thin sheets (Figure 10). We believe that this is mainly because characteristic mapping preserves highly thin sheets that may not be captured by some grid cells due to the Nyquist limit. We find that simply eroding the level set by the shift of $\Delta x/4$ effectively reduces the artifacts (to counteract the effect of erosion, one may dilate the level set by $-\Delta x/4$ at the beginning of the simulation). A further post-process (e.g., smoothing [FF01]) might be done if desired.

4.10. Reducing Redundant Space

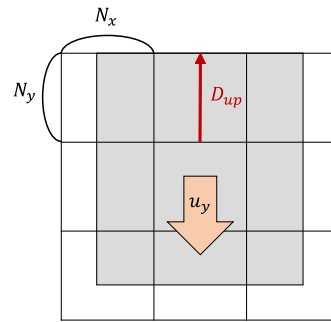


Figure 8: Anisotropic tile extension for redundancy reduction. In this example, velocity is pointing downward, making only upward margins sufficient while leaving other directions tight.

Our characteristic maps contain some redundant space to allow longer back-tracing (out-of-tile back-tracing) of characteristic maps. However, such a redundancy also comes at a cost of both extra memory requirements and the additional advection overhead. This redundancy may be optionally reduced by anisotropically designing redundant margins. Our basic idea is to extend redundant space only towards upwind directions.

For example, if a tile contains velocity components that are all pointing downward directions, the back-traced positions along this velocity will drift only upward directions (Figure 8). In this specific case, more redundant space will be allocated towards upward, while making other directions relatively tighter. In our implementation, the amount of upward extension D_{up} is computed by

$$D_{up} = \min(D_{band}, \max(-\alpha\Delta t u_y, 0) + (\beta + D_{band}/2)\Delta x), \quad (7)$$

where α and β are constant parameters dictating scaling and an additional extension width, which we set $\alpha = 10$ and $\beta = 3$. Note that the minimal amount $N_y/2$ is required to ensure smooth interpolation between tiles. D_{right} , D_{left} and D_{bottom} can be computed likewise.

5. Results

We performed four sets of experiments: level set method, particle level set method, a globally defined characteristic map, and the tiled characteristic maps (our method). As for the level set method we implemented, we performed re-distancing (WENO5/PDE) every 3 steps, and used a tri-linearly interpolated semi-Lagrangian method. We also performed Zalesak's disk rotation and the Enright's test. Time per video frame is available in the associated figure captions.

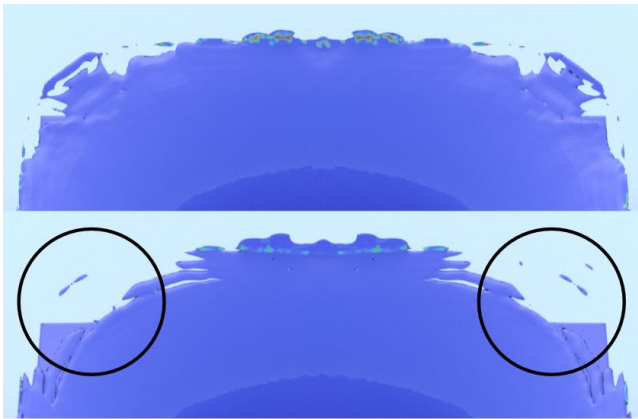


Figure 9: Re-initialization artifacts of a globally defined characteristic map. Before being re-initialized (top). After the re-initialization (bottom). When re-initialization is triggered, a noticeably large part of thin sheets are deleted within a single step.

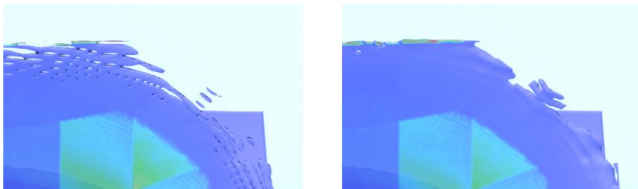


Figure 10: Effect of our post-processing. Without dilation (left). With our dilation (right). Notice that many slits and small holes are removed on the right.

