UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Boundary particle resampling for surface reconstruction in particle-based fluids

Nicolas Masanori Shimizu Oe

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Nicolas Masanori Shimizu Oe

Boundary particle resampling for surface reconstruction in particle-based fluids

Dissertation submitted to the Institute of Mathematics
and Computer Sciences – ICMC-USP – in
accordance with the requirements of the Computer
and Mathematical Sciences Graduate Program, for
the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Prof. Dr. Afonso Paiva Neto

USP – São Carlos
December 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

O28b Oe, Nicolas Masanori Shimizu
 Boundary particle resampling for surface
 reconstruction in particle-based fluids / Nicolas
 Masanori Shimizu Oe; orientador Afonso Paiva Neto. -
 - São Carlos, 2018.
 55 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Ciências de Computação e Matemática
 Computacional) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2018.

 1. Particle resampling. 2. Boundary particles.
 3. Surface reconstruction. 4. Particle-based
 fluids. 5. Liquid animation. I. Paiva Neto, Afonso,
 orient. II. Título.

Nicolas Masanori Shimizu Oe

**Reamostragem de partículas de borda para reconstruções
de superfícies em fluidos baseados em partícula**

Dissertação apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Mestre em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Supervisor: Prof. Dr. Afonso Paiva Neto

**USP – São Carlos
Dezembro de 2018**

*This work is dedicated to the competitive programing study group (GEMA) that,
was the main motive to walk into this stage of life,
and made me wish it lasts forever.*

ACKNOWLEDGEMENTS

Firstly, I would like to give my thanks to my mother Sandra Oe, my father Carlos Oe, my sisters Bianca Oe and Marye Oe, my grandparents and all other people that raised me and made me reach where I am. My thanks to Bianca Oe again for the patience while helping me complete this thesis. Secondly, thanks to Instituto de Ciências Matemáticas e Computação (ICMC) for the opportunity to realize this work and complete another step of my life. Thanks to Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support and thanks to professor Afonso Paiva Neto for tutoring me for my Master's Degree and to my professors and ICMC's Geometric Processing group for the knowledge I've received.

I want to thank the competitive programming study group (GEMA) and all other friendships that motivated and accompanied me. A special thanks to professor João Batista for all support and dedication to the study group and to Lucas Pacheco for the extra effort in helping me during this work. Finally, I thank all my friendships that weren't part of my academic life but were with me nevertheless.

ABSTRACT

OE, N. M. **Boundary particle resampling for surface reconstruction in particle-based fluids.** 2018. 55 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

In this work, it's presented a novel particle resampling method for free-surface fitting of liquids from particle-based fluid simulations. The proposed approach is simple and easy to implement, and only requires the positions of the particles to properly identify and refine regions with small-scale features. The method comprises three main stages: boundary detection, feature classification, and particle refinement. For each simulation frame, firstly the free-surface is captured through a boundary detection scheme as chosen by the user. Then, the boundary particles are classified and labeled according to the deformation and the stretching of the free-surface computed from the Principal Component Analysis (PCA) of the particle positions. Finally, particles placed at feature regions are refined according to their feature classification. In order to render the free-surface, it's demonstrated how the traditional methods of free-surface fitting in Computer Graphics and Computational Physics literature can be benefited by the proposed resampling method. Moreover, the results shown in this work attest the effectiveness and robustness of the method when compared against state-of-the-art adaptive particle sampling techniques.

Keywords: Particle resampling, Boundary particles, Surface reconstruction, Particle-based fluids, Liquid animation.

RESUMO

OE, N. M. **Reamostragem de partículas de borda para reconstruções de superfícies em fluidos baseados em partícula**. 2018. 55 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Neste trabalho é apresentado um novo método de re-amostragem para a superfície livre de líquidos de simulações de fluidos baseados em partícula. A abordagem proposta é simples e fácil de ser implementada, e requer apenas as posições das partículas para identificar e refinar devidamente regiões com características de baixa escala. O método consiste de três estágios principais: detecção de borda, classificação de características e refinamento de partículas. Para cada quadro da simulação, primeiro a superfície livre é identificada a partir da detecção de borda, cujo método pode ser escolhido pelo usuário. Então, as partículas de borda são classificadas e recebem um valor de acordo com a deformação e alongamento da superfície livre calculada a partir da Análise de Componentes Principais (PCA) das posições das partículas. Por fim, partículas que pertencem às camadas finas são refinadas de acordo com suas classificações. Para renderizar a superfície livre, é demonstrado como métodos tradicionais de "fitting" de superfície em Computação Gráfica e Física computacional podem ser beneficiadas pelo método de re-amostragem proposta. Por fim, os resultados mostrados neste trabalho certificam a efetividade e a robustez do método quando comparado com as técnicas adaptativas de amostragem de partículas do estado-da-arte.

Palavras-chave: Re-amostragem de partículas, Partículas de borda, Reconstrução da superfície, Fluidos baseados em partículas, Animação de líquidos.

LIST OF FIGURES

1.	Comparison between different surface reconstruction approaches on the Double dam break animation. (YU; TURK, 2010).	18
2.	Free-surface fitting of a liquid splashing by using the method proposed by Sandim <i>et al.</i> (2016), without (left) and with our resampling method (right). Notice that the ligaments and thin sheets of fluid are gracefully preserved with our method.	19
3.	Flooding a valley. The particle density is dynamically adapted based on geometric complexity and visual importance (as color coded on the particles). The bottom row shows the actual camera views. The top row shows cross sections of the whole simulation domain (ADAMS <i>et al.</i> , 2007).	22
4.	Zhang, Solenthaler and Pajarola (2008) rendering results. Top row: Rendering with only a few surface particles. Bottom row: Rendering with more surface particles.	22
5.	Water drop on cubic tower. Top row: liquid solver only. Bottom row: the same simulation with our sheet preserving algorithm. The left hand side visualizes the particles, while the right hand side shows the transparently rendered mesh.	23
6.	Ando, Thürey and Wojtan (2013) adaptive simulation framework allows to efficiently simulate highly detailed splashes on large open surfaces.	24
7.	A sequence of images captured at (a, b) the 40th frame and (c, d) the 80th frame of a simulation. The surfaces reconstructed using Jang and Noh (2015) method (b, d) have fewer holes than that which underwent smooth surface reconstruction (ZHU; BRIDSON, 2005) (a, c) due to the contributions of the interparticle connections at the reconstruction stage.	25
8.	Illustration of a particle model and it's point-set. The left figure is a gingerbread with volumetric data and the right figure is it's point-set, without volumetric data.	27
9.	Overview of the pipeline of the method in a double dam breaking simulation.	31
10.	Sketch of the regions used in the <i>Sregion</i> (MARRONE <i>et al.</i> , 2010).	35
11.	The color encodes the boundary particles (blue) and the features particles in a water crown splash: small droplets (cyan), ligaments (yellow) and thin sheets (magenta).	37

12.	Particle refinement: a feature particle i (blue) is replaced by new particles (orange) according to its label ℓ_i	38
13.	Highlighted regions of the free-surface (magenta) are corrected by shifting the boundary particles of label $\ell_i = 3$	39
14.	A sampled particle (green) from a feature particle (blue) is rejected by virtue of its safe region (yellow) is not empty.	40
15.	Free-surface is enhanced by a mesh smoothing method.	42
16.	Penalty calculation diagram. The green particle had to be a mesh (in red) vertex and its penalty is the distance to the closest mesh particle. The purple particles are mesh vertexes that weren't supposed to exist, and its penalty is the distance to the closest boundary particle.	45
17.	Surface reconstruction in a toy example using RBF implicits: (a) boundary particles (black) with oriented normals (red) and (b) the same dataset after the resampling. The signed distance field ϕ : (c) without resampling and (d) with resampling. The values of ϕ vary from negative (red) to positive (blue) values and its zero level-set (black).	46
18.	Water crown splash. Free-surface fitting without (left) and with (right) the resampling method.	46
19.	Liquid drop on a cube. Comparison between the method (right) and the methods of Ando, Thürey and Tsuruno (2012) (left) and Zhang, Solenthaler and Pajarola (2008) (middle). The boundary particles (black dots) fitted by the surface are illustrated at the top-row. Notice that the surface reconstruction using the resampling is more resistant to the deterioration of the small-scale features during a violent splash.	47
20.	The number of new particles per frame in the liquid drop simulation (Fig. 19) in each method: Zhang, Solenthaler and Pajarola (2008) (yellow), Ando, Thürey and Tsuruno (2012) (blue) and the method (red).	48
21.	The effect of the resampling method in the surface reconstruction schemes proposed by Sandim <i>et al.</i> (2016) and Marrone <i>et al.</i> (2010).	48
22.	Enright test at maximum stretching time. From left to right: free-surface fitting without the resampling in high-resolution, without the resampling in low-resolution, with the resampling in low-resolution and the new particles (red) inserted by the method.	49

CONTENTS

1	INTRODUCTION	17
1.1.	Introduction	17
1.1.1.	<i>Contributions</i>	19
1.1.2.	<i>Thesis outline</i>	19
2	RELATED WORKS	21
2.1.	Related Works	21
3	BASIC CONCEPTS	27
3.1.	Surface Reconstruction	27
3.2.	Principal Components Analysis	29
4	THE METHOD	31
4.1.	Particle resampling for free-surface fitting	31
4.1.1.	<i>Boundary detection</i>	31
4.1.1.1.	<i>Bpart</i>	32
4.1.1.2.	<i>Sregion</i>	33
4.1.2.	<i>Feature classification</i>	36
4.1.3.	<i>Particle refinement</i>	37
4.2.	Surface reconstruction	41
4.2.1.	<i>Screened Poisson</i>	42
4.2.2.	<i>Marrone's free-surface fitting</i>	43
4.3.	Results	44
4.3.1.	<i>Quantitative analysis</i>	44
4.3.2.	<i>Qualitative analysis</i>	46
5	CONCLUSION	51
5.1.	Conclusion	51
5.2.	Bibliographic production related to this thesis	52
	BIBLIOGRAPHY	53

INTRODUCTION

1.1. Introduction

An interface between a liquid and a gas (e.g., water and air) is known as a free-surface. The fragmentation of the free-surface caused by a liquid splashing is a ubiquitous phenomenon which generates not only an intricate surface but also important small-scale features of the liquid, such as thin sheets, liquid streams or ligaments, and small droplets. A plausible representation of these features is essential in fluid animation, demanding an intense research activity in the field of computer animation.

