

Boundary Detection in Particle-based Fluids

Marcos Sandim¹, Douglas Cedrim¹, Luis Gustavo Nonato¹, Paulo Pagliosa² and Afonso Paiva¹

¹ICMC-USP, São Carlos, Brazil

²FACOM-UFMS, Campo Grande, Brazil

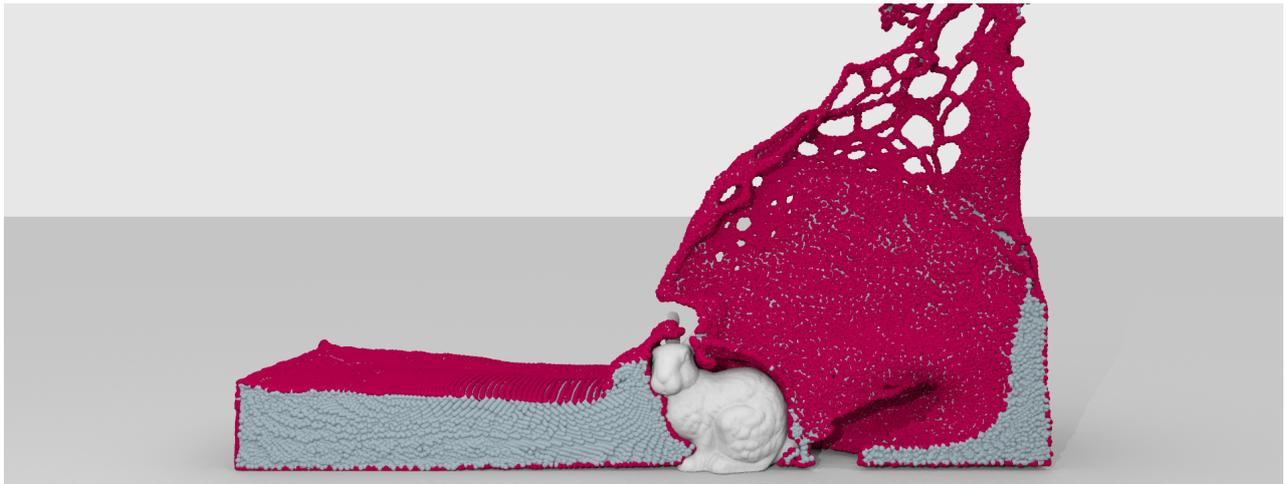


Figure 1: The boundary particles (red) detected by our method in a cutaway view of a dam break with Bunny as obstacle.

Abstract

This paper presents a novel method to detect free-surfaces on particle-based volume representation. In contrast to most particle-based free-surface detection methods, which perform the surface identification based on physical and geometrical properties derived from the underlying fluid flow simulation, the proposed approach only demands the spatial location of the particles to properly recognize surface particles, avoiding even the use of kernels. Boundary particles are identified through a Hidden Point Removal (HPR) operator used for visibility test. Our method is very simple, fast, easy to implement and robust to changes in the distribution of particles, even when facing large deformation of the free-surface. A set of comparisons against state-of-the-art boundary detection methods show the effectiveness of our approach. The good performance of our method is also attested in the context of fluid flow simulation involving free-surface, mainly when using level-sets for rendering purposes.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary representations; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Particle-based fluid representation has an important role in computer animation, enabling realistic physical simulations of water and blood flows, honey pouring, and smoke dispersion. In fact, meshfree methods such as Smoothed Particle Hydrodynamics (SPH) [IOS*14], Moving-Particle Semi-implicit (MPS) [PTB*03], Fluid-Implicit-Particle (FLIP) [ZB05], and Position Based Fluids (PBF) [MM13] strongly rely on particle representation to simulate topologically intricate fluid flows in complex geometries.

A main issue for meshfree methods is the imposition of boundary conditions, a task that demands accurate identification of *boundary particles*, that is, particles that belong to the free-surface. Inaccuracies in the detection of boundary particles can lead to undesirable results such as unrealistic artifacts during simulation. Moreover, rendering mechanisms can greatly benefit from the information of boundary particles, reinforcing the need of an effective detection of boundary particle.

Despite its importance, the problem of accurately detecting

boundary particles has not been extensively addressed in the literature. One of the first attempt to properly tackle the problem is the work by Dilts [Dil00], where a geometric approach is proposed to identify boundary particles in 2D meshfree simulations. However, Dilts' method is computationally involved, making its extension to 3D not so straightforward [HD07]. Müller et al. [MCG03] rely on a simple approach based on the SPH gradient of a color field defined on the particles. A main drawback of using color field gradients is its instability when facing non-uniform particle distribution, making hard the task of finding a global gradient magnitude threshold to properly classify particles as boundary or non-boundary. Zhang et al. [ZSP08] proposed to identify boundary particles based on the distance between a particle and the center of mass of its neighbors. He et al. [HLW*12] improved Zhang's method by adding the SPH particle density information in the boundary/non-boundary test. However, the use of a kernel function to estimate densities and center of masses makes the method sensitive to uneven particle distribution. In order to simulate fluid surface transport, Orthmann et al. [OHB*13] compute surface particles using an SPH approximation of the surface area. Their method classifies surface particles using an area threshold, resulting in layers of particles around the surface. In computational physics, Marrone et al. [MCLTG10] performs boundary/non-boundary particle classification based on the spectrum of the SPH kernel correction matrix combined with a conical region scan defined from normal vector information. However, Marrone's methodology is restricted to SPH-based numerical simulation with fairly uniform distribution of particles.

In this work we propose a novel method to detect boundary particles in meshfree fluid flow simulations. Our approach converts the boundary particle identification problem in a visibility test problem, employing the well-known *Hidden Point Removal* (HPR) operator proposed by Katz et al. [KTB07] to identify the particles that belong to the free-surface of the fluid. Figure 1 shows our method in action.

Contributions. In contrast to previous techniques, the proposed method is purely geometric, requiring only particle positions (without normal) to perform the boundary/non-boundary particle classification. In other words, our methodology does not demand any kernel-based interpolation of physical and geometrical properties. Moreover, besides being computationally efficient and easy to implement, our approach is able to deal with non-uniform particle distributions and it is robust to large free-surface deformations.

The effectiveness of our approach is shown through a set of comparisons against state-of-the-art of boundary particle detection techniques. The good performance of the proposed method is also confirmed in experiments involving free-surfaces generated from level-sets defined from boundary particles.

