

Reconstructing Surfaces of Particle-Based Fluids Using Anisotropic Kernels

Jihun Yu¹ and Greg Turk²

¹New York University

²Georgia Institute of Technology

Abstract

In this paper we present a novel surface reconstruction method for particle-based fluid simulators such as Smoothed Particle Hydrodynamics. In particle-based simulations, fluid surfaces are usually defined as a level set of an implicit function. We formulate the implicit function as a sum of anisotropic smoothing kernels, and the direction of anisotropy at a particle is determined by performing Principal Component Analysis (PCA) over the neighboring particles. In addition, we perform a smoothing step that re-positions the centers of these smoothing kernels. Since these anisotropic smoothing kernels capture the local particle distributions more accurately, our method has advantages over existing methods in representing smooth surfaces, thin streams and sharp features of fluids. Our method is fast, easy to implement, and our results demonstrate a significant improvement in the quality of reconstructed surfaces as compared to existing methods.

Categories and Subject Descriptors (according to ACM CCS): 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

It is becoming increasingly popular to create animated liquids using physics-based simulation methods for feature film effects and interactive applications. There exist two broad categories for simulation methods based on their different approaches to spatial discretization: mesh-based methods and mesh-free methods. In mesh-based methods, the simulation domain is discretized into mesh grids and the values of physical properties on grid points are determined by solving the governing equations. In mesh-free methods, on the other hand, the fluid volume is discretized into sampled particles that carry physical properties and that are advected in space by the governing equations. In recent years, mesh-free methods have become a competitive alternative to mesh-based methods due to various advantages such as their inherent mass conservation, the flexibility of simulation in unbounded domains, and ease of implementation. Among various mesh free methods, Smoothed Particle Hydrodynamics (SPH) is the most popular approach for simulating fluid since it is computationally simple and efficient

compared to others. In computer graphics, SPH has been successfully used for the simulation of free-surface fluids [MCG03], fluid interface [MSKG05, SP08], fluid-solid coupling [MST*04, LAD08, BTT09], deformable body [BIT09], multi-phase fluid [MKN*04, KAG*05, SSP07] and fluid control [TKPR06].

Although SPH has been used to simulate various fluid phenomena, extracting high quality fluid surfaces from the particle locations is not straightforward. Classical surface reconstruction methods have difficulties in producing smooth surfaces due to irregularly placed particles. Few researchers have successfully addressed this issue of reconstructing smooth fluid surfaces from particles. In this paper, we propose a novel surface extraction method that significantly improves the quality of the reconstructed surfaces. Our new method can create smooth surfaces and thin streams along with sharp features such as edges and corners. The key to our method is to use a stretched, anisotropic smoothing kernel to represent each particle in the simulation. The orientation and scale of the anisotropy is determined by captur-