In the context of particle-based fluids, rendering of the free-surface usually involves the employment of a surface reconstruction technique which transforms particles into a polygonal mesh. Several surface reconstruction methods have been proposed in the Computer Graphics literature, not only for particle-based fluids but also for point clouds in general. The concern of point cloud surface reconstruction techniques (BERGER *et al.*, 2014) is to create a watertight surface from oriented point cloud (i.e., a subset of 3D points that sample the surface) typically acquired with 3D range scanners. On the other hand, in particle-based fluids, such as *Smoothed Particle Hydrodynamics* (SPH) (IHMSSEN *et al.*, 2014) and *Fluid-Implicit-Particle* (FLIP) (ZHU; BRIDSON, 2005), the main purpose is to extract a high-quality smooth free-surface from the particle positions. In this case, the free-surface is represented implicitly by a blobby-like model, i.e., the zero set of a signed distance field computed from a weighted sum of *kernel* evaluations over the distances between the particle positions. However, traditional surface reconstruction methods for particle-based fluids have difficulty in generating smooth surfaces with small-scale features due to an irregular distribution of the particles on the free-surface (see Figure 1). For this reason, many works in the literature address the low particle resolution on the free-surface by using adaptive sampling methods (SOLENTHALER; ZHANG; PAJAROLA, 2007; ZHANG; SOLENTHALER; PAJAROLA, 2008; ANDO; THÜREY; TSURUNO, 2012; JANG; NOH, 2015).

Figure 1 – Comparison between different surface reconstruction approaches on the Double dam break animation. (YU; TURK, 2010).

Source: Yu and Turk (2010)

Recently, Sandim *et al.* (2016) proposed a framework for free-surface reconstruction based on robust methods for point cloud surface reconstruction. That method relies on the conversion of the volumetric shape defined by the fluid particles into a point cloud. The point cloud is achieved through of an accurate method for detection of particles on the free-surface, which are known as *boundary particles* (or *surface particles*). In order to avoid the blooby artifacts (e.g., bumps or indentations) caused either by the free-surface fragmentation or by the sparsity of the particle distribution, the free-surface reconstruction is generated using a surface fitting algorithm, such as *Radial Basis Functions* (RBF) implicits (CARR *et al.*, 2001a), *Multi-level Partition of Unity* (MPU) implicits (OHTAKE *et al.*, 2003), and *Screened Poisson surface reconstruction* (KAZHDAN; HOPPE, 2013).

Despite reconstructing a watertight surface from the boundary particles, this method also suffers from the problem of particle sampling with the disadvantage of lacking some adaptive resampling method tailored for free-surface fitting. Small scale features are lost where there aren't enough particles to define a volume due to surface fitting algorithms limitations. The reason for that is the goal of these algorithms: they're designed to reconstruct volume data and thin sheets, for example, lacks volume. Here, it's presented a novel particle resampling method for surface reconstruction of a point cloud which represents the free-surface of a liquid from a particle-based simulation, providing a significant improvement in the free-surface fitting proposed by Sandim *et al.* (2016). The method is based on an adaptive particles refinement of the free-surface according to its small-scale features, and can preserve small droplets, streams, and thin sheets due to the insertion of new particles in poorly sampled regions. The detection of the small features and the positioning of the new sampled particles are guided by the deformation and the stretching of the free-surface computed from the *Principal Component Analysis* (PCA) (JOLLIFFE, 2002) of the particle positions. Figure 2 depicts the method in action.

Figure 2 – Free-surface fitting of a liquid splashing by using the method proposed by Sandim *et al.* (2016), without (left) and with our resampling method (right). Notice that the ligaments and thin sheets of fluid are gracefully preserved with our method.

1.1.1. Contributions

In contrast to previous methods, the proposed method is well suited for surface fitting of dynamic point clouds which sample the free-surface of a liquid. The method can capture the small-scale features representation along the simulation. Moreover, besides being computationally efficient, simple and easy to implement, the resampling is kernel-free, i.e., it does not require any kernel-based interpolation of geometrical or physical quantities.

The effectiveness of the method is also attested by a set of experiments and comparisons against the state-of-the-art methods in particle resampling. This thesis method gives a better result in comparison with those other methods and it's capable of reproducing high quality effects over low quality simulations.

1.1.2. Thesis outline

The remainder of the thesis is organized as follows. Chapter 2 presents a brief review of related methods existing in the literature. Some basic concepts for better understanding this thesis is given in Chapter 3. The proposed method is described in Section 4, and the surface reconstruction technique used is described in Section 4.2. Results are shown and discussed in Section 4.3. Chapter 5 concludes the thesis giving a glimpse of future work.

RELATED WORKS

2.1. Related Works

There are many works related to particle resampling, such as [Alexa *et al.* \(2003\)](#), resampling for representing solid shapes, and [Gois and Buscaglia \(2010\)](#), resampling on Moving-Least-Squares Surfaces. In order to better contextualize the method and highlight its properties, this chapter focuses on reviewing previous works closely related to particle resampling methods in fluid animation.

[Adams *et al.* \(2007\)](#) proposed an adaptive sampling method for SPH fluids based on geometric local feature size, computed using a particle approximation of the medial axis of the free-surface (see [Figure 3](#)). Their method allows increasing the number of the particles in geometrically complex regions near from the free-surface, while reducing the number of particles deep inside the fluid. Concerning surface reconstruction, they have improved the implicit surface definition proposed by [Zhu and Bridson \(2005\)](#) replacing the distance between the particles by a particle-to-surface distance.

Figure 3 – Flooding a valley. The particle density is dynamically adapted based on geometric complexity and visual importance (as color coded on the particles). The bottom row shows the actual camera views. The top row shows cross sections of the whole simulation domain ([ADAMS *et al.*, 2007](#)).

Source: [Adams *et al.* \(2007\)](#)

[Solenthaler, Zhang and Pajarola \(2007\)](#) introduced an upsampling method to enhance the SPH particle resolution of the free-surface of a liquid. Although refining the particles near the free-surface, this method can lead to surface artifacts due to the coarse approximation of the boundary particles based on a user threshold of the distance between the particle and the centroid of its neighborhood. [Zhang, Solenthaler and Pajarola \(2008\)](#) proposed operations of split and merge of SPH particles on GPU. Taking the previous boundary detection approach as a resampling criteria, their method splits a boundary particle into four sub-particles placed at the corners of a tetrahedron, while interior particles are merged to a single particle located at their centroid (Figure 4).

Figure 4 – [Zhang, Solenthaler and Pajarola \(2008\)](#) rendering results. Top row: Rendering with only a few surface particles. Bottom row: Rendering with more surface particles.

Source: [Zhang, Solenthaler and Pajarola \(2008\)](#)

[Hong, House and Keyser \(2008\)](#) introduced an adaptive particle sampling method for incompressible fluids coupling into the FLIP, where the refinement and simplification criteria are defined using the Reynolds number and the particle-to-surface distance. In order to preserve fluid thin sheets on FLIP simulations, [Ando, Thürey and Tsuruno \(2012\)](#) provided a remarkable particle resampling method (see Figure 5) based on the anisotropic kernels introduced by [Yu and Turk \(2010\)](#).

Figure 5 – Water drop on cubic tower. Top row: liquid solver only. Bottom row: the same simulation with our sheet preserving algorithm. The left hand side visualizes the particles, while the right hand side shows the transparently rendered mesh.

Source: [Ando, Thürey and Tsuruno \(2012\)](#)

[Solenthaler and Gross \(2011\)](#) introduced a multi-scale method for SPH fluids. The key idea of their method is to blend individual simulations with distinct resolution levels regarding the complexity of the flow behavior. Similar to the previous method, [Orthmann and Kolb \(2012\)](#) presented an adaptive incompressible SPH framework for convection-diffusion simulations with a temporal blending of the particle quantities in order to enable adaptive time-steps. The particle resampling is controlled by user thresholds for the magnitude of the volume gradient and the concentration gradient. [Ando, Thürey and Wojtan \(2013\)](#) integrate an adaptive tetrahedral BCC (Body-Centered Cubic) mesh with adaptive FLIP particles to allow varying the resolution during the simulation (see Figure 6).

Figure 6 – [Ando, Thürey and Wojtan \(2013\)](#) adaptive simulation framework allows to efficiently simulate highly detailed splashes on large open surfaces.