2. Boundary and Visibility

Let \mathcal{P} be a set of points (particles) sampling a compact region $\Omega \subset \mathbb{R}^d$ and $S = \partial\Omega$ be the boundary surface (free-surface in the case of a fluid) of Ω . Our main goal is to identify the subset of points $\mathbf{p} \in \mathcal{P}$ that lie on S , called *boundary points*. Boundary points can be characterized by their visibility, that is, a point \mathbf{p} is a boundary point if there is a viewpoint $\mathbf{V} \notin \Omega$ such that the line segment connecting

\mathbf{p} and \mathbf{V} intersects Ω only in \mathbf{p} . Intuitively, the visibility criterion says that points $\mathbf{p} \in \text{int}(\Omega)$ are "hidden" from \mathbf{V} by S , therefore, only points on S are visible.

Although intuitive, the visibility test as stated above is not useful in practice, as the region defined by Ω is unknown (only information about the particles are available). In order to precisely define boundary particles in a meshfree context we rely on the concept of ρ -sampling [KTB07]. A point set \mathcal{P} is a ρ -sampling of a domain Ω if for any $\mathbf{x} \in \Omega$ there exist $\mathbf{p} \in \mathcal{P}$, such that $\|\mathbf{x} - \mathbf{p}\| < \rho$. Assuming a ρ -sampling \mathcal{P} , consider the set

$$M_\rho = \{\mathbf{x} \in \mathbb{R}^d \mid \text{dist}(\mathbf{x}, \mathcal{P}) < \rho, \mathcal{P} \subset \Omega\}. \quad (1)$$

Let $B_\rho(\mathbf{p})$ denotes an open ball of radius ρ centered at \mathbf{p} . A point $\mathbf{p} \in \mathcal{P}$ is called ρ -interior if $\partial B_\rho(\mathbf{p}) \subset M_\rho$, where $\partial B_\rho(\mathbf{p})$ is the boundary of $B_\rho(\mathbf{p})$. Otherwise, \mathbf{p} is called ρ -boundary and we denote by $\mathcal{B}_\rho \subset \mathcal{P}$ the set of all ρ -boundary particles (see Figure 2).

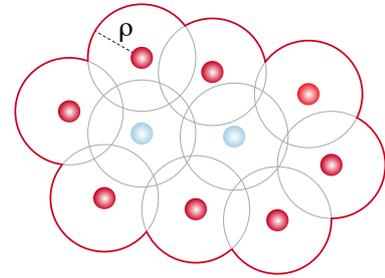


Figure 2: The ρ -boundary particles (red) and ρ -interior particles (blue) from \mathcal{P} .

Computing the ρ -boundary set \mathcal{B}_ρ is a laborious, intricate, and computationally intensive task [HD07]. An alternative is to approximate \mathcal{B}_ρ rather than compute it directly. A tenable option is to approximate \mathcal{B}_ρ using the hidden point removal (HPR) operator [KTB07], which aims at identifying visible points on point-set surfaces.

The HPR operator comprises two main steps: inversion and convex hull computation. In the inversion step each point $\mathbf{p} \in \mathcal{P}$ is mapped to an "inverted" domain as to a viewpoint \mathbf{V} , that is, points closer to \mathbf{V} are mapped far away while points distant from \mathbf{V} tend to be placed closer to \mathbf{V} . Denoting the set of inverted points as $\hat{\mathcal{P}}$, a point \mathbf{p} is labeled as visible if its inverted image $\hat{\mathbf{p}}$ is a vertex of the convex hull of $\hat{\mathcal{P}} \cup \{\mathbf{V}\}$. The second step of the HPR algorithm computes the convex hull of inverted points, returning the vertices of the resulting polyhedron as the visible points.

Spherical flipping [KTB07] is possibly the most widely used inversion mapping in the context of point-set surfaces. However, when dealing with volumetric particle distributions, the spherical flipping does not present satisfactory results, as illustrated in Figure 3a, where ρ -interior points end up being labeled as visible when the spherical flipping mapping is employed.

In order to better handle volumetrically distributed particles we opt to the *exponential flipping* inversion mapping. Assuming a viewpoint placed at origin $\mathbf{0}$ and that $\|\mathbf{p}\| < 1$, the exponential flip-

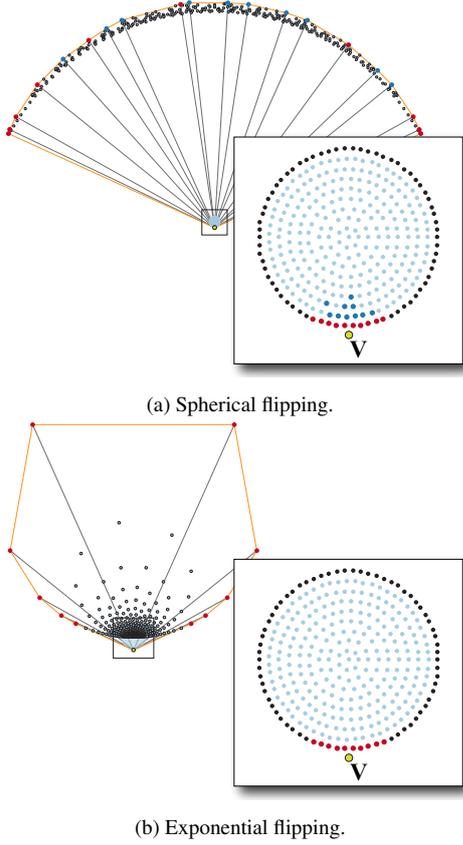


Figure 3: Boundary particles (red) detected from a viewpoint \mathbf{V} (green): (a) the effect of HPR operator using spherical flipping, (b) HPR operator using exponential flipping and their correspondent convex hull of $\hat{\mathcal{P}} \cup \{\mathbf{V}\}$ (orange) on top left. Note that, the spherical flipping can produce misclassified points (dark blue).

ping is defined as [KTB07]:

$$f(\mathbf{p}) = \begin{cases} \frac{\mathbf{p}}{\|\mathbf{p}\|^\gamma} & \text{if } \|\mathbf{p}\| \neq 0 \\ \mathbf{0} & \text{otherwise} \end{cases}, \quad (2)$$

where $\gamma > 1$ is a parameter ($\gamma = 1.3$ in our implementation). Figure 3b shows particles labeled as visible when using the exponential flipping inversion mapping, a result clearly better than in Figure 3a. As depicted in Figure 3b, the exponential flipping performs better because it generates larger and more “tangential” displacements, tending to place visible points on the border of the convex hull of inverted points.

An important aspect to be analyzed is whether the set of visible points obtained from the HPR algorithm is indeed a good approximation to the ρ -boundary set \mathcal{B}_ρ . The following lemma provides an answer to this question by establishing a relation between visible and ρ -boundary points (see proof in Appendix):

Lemma 2.1 Let \mathcal{P} be a ρ -sampling of a domain Ω and \mathbf{p} be a ρ -boundary point, that is, $\mathbf{p} \in \mathcal{B}_\rho$. There exists a viewpoint \mathbf{V} such that \mathbf{p} is in the set of visible points $\mathcal{H}_V(\mathcal{P})$ resulting from HPR algorithm.