5.1. Water Drop

Figure 2 primarily demonstrates that our method preserves crown sheets thinner than the level set method. As can be seen from the figure, both the level set and the particle level methods fail to keep such sheets. Globally defined characteristic map may also better preserve thin sheets for a longer period of time, global re-initialization immediately removes large parts of sheets within a single step, resulting in temporally noticeable artifacts (Figure 9). $250 \times 200 \times 250$ resolution, $10 \times 8 \times 10$ tiles. Simulated on an AMD Ryzen 9 3900X CPU workstation.

5.2. Breaking Dam

Figure 4 is an example of breaking dam interacting with spherical obstacle. Like the traditional level set method, we extrapolate level set towards solid to handle the presence of solid obstacles. Notice that our method remains robust without special care in such cases. $250 \times 250 \times 125$ resolution, $10 \times 10 \times 5$ tiles. Simulated on an AMD Ryzen 9 3900X CPU workstation.

5.3. Corner Pouring

Figure 3 shows an example of pouring water onto a box. This is particularly challenging with the level set method since the wide

spreading sheets are quickly wiped away due to the numerical diffusion. Our method on the other hand allows the sheets to land the floor. $200 \times 200 \times 200$ resolution, $8 \times 8 \times 8$ tiles. Simulated on an AMD Ryzen 9 3900X CPU workstation.

5.4. Fountain

Finally, Figure 1 shows a case with complicated sheets and solids. Notice that when multiple columns of water form consistent thin sheets that are merged to the pool on the ground. Traditional level set, particle level set method, on the other hand, fail to maintain such thin sheets. $300 \times 200 \times 300$ resolution, $12 \times 8 \times 12$ tiles. Simulated on an AMD Ryzen 9 3900X CPU workstation. Comparisons are available in the supplemental video.

5.5. Timings and Memory Reduction

In our implementation, we used OpenVDB[Mus13] where applicable. As shown in Table 1, the run-time overhead ranges from a half to up to twice the cost of the level set method. It is expected that our method runs as fast as particle level set method for cases where observed visual complexity is almost the same.

As for the memory consumption, we observed that the total amount is similar to the particle level set method. Particularly, this amount and also the run-time are closely linked to the added visual complexity in the resulting simulation, indicating that added overheads are somehow justified. Table 2 shows a reduction rate of our anisotropic tile extension. With our extension, our method cuts the overall memory use into (up to) half.

5.6. Zalesak's Disk Rotation and Enright's Test

To quantitatively evaluate the accuracy of advection, we performed both the Zalesak's disk rotation experiment and Enright's deformation test as also done by previous work [KTT13; WTGT09] shown in Figure 11. Our method better maintains sharp corners and thin sheets than both the level set and the particle level set methods. For the level set method and our method, we used semi-Lagrangian advection and the WENO5/PDE re-distancing schemes [Mus13]. The only difference is the presence of our extension.

6. Discussion

6.1. Advanced Characteristic Maps

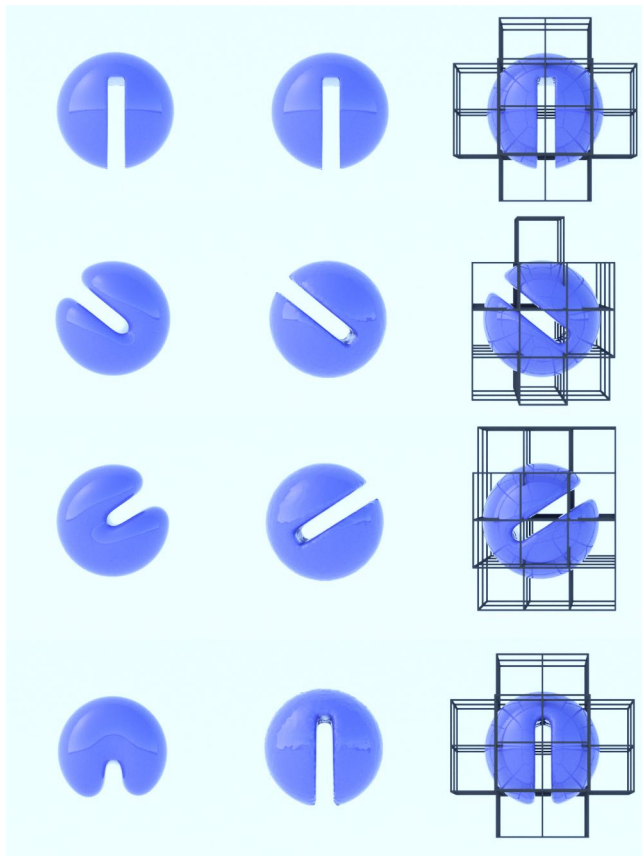
We based a (comparably) primitive characteristic mapping [TP11] because of its simplicity. This can be further improved by Qu et al. [QZG*19], which we leave for future work.