Source: [Ando, Thürey and Wojtan \(2013\)](#)

Recently, [Jang and Noh \(2015\)](#) proposed a particle refinement method for SPH fluids which relies on the analysis of the geometric structure between surface particles. New particles are inserted into the interparticle connections using *Poisson disk* sampling performed on a regular grid. However, this method in some cases does not preserve the temporal coherence of the surface reconstruction (see Figure 7).

Figure 7 – A sequence of images captured at (a, b) the 40th frame and (c, d) the 80th frame of a simulation. The surfaces reconstructed using [Jang and Noh \(2015\)](#) method (b, d) have fewer holes than that which underwent smooth surface reconstruction ([ZHU; BRIDSON, 2005](#)) (a, c) due to the contributions of the interparticle connections at the reconstruction stage.

Source: [Jang and Noh \(2015\)](#)

It is important to point out that the current state-of-the-art methods of particle resampling were developed for surface reconstruction methods (for instance, [Zhu and Bridson \(2005\)](#) and [Yu and Turk \(2010\)](#)) based on a smoothed signed distance function defined over all particles of the system, i.e., without distinguishing whether a particle belongs to the free-surface or not. Therefore, one drawback of these methods mentioned above is that they are not capable of preserving the temporal coherence of the free-surface fitting, i.e., the small-scale features may appear or disappear along with the animation. This thesis fills this gap in the literature.

BASIC CONCEPTS

3.1. Surface Reconstruction

Surface reconstruction from a point cloud is a process of finding a surface model that approximates an unknown surface for a given set of sample points lying on or near the unknown surface (JEONG *et al.*, 2018). This approximation can be in the form of an implicit object, defined by a set of solutions of $f(p) = 0$ where $f : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$. If f is well behaved, this set will be a surface of dimension $n - 1$ in $\mathbb{R}^n$. The method that solves this problem is called implicit method, e.g. *Radial Basis Function Implicit* (CARR *et al.*, 2001b) and *Screened Poisson Surface Reconstruction* (KAZHDAN; HOPPE, 2013) and the difference between implicit methods are the way they find f (HOPPE *et al.*, 1992). The function f is called level-set and the set of points $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]$ and we call *point-set* the set of all $f(\mathbf{p}_i) = 0$ where $1 \leq i \leq n$. The point-set contains only the surface points of the model therefore it represents a model without volumetric data, as shown in Figure 8.

Figure 8 – Illustration of a particle model and it's point-set. The left figure is a gingerbread with volumetric data and the right figure is it's point-set, without volumetric data.

Another way to reconstruct a surface is by using SPH approximation, a interpolation method used to simulate particle based fluids. There the fluid continuous flow is represented by particles as discrete elements. Each particle is defined by its position $\mathbf{x}$, support radius r and mass m . Despite r and m being considered equal to every particle, some adaptive methods uses different mass and radius to each particle (AKINCI, 2010).

In SPH context, fluid proprieties are smoothed by a kernel function inside an influence radius, known as smoothing length h , usually chosen as twice the initial distance between adjacent particles. The smoothing operation can be explained as calculating any physical quantity of the particle by using the weighted sum of the same quantity of the neighboring particles that lie in the range of the kernel function. The contributions of the neighboring particles are governed by kernel functions (commonly denoted as W) (AKINCI, 2010). The set of particles that lies in the range of W is called neighborhood, denoted by $\mathcal{N}_i$. The equation to calculate the scalar value A_i of an arbitrary particle in the position $\mathbf{x}_i$ is defined by:

$$A_i = \sum_{j \in \mathcal{N}_i} m_j \frac{A_j}{\rho_j} W(\mathbf{x}_i - \mathbf{x}_j, h).$$

In this case, j is an iterator on all neighborhood particles, ρ_j is the particle density, W is the kernel function (e.g. *Gaussian*, *Cubic Spline*) and A_k is called field quantity of a particle. By the above denition, only the particles whose distance to the particle of interest is smaller than the smoothing length h are considered to be on the neighborhood (AKINCI, 2010). The surface of the model can be found using an additional field quantity with value 1 in positions where there are particles and 0 where there aren't. This field quantity is called *color field* (MÜLLER; CHARYPAR; GROSS, 2003). The smoothened color field is calculcd by:

$$c_i = \sum_{j \in \mathcal{N}_i} m_j \frac{1}{\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h).$$

The normal vector $\mathbf{n}_i$ of a particle in the position $\mathbf{x}_i$, on the surface of the model, is given by the gradient $\mathbf{u}_i$ of the color field. As said by Müller, Charypar and Gross (2003), $\mathbf{n}_i$ is given by:

$$\mathbf{n}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|} \quad \text{with} \quad \mathbf{u}_i = \sum_{j \in \mathcal{N}_i} m_j \frac{1}{\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h),$$

and the gradient will be pointing to the inside of the model, since the particles has more interaction inside of the fluid.

3.2. Principal Components Analysis

The *Principal Components Analysis* (PCA) (SHLENS, 2014) is a technique used to reduce a data table dimension without much information loss, so that relevant data can be extracted by the user. Its geometric interpretation is usefull in identifying anisotropy of a region. The main goal is to sort the directions with more variance, decreasingly, and each direction is orthogonal with each other. This is done by calculating the covariance matrix of a dataset and maximizing the value of each element in it's diagonal, maximizing the variance of each axis, and then sorting the eigenvectors of the covariance matrix by its eigenvalue. Each eigenvector will be a principal component (direction with more variance) of the dataset and the higher its eigenvalue is, the higher the variance in that direction is.

The variance of a dataset $\mathbf{X}^\top = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ is defined as $\sigma_{\mathbf{X}}^2 = \frac{1}{n} \sum \mathbf{x}_i^2$, and its covariance matrix $\mathbf{C}_{\mathbf{X}}$ is given as:

$$\mathbf{C}_{\mathbf{X}} = \frac{1}{n} \mathbf{X}^\top \mathbf{X}.$$

where $\mathbf{x}_i$ is a vector of dimension p and corresponds to an instance on the data table.

Observe that $\mathbf{C}_{\mathbf{X}}$ is a matrix of dimension $p \times p$ and the scalar value in the position ij of the matrix is the covariance between a measure i and a measure j . When $i = j$, this scalar value is the variance of i . So this problem turns out to be a maximization problem, where the diagonal positions needs to be maximized thus maximizing the variance of each measure. The reason for that is that by maximizing the variance on each axis, the size of each principal component is maximized and then it's possible to sort them accordingly.

Since $\mathbf{X}$ have dimension $n \times p$, where n is the number of elements and p the dimension of the data, then its projection is given by $\mathbf{X}\mathbf{w}$, where $\mathbf{w}$ is a unitary vector $p \times 1$. Therefore the variance becomes:

$$\begin{aligned} \sigma_{\mathbf{w}}^2 &= \frac{1}{n} \sum (\mathbf{x}_i \cdot \mathbf{w})^2 \\ &= \frac{1}{n} (\mathbf{X}\mathbf{w})^\top (\mathbf{X}\mathbf{w}) \\ &= \frac{1}{n} \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} \\ &= \mathbf{w}^\top \frac{\mathbf{X}^\top \mathbf{X}}{n} \mathbf{w} \\ &= \mathbf{w}^\top \mathbf{C}_{\mathbf{X}} \mathbf{w}. \end{aligned}$$

As stated before, the variance $\sigma_{\mathbf{w}}^2$ needs to be maximized in function of a unitary vector $\mathbf{w}$. To guarantee that $\mathbf{w}$ is unitary a restriction $\mathbf{w} \cdot \mathbf{w} = 1$ is added. Then, this problem becomes an optimization problem where $\mathbf{w}$ needs to be maximized subject to $\mathbf{w} \cdot \mathbf{w} = 1$. For that, Lagrange Multipliers can be used, represented by λ , such that:

$$\mathcal{L}(\mathbf{w}, \lambda) \equiv \sigma_{\mathbf{w}}^2 - \lambda(\mathbf{w}^\top \mathbf{w} - 1)$$

$$\frac{\partial L}{\partial \lambda} = \mathbf{w}^\top \mathbf{w} - 1$$

$$\frac{\partial L}{\partial \mathbf{w}} = 2\mathbf{C}_X \mathbf{w} - 2\lambda \mathbf{w},$$

solving for the derivatives equal to 0:

$$\mathbf{C}_X \mathbf{w} = \lambda \mathbf{w}. \quad (3.1)$$

By the equation 3.1, it can be concluded that $\mathbf{w}$ needs to be a eigenvector of $\mathbf{C}_X$ and the $\mathbf{w}$ that maximizes its value is the eigenvalue λ . Thus, the eigenvectors of $\mathbf{C}_X$ are the principal components and the most relevant one is associated to the higher eigenvalue. This is very convenient, once that calculating eigenvalues and eigenvectors are a very known problem with a relatively fast solution and with good mathematical proprieties. A particular propriety is that because $\mathbf{C}_X \mathbf{w}$ is symmetric semi-definite positive, the eigenvalues will be greater or equal to zero, once that $\mathbf{C}_X = \frac{1}{n} \mathbf{X}^\top \mathbf{X}$ so $\frac{1}{n} \mathbf{u}^\top \mathbf{X}^\top \mathbf{X} \mathbf{u} = \frac{1}{n} Z_u^2 \geq 0$ to any vector $\mathbf{u}$ where Z_u is a real value. Besides this, the principal components are orthogonal, once they are eigenvectors of $\mathbf{C}_X$ and its projections won't have correlation. Besides this, eigenvalues gives the variance each component describes.