Lemma 2.1 provides theoretical guarantees on the existence of viewpoints from which one can obtain approximations to \mathcal{B}_ρ , however, its implementation is cumbersome and computationally intensive. Next section tackles the problem of creating viewpoints that produce satisfactory results in a computationally efficient manner.

3. The Proposed Method

Given a ρ -sampling set of points \mathcal{P} , we have to choose viewpoint positions from which the HPR algorithm can properly identify boundary points. Since our approach rely on HPR to perform the boundary/non-boundary classification, it only requires the coordinates of the points as input, disregarding physical attributes typically used by other approaches.

As illustrated in Figure 4, two main steps are involved in our boundary identification process: viewpoint placement (green dots in Figure 4) and the visibility test (red dots in Figure 4). The following subsections detail both steps of our methodology.

3.1. Viewpoint Placement

In order to place viewpoints in appropriate locations, we first define a Cartesian uniform grid \mathcal{G} covering the bounding box of \mathcal{P} , setting the edge length of each cubic (square in 2D) cell equal to 2ρ . A grid cell C is labeled *empty* if it contains no particles in its interior, otherwise, C is labeled *full*.

Viewpoints must be placed outside the domain Ω (the fluid domain in our case). For this reason, empty cells adjacent to at least one full cell become natural candidates to host viewpoints. Therefore, we place a viewpoint in the centroid of each empty cell adjacent to a full cell. However, cavities with a diameter at least 2ρ may be present inside or between full cells and the points on the boundary of those cavities must also be identified. We mark a full cell C as a *cavity cell* if it satisfies the two conditions below:

- (a) all cells adjacent to C are full,
- (b) there is a ball with center in C and radius ρ that contains no point of \mathcal{P} in its interior.

Condition (a) is straightforward to check. In order to verify condition (b), for each point \mathbf{p}_i inside C a candidate viewpoint \mathbf{V}_i is located as follows:

$$\mathbf{V}_i = \begin{cases} \mathbf{p}_i + \rho \frac{\boldsymbol{\delta}_i}{\|\boldsymbol{\delta}_i\|} & \text{if } \|\boldsymbol{\delta}_i\| \neq 0 \\ \mathbf{p}_i & \text{otherwise} \end{cases} \quad (3)$$

with

$$\boldsymbol{\delta}_i = \mathbf{p}_i - \frac{1}{|N_i|} \sum_{j \in N_i} \mathbf{p}_j,$$

where j indexes the points \mathbf{p}_j lying in the neighborhood N_i of radius 2ρ from \mathbf{p}_i . Then, we check the emptiness of each ball $B_\rho(\mathbf{V}_i)$. If $B_\rho(\mathbf{V}_i)$ is empty, its center is chosen as a viewpoint. In terms of computational aspects, we decrease the diameter of $B_\rho(\mathbf{V}_i)$ by multiplying a scale factor of 0.95 due to the non-uniform particle distribution. Figure 5 shows the viewpoints generated in the cavity cells, note that a cavity cell can admit multiple viewpoints.

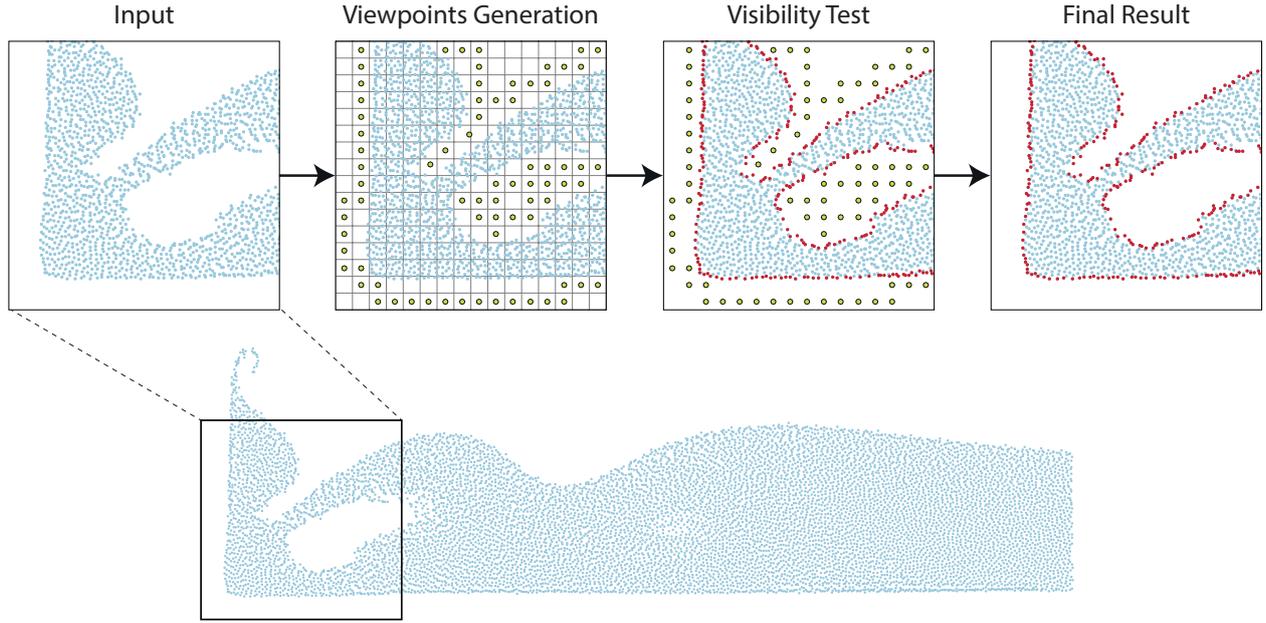


Figure 4: Overview of our pipeline in a dam break simulation.

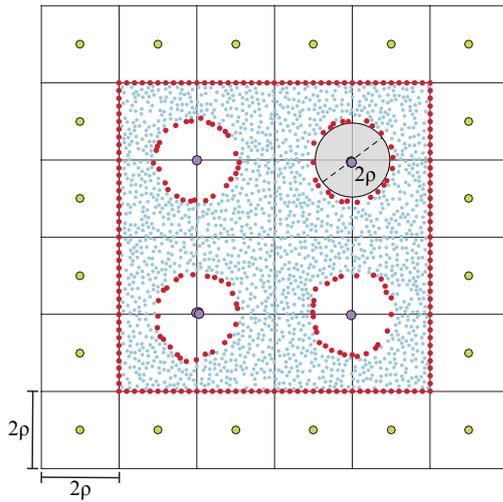


Figure 5: Viewpoints generated in empty cells (green) are not enough to detect boundary particles (red) in internal cavities with diameter at least 2ρ (gray). While, viewpoints generated in cavity cells (purple) are able to detect the remaining boundary particles.