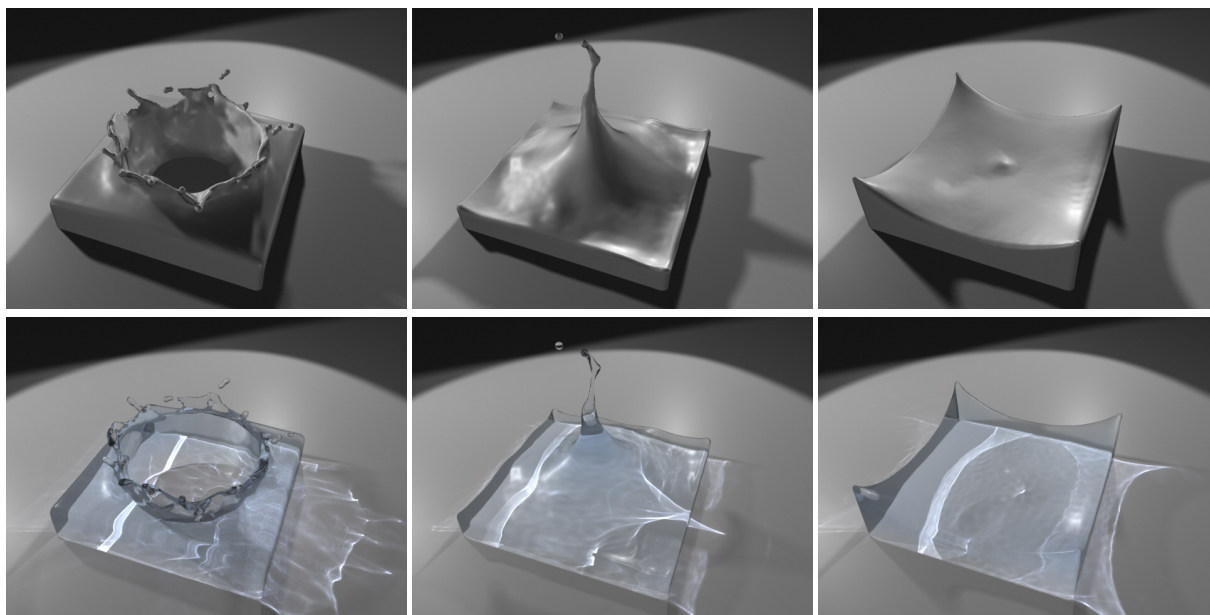


Figure 1: Water splash.

ing each particle's neighborhood spatial distribution. We obtain the neighborhood distribution in the form of covariance tensor and analyze it through Principle Component Analysis (PCA). We then use these principal components to orient and scale the anisotropic kernel. We adjust the centers of these kernels using a variant of Laplacian smoothing to counteract the irregular placement of particles. A new density field is then constructed by the weighted mass contribution from the smoothing kernels. Finally, the renderable surface is reconstructed from the iso-surface of the given density field. We show that our new method leads to the realistic visualization of fluid surfaces and that it outperforms existing methods for handling smooth and thin surfaces with sharp features. The simplicity and efficiency of our method facilitates the incorporation of our method with existing SPH simulation schemes with little additional effort.

2. Related Work

Because the surface representation of a fluid is crucial for realistic animation, methods for reconstructing and tracking fluid surfaces have been a topic of research since the fluid simulation was first introduced in computer graphics. In non-Lagrangian simulation frameworks, numerous methods has been proposed such as level-set methods [OF02], particle level-set methods [EFFM02, EMF02, ELF05], semi-Lagrangian contouring [Str01, BGOS06], volume-of-fluid methods (VOF) [HN81] and explicit surface tracking [BB06, BB09, WTGT09, Mül09].

In mesh-free (Lagrangian) simulation frameworks,

Blinn [Bli82] introduced the classic blobby spheres approach. In this method, an isosurface is extracted from a scalar field that is constructed from a sum of radial basis functions that are placed at each particle center. One of the drawbacks of Blinn's original formulation is that high or low densities of particles will cause bumps or indentations on the surface. Noting this problem, Zhu and Bridson [ZB05] modify this basic algorithm to compensate for local particle density variations. They calculate a scalar field from the particle positions that is much like a radial basis function that is centered at a particle. For a given location in space, they calculate a scalar value from a basis function whose center is a weighted sum of nearby particle centers, and whose radius is a weighted sum of particle radii. They then sample this scalar distance function on a grid, perform a smoothing pass over the grid, and then extract an isosurface mesh from the grid. Their results are considerably smoother than the classic blobby spheres surface. Adams et al. [APKG07] further improved upon the method of Zhu and Bridson by tracking the particle-to-surface distances over time. Specifically, they retain a sampled version of a signed distance field at each time step, and they use this to adjust per-particle distances to the surface. They perform particle redistancing by propagating the distance information from surface particles to interior particles using a fast marching scheme. The final surface is from the Zhu and Bridson scalar field, but calculated using these new per-particle distances. This method is successful at generating smooth surfaces both for fixed-radius and adaptively-sized particles.