6.2. Post Process

In our current implementation, we employed a simple erosion algorithm to clean up small slit artifacts. This could be also further improved by a dilation-erosion-smooth algorithm [Mus13]. In the supplemental video, we provide a simulation without post processing to illustrate how this post process improves the visual animation.

Table 1: Timings of our method, a level set and the particle level set. Postprocess includes dilation, map distortion computation, tile allocation/deletion (these three steps only exist in our method), surface meshing, and the re-distancing. All numbers are given by seconds.

Scene	Grid Resolution	Tile Resolution	Method	Vel Step			LS Step	Postprocess	Total
				Project	Rest	Total			
Fig. 2 / Drop	$250 \times 200 \times 250$	-	LSM	1.5	0.9	2.4	0.6	1.3	4.4
		-	PLS	1.4	0.8	2.2	4.3	1.3	7.9
		$10 \times 8 \times 10$	Ours	1.5	2.0	3.5	0.6	2.6	6.8
Fig. 4 / Dam	$250 \times 250 \times 125$	-	LSM	1.6	0.6	2.2	0.5	1.1	3.8
		-	PLS	1.1	0.5	1.6	3.6	1.1	6.4
		$10 \times 10 \times 5$	Ours	1.5	1.9	3.4	0.5	2.4	6.4
Fig. 3 / Pouring	$200 \times 200 \times 200$	-	LSM	1.1	0.6	1.7	0.8	1.6	4.3
		-	PLS	0.2	0.7	0.9	3.6	1.2	5.8
		$8 \times 8 \times 8$	Ours	1.0	1.9	2.9	0.8	3.9	8.0
Fig. 1 / Fountain	$300 \times 200 \times 300$	-	LSM	0.7	1.4	2.1	1.0	2.3	5.6
		-	PLS	0.7	1.3	2.0	7.7	2.4	12.3
		$12 \times 8 \times 12$	Ours	2.2	3.8	6.0	1.3	6.0	13.7

**Figure 11:** Zalesak's disk rotation. Level set (left column), particle level set (middle column) and our method (right column). $200 \times 200 \times 200$ resolution, $8 \times 8 \times 8$ tiles.**Table 2:** Memory reduction rate with/without the anisotropic tile extension. The memory consumption includes all the components of the solver, such as velocity and pressure.

Scene	w/o (GB)	w (GB)	Reduction Rate (%)
Fig. 2 / Drop	0.95	0.5	52.4
Fig. 4 / Dam	0.88	0.39	44.3
Fig. 3 / Pouring	1.19	0.56	46.5
Fig. 1 / Fountain	1.94	0.94	48.7

6.3. FLIP/APIC

In this paper we have not incorporated FLIP/APIC [JSS*15] methods because focusing on the purely level set method more clearly demonstrates the advantages of our method. If desired, our method may be combined with these FLIP-based extensions [FAW*16; SWT*18].

6.4. Volume Loss

We showed that our method better preserves sheets and splashes, but like the level set method we do not guarantee volume preservation. This may be corrected by global [KLL*07] or local [GAB20] volume preservation algorithms.

6.5. Global Characteristic Maps

In the case of Figure 2, the global characteristic maps run 13% slower than our approach. This is because the global characteristic maps require that the velocity field must be extrapolated in the whole domain while our method does not.

6.5.1. Tile Size

We chose N_x, N_y and N_z in a heuristic manner. For example, if tiles are too large, the (undesired) impact of re-initialization due to the local distortions would reach many nearby cells. If tiles are too small, our characteristic maps become less effective. In our experiments, we find that a size of 25^3 was a good balance.