3.2. Visibility Test

Once the viewpoints have been settled, the HPR operator should be locally applied as discussed in Section 2. For each viewpoint \mathbf{V}_i , the set of points \mathcal{P}_i within the ball centered in \mathbf{V}_i with radius 4ρ are picked out for inversion. In order to ensure the HPR requirements, apply an affine transformation to $\mathcal{P}_i \cup \{\mathbf{V}_i\}$ such that \mathbf{V}_i is moved to the origin and \mathcal{P}_i is scaled by a factor of $1/4\rho$. After inversion, the convex hull of the inverted points $\hat{\mathcal{P}}_i \cup \{\mathbf{V}_i\}$ is computed using

Algorithm 1: Boundary Particle Detection

Data: (\mathcal{P}, ρ)

Result: \mathcal{B}_ρ

build a uniform grid \mathcal{G} of cell size 2ρ ;

insert the points of \mathcal{P} in \mathcal{G} ;

foreach cell $C \in \mathcal{G}$ **do**

if C is empty and has an adjacent cell that is full **then**
 place a new viewpoint at the centroid of C ;

else if C is a cavity cell **then**

foreach point $\mathbf{p}_i \in C$ **do**

 place a viewpoint \mathbf{V}_i according to Eq. (3) which satisfies the condition (b);

end

end

end

foreach viewpoint \mathbf{V}_i **do**

 run the visibility test with all points within radius 4ρ of \mathbf{V}_i ;
 insert the visible points in \mathcal{B}_ρ ;

end

return \mathcal{B}_ρ

the QuickHull algorithm [BDH96]. The union of all points marked as visible after applying the HPR operator from each \mathbf{V}_i gives rise to the set of the boundary points \mathcal{B}_ρ .

The proposed method can be summarized by the Algorithm 1.

4. Results

We implemented our technique in C++ using OpenMP [Ope11] under the Microsoft Visual Studio 2013 for Windows. All experiments

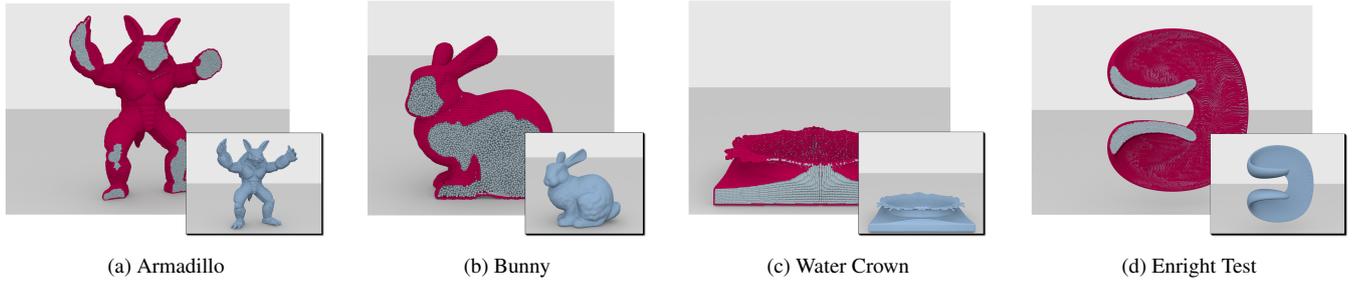


Figure 6: Boundary particles (red) detected by our method. The interior points (gray) are shown in a cutaway view. At bottom-right, their reconstructed surface.

have been performed on an Intel Xeon processor E5620 with four 2.4 GHz cores and 12 GB RAM. Table 1 shows statistics as to the number of particles ($|\mathcal{P}|$), the number of viewpoints (ϑ) generated by our method, the ratio between the number of boundary particles (Bd) and interior particles (Int), and computational times for all 3D experiments presented in this and the following section.

Table 1: Average statistics and computational times (in seconds) per frame.

Dataset	$ \mathcal{P} $	ϑ	Ratio (Bd:Int)	Exec. Time
Bunny	114K	4K	1:2.3	0.9
Double Dam Break	275K	8K	1:9.3	1.6
Water Crown	297K	8K	1:5.7	1.8
Armadillo	797K	18K	1:3.7	7.5
Enright Test	1904K	13K	1:16.8	44.7

The parameter ρ used to define the underlying grid (see subsection 3.1) depends on numerical and spatial resolution of the problem. In SPH simulations a good choice for ρ is the *smoothing length*, which defines the radius of influence of the kernel function [HD07]. In our experiments, all particle-based fluid simulations were produced using a weakly compressible SPH formulation implemented in SPHysics [GGRC*12]. The volumetric particle models for Armadillo and Bunny were created by first generating tetrahedral meshes from the corresponding triangular surface meshes using Gmsh [GR09], getting the vertices of the tetrahedra as particles. Tetrahedron element size and the ρ parameter were chosen as the average edge length of the input triangular surface meshes (see Figure 6).

The effectiveness of our approach (denoted by “Ours”) is assessed through a set of qualitative and quantitative comparisons against four existing surface detection techniques. More precisely, we compare our approach against the methods proposed by Müller et al. [MCG03] (Müller), He et al. [HLW*12] (He) and Orthmann et al. [OHB*13] (Orthmann), which are well known by the computer graphics community. We also compare against the method proposed by [MCLTG10] (Marrone), which is the state-of-art in computational physics. All those four techniques were applied using parameters as suggested in the corresponding papers. Figures 7 and 8 enable qualitative comparisons by showing boundary points resulting from each technique. Notice that our approach is able to

capture sharp and thin features better than other methods while producing a reduced number of misclassified interior points.

For the sake of quantitative comparisons, we apply the method proposed by Haque and Dilts [HD07] (and its 2D version [Di00]) to label boundary/non-boundary particle, which we will consider as ground truth. The classification provided by Haque and Dilts is fairly reliable, as the method computes, for a given ρ , the “exact” intersection of spheres with radius ρ centered at each particle, identifying ρ -interior and ρ -boundary particles as defined in Section 2. Although very accurate, Haque and Dilts’ method is not scalable for practical applications, as its computational complexity is $O(|\mathcal{P}|n^\epsilon)$, where n is the average number of particles inside spheres with radius 2ρ centered at the particles and $\epsilon \in [1.6, 2.0]$