One drawback of these aforementioned methods is a typ-

ical assumption that the smoothing kernel of each particle is isotropic, and the spherical shape of the kernel makes it difficult to produce flat surfaces and sharp features. In contrast to these methods, our approach uses anisotropic kernels to stretch spheres into ellipsoids in order to alleviate those limitations.

Desbrun and Cani-Gascuel [DCG98] and Premoze et al. [PTB*03] use a different surface tracking method in which a scalar field is advected on a Eulerian grid. Unfortunately this approach is more difficult to use for unbounded simulations.

Recently, an alternative method of surface reconstruction was proposed by Williams [Wi08]. In his method, a non-linear optimization problem is solved iteratively to achieve global smoothness on surface mesh. The important contribution of the method is that the perfectly flat surfaces can be generated under certain conditions. Sin et al. [SBH09] use level-set variants of the original method to alleviate the problem of temporal coherence.

Our anisotropic kernel approach is inspired by the work of Owen et al. [OVSM95] and Liu et al. [LLL06]. They adapt anisotropic kernels to simulate large deformations of materials in the SPH framework, and their primary interest is in simulation accuracy. In their approach, the axes of their anisotropic kernels evolve in time according to the strain-rate tensor estimates. Our approach is also related to the work of Kalaiah and Varshney [KV03] and Dinh et al. [DTS01]. Kalaiah and Varshney apply PCA to point clouds for point-based modeling. Dinh et al. reconstruct surfaces from voxel carving data by combining anisotropic kernels with variational implicit surfaces.

3. SPH Framework

In SPH, the fluid volume is described as a set of particles with prescribed masses. In a given simulation step, physical quantities such as density and pressure are represented by values that are associated with each particle. For particle i at location \mathbf{x}_i , the density ρ_i is interpolated by a sum of the weighted contributions of nearby particle masses m_j :

$$\rho_i = \sum_j m_j W(\mathbf{x}_j - \mathbf{x}_i, h_j), \quad (1)$$

where W is the smoothing kernel and h_j is the smoothing radius associated with particle j . The pressure p_i of particle i is typically described as a function of the density of the fluid such as given by the Tait equation [Mon94], which is

$$p_i = k\rho_0 \left(\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right), \quad (2)$$

where k and γ are stiffness parameters and ρ_0 is the rest density of the fluid. In the SPH framework, the Navier Stokes

equation, discretized on particle locations, becomes an Ordinary Differential Equation (ODE) of the form:

$$\rho_i \frac{\partial \mathbf{v}_i}{\partial t} = -\langle \nabla p \rangle(\mathbf{x}_i) + \mu \langle \Delta \mathbf{v} \rangle(\mathbf{x}_i) + \mathbf{f}_i^{ext}, \quad (3)$$

where \mathbf{v}_i is the velocity, \mathbf{f}_i^{ext} is an external force (such as gravity), μ is the viscosity constant, and $\langle \nabla p \rangle(\mathbf{x}_i)$ and $\langle \Delta \mathbf{v} \rangle(\mathbf{x}_i)$ are approximations of the pressure gradient and the velocity Laplacian at \mathbf{x}_i in the SPH framework. For various approximations of differential operators in SPH framework, we refer the readers to [AW09].

In this paper, we simulate all of our examples using the Weakly Compressible SPH (WCSPH) framework [BT07]. Note that our method can be used with any SPH framework, as long as the simulator provides particle positions, radii, masses and densities. In the case that mass and radius are global constants, our approach can be generalized to work with other Lagrangian simulation frameworks such as Particle-in-Cell (PIC) and Fluid-Implicit Particle (FLIP) [ZB05].