THE METHOD

4.1. Particle resampling for free-surface fitting

In this section, the proposed particle resampling method is described. As illustrated in Figure 9, the method comprises three main stages: boundary detection, feature classification, and particle refinement. For each time-step of a particle-based fluid simulation, since the thesis is focused in rendering the free-surface, firstly the free-surface is captured through a boundary particle detection. Then, the boundary particles are classified and labeled depending on which region they are located regarding the small-scale feature. Finally, particles placed at feature regions are refined according to their feature classification.

Figure 9 – Overview of the pipeline of the method in a double dam breaking simulation.

4.1.1. Boundary detection

The first step of the algorithm is the boundary detection stage, and it can be any chosen by the user. This thesis focuses on using a method proposed by [Sandim *et al.* \(2016\)](#), where according to the experiments reported by them their method outperforms in performance and accuracy other boundary detection methods, and later comparisons with the method proposed by [Marrone *et al.* \(2010\)](#), considered the state of art, are made. For simplicity we will use the term

Sregion to refer to the method by Marrone *et al.* (2010), and *Bpart* for the method by Sandim *et al.* (2016). In this section, both methods will be detailed for best understanding of the thesis results.

4.1.1.1. *Bpart*

The *Bpart* method demands only particle positions (without normals) to perform the boundary/non-boundary particle classification. In addition to being purely geometric, it's adopted *Bpart* since it is easy to implement and computationally efficient. Indeed, according to the experiments reported by Sandim *et al.* (2016), the method outperforms in performance and accuracy other boundary detection techniques both in Computer Graphics (MÜLLER; CHARYPAR; GROSS, 2003; HE *et al.*, 2012; ORTHMANN *et al.*, 2013) and outperforms *Sregion* Computational Physics (e.g. Marrone *et al.* (2010), considered the state-of-art in this field).

In order to precisely define boundary particles, the method employs the concept of δ -sampling (KATZ; TAL; BASRI, 2007). A point set $\mathcal{P}$ is a δ -sampling of a domain $\Omega \subset \mathbb{R}^3$ if for any point $\mathbf{p} \in \Omega$ there is a particle i located at $\mathbf{x}_i \in \mathcal{P}$ such that $\|\mathbf{p} - \mathbf{x}_i\|_2 < \delta$. Assuming a δ -sampling $\mathcal{P}$, consider the set

$$\mathcal{M}_\delta = \left\{ \mathbf{q} \in \mathbb{R}^3 \mid \inf_{\mathbf{x}_i \in \mathcal{P}} \{\|\mathbf{q} - \mathbf{x}_i\|_2\} < \delta \right\}.$$

Let $\mathcal{B}_\delta(\mathbf{x}_i)$ be an open ball of radius δ centered at $\mathbf{x}_i$. The particle i is called δ -interior if $\partial\mathcal{B}_\delta(\mathbf{x}_i) \subset \mathcal{M}_\delta$, where $\partial\mathcal{B}_\delta(\mathbf{x}_i)$ is the boundary of $\mathcal{B}_\delta(\mathbf{x}_i)$; otherwise, i is called δ -boundary.

Let $\mathcal{S}_\delta \subset \mathcal{P}$ denote the point set corresponding to all δ -boundary particles. *Bpart* approximates $\mathcal{S}_\delta$ by using the *hidden point removal* (HPR) operator (KATZ; TAL; BASRI, 2007), which aims at identifying visible points on point-set surfaces. A particle i is classified as a boundary particle if there is a viewpoint $\mathbf{v} \notin \mathcal{P}$ such that the line segment connecting $\mathbf{x}_i$ and $\mathbf{v}$ intersects $\mathcal{P}$ only at $\mathbf{x}_i$. Indeed, Sandim *et al.* (2016) provide theoretical guarantees on the existence of viewpoints from which one can obtain approximations to $\mathcal{S}_\delta$.

The HPR operator comprises two main steps: inversion and convex hull computation. Given a viewpoint $\mathbf{v}$, in the inversion step each point $\mathbf{x}_i \in \mathcal{P}$ is mapped to an inverted domain in relation to $\mathbf{v}$, i.e., points closer to $\mathbf{v}$ are mapped far away while points distant from $\mathbf{v}$ are placed closer to the viewpoint. Let $\hat{\mathcal{P}}$ denote the set of inverted points. A point $\mathbf{x}_i$ is labeled as visible if its inverted image $\hat{\mathbf{x}}_i$ 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 and returns the vertices of the resulting polyhedron as the visible points. Following Sandim *et al.* (2016), the work uses the *exponential flipping* inversion mapping in order to better handle volumetrically distributed particles (see (KATZ; TAL; BASRI, 2007) for details).

Given a δ -sampling set of points $\mathcal{P}$, the boundary detection stage encompasses two steps: viewpoint placement and the visibility test. In order to place viewpoints in locations from which the HPR algorithm can properly identify boundary particles, the bounding box of $\mathcal{P}$ is covered by a regular grid $\mathcal{G}$ whose edge length of each cubic cell is set to 2δ . A grid cell C is labeled *empty* if it contains no particles in its interior, otherwise, C is labeled *full*. As viewpoints must be placed outside $\mathcal{P}$ (i.e., the fluid domain), empty cells adjacent to at least one full cell are natural candidates to host viewpoints. Thus, a viewpoint is placed in the centroid of each empty cell adjacent to a full cell. However, cavities with a diameter at least 2δ may be present inside or between full cells. In order to identify the points on the boundary of those cavities, a full cell C is marked as a *cavity cell* if it satisfies the following two conditions:

1. all cells adjacent to C are full, and
2. there is a ball with center in C and radius δ that contains no point of $\mathcal{P}$ in its interior.

Then, for each particle i inside C , a candidate viewpoint $\mathbf{v}_i$ is located as follows:

$$\mathbf{v}_i = \begin{cases} \mathbf{x}_i + \delta \frac{\bar{\mathbf{x}}_i}{\|\bar{\mathbf{x}}_i\|_2} & \text{if } \|\bar{\mathbf{x}}_i\|_2 \neq 0 \\ \mathbf{x}_i, & \text{otherwise} \end{cases},$$

with

$$\bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{x}_j, \quad (4.1)$$

where $\mathcal{N}_i$ is the local neighborhood of radius 2δ of the particle i . Next, the emptiness of each ball $\mathcal{B}_\delta(\mathbf{v}_i)$ is verified. If $\mathcal{B}_\delta(\mathbf{v}_i)$ is empty, then its center is chosen as a viewpoint. In practice, the diameter of $\mathcal{B}_\delta(\mathbf{v}_i)$ is decreased by a scale factor of 0.95 due to the non-uniform particle distribution.

Once the viewpoints have been defined, the last step is the visibility test. The HPR operator is locally applied such that, for each viewpoint $\mathbf{v}_i$, the set of points $\mathcal{P}_i \subset \mathcal{P}$ within the ball centered in $\mathbf{v}_i$ with radius 4δ is picked out for inversion. The set of the boundary particle positions $\mathcal{B}_\delta$ is the union of all points marked as visible after applying the HPR operator from each $\mathbf{v}_i$.

4.1.1.2. *Sregion*

As *Bpart*, this method requires only particle positions to perform the boundary/non-boundary particle classification. *Sregion* is composed of two steps: in the first one a renormalization matrix is used to find particles next to the free-surface. This first step strongly decreases the number of particles that will be processed in the second step. In the second step the algorithm, by means of geometric properties, detects particles that actually belong to the free-surface and

evaluates their local normals. This renormalization matrix is defined as:

$$\mathbf{B}(\mathbf{x}_i) = \left[\sum_j \nabla W_j(\mathbf{x}_i) \otimes (\mathbf{x}_j - \mathbf{x}_i) \Delta V_j \right]^{-1}, \quad (4.2)$$

where ΔV_j is the volume of the j -th particle and $W_j(\mathbf{x}_i)$ is the interpolating kernel centered on particle j and evaluated in the position $\mathbf{x}_i$. The spatial derivatives of $W_j(\mathbf{x}_i)$ are referred with respect to the position $\mathbf{x}_i$.

The value of the minimum eigenvalue, λ , of the matrix $\mathbf{B}^{-1}$ depends on the spatial organization of the particles j in the neighborhood of the considered calculation point i . When going away from the fluid domain this eigenvalue λ tends theoretically to 0 while inside this domain λ tends theoretically to 1. This allows to determine regions of the fluid domain where the free-surface can lie or not.

Define $\mathbb{N}$ as the set of all the fluid particles and $\mathcal{P}$ as the set of the free surface particles on $\mathbb{N}$ (i.e. the point-set). By the value of λ in some particle, it is possible to label it as interior particles (those belongs to the subset $\mathbb{I}$ of $\mathbb{N}$), close-to-the-free-surface particles (belongs to the subset $\mathbb{B}$), particles in regions of the domain where particles are not uniformly spread (belongs to the subset $\mathbb{B}$ as well) and particles in thin sheets and drops (belongs to the subset $\mathbb{E}$. Those are classified by a high value of λ , intermediate value of λ and low value of λ respectively. So, $\mathbb{N} = \mathbb{E} \cup \mathbb{I} \cup \mathbb{B}$. The point-set $\mathcal{P}$ is thus composed of all the particles of subset $\mathbb{E}$ and a part of the elements of subset $\mathbb{B}$.