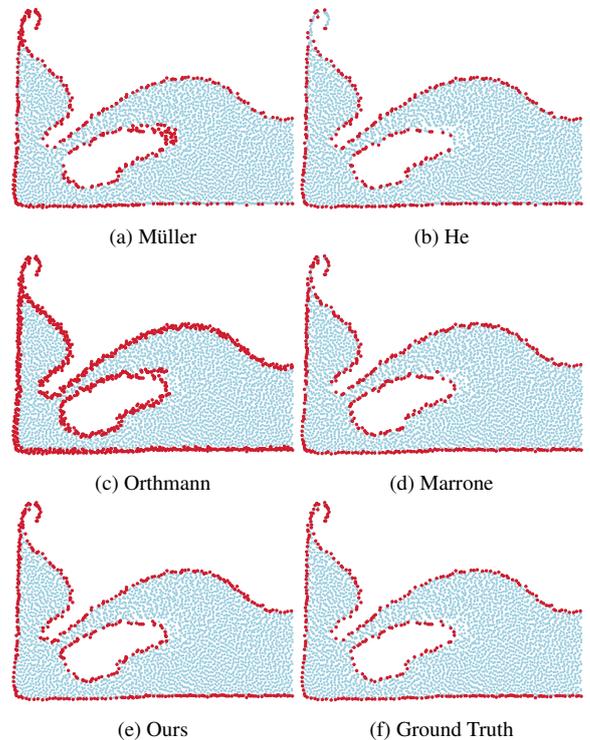


Figure 7: Comparison between different boundary detection methods in a 2D dam break simulation.

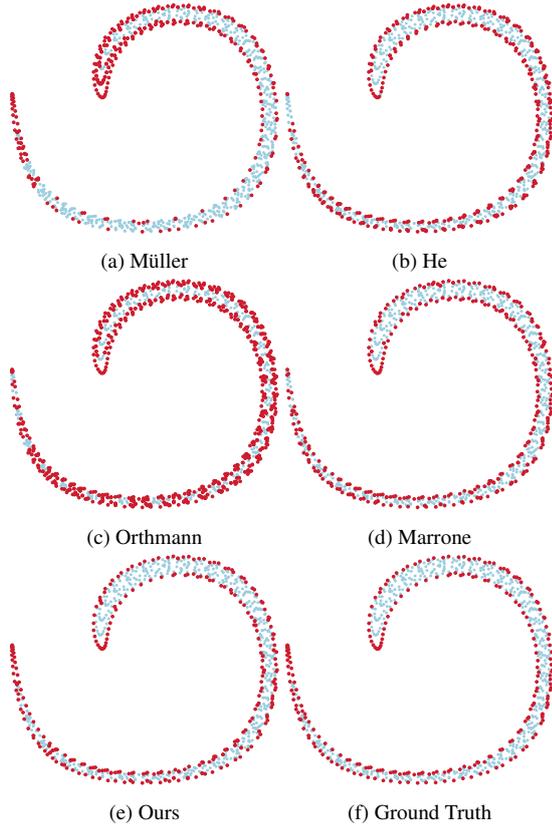


Figure 8: Comparison between different boundary detection methods in a 2D *single vortex* experiment, as detailed in [EFFM02].

is a problem-dependent constant. Our approach, in contrast, inherits the complexity of the local convex hull computations, resulting in a complexity $O(\vartheta k \log k)$, where ϑ is the number of viewpoints and k is the maximum number of particles used as input the convex hull. Although in the worst case (a full convex hull for each point) our approach is also costly, that worst case is very rare, specially in real applications involving a large number of particles. Given a reference classification, we can assess the accuracy of each method using quantitative metrics commonly employed in data classification [Pow11].

Specifically, we assign each particle to one of four categories:

- *True Positive (TP)*: a ρ -boundary point correctly classified;
- *True Negative (TN)*: a ρ -interior point correctly classified;
- *False Positive (FP)*: a ρ -interior point classified as ρ -boundary;
- *False Negative (FN)*: a ρ -boundary point classified as ρ -interior.

The *Recall*, is given by

$$Rec = \frac{TP}{TP + FN}.$$

measures how well a technique performs when detecting boundary particles among the actual set of ρ -boundary particles. The best possible result is $Rec = 1$, meaning that all boundary particles were detected correctly, although false positives can be included. The main issue caused by false positives is that they thicken the

boundary surface, thus hampering tasks such as surface reconstruction. On the other hand, the *False Positive Rate (FPR)* quantifies how many interior particles were classified as boundary. Mathematically, FPR is defined as

$$FPR = \frac{FP}{FP + TN}.$$

The best scenario occurs when $FPR = 0$, i.e., no interior particle has been classified as boundary. When employed separately those two metrics may lead to a wrong analysis. For instance, the maximum recall can be achieved by classifying every particle in \mathcal{P} as boundary, regardless whether this is true or not. FPR can be minimized when the whole set of particles is classified as interior. Therefore, we combined both metrics as follows:

$$M_C = Rec * (1 - FPR).$$

The combined metric M_C reaches its maximum value 1 when the Recall is maximum and FPR is minimum simultaneously. Table 2 shows M_C scores resulting from each of the five techniques. Notice that our approach outperforms the four other methods in all experiments.

Table 2: Quantitative analysis between different boundary detection techniques (best results are shown in bold).

	Dataset	Method	M_C
2D cases	Dam Break (Fig. 7)	Müller	0.796
		He	0.795
		Orthmann	0.885
		Marrone	0.949
		Ours	0.978
	Single Vortex (Fig. 8)	Müller	0.256
		He	0.708
		Orthmann	0.596
		Marrone	0.873
		Ours	0.944
3D cases	Armadillo (Fig. 6a)	Müller	0.000
		He	0.078
		Orthmann	0.822
		Marrone	0.947
		Ours	0.995
	Bunny (Fig. 6b)	Müller	0.017
		He	0.027
		Orthmann	0.740
		Marrone	0.921
		Ours	0.994
	Water Crown (Fig. 6c)	Müller	0.058
		He	0.848
		Orthmann	0.931
		Marrone	0.957
		Ours	0.971
	Enright Test (Fig. 6d)	Müller	0.926
		He	0.938
		Orthmann	0.904
Marrone		0.968	
Ours		0.989	
Double Dam Break (Fig. 11)	Müller	0.589	
	He	0.074	
	Orthmann	0.919	
	Marrone	0.892	
	Ours	0.974	

5. Applications

One of the main advantages of correctly classifying boundary particles is the possibility of using point-set surface reconstruction algorithms [BTS*14] to generate the free-surface of particle-based fluids, instead of SPH-based surface reconstruction [MCG03, SSP07, YT13] which requires a weighted summation over all neighboring particles. The free-surface S is given implicitly by the zero level-set of a signed distance field $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$. Besides the identified boundary particles, the construction of ϕ demands oriented surface normals. Many algorithms have been proposed to compute oriented normals. In our case, the traditional SPH approximation proposed by Müller et al. [MCG03] is used for estimating oriented normals. Thus, the normal at a boundary particle i is computed as

$$\mathbf{n}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|} \quad \text{with} \quad \mathbf{u}_i = \sum_{j \in \mathcal{N}_i} \nabla W_h(\|\mathbf{p}_i - \mathbf{p}_j\|),$$

where W_h is the Gaussian kernel function with smoothing length h . Figure 9 shows a 2D example of our boundary detection (BD) method and the computed surface normals (Figure 9a). The smooth signed distance field ϕ defined using *Radial Basis Functions* (RBF) [CBC*01] and its corresponding surface $S = \phi^{-1}(0)$ is depicted in Figure 9b.