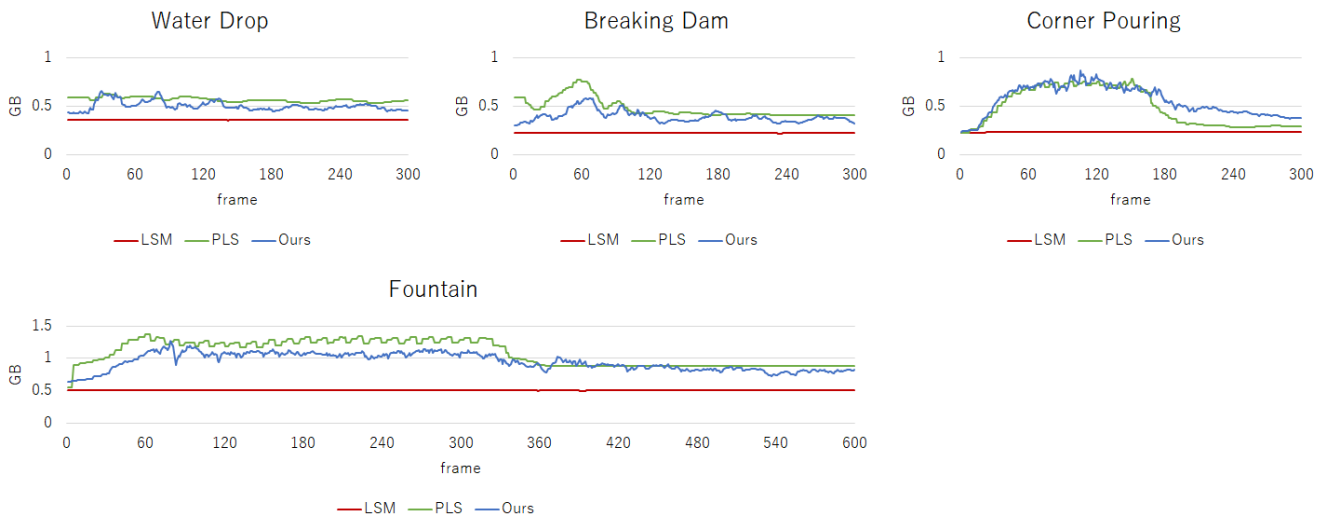


Figure 12: Memory consumption overview of a level set method, the particle level set method, and our method, respectively. Note that we used OpenVDB where possible to reduce explicit grid allocation.

6.5.2. Particle Level Set Method

Our implementation of the particle level set method is based on the work of Enright et al. [ENGF03]. We note that this particle level set method can be extended to better preserve splash and sheets [GSLF05].

6.5.3. Advection

We used semi-Lagrangian advection because of its simplicity. A more accurate advection scheme such as BFECC, MacCormack could be used with special care near free surfaces [SFK*08].

6.6. Limitations

6.6.1. Source and Sink

Currently, our method only supports the advection of the form $D\phi/Dt = 0$; indicating that no source or sink terms should be introduced. In the context of characteristic mapping methods, it is shown that such a source term may be taken into account by performing extra forward tracing [Hac05; TP11; QZG*19]. In our implementations, we simply reset mapping if a source term such as water injection is needed.

Note that re-distancing level set is not affected by this limitation because it is only performed on a scratch grid (a globally flattened level set field), and will not be carried back to the tile's level set grids, except when the characteristic map is reset.

6.6.2. Nyquist Limit

It should be noted that our method may suffer from the Nyquist limit when a strong deformation is allowed to be generated. For example, when two virtual back-tracked particles originated from adjacent cells travel too far from each other, some highly detailed geometries residing between two particles will be overlooked. This

limitation may be alleviated by super-sampling of back-tracking; which would add more extra computation overheads.

7. Conclusions

In this paper, we proposed new tiled characteristic maps for level set method. The principle idea of our method is to assign locally defined characteristic maps to individual tiles. Unlike the globally defined characteristic map, which responds to locally strong distortions and deletes all the small-scale features (thin sheets and spindles) within a single step, our method re-initializes maps only where necessary, leaving other parts of liquid geometry untouched.

We ran a number of examples and demonstrated that our method is robust against a wide variety of liquid simulation scenarios, including the presence of obstacles and a water pouring on a flat container. Considering the increased visually rich complexity, we believe that overhead of our method is within a reasonable range (e.g., additional run-time is up to twice of the level set method). We also validated the quantitative accuracy via Zalesak's disk rotation and the Enright's deformation test.