4. Surface Reconstruction

4.1. Surface Definition

Our surface definition is based on the approach proposed in [MCG03], where the surface is defined as an isosurface of a scalar field

$$\phi(\mathbf{x}) = \sum_j \frac{m_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h_j), \quad (4)$$

and W is an isotropic smoothing kernel of the form

$$W(\mathbf{r}, h) = \frac{\sigma}{h^d} P \left(\frac{\|\mathbf{r}\|}{h} \right). \quad (5)$$

In the above equation, σ is a scaling factor, d is the dimension of the simulation, \mathbf{r} is a radial vector and P is a symmetric decaying spline with finite support. The scalar field $\phi(\mathbf{x})$ is designed as a normalized density field that smooths out the scalar value of 1 at each particle's position over a continuous domain, and an isosurface from $\phi(\mathbf{x})$ gives a surface representation that coats the particles. However, the resulting surfaces often have bumps, and there are two reasons for this. First, the irregular placement of particles makes it difficult to represent an absolutely flat surface. Although irregular sampling is an essential feature of any Lagrangian scheme, this irregularity of the positions of the boundary particles can make surfaces appear blobby. Second, the spherical shape of the smoothing kernels is not suitable to describe the density distribution near a surface. That is, in order to correctly model surface geometry, it is necessary for the density of the near-surface particles to decrease at different rates in different directions.

To resolve the problem of irregular particle placement, we apply one step of diffusion smoothing to the location of the kernel centers. This process can be interpreted as a 3D variant of Laplacian smoothing as described in [Tau00], and has an effect of denoising point clouds. The updated kernel centers $\bar{\mathbf{x}}_i$ are calculated by

$$\bar{\mathbf{x}}_i = (1 - \lambda)\mathbf{x}_i + \lambda \frac{\sum_j w_{ij}\mathbf{x}_j}{\sum_j w_{ij}}, \quad (6)$$

where w is a suitable finite support weighting function and λ is a constant with $0 < \lambda < 1$. We use λ between 0.9 and 1 in our examples to maximize the smoothing effect. Note that this smoothing process is used only for surface reconstruction, and the averaged positions are not carried back into the simulation. Typically, Laplacian smoothing results in volume shrinking, and our approach also shrinks the fluid volume slightly by moving the kernels for boundary particles towards the inside. However, in contrast to level-set methods, our approach does not shrink volume continuously as the simulation evolves. Furthermore, the analysis in Appendix A shows that the maximum distance from our reconstructed surfaces to the original particle positions is within a small constant of the particle radius scale.

To cope with the problem of density distributions near the surface, our new approach is designed to capture the density distribution more accurately by allowing the smoothing kernels to be anisotropic. By replacing h with a $d \times d$ real positive definite matrix \mathbf{G} , we can simply redefine W to be an anisotropic kernel

$$W(\mathbf{r}, \mathbf{G}) = \sigma \|\mathbf{G}\| P(\|\mathbf{G}\mathbf{r}\|). \quad (7)$$

The linear transformation \mathbf{G} rotates and stretches the radial vector \mathbf{r} . Therefore $W(\mathbf{r}, \mathbf{G})$ becomes an anisotropic kernel, and isosurfaces of W are ellipsoids instead of spheres. Note that the isotropic kernel can be treated as a special case of the anisotropic kernel by letting $\mathbf{G} = h^{-1}\mathbf{I}$ where \mathbf{I} is an identity matrix. The key idea of our new method is to associate an anisotropy matrix \mathbf{G} with each particle so that for particle j , \mathbf{G}_j describes better the neighborhood density distribution.

Once All \mathbf{G}_j 's and $\bar{\mathbf{x}}_j$'s are computed, we extract an iso-surface from a redefined scalar field

$$\phi_{new}(\mathbf{x}) = \sum_j \frac{m_j}{\rho_j} W(\mathbf{x} - \bar{\mathbf{x}}_j, \mathbf{G}_j). \quad (8)$$

It is necessary to point out that the equation of W is depending on the SPH simulation, since different SPH schemes can use different W 's for density computation. For our examples, we use a B-cubic spline kernel from [BT07].