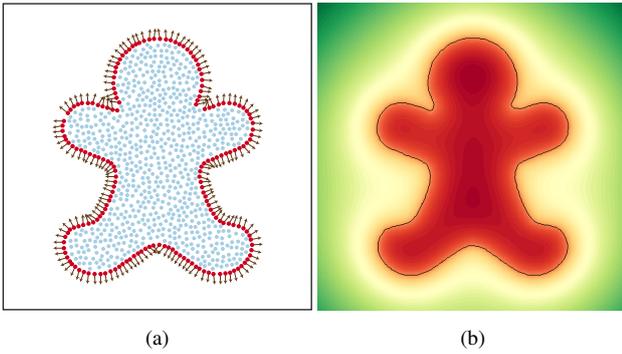


Figure 9: Surface reconstruction of gingerbread man: (a) boundary particles (red) with oriented normals (brown) and (b) the signed distance field ϕ using RBF, varying from negative (red) to positive (green) values and its zero level-set (black).

Figure 10 compares level-sets generated by our BD with RBF implicit [CBC*01] against the *Particle Level Set* (PLS) method [EFFM02], a well known level-set method employed in Eulerian grid-based simulations. The experiment shows the resulting boundary surface after a 360° degree rotation of the classical *Zalesak's disc* with fourth-order Runge-Kutta for time integration [Zal79]. Notice that the surface resulting from our method is quite close to that generated by PLS, preserving the sharp features and avoiding numerical diffusion (mass loss) at the interface. Moreover, in contrast to PLS, our approach does not demand any level-set re-initialization nor outer marker particles, making it a good asset for any point-set surface reconstruction method.

Figure 11 shows our BD method combined with *Screened Poisson surface reconstruction* [KH13] to compute the level-set surface that represents the liquid interface. Notice that due to the good performance of our BD method, the level-set obtained by the Screened

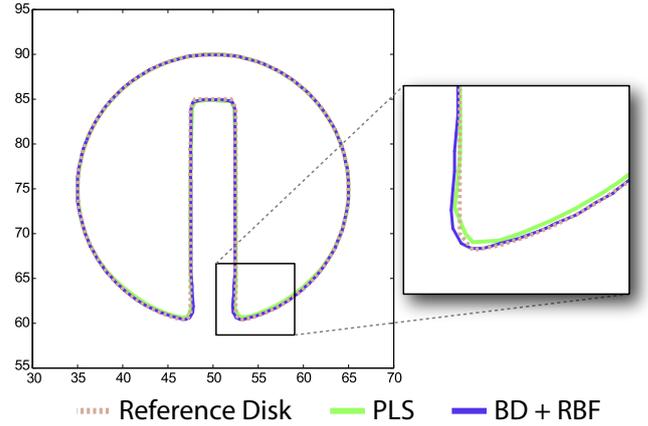


Figure 10: Zalesak's disk after one full rotation: our method with 800 particles and PLS on a 100×100 resolution grid.

Poisson method nicely captures droplets and thin sheets of water (usually formed by a thin layer of particles) in a double dam break simulation. In all 3D experiments shown in this section we used the Screened Poisson as surface reconstruction method (notice that Screened Poisson can not operate in volumes).

Figure 12 shows the effectiveness of our approach when facing the *Enright test* [EFFM02], where a sphere is advected by a velocity field which induces large shape deformation. Notice that the sphere is preserved even after a stretching followed by a compression. In this example, we used fourth-order Runge-Kutta to compute particles trajectories.

Figure 13 shows a comparison between our approach and the surface reconstruction method proposed by Zhu and Bridson [ZB05], which also operates on the particle volume. One can easily see that Zhu and Bridson's method introduces surface bumps due to the irregular particle distribution. Those artifacts are not present in the reconstruction generated from our approach. We use an implementation of Marching Cubes (MC) algorithm [LLVT03] to extract the isosurface from the output of Zhu and Bridson's method, defining the MC grid size from the particle spacing [AIAT12]. Moreover, the surface reconstruction with Screened Poisson took a similar time of Zhu and Bridson's method, approximately 5 seconds per frame.

A common strategy when dealing with particle volumes is to perform surface reconstruction directly from the volume defined by the particles. Bhattacharya et al. [BGB15] proposed a technique for skinning particle data that relies on a biharmonic-smoothed distance field constrained between two-offset surfaces obtained from the particles. Figure 14 presents surfaces reconstructed from a volumetric Armadillo model using our approach and the particle skinning proposed by Bhattacharya. Notice that the reconstruction generated by Screened Poisson from the boundary surface captured by our method contains most of the details and sharp features present in the original model. Bhattacharya et al. technique, in contrast, produced a smoother reconstruction. In this experiment, our BD method with Screened Poisson took approximately 16 seconds (BD corresponds to $\sim 47\%$ of the time) against 105 seconds of Bhattacharya's method.

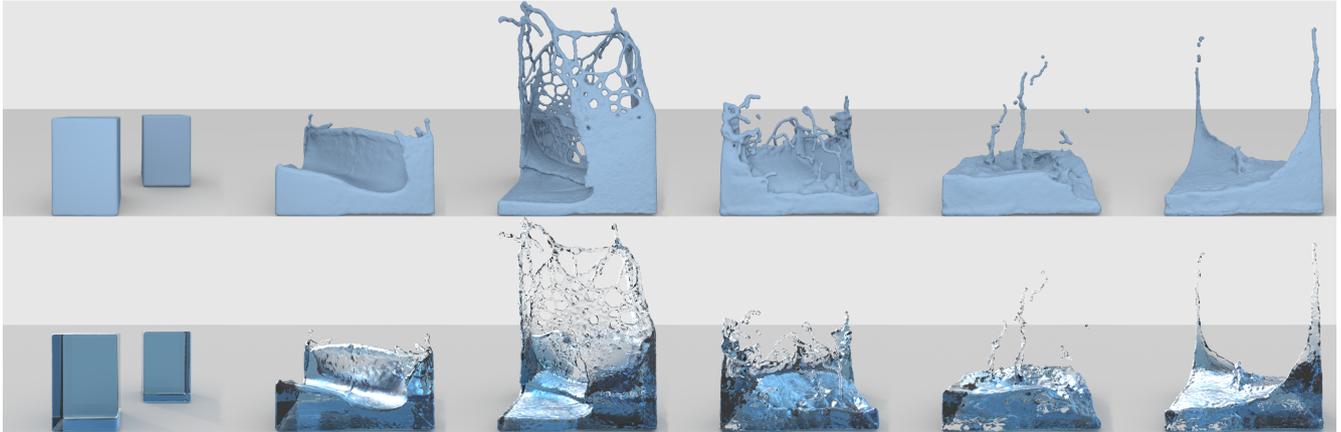


Figure 11: The free-surface reconstruction of a double dam break simulation with opaque (top) and transparent (bottom) renderings.