In future work, we are interested in exploring our method not only (more advanced) liquid (e.g., surface tension) but also smoke, fire, viscoelastic materials and highly viscous fluid, which all involve non-trivial material transport. We believe that our method can be also applied for engineering and medical purposes outside graphics, such as shape extraction [ZM10; BN14], boundary detection [AUS13] and MRI image segmentation [Bar11].

Acknowledgements

This research was supported by the JSPS Grant-in-Aid for Young Scientists (18K18060).

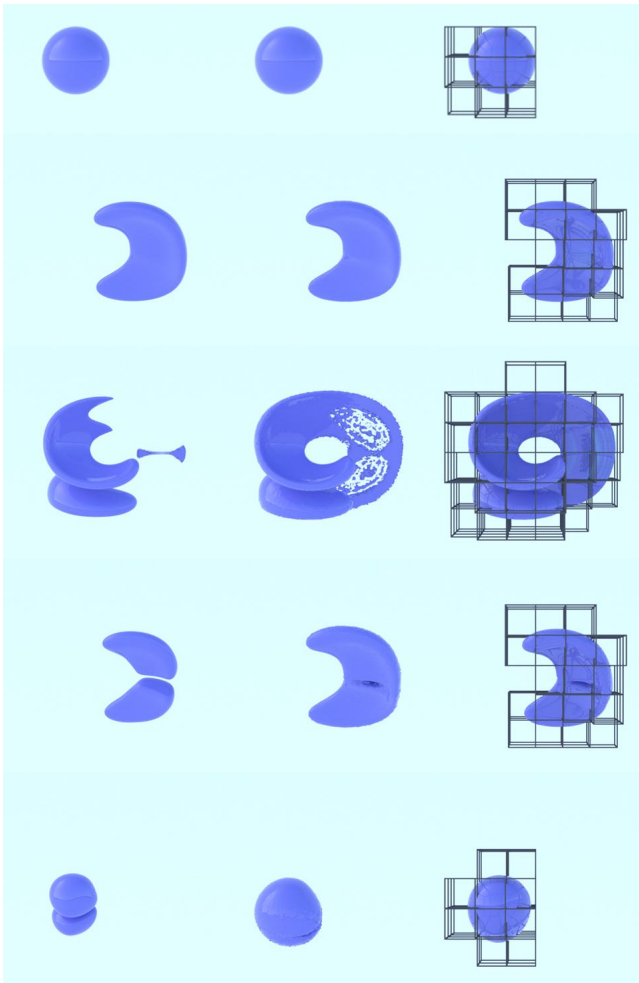


Figure 13: Enright's test [EFFM02]. Level set (left column), particle level set (middle column) and our method (right column). $200 \times 200 \times 200$ resolution, $8 \times 8 \times 8$ tiles.