The subset identification is made by the definition of threshold values for λ which depend on the considered kernel. *Sregion* uses a renormalized gaussian kernel shape defined as:

$$W(s, h, \delta) = \frac{e^{(-s/h)^2} - e^{(-\delta/h)^2}}{2\pi \int_0^\delta s(e^{(-s/h)^2} - e^{(-\delta/h)^2}) ds}, \quad (4.3)$$

where s is the distance between two particles, h is the smoothing length and δ is a support radius chosen as $\delta = 3h$. h is equal to $1.33dx$, where dx is the average particle spacing, which means that in two dimensions a particle has $\simeq dx^2$ for volume and an average number of neighbors equal to 50 particles. Being i the particle under examination, the values found by testing thresholds for λ are the following:

$$\begin{cases} i \in \mathbb{E}, & \lambda \leq 0.20 \\ i \in \mathbb{B}, & 0.20 < \lambda \leq 0.75 \\ i \in \mathbb{I}, & 0.75 < \lambda. \end{cases} \quad (4.4)$$

Particles closer to the free-surface and near cavities are correctly detected by this step but this also happens for some particles within internal fluid regions characterized by non-uniform distribution. Conversely, particles which belong to drops and thin jets are easily captured by the lower threshold and directly identified as free-surface particles.

In the second step of *Sregion*, the free-surface particles in $\mathbb{B}$ are filtered using the fact that inside the fluid domain, the sum of the kernel gradient over neighbors is very close to zero. When a particle, instead, is near the free-surface, such sum is a good approximation of the local normal $\mathbf{n}$ to the free-surface. Since the accuracy of the evaluation of this vector depends on the particle disorder, it is possible to get a more accurate evaluation by using again the renormalization matrix:

$$\mathbf{n}(\mathbf{x}_i) = \frac{\mathbf{v}(\mathbf{x}_i)}{v(\mathbf{x}_i)}; \quad \mathbf{v}(\mathbf{x}_i) = -\mathbf{B}(\mathbf{x}_i) \sum_j \nabla W_j(\mathbf{x}_i) \Delta V_j. \quad (4.5)$$

This is a standard way to improve accuracy of the SPH interpolation within the fluid domain. The same principle is used to improve the accuracy of the evaluation of the local normal. Once this vector $\mathbf{n}$ is known, it is possible to define a region where a scan is made to check whether or not at least one neighbor particle lies in it, hereinafter referred to as scan region (Figure 10). If no neighbor is found inside it, the candidate particle belongs to the free surface. This scan region classifies a particle $i \notin \mathcal{P}$ if:

$$\begin{cases} |\mathbf{x}_{ij}| \geq \sqrt{2}h, & |\mathbf{x}_{iT}| < h \\ |\mathbf{x}_{ij}| < \sqrt{2}h, & \arccos\left(\frac{\mathbf{n} \cdot \mathbf{x}_{ji}}{|\mathbf{x}_{ji}|}\right) < \frac{\pi}{4} \end{cases}, \quad (4.6)$$

where $\mathbf{x}_{ji}$ is defined as $\mathbf{x}_j - \mathbf{x}_i$, j is a neighbor particle of i in a $3h$ radius distance and T is the point at distance h from i in the normal direction. Thus, all particles that doesn't satisfy any of these equations are free-surface particles (MARRONE *et al.*, 2010).

Figure 10 – Sketch of the regions used in the *Sregion* (MARRONE *et al.*, 2010).

Source: Marrone *et al.* (2010)

Despite of the effectiveness of the boundary detection method adopted in this stage, it is worth noting that, without resampling, the resulting boundary particles may not be sufficient enough for reconstruction of a smooth free-surface in regions with small-scale features.

4.1.2. Feature classification

Small-scale features appear in free-surface regions that suffer large stretching. One way to measure the displacement anisotropy of the particles' positions caused by stretching is using PCA. There was the possibility of using WPCA or PCA, as they both works in similar ways. Since there was no expressive difference in the results, PCA was used in this thesis for its faster execution time. In order to reduce more the computational effort, PCA is only computed around the neighborhood of the boundary particles.

Given a boundary particle i located at $\mathbf{x}_i \in \mathbb{R}^3$, PCA is performed in the neighborhood $\mathcal{N}_i$ which may contain boundary and interior particles. PCA computes the spectral decomposition of the covariance matrix $\mathbf{C}_i$:

$$\mathbf{C}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_i)^\top \quad (4.7)$$

with $\bar{\mathbf{x}}_i$ given by Equation (4.1). The eigendecomposition of $\mathbf{C}_i$ is obtained using the *Singular Value Decomposition* (SVD) (JIA, 2013). This algorithm decomposes $\mathbf{C}_i$ as follows:

$$\mathbf{C}_i = \mathbf{U}_i \Sigma_i \mathbf{U}_i^\top,$$

where $\mathbf{U}_i$ denotes the orthonormal eigenvector matrix and $\Sigma_i = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$ is the eigenvalue matrix with the (positive) eigenvalues sorted in ascending order, i.e., $\sigma_1 \leq \sigma_2 \leq \sigma_3$. This is only possible because $\mathbf{C}_i$ is symmetric.

The eigenvalue σ_k of $\mathbf{C}_i$ measures the variation of the particle positions along the direction of its corresponding eigenvector $\mathbf{u}_k$ (k -th column vector of $\mathbf{U}_i$), which means the particles are completely isotropically distributed when all eigenvalues are equal. In this implementation, the *Eigen* library (GUENNEBAUD; JACOB *et al.*, 2010) is used to compute the SVD decomposition.

The features are classified according to the deformation of the free-surface and the cardinality of the particle neighborhood. Given a boundary particle i , before computing the covariance matrix $\mathbf{C}_i$, the cardinality of $\mathcal{N}_i$ is analyzed. The particle i receives the label $\ell_i = 0$ if $|\mathcal{N}_i|$ is less than a certain number K of particles, including the particle i itself. In this case, the SVD decomposition is bypassed and it's assumed that $\mathbf{U}_i$ is the identity matrix $\mathbf{I}$ of order 3. Otherwise, the deformation of the free-surface at $\mathbf{x}_i$ is estimated using the spectrum of $\mathbf{C}_i$. Inspired by Ando, Thürey and Tsuruno (2012), the particle i is labeled as follows:

$$\ell_i = \begin{cases} 1 & \text{if } \sigma_2 \leq \alpha \sigma_3 \\ 2 & \text{if } \sigma_1 \leq \alpha \sigma_3 \text{ and } \sigma_2 > \alpha \sigma_3 \\ 3 & \text{if } \sigma_1 > \alpha \sigma_3 \end{cases} \quad (4.8)$$

The value α denotes a threshold which defines the degree of deformation (or thickness). In the experiments, it's adopted $K = 4$ and $\alpha = 0.2$. Summarizing the classification scheme, the label $\ell_i = 0$ refers to droplets, while $\ell_i = 1$ refers to liquid streams or ligaments, and $\ell_i = 2$ refers to thin sheets, as illustrated in Figure 11. It's assumed that label $\ell_i = 3$ does not represent a feature. A particle labeled as a feature is called *feature particle*.

Figure 11 – The color encodes the boundary particles (blue) and the features particles in a water crown splash: small droplets (cyan), ligaments (yellow) and thin sheets (magenta).

4.1.3. Particle refinement

Usually, surface fitting algorithms interpolate a point cloud using the valuable information of the surface normals. A consistent orientation for the normal vector field provides a comprehension of the exterior and interior of the surface. It can be used to define a signed

distance field ϕ over the domain, where ϕ takes negative values in the interior and positive values in the exterior. The surface S is then represented implicitly by the zero set of the signed distance field, i.e., $S = \phi^{-1}(0)$. An orientable surface provides a useful hint not only for a surface reconstruction method but also for the particle refinement scheme.

After the feature classification stage, a feature particle i (i.e., $\ell_i \neq 3$) is refined in order to benefit the surface fitting algorithms, where the new particles are inserted such that they surround the particle i . In the proposed method, the particle i is replaced by new particles placed at $\tilde{\mathbf{x}}$ regarding the label ℓ_i (Figure 12) as follows:

$$\tilde{\mathbf{x}} = \mathbf{x}_i \pm 0.5 \delta \mathbf{u}_k, \quad k = 1, \dots, 3 - \ell_i. \quad (4.9)$$

Note that the thickness of the features reconstructed by the method is at least δ (matching with

Figure 12 – Particle refinement: a feature particle i (blue) is replaced by new particles (orange) according to its label ℓ_i .

the numerical resolution used in the boundary particle detection stage) due to the displacement of size 0.5δ in both directions of the eigenvector $\mathbf{u}_k$ in Equation (4.9).

Since the normal vector is orthogonal to the tangent space at the position of a boundary particle and the PCA defines an estimation of a tangent plane in a least-squares sense, the PCA can be reused to compute the oriented normal (pointing outward of the fluid) of each particle sampled by the method. The normal $\tilde{\mathbf{n}}$ of a new particle generated from a particle i is given by:

$$\tilde{\mathbf{n}} = \frac{\tilde{\mathbf{x}} - \mathbf{x}_i}{\|\tilde{\mathbf{x}} - \mathbf{x}_i\|_2}, \quad (4.10)$$

the normals of the remaining boundary particles with $\ell_i = 3$ are computed as follows:

$$\mathbf{n}_i = \text{sign}((\mathbf{x}_i - \bar{\mathbf{x}}_i) \cdot \mathbf{u}_1) \mathbf{u}_1. \quad (4.11)$$

However, the insertion of the new sampled particles can introduce discontinuities in the free-surface (Figure 13); this can be remedied by changing the position of these boundary particles with $\ell_i = 3$ as follows:

$$\mathbf{x}_i = \mathbf{x}_i + 0.5 \delta \mathbf{n}_i. \quad (4.12)$$

Figure 13 – Highlighted regions of the free-surface (magenta) are corrected by shifting the boundary particles of label $\ell_i = 3$.