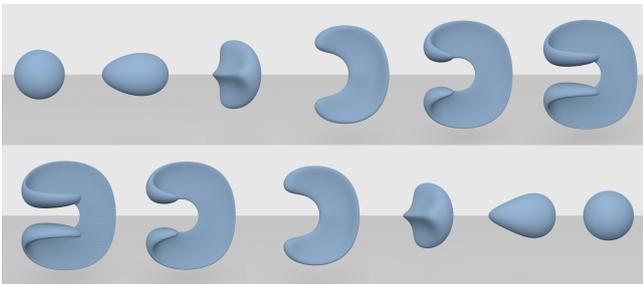


Figure 12: Enright test: the shape of sphere is preserved by our approach despite large deformations (start: top left, end: bottom right).

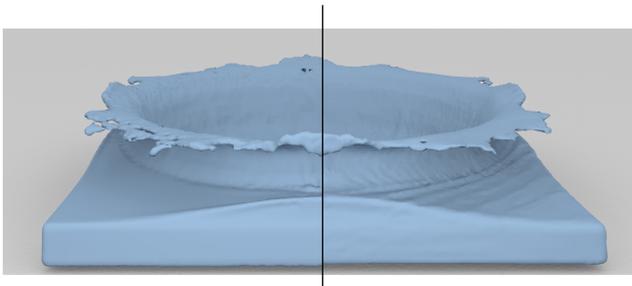


Figure 13: Comparison with our method (left) and the surface reconstruction proposed by Zhu and Bridson [ZB05] (right) on the water crown formed by the impact of a drop on a water layer.

6. Discussion and Limitations

We have experimentally found that good results are reached when setting $\gamma = 1.3$ in the exponential flipping. We use this parameter value in all experiments presented in this paper. However, misclassification can show up in concave regions with high curvature, resulting in false negatives. A theoretical upper-bound for the curvature (cf. Lemma 3.1 in [KT13]) can be established in terms of γ and surface normals, i.e., HPR is guaranteed to produce correct results if the curvature is not larger than an upper-bound.

The parameter ρ defining the underlying grid size employed our BD method can also be tricky to be specified when information about particle spacing is not available. If particles are provided with no additional information, it is very hard to make any assumption about the topology of the surface and thus proper value to set ρ . We believe that it is possible to establish a relation between ρ and the *local feature size* [AB98] of the points, but this is not a straightforward task.

7. Conclusion and Future Work

We presented a novel method to detect free-surface on particle-base volumes. The method relies on HPR visibility test and in the concept of ρ -sampling to identify particles laying on the boundary of a volume sampled by particles. Comparisons against state-of-the-art methods show the effectiveness of our methodology, which can be used together with most point set-surface reconstruction method to produce accurate surface models.

As future work, we would like to build our boundary detection method into meshfree solvers, especially in simulations of physical phenomena involving complex free-surface phenomena. In particular, we believe that particle-based simulations involving surface tension [AAT13] and surface turbulence [MBT⁺15] can greatly benefit from our methodology. In terms of rendering, we want to investigate the surface reconstruction in screen space using an image-based implementation of the HPR operator [MeSEMO14]. Finally, since our method is built upon local visibility tests, adapting it to GPU architecture is feasible, being another direction for future work.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and the Center for Mathematical Sciences Applied to Industry (CEPID-CeMEAI) for providing the computational resources. The authors are supported by CNPq (fellowships #302643/2013-3 and #305796/2013-5) and São Paulo Research Foundation (FAPESP) grants #11/12263-0, #11/22749-8 and #14/09546-9.

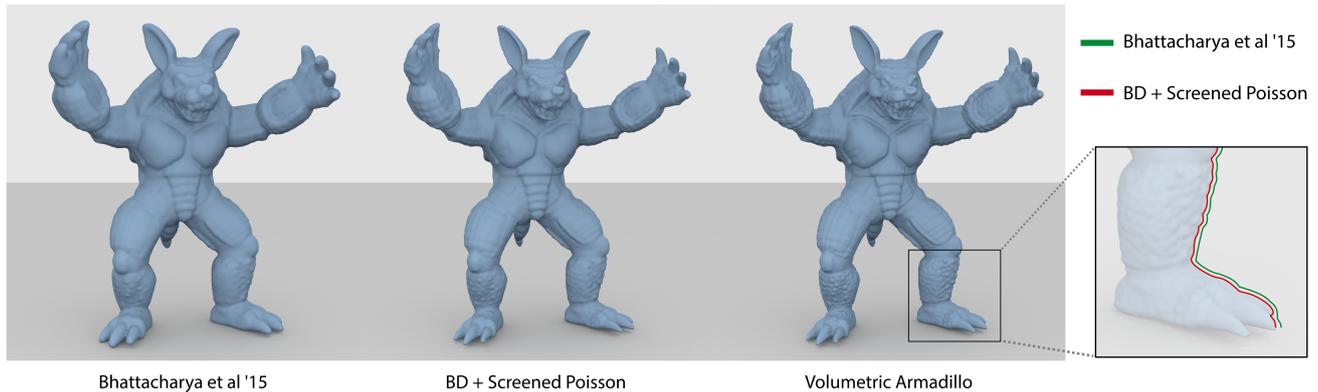


Figure 14: Comparison of our strategy (middle) against Particle Skinner (left) proposed by Bhattacharya et al. [BGB15]. At the rightmost, the silhouette profiles of each reconstruction strategy is shown in detail against the reference volumetric Armadillo.