References

- [AUS13] ANAM, S., UCHINO, E., and SUETAKE, N. "Image Boundary Detection Using the Modified Level Set Method and a Diffusion Filter". *Procedia Computer Science* 22 (2013). 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013, 192–200 10.
- [Bar11] BARMAN, PARESH. "MRI Image Segmentation Using Level Set Method and Implement an Medical Diagnosis System". *Computer Science Engineering: An International Journal* 1 (Dec. 2011), 11–10 10.
- [BB09] BROCHU, TYSON and BRIDSON, ROBERT. "Robust Topological Operations for Dynamic Explicit Surfaces". *SIAM J. Sci. Comput.* 31.4 (June 2009), 2472–2493 2, 3.
- [BBB10] BROCHU, TYSON, BATTY, CHRISTOPHER, and BRIDSON, ROBERT. "Matching Fluid Simulation Elements to Surface Geometry and Topology". *ACM Trans. Graph.* 29.4 (July 2010) 3, 6.
- [BGB11] BHATACHARYA, HAIMASREE, GAO, YUE, and BARGTEIL, ADAM. "A Level-Set Method for Skinning Animated Particle Data". *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '11. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2011, 17–24 3.
- [BGOS06] BARGTEIL, ADAM W., GOKTEKIN, TOLGA G., O'BRIEN, JAMES F., and STRAIN, JOHN A. "A Semi-Lagrangian Contouring Method for Fluid Simulation". *ACM Trans. Graph.* 25.1 (Jan. 2006), 19–38 2.
- [BLMB13] BUDSBERG, JEFF, LOSURE, MICHAEL, MUSETH, KEN, and BAER, MATT. *Liquids in The Croods*. 2013 2.
- [BN14] BINDU, V R and NAIR, K N RAMACHANDRAN. "A fast narrow band level set formulation for shape extraction". *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*. 2014, 137–142 10.
- [Bri08] BRIDSON, ROBERT. *Fluid Simulation*. USA: A. K. Peters, Ltd., 2008 4.
- [CFL*07] CHENTANEZ, NUTTAPONG, FELDMAN, BRYAN E., LABELLE, FRANÇOIS, et al. "Liquid Simulation on Lattice-Based Tetrahedral Meshes". *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '07. San Diego, California: Eurographics Association, 2007, 219–228 2.
- [CMMK15] CHENTANEZ, NUTTAPONG, MÜLLER, MATTHIAS, MACKLIN, MILES, and KIM, TAE-YONG. "Fast Grid-Free Surface Tracking". *ACM Trans. Graph.* 34.4 (July 2015) 3.
- [CWSO13] CLAUSEN, PASCAL, WICKE, MARTIN, SHEWCHUK, JONATHAN R., and O'BRIEN, JAMES F. "Simulating Liquids and Solid-Liquid Interactions with Lagrangian Meshes". *ACM Trans. Graph.* 32.2 (Apr. 2013) 3.
- [DBG14] DA, FANG, BATTY, CHRISTOPHER, and GRINSPUN, EITAN. "Multimaterial Mesh-Based Surface Tracking". *ACM Trans. Graph.* 33.4 (July 2014) 3.
- [DHB*16] DA, FANG, HAHN, DAVID, BATTY, CHRISTOPHER, et al. "Surface-Only Liquids". *ACM Trans. Graph.* 35.4 (July 2016) 3.
- [EFFM02] ENRIGHT, DOUGLAS, FEDKIW, RONALD, FERZIGER, JOEL, and MITCHELL, IAN. "A Hybrid Particle Level Set Method for Improved Interface Capturing". *Journal of Computational Physics* 183.1 (2002), 83–116 11.
- [ENGF03] ENRIGHT, DOUG, NGUYEN, DUC, GIBOU, FREDERIC, and FEDKIW, RON. "Using the Particle Level Set Method and a Second Order Accurate Pressure Boundary Condition for Free Surface Flows". Jan. 2003 2, 5, 7, 10.
- [EQYF13] ENGLISH, R. ELLIOT, QIU, LINHAI, YU, YUE, and FEDKIW, RONALD. "Chimera Grids for Water Simulation". *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '13. Anaheim, California: Association for Computing Machinery, 2013, 85–94 2.
- [FAW*16] FERSTL, FLORIAN, ANDO, RYOICHI, WOJTAN, CHRIS, et al. "Narrow Band FLIP for Liquid Simulations". *Computer Graphics Forum* 35.2 (2016), 225–232 2, 9.
- [FF01] FOSTER, NICK and FEDKIW, RONALD. "Practical Animation of Liquids". *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, 23–30 2, 5, 7.
- [FWD14] FERSTL, FLORIAN, WESTERMANN, RÜDIGER, and DICK, CHRISTIAN. "Large-Scale Liquid Simulation on Adaptive Hexahedral Grids". *IEEE Trans. Vis. Comput. Graph.* 20.10 (2014), 1405–1417 2.
- [GAB20] GOLDADE, RYAN, AANJANEYA, MRIDUL, and BATTY, CHRISTOPHER. "Constraint Bubbles and Affine Regions: Reduced Fluid Models for Efficient Immersed Bubbles and Flexible Spatial Coarsening". *ACM Trans. Graph.* 39.4 (July 2020) 9.
- [GFO18] GIBOU, FREDERIC, FEDKIW, RONALD, and OSHER, STANLEY. "A review of level-set methods and some recent applications". *Journal of Computational Physics* 353 (2018), 82–109 2.
- [GSLF05] GUENDELMAN, ERAN, SELLE, ANDREW, LOSASSO, FRANK, and FEDKIW, RONALD. "Coupling Water and Smoke to Thin Deformable and Rigid Shells". *ACM SIGGRAPH 2005 Papers*. SIGGRAPH '05. Los Angeles, California: Association for Computing Machinery, 2005, 973–981 10.