(a) without correction

(b) with correction

In some cases, a boundary particle can be misclassified as a feature particle, like in Figure 11, where some particles inside the water crown are classified as $\ell_i = 2$. Following the refinement process, this misclassified particle can give rise to new particles located inside the fluid. In order to avoid this drawback, for each new sampled particle, the emptiness of its *safe region* is checked, defined as being the open ball $\mathcal{B}_r(\tilde{\mathbf{x}})$ with $r = 0.45\delta$. In fact, a new sampled particle is only inserted into the system if there are no particles inside of its corresponding safe region, as illustrated in Figure 14. This procedure ensures that an added particle is a boundary particle, thus avoiding to run the boundary detection again.

Figure 14 – A sampled particle (green) from a feature particle (blue) is rejected by virtue of its safe region (yellow) is not empty.

The resampling method can be summarized in Algorithm 1.

Algorithm 1 – Particle resampling for free-surface

Data: $(\mathcal{P}, \delta)$
Result: oriented point cloud $\mathcal{C}$
 $\mathcal{S}_\delta \leftarrow \text{boundary}(\mathcal{P}, \delta)$ (Section 4.1.1)

 $\mathcal{C}.\text{positions} \leftarrow \mathcal{S}_\delta$
 $\mathcal{C}.\text{normals} \leftarrow \emptyset$
foreach oriented point $i \in \mathcal{C}$ **do**

/* classification */

if $|\mathcal{N}_i| < K$ **then**
 $\ell_i \leftarrow 0$
 $\mathbf{U}_i \leftarrow \mathbf{I}$
else

 Compute the matrix $\mathbf{C}_i$ using Equation (4.7)

 $[\Sigma_i, \mathbf{U}_i] \leftarrow \text{SVD}(\mathbf{C}_i)$
 $\ell_i \leftarrow \text{classify}(\Sigma_i)$ (Equation (4.8))

end
end
foreach oriented point $i \in \mathcal{C}$ **do**

/* refinement */

if $\ell_i \neq 3$ **then**
 $\mathcal{X} \leftarrow \text{refine}(\mathbf{x}_i, \ell_i, \delta, \mathbf{U}_i)$ (Equation (4.9))

 Remove $\mathbf{x}_i$ from $\mathcal{C}.\text{positions}$
foreach $\tilde{\mathbf{x}} \in \mathcal{X}$ **do**
if the safe region of $\tilde{\mathbf{x}}$ is empty **then**

 Insert $\tilde{\mathbf{x}}$ to $\mathcal{C}.\text{positions}$

 Compute normal $\tilde{\mathbf{n}}$ using Equation (4.10)

 Insert $\tilde{\mathbf{n}}$ to $\mathcal{C}.\text{normals}$
end
end
else

 Compute normal $\mathbf{n}_i$ using Equation (4.11)

 Insert $\mathbf{n}_i$ to $\mathcal{C}.\text{normals}$ Update $\mathbf{x}_i$ in $\mathcal{C}.\text{positions}$ using Equation (4.12)

end
end
return $\mathcal{C}$

4.2. Surface reconstruction

After performing the particle resampling method, an oriented point cloud is obtained where the normal vectors of the sampled particles are computed by PCA. This thesis used two different free-surface fitting, one proposed by Sandim *et al.* (2016) using the Screened Poisson method (KAZHDAN; HOPPE, 2013) and the other proposed by Marrone *et al.* (2010). This section details each method for best understanding of the thesis results.

4.2.1. Screened Poisson

Given a surface normal field $\vec{V} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, the key idea of Screened Poisson method is to find a smoothed approximation of an indicator function $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$, defined as having the values $-1/2$ inside and $1/2$ outside the surface, by solving the following Poisson equation:

$$\Delta\phi = \nabla \cdot \vec{V}. \quad (4.13)$$

In the application, the *mixed boundary conditions* is used in the Screened Poisson method, i.e., Neumann boundary conditions on droplets and Dirichlet boundary conditions are imposed on the remaining feature particle to enforce the reconstruction of the small-scale features and closed surfaces.

The solution of Equation (4.13) is achieved solving a Laplacian linear system derived from the Galerkin approximation of the Equation (4.13), over a multiresolution B-spline basis, built under an adaptive octree. Then, the isosurface $S = \phi^{-1}(0)$ is extracted using an octree-based Marching Cubes algorithm, where the size of octree cells at the finer resolution matches with the particle spacing (AKINCI *et al.*, 2012).

In order to alleviate the noise and oscillations of the free-surface mesh while respecting the preservation of its small-scale features, further improvements in the surface can be done using a mesh smoothing method (e.g., Taubin (1995) and Belyaev and Ohtake (2003)). Figure 15 shows the effect of this geometry processing applied on the free-surface.

Figure 15 – Free-surface is enhanced by a mesh smoothing method.

4.2.2. Marrone's free-surface fitting

The key idea of this algorithm is to use a scalar quantity based on the direction of the vector from a grid node to the nearest free-surface particle to define a level-set function. Consider a regular Cartesian grid of spatial resolution equal to the average particle spacing dx as defined in Section 4.1.1.2, and which encloses all the computational domain. For each grid node i close to free-surface particles, the nearest free-surface particle j is detected and the scalar quantity d_{ij} is evaluated through:

$$d_{ij} = (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_j \quad (4.14)$$

where $\mathbf{n}_j$ is the normal to the free-surface evaluated in j . For each particle it is now possible to define a level-set function $\phi(x_i)$:

$$\phi(\mathbf{x}_i) = \begin{cases} -1, & d_{ij} \leq -2h \\ d_{ij}/2h, & -2h < d_{ij} < 2h \\ +1, & d_{ij} \geq 2h \end{cases} \quad (4.15)$$

Marrone *et al.* (2010) assumes that the actual free-surface location is at a distance $dx/2$ from the particles belonging to the point-set $\mathbb{F}$, thus the value of ϕ on the free-surface is $\phi = -dx/4h$ and not zero anymore. This propriety simulates a particle radius and creates more volume in thin sheet cases but for this thesis it won't be needed as it solves particle deficiency problems and $\phi = 0$ was set as the free-surface location.

At the computational level, the procedure to evaluate ϕ can be executed in a fast manner. At each particle, all the other particles present within a $2h$ distance is identified first. Those are denoted as $\mathbb{N}_{2h}$ and $\phi(x_i)$ can be evaluated through then as:

$$\phi(x_i) = \begin{cases} -1, & \mathbb{N}_{2h} = \emptyset \\ +1, & \mathbb{N}_{2h} \neq \emptyset, \mathbb{N}_{2h} \cap \mathbb{F} = \emptyset \\ d_{ij}/2h, & \mathbb{N}_{2h} \neq \emptyset, \mathbb{N}_{2h} \cap \mathbb{F} = \emptyset, \lambda_j \leq 0.1 \\ -\|\mathbf{x}_i - \mathbf{x}_j\|_2/2h, & \mathbb{N}_{2h} \neq \emptyset, \mathbb{N}_{2h} \cap \mathbb{F} = \emptyset, \lambda_j < 0.1 \end{cases} \quad (4.16)$$

where λ_j is the minimum eigenvalue of the renormalization matrix defined in Equation 4.2 for the particle j . When $\lambda_j < 0.1$ the nearest free-surface particle for the node i is a solitary particle, then the vector $\mathbf{n}_j$ is null (see Equation 4.5) and consequently d_{ij} is zero. In this case, $\phi(\mathbf{x}_i)$ is evaluated in function of the distance $\|\mathbf{x}_i - \mathbf{x}_j\|_2$. Further, to smooth out the function ϕ on the whole mesh, a Gaussian filter on the nodes is performed once ϕ is evaluated by Equation 4.16.

4.3. Results

The method was implemented in C++ using OpenMP multithreading. The particle-based fluid simulations were generated using a weakly compressible SPH implementation provided by DualSPHysics (CRESPO *et al.*, 2015). For each simulation, the value of the parameter δ is defined as the SPH smoothing length, as suggested by Sandim *et al.* (2016). All experiments have been performed on a 4-core 2.8 GHz Intel i7-4980HQ with 16 GB of RAM. In order to validate the effectiveness of the proposed approach, two different types of analyses were carried out.

4.3.1. Quantitative analysis

The main goal of this analysis is to quantify two different aspects: an eventual increase in number of boundary particles after resampling; and the reliability of the resampling. Both aspects were also compared against state-of-art methods.

The first aspect is summarized in Table 1, that shows the statistics of the number of particles ($|\mathcal{P}|$), the number of boundary particles ($|\mathcal{S}_\delta|$), the number of the boundary particles after the resampling ($|\mathcal{C}|$), the rate (in percentage) of new sampled points inserted by the method in the free-surface ($(|\mathcal{C}| - |\mathcal{S}_\delta|) / |\mathcal{S}_\delta|$), and computational times for all experiments.