References

- [AAT13] AKINCI N., AKINCI G., TESCHNER M.: Versatile surface tension and adhesion for sph fluids. *ACM Trans. Graph.* 32, 6 (2013), 182:1–182:8. 8
- [AB98] AMENTA N., BERN M.: Surface reconstruction by voronoi filtering. In *Proceedings of Symposium on Computational Geometry (SCG '98)* (1998), pp. 39–48. 8
- [AIAT12] AKINCI G., IHMSEN M., AKINCI N., TESCHNER M.: Parallel surface reconstruction for particle-based fluids. *Computer Graphics Forum* 31, 6 (2012), 1797–1809. 7
- [BDH96] BARBER C. B., DOBKIN D. P., HUHDANPAA H.: The quick-hull algorithm for convex hulls. *ACM Trans. Math. Softw.* 22, 4 (1996), 469–483. 4
- [BGB15] BHATTACHARYA H., GAO Y., BARGTEIL A. W.: A level-set method for skinning animated particle data. *IEEE Trans. Vis. Comput. Graph.* 21 (2015), 315–327. 7, 9
- [BTS*14] BERGER M., TAGLIASACCHI A., SEVERSKY L. M., ALLIEZ P., LEVINE J. A., SHARF A., SILVA C.: State of the art in surface reconstruction from point clouds. In *Eurographics STAR (Proc. of EG'14)* (2014). 7
- [CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of SIGGRAPH '01* (2001), pp. 67–76. 7
- [Dil00] DILTS G. A.: Moving least-squares particle hydrodynamics II: conservation and boundaries. *Int. J. Numer. Meth. Eng.* 48, 10 (2000), 1503–1524. 2, 5
- [EFFM02] ENRIGHT D., FEDKIW R., FERZIGER J., MITCHELL I.: A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* 183, 1 (2002), 83–116. 6, 7
- [GGRC*12] GOMEZ-GESTEIRA M., ROGERS B. D., CRESPO A. J. C., DALRYMPLE R. A., NARAYANASWAMY M., DOMINGUEZ J. M.: SPHysics - development of a free-surface fluid solver - part 1: theory and formulations. *Comput. Geosci.* 48 (2012), 289–299. 5
- [GR09] GEUZAIN C., REMACLE J.-F. A.: Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Meth. Eng.* 79, 11 (2009), 1309–1331. 5
- [HD07] HAQUE A., DILTS G. A.: Three-dimensional boundary detection for particle methods. *J. Comput. Phys.* 226, 2 (2007), 1710 – 1730. 2, 5
- [HLW*12] HE X., LIU N., WANG G., ZHANG F., LI S., SHAO S., WANG H.: Staggered meshless solid-fluid coupling. *ACM Trans. Graph.* 31, 6 (2012), 149:1–149:12. 2, 5
- [IOS*14] IHMSEN M., ORTHMANN J., SOLENTHALER B., KOLB A., TESCHNER M.: SPH Fluids in Computer Graphics. In *Eurographics STAR (Proc. of EG'14)* (2014). 1
- [KH13] KAZHDAN M., HOPPE H.: Screened poisson surface reconstruction. *ACM Trans. Graph.* 32, 3 (2013), 29:1–29:13. 7
- [KT13] KATZ S., TAL A.: Improving the visual comprehension of point sets. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on* (2013), pp. 121–128. 8
- [KTB07] KATZ S., TAL A., BASRI R.: Direct visibility of point sets. *ACM Trans. Graph.* 26, 3 (2007). 2, 3
- [LLVT03] LEWINER T., LOPES H., VIEIRA A. W., TAVARES G.: Efficient implementation of marching cubes cases with topological guarantees. *Journal of Graphics Tools* 8, 2 (2003), 1–15. 7
- [MBT*15] MERCIER O., BEAUCHEMIN C., THUREY N., KIM T., NOWROUZEZAHRAI D.: Surface turbulence for particle-based liquid simulations. *ACM Trans. Graph.* 34, 6 (2015), 202:1–202:10. 8
- [MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'03)* (2003), pp. 154–159. 2, 5, 7
- [MCLTG10] MARRONE S., COLAGROSSI A., LE TOUZÉ D., GRAZIANI G.: Fast free-surface detection and level-set function definition in SPH solvers. *J. Comput. Phys.* 229, 10 (2010), 3652–3663. 2, 5
- [MeSEMO14] MACHADO E SILVA R., ESPERANÇA C., MARROQUIM R., OLIVEIRA A. A. F.: Image space rendering of point clouds using the HPR operator. *Computer Graphics Forum* 33, 1 (2014), 178–189. 8
- [MM13] MACKLIN M., MÜLLER M.: Position based fluids. *ACM Trans. Graph.* 32, 4 (2013), 104:1–104:12. 1
- [OHB*13] ORTHMANN J., HOCHSTETTER H., BADER J., BAYRAKTAR S., KOLB A.: Consistent surface model for SPH-based fluid transport. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'13)* (2013), pp. 95–103. 2, 5
- [Ope11] OPENMP ARCHITECTURE REVIEW BOARD: OpenMP application program interface version 3.1, July 2011. 4
- [Pow11] POWERS D. M. W.: Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies* 2, 1 (2011), 37–63. 6
- [PTB*03] PREMZOE S., TASDIZEN T., BIGLER J., LEFOHN A., WHITAKER R. T.: Particle-based simulation of fluids. *Computer Graphics Forum* 22, 3 (2003), 401–410. 1

- [SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid–solid interactions. *Comput. Animat. Virtual Worlds* 18, 1 (2007), 69–82. 7
- [YT13] YU J., TURK G.: Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.* 32, 1 (2013), 5:1–5:12. 7
- [Zal79] ZALESAK S. T.: Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* 31, 3 (1979), 335–362. 7
- [ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph.* 24, 3 (2005), 965–972. 1, 7, 8
- [ZSP08] ZHANG Y., SOLENTHALER B., PAJAROLA R.: Adaptive sampling and rendering of fluids on the GPU. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on Point-Based Graphics (PBG'08)* (2008), pp. 137–146. 2

Appendix

Lemma 2.1 Let \mathcal{P} be a ρ -sampling of a domain Ω and \mathbf{p} be a ρ -boundary point, that is, $\mathbf{p} \in \mathcal{B}_\rho$. There exists a viewpoint \mathbf{V} such that \mathbf{p} is in the set of visible points $\mathcal{H}_\mathbf{V}(\mathcal{P})$ resulting from HPR algorithm.

Proof Since $\mathbf{p} \in \mathcal{B}_\rho$, there is at least one point $\mathbf{x} \in \partial B_\rho(\mathbf{p})$ that is not covered by any other ball $B_\rho(\mathbf{q})$ centered in $\mathbf{q} \in \mathcal{P} \setminus \{\mathbf{p}\}$. Therefore, a viewpoint \mathbf{V} placed on \mathbf{x} is closer to \mathbf{p} than any other $\mathbf{q} \in \mathcal{P} \setminus \{\mathbf{p}\}$. By construction, the exponential flipping f is strictly monotonically decreasing along each ray from \mathbf{V} , then the inversion mapping will take \mathbf{p} farther away from \mathbf{V} than other point in $\mathcal{P} \setminus \{\mathbf{p}\}$. Thus, ensuring that $\hat{\mathbf{p}} = f(\mathbf{p})$ lies on the boundary of the convex hull of $\hat{\mathcal{P}} \cup \{\mathbf{V}\}$, i.e., $\mathbf{p} \in \mathcal{H}_\mathbf{V}(\mathcal{P})$. \square