- [Hac05] HACHISUKA, TOSHIYA. “Combined Lagrangian-Eulerian Approach for Accurate Advection”. *ACM SIGGRAPH 2005 Posters*. SIGGRAPH '05. Los Angeles, California: Association for Computing Machinery, 2005, 114–es 2, 4, 5, 10.
- [HJST13] HEGEMANN, JAN, JIANG, CHENFANFU, SCHROEDER, CRAIG, and TERAN, JOSEPH M. “A Level Set Method for Ductile Fracture”. *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '13. Anaheim, California: Association for Computing Machinery, 2013, 193–201 2.
- [HK10] HEO, NAMBIN and KO, HYEONG-SEOK. “Detail-Preserving Fully-Eulerian Interface Tracking Framework”. *ACM Trans. Graph.* 29.6 (Dec. 2010) 2.
- [IGLF06] IRVING, GEOFFREY, GUENDELMAN, ERAN, LOSASSO, FRANK, and FEDKIW, RONALD. “Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques”. *ACM Trans. Graph.* 25.3 (July 2006), 805–811 2.
- [IOS*14] IHMSEN, MARKUS, ORTHMANN, JENS, SOLENTHALER, BARBARA, et al. “SPH Fluids in Computer Graphics”. *Eurographics 2014 - State of the Art Reports*. Ed. by LEFEBVRE, SYLVAIN and SPAGNUOLO, MICHELA. The Eurographics Association, 2014 3.
- [JSS*15] JIANG, CHENFANFU, SCHROEDER, CRAIG, SELLE, ANDREW, et al. “The Affine Particle-in-Cell Method”. *ACM Trans. Graph.* 34.4 (July 2015) 9.
- [JST*16] JIANG, CHENFANFU, SCHROEDER, CRAIG, TERAN, JOSEPH, et al. “The Material Point Method for Simulating Continuum Materials”. *ACM SIGGRAPH 2016 Courses*. SIGGRAPH '16. Anaheim, California: Association for Computing Machinery, 2016 2.
- [KLL*07] KIM, BYUNGMOON, LIU, YINGJIE, LLAMAS, IGNACIO, et al. “Simulation of Bubbles in Foam with the Volume Control Method”. *ACM Trans. Graph.* 26.3 (July 2007), 98–es 9.
- [KMA20] KOIKE, T., MORISHIMA, S., and ANDO, R. “Asynchronous Eulerian Liquid Simulation”. *Computer Graphics Forum* 39.2 (2020), 1–8 2, 6.
- [KTT13] KIM, THEODORE, TESSENDORF, JERRY, and THÜREY, NILS. “Closest Point Turbulence for Liquid Surfaces”. *ACM Trans. Graph.* 32.2 (Apr. 2013) 2, 8.
- [MEB*14] MISZTAL, MAREK KRZYSZTOF, ERLEBEN, KENNY, BARGTEIL, ADAM W., et al. “Multiphase Flow of Immiscible Fluids on Unstructured Moving Meshes (TVCG)”. *IEEE Trans. Vis. Comput. Graph.* 20.1 (2014), 4–16 3.
- [MEM*20] MACKLIN, MILES, ERLEBEN, KENNY, MÜLLER, MATTHIAS, et al. “Local Optimization for Robust Signed Distance Field Collision”. *Proc. ACM Comput. Graph. Interact. Tech.* 3.1 (Apr. 2020) 2.
- [MG07] MIN, CHO HONG and GIBOU, FRÉDÉRIC. “A second order accurate level set method on non-graded adaptive cartesian grids”. *Journal of Computational Physics* 225.1 (2007), 300–321 2.
- [Mül09] MÜLLER, MATTHIAS. “Fast and Robust Tracking of Fluid Surfaces”. *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '09. New Orleans, Louisiana: Association for Computing Machinery, 2009, 237–245 3.
- [Mus13] MUSETH, KEN. “VDB: High-Resolution Sparse Volumes with Dynamic Topology”. *ACM Trans. Graph.* 32.3 (July 2013) 2, 8.
- [OF03] OSHER, STANLEY and FEDKIW, RONALD. *Level Set Methods and Dynamic Implicit Surfaces*. Springer New York, 2003 2.
- [QZG*19] QU, ZIYIN, ZHANG, XINXIN, GAO, MING, et al. “Efficient and Conservative Fluids Using Bidirectional Mapping”. *ACM Trans. Graph.* 38.4 (July 2019) 2, 4, 5, 8, 10.