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

Experiment	Figure	$ \mathcal{P} $	$ \mathcal{S}_\delta $	$ \mathcal{C} $	Rate(%)	Time
DDB	9	275K	46.4K	48.6K	4.84	1.3
Water crown	18	298K	52.8K	53.1K	0.60	1.5
BoxDrop	19	106K	51.2K	64.3K	25.60	1.3
Waterfall	21	129K	36.0K	43.7K	21.32	1.2

Once the resampling is performed, its reliability is attested by checking whether the new particles allow for a proper surface reconstruction. It's expected that S properly reconstructs the boundary particles if it locally approximates them, i.e., the surface S lie as close as possible to the boundary particles.

In order to quantify the reliability of a method, the sum of all distances between each boundary particles and the mesh and the sum of all distances between each mesh vertexes and the nearest boundary particle is calculated. Combining both values (by summing them), a unique value is obtained for comparison. The lower this value is, best the method is, making this result act as a penalty.

For the distance of the boundary particle to the mesh case, given that $\mathcal{P}$ is the set of boundary particles, for each $\mathbf{x}_i \in \mathcal{P}$ its distance to its projection on the closest triangle in S is computed, i.e., $d(\mathbf{x}_i, S) = \min_{f \in \mathcal{F}(S)} d(\mathbf{x}_i, \pi(\mathbf{x}_i, f))$, where $\mathcal{F}(S)$ is the set of triangles f of S and π a projection of $\mathbf{x}_i$ into f . For the distance of the mesh vertexes to the boundary particles

case, for each vertex $\mathbf{v}_i \in S$ its distance to the closest boundary particle $\mathbf{x}_i$ is computed, i.e., $d(\mathbf{v}_i, \mathcal{P}) = \min_{\mathbf{x}_i \in \mathcal{P}} d(\mathbf{v}_i, \mathbf{x}_i)$ (Figure 16).

Figure 16 – Penalty calculation diagram. The green particle had to be a mesh (in red) vertex and its penalty is the distance to the closest mesh particle. The purple particles are mesh vertexes that weren't supposed to exist, and its penalty is the distance to the closest boundary particle.

Then, the mean of all $d(\mathbf{x}_i, S)$ and $d(\mathbf{v}_i, \mathcal{P})$ are done by dividing their values by their number of samples. A sample for the boundary particle to mesh case is a boundary particle, and to the other case a sample is a mesh vertex. Then, all these values are summed, resulting in the mean distance error of a frame in a simulation, and then the *Area Under the Curve* (AUC) in the distance mean by frame graph is finally calculated and the reconstruction error is defined as $Error = AUC(d(\mathbf{x}, S)) + AUC(d(\mathbf{v}, \mathcal{P}))$ and is used as comparison as shown in Table 2.

Table 2 – Reconstruction error using Screened Poisson and Marrone's surface reconstruction. The best values for each experiment and reconstruction method are in bold.

Experiment	Resampling Method	Screened Poisson	Marrone
DDB	No Resamp	0.439	1.052
	Ando	0.438	1.057
	Zhang	0.432	0.964
	Proposed Method	0.402	0.914
Water Crown	No Resamp	0.403	0.847
	Ando	0.413	0.850
	Zhang	0.404	0.858
	Proposed Method	0.380	0.839
BoxDrop	No Resamp	0.489	1.152
	Ando	0.514	1.130
	Zhang	0.507	0.885
	Proposed Method	0.437	0.803
Waterfall	No Resamp	0.364	0.991
	Ando	0.426	1.054
	Zhang	0.330	0.810
	Proposed Method	0.280	0.778

By the table above (2), the error in all experiments using both surface reconstruction methods is lower for the proposed methods. By this results, it can be concluded that the proposed method is expected to be more suitable for this type of application.

4.3.2. Qualitative analysis

In order to further explore the results of the experiments, visual analyses together with the quantitative analyses has been performed.

Figure 17 shows a toy example in which the normals are computed using the SPH approximation proposed by Müller, Charypar and Gross (2003). Due to the deficiency of particles in the thin features (isolated points and line segments), the normals are not consistently oriented; as a consequence, the surface reconstruction obtained using RBF implicits (CARR *et al.*, 2001a) provides a spurious result (Figure 17c). This issue is easily remedied by the particle resampling method (Figure 17d), where the thin features are satisfactorily reconstructed.

Figure 17 – Surface reconstruction in a toy example using RBF implicits: (a) boundary particles (black) with oriented normals (red) and (b) the same dataset after the resampling. The signed distance field ϕ : (c) without resampling and (d) with resampling. The values of ϕ vary from negative (red) to positive (blue) values and its zero level-set (black).

Figure 18 shows a water crown splash caused by the collision of a water drop with a liquid tank. The free-surface fitting is benefited by the resampling method regarding the preservation of the thin sheets. While it is able to capture the thin layers of liquid, on the other hand, the deterioration of the free-surface in the absence of the method caused by the particle deficiency in regions containing small-scale features can be observed.

Figure 18 – Water crown splash. Free-surface fitting without (left) and with (right) the resampling method.

Figure 19 shows a comparison between the method against other state-of-art particle resampling methods. The comparison is performed on a liquid drop simulation, where a thin film of liquid arises from the impact of the drop on a cube. For this experiment, the particle refinement strategies proposed by [Ando, Thürey and Tsuruno \(2012\)](#) and [Zhang, Solenthaler and Pajarola \(2008\)](#) has been implemented. In order to perform a visual inspection, the boundary particles as black dots is shown to better assess the aspect of the surface fitting after the particle refinement step. In this sense, a smaller amount of boundary particles visualized implies in a better fitting.

Figure 19 – Liquid drop on a cube. Comparison between the method (right) and the methods of [Ando, Thürey and Tsuruno \(2012\)](#) (left) and [Zhang, Solenthaler and Pajarola \(2008\)](#) (middle). The boundary particles (black dots) fitted by the surface are illustrated at the top-row. Notice that the surface reconstruction using the resampling is more resistant to the deterioration of the small-scale features during a violent splash.

The method of [Ando, Thürey and Tsuruno \(2012\)](#) fails to preserve the free-surface details due to insertion of new sampled particles tangentially to the surface; thus, the poorly sampling along the normal direction compromises the quality of the surface fitting producing an unsatisfactory visual result. Despite the acceptable result produced by the method of [Zhang, Solenthaler and Pajarola \(2008\)](#), their method extracts a thin sheet of liquid with many indent holes, even adding a large number of new sampled particles, as can be seen in the Figure 20. In turn, the method successfully generates a smooth free-surface preserving the small-scale features in an efficient manner.

Figure 20 – The number of new particles per frame in the liquid drop simulation (Fig. 19) in each method: Zhang, Solenthaler and Pajarola (2008) (yellow), Ando, Thürey and Tsuruno (2012) (blue) and the method (red).

This method can be adapted to another surface reconstruction frameworks based on boundary particle detection. For instance, Marrone *et al.* (2010) surface fitting algorithm may be used, as defined in Section 4.2.2

Figure 21 illustrates the effectiveness of the resampling method when it is applied to the surface reconstruction methods proposed by Sandim *et al.* (2016) and Marrone *et al.* (2010) in a domain with complex geometry. The adaptive sampling is capable of capturing the thin sheet produced when a liquid flows through a narrow gap imitating a waterfall, without introducing spurious artifacts in the surface reconstruction. Notice that the surface fitting without the method produces small holes and a huge bump below the thin sheet or spikes on the surface, caused by the inconsistent orientation of the normal vectors due to the poorly sampled regions in the free-surface.

Figure 21 – The effect of the resampling method in the surface reconstruction schemes proposed by Sandim *et al.* (2016) and Marrone *et al.* (2010).

Figure 22 shows the robustness of the method regarding the particle resolution. For that, two sphere-shaped particle sets are employed, one in high-resolution ($\approx 1.9 \times 10^6$ particles) and the other in low-resolution ($\approx 2.4 \times 10^5$ particles). The comparison is made using the Enright deformation test (ENRIGHT *et al.*, 2002), a classical benchmark for level-set methods and surface tracking algorithms. The test consists in deforming each sphere according to an artificial velocity field. After the sphere reaches its maximum stretching, the direction of the velocity field is reversed to restore its original shape. Performing the traditional free-surface fitting, the surface in high-resolution is successfully reconstructed, while in low-resolution the surface breaks up in the thin sheet region. Applying the resampling method to the low-resolution model, the surface remains intact and the number of additional particles created by the method represents less than 0.5% of the total amount of the original boundary particles, which demonstrates its efficiency.

Figure 22 – Enright test at maximum stretching time. From left to right: free-surface fitting without the resampling in high-resolution, without the resampling in low-resolution, with the resampling in low-resolution and the new particles (red) inserted by the method.

CONCLUSION

5.1. Conclusion

A novel particle resampling method for the free-surface fitting of liquids from particle-based fluid simulations has been presented. Given a particle set representing a particle-based fluid, firstly the method counts with any boundary detection method, such as proposed by [Sandim *et al.* \(2016\)](#) and [Marrone *et al.* \(2010\)](#). Each boundary particle is then classified according to the deformation of the free-surface and the cardinality of its local neighborhood by using PCA. After that, the feature particle is replaced by new particles which surround it, according to its label as illustrated in [Figure 12](#) with particle misclassification correction using safe regions.

The proposed method is simple, fast, easy to implement, and only requires the particle positions to accurately identify and refine regions with small-scale features, i.e., it does not demand any kernel-based interpolation of geometric or physical quantities.