- [RS00] RUSSO, GIOVANNI and SMEREKA, PETER. “A Remark on Computing Distance Functions”. *Journal of Computational Physics* 163.1 (2000), 51–67 2.
- [Set99] SETHIAN, J.A. “Advancing Interfaces: Level Set and Fast Marching Methods”. *Dept. of Mathematics, University of California, Berkeley* (1999) 2, 4.
- [SFK*08] SELLE, ANDREW, FEDKIW, RONALD, KIM, BYUNGMOON, et al. “An Unconditionally Stable MacCormack Method”. *J. Sci. Comput.* 35 (June 2008), 350–371 10.
- [SKK07] SONG, OH-YOUNG, KIM, DOYUB, and KO, HYEONG-SEOK. “Derivative Particles for Simulating Detailed Movements of Fluids”. *IEEE Trans. Vis. Comput. Graph.* 13.4 (2007), 711–719 2.
- [SSO94] SUSSMAN, MARK, SMEREKA, PETER, and OSHER, STANLEY. “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow”. *Journal of Computational Physics* 114.1 (1994), 146–159 2.
- [Sta99] STAM, JOS. “Stable Fluids”. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 1999, 121–128 6.
- [SWT*18] SATO, T., WOJTAN, C., THUREY, N., et al. “Extended Narrow Band FLIP for Liquid Simulations”. *Computer Graphics Forum* 37.2 (2018), 169–177 2, 4, 9.
- [TP11] TESSENDORF, J. and PELFREY, BRANDON. “The Characteristic Map for Fast and Efficient VFX Fluid Simulations”. In *Proceedings of the Computer Graphics International Workshop on VFX*. 2011 4–6, 8, 10.
- [WFL*19] WOLPER, JOSHUAH, FANG, YU, LI, MINCHEN, et al. “CD-MPM: Continuum Damage Material Point Methods for Dynamic Fracture Animation”. *ACM Trans. Graph.* 38.4 (July 2019) 2.
- [WJL*20] WANG, HUI, JIN, YONGXU, LUO, ANQI, et al. “Codimensional Surface Tension Flow Using Moving-Least-Squares Particles”. *ACM Trans. Graph.* 39.4 (July 2020) 3.
- [WMB11] WOJTAN, CHRIS, MÜLLER-FISCHER, MATTHIAS, and BROCHU, TYSON. “Liquid Simulation with Mesh-Based Surface Tracking”. *ACM SIGGRAPH 2011 Courses*. SIGGRAPH '11. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2011 2.
- [WTGT09] WOJTAN, CHRIS, THÜREY, NILS, GROSS, MARKUS, and TURK, GREG. “Deforming Meshes That Split and Merge”. *ACM Trans. Graph.* 28.3 (July 2009) 4, 8.
- [WTGT10] WOJTAN, CHRIS, THÜREY, NILS, GROSS, MARKUS, and TURK, GREG. “Physics-Inspired Topology Changes for Thin Fluid Features”. *ACM Trans. Graph.* 29.4 (July 2010) 4.
- [WWYW20] WU, LONGHUA, WU, BOTAO, YANG, YIN, and WANG, HUAMIN. “A Safe and Fast Repulsion Method for GPU-based Cloth Self Collisions”. *ACM Trans. Graph.* 40.1 (2020), 5:1–5:18 2.
- [YT10] YU, JIHUN and TURK, GREG. “Reconstructing Surfaces of Particle-Based Fluids Using Anisotropic Kernels”. *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '10. Madrid, Spain: Eurographics Association, 2010, 217–225 3.
- [YWTY12] YU, JIHUN, WOJTAN, CHRIS, TURK, GREG, and YAP, CHEE. “Explicit Mesh Surfaces for Particle Based Fluids”. *Comput. Graph. Forum* 31.2pt4 (May 2012), 815–824 4.
- [Zha04] ZHAO, HONGKAI. “A fast sweeping method for Eikonal equations”. *Mathematics of Computation* 74.250 (May 2004), 603–628 2.
- [ZM10] ZHOU, BIN and MU, CHUN-LAI. “Level set evolution for boundary extraction based on a p-Laplace equation”. *Applied Mathematical Modelling* 34.12 (2010), 3910–3916 10.