In order to perform the surface fitting, the Screened Poisson method and Marrone's Surface Fitting method on point cloud formed by the boundary particles and surface normals has been applied.

As a result, a framework that can be effectively used for rendering of free-surfaces in particle-based fluids has been developed, as attested by the experiments and comparisons carried out in this thesis.

As future work, it's intended to integrate the framework into SPH and FLIP fluid solvers, thus enabling the particles resulting from resampling to be incorporated into the fluid for simulation of free-surface phenomena, e.g., involving surface tension. Also, the method can be properly adapted to use graphics processing units (GPUs) as parallel coprocessors, since computations at all stages of the method are performed locally and independently for each particle.

5.2. Bibliographic production related to this thesis

These papers were made and submitted during this master course.

- M. Sandim, N. Oe, D. Cedrim, P. Pagilosa, A. Paiva - Particle resampling for free-surface fitting Submitted to Computers & Graphics

BIBLIOGRAPHY

ADAMS, B.; PAULY, M.; KEISER, R.; GUIBAS, L. J. Adaptively sampled particle fluids. **ACM Trans. Graph.**, v. 26, n. 3, 2007. Citations on pages [13](#), [21](#), and [22](#).

AKINCI, G. **Smooth Surface Reconstruction for SPH**. Master's Thesis (Master's Thesis) — Albert-Ludwig-University of Freiburg, 2010. Citation on page [28](#).

AKINCI, G.; IHMSEN, M.; AKINCI, N.; TESCHNER, M. Parallel surface reconstruction for particle-based fluids. **Computer Graphics Forum**, v. 31, n. 6, p. 1797–1809, 2012. Citation on page [42](#).

ALEXA, M.; BEHR, J.; COHEN-OR, D.; FLEISHMAN, S.; LEVIN, D.; SILVA, C. T. Computing and rendering point set surfaces. **IEEE Transactions on visualization and computer graphics**, IEEE, v. 9, n. 1, p. 3–15, 2003. Citation on page [21](#).

ANDO, R.; THÜREY, N.; TSURUNO, R. Preserving fluid sheets with adaptively sampled anisotropic particles. **IEEE Trans. Vis. Comput. Graph.**, v. 18, n. 8, p. 1202–1214, 2012. Citations on pages [14](#), [17](#), [23](#), [36](#), [47](#), and [48](#).

ANDO, R.; THÜREY, N.; WOJTAN, C. Highly adaptive liquid simulations on tetrahedral meshes. **ACM Trans. Graph.**, v. 32, n. 4, p. 103:1–103:10, 2013. Citations on pages [13](#) and [24](#).

BELYAEV, A.; OHTAKE, Y. A comparison of mesh smoothing methods. In: **Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics**. [S.l.: s.n.], 2003. p. 83–87. Citation on page [42](#).

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 2014 - State of the Art Reports**. [S.l.: s.n.], 2014. Citation on page [17](#).

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**. [S.l.: s.n.], 2001. p. 67–76. Citations on pages [18](#) and [46](#).

_____. Reconstruction and representation of 3d objects with radial basis functions. In: **ACM. Proceedings of the 28th annual conference on Computer graphics and interactive techniques**. [S.l.], 2001. p. 67–76. Citation on page [27](#).

CRESPO, A.; DOMÍNGUEZ, J.; ROGERS, B.; GÓMEZ-GESTEIRA, M.; LONGSHAW, S.; CANELAS, R.; VACONDIO, R.; BARREIRO, A.; GARCÍA-FEAL, O. DualSPHysics: Open-source parallel CFD solver based on smoothed particle hydrodynamics (SPH). **Computer Physics Communications**, v. 187, p. 204–216, 2015. Citation on page [44](#).

ENRIGHT, D.; FEDKIW, R.; FERZIGER, J.; MITCHELL, I. A hybrid particle level set method for improved interface capturing. **J. Comput. Phys.**, v. 183, n. 1, p. 83–116, 2002. Citation on page [49](#).

GOIS, J. P.; BUSCAGLIA, G. C. Resampling strategies for deforming mls surfaces. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**. [S.l.], 2010. v. 29, n. 6, p. 1969–1980. Citation on page 21.

GUENNEBAUD, G.; JACOB, B. *et al.* **Eigen v3**. 2010. <<http://eigen.tuxfamily.org>>. Citation on page 36.

HE, X.; LIU, N.; WANG, G.; ZHANG, F.; LI, S.; SHAO, S.; WANG, H. Staggered meshless solid-fluid coupling. **ACM Trans. Graph.**, v. 31, n. 6, p. 149:1–149:12, 2012. Citation on page 32.

HONG, W.; HOUSE, D. H.; KEYSER, J. Adaptive particles for incompressible fluid simulation. **The Visual Computer**, v. 24, n. 7, p. 535–543, 2008. Citation on page 23.

HOPPE, H.; DEROSE, T.; DUCHAMP, T.; MCDONALD, J.; STUETZLE, W. **Surface reconstruction from unorganized points**. [S.l.]: ACM, 1992. Citation on page 27.

IHMSEN, M.; ORTHMANN, J.; SOLENTHALER, B.; KOLB, A.; TESCHNER, M. SPH Fluids in Computer Graphics. In: **Eurographics 2014 - State of the Art Reports**. [S.l.: s.n.], 2014. Citation on page 17.

JANG, T.; NOH, J. A geometric approach to animating thin surface features in smoothed particle hydrodynamics water. **Computer Animation and Virtual Worlds**, v. 26, n. 2, p. 161–172, 2015. Citations on pages 13, 17, 24, and 25.

JEONG, D.; LI, Y.; LEE, H. J.; LEE, S. M.; YANG, J.; PARK, S.; KIM, H.; CHOI, Y.; KIM, J. Efficient 3d volume reconstruction from a point cloud using a phase-field method. **Mathematical Problems in Engineering**, Hindawi, v. 2018, 2018. Citation on page 27.

JIA, Y.-B. Singular value decomposition. **Department of Computer Science, Iowa State University, Ames, Iowa, USA**, 2013. Citation on page 36.

JOLLIFFE, I. **Principal Component Analysis**. 2nd. ed. [S.l.]: Springer, 2002. Citation on page 18.

KATZ, S.; TAL, A.; BASRI, R. Direct visibility of point sets. **ACM Trans. Graph.**, v. 26, n. 3, 2007. Citation on page 32.

KAZHDAN, M.; HOPPE, H. Screened poisson surface reconstruction. **ACM Trans. Graph.**, v. 32, n. 3, p. 29:1–29:13, 2013. Citations on pages 18, 27, and 41.

MARRONE, S.; COLAGROSSI, A.; TOUZÉ, D. L.; GRAZIANI, G. Fast free-surface detection and level-set function definition in SPH solvers. **J. Comput. Phys.**, v. 229, n. 10, p. 3652–3663, 2010. Citations on pages 13, 14, 31, 32, 35, 41, 43, 48, and 51.

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)**. [S.l.: s.n.], 2003. p. 154–159. Citations on pages 28, 32, and 46.

OHTAKE, Y.; BELYAEV, A.; ALEXA, M.; TURK, G.; SEIDEL, H.-P. Multi-level partition of unity implicits. **ACM Trans. Graph.**, v. 22, n. 3, p. 463–470, 2003. Citation on page 18.

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)**. [S.l.: s.n.], 2013. p. 95–103. Citation on page [32](#).

ORTHMANN, J.; KOLB, A. Temporal blending for adaptive SPH. **Computer Graphics Forum**, v. 31, n. 8, p. 2436–2449, 2012. Citation on page [24](#).

SANDIM, M.; CEDRIM, D.; NONATO, L. G.; PAGLIOSA, P.; PAIVA, A. Boundary detection in particle-based fluids. **Computer Graphics Forum**, v. 35, n. 2, p. 215–224, 2016. Citations on pages [13](#), [14](#), [18](#), [19](#), [31](#), [32](#), [41](#), [44](#), [48](#), and [51](#).

SHLENS, J. A tutorial on principal component analysis. **arXiv preprint arXiv:1404.1100**, 2014. Citation on page [29](#).

SOLENTHALER, B.; GROSS, M. Two-scale particle simulation. **ACM Trans. Graph.**, v. 30, n. 4, p. 81:1–81:8, 2011. Citation on page [24](#).

SOLENTHALER, B.; ZHANG, Y.; PAJAROLA, R. Efficient refinement of dynamic point data. In: **Eurographics Symposium on Point-Based Graphics**. [S.l.: s.n.], 2007. Citations on pages [17](#) and [22](#).

TAUBIN, G. A signal processing approach to fair surface design. In: **SIGGRAPH '95**. [S.l.: s.n.], 1995. p. 351–358. Citation on page [42](#).

YU, J.; TURK, G. Reconstructing surfaces of particle-based fluids using anisotropic kernels. In: **Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation**. [S.l.: s.n.], 2010. p. 217–225. Citations on pages [13](#), [18](#), [23](#), and [25](#).

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)**. [S.l.: s.n.], 2008. p. 137–146. Citations on pages [13](#), [14](#), [17](#), [22](#), [47](#), and [48](#).

ZHU, Y.; BRIDSON, R. Animating sand as a fluid. **ACM Trans. Graph.**, v. 24, n. 3, p. 965–972, 2005. Citations on pages [13](#), [17](#), [21](#), and [25](#).