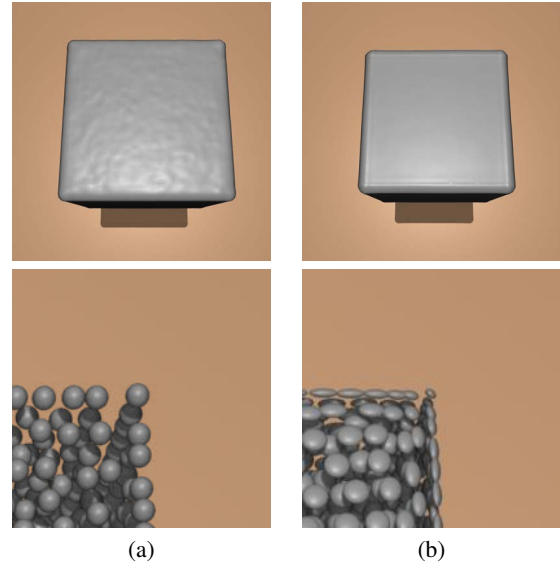


Figure 2: A comparison between the surface reconstruction using isotropic kernels (a) and our anisotropic kernels (b). Top row: the surface of SPH particles from a single dam break simulation. Bottom row: Illustration of particles at the top left corner. The shape of a particle in (b) represents the anisotropy of the corresponding smoothing kernel. Note that our approach constructs a flat surface with sharp edges and corners from properly stretched particles.

4.2. Determining the Anisotropy

As mentioned in the previous subsection, our new approach determines an anisotropy matrix \mathbf{G} for each particle in order to more accurately describe the density distribution around the particle. For example, in the neighborhood of a particle that is inside the fluid volume, the density is likely to be constant in all directions, making the corresponding \mathbf{G} a scalar multiple of an identity matrix to keep the smoothing kernel W isotropic. On the other hand, around a particle that is near a flat surface, the particle density will decay faster along the normal axis than along the tangential axes. Then \mathbf{G} should stretch W along the tangential axes and shrink W along the normal axis. At a sharp feature, the density will decay sharply in several directions, and \mathbf{G} should shrink W in order to capture the sharp feature. See Figure 2 for a comparison between isotropic and anisotropic kernels that are near the surface of a region of fluid.

In order to determine \mathbf{G} , we apply the weighted version of Principal Component Analysis (WPCA) that is proposed in [KC03] to the neighborhood particle positions. A drawback of the conventional PCA is its sensitivity to outliers, and it often produces inaccurate information when the number of samples is small and the sample positions are noisy, which commonly happens in particle-based fluids. In contrast, WPCA achieves significant robustness towards outliers

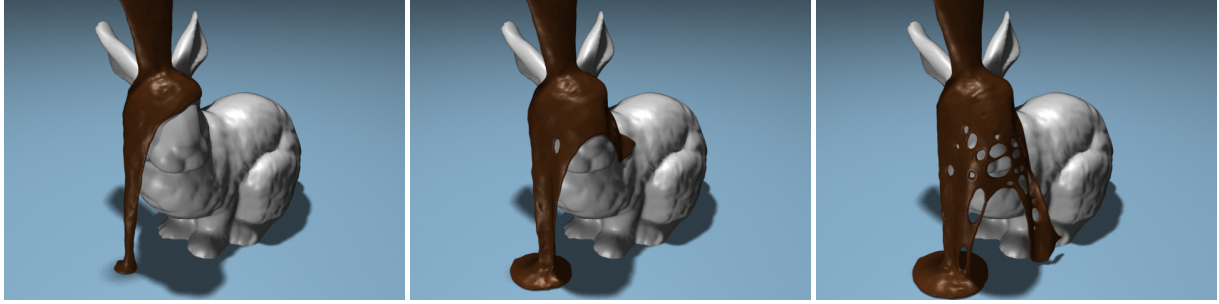


Figure 3: Melted chocolate falling on a Bunny.

and noisy data by assigning appropriate weights to the data points. Specifically, WPCA begins by computing a weighted mean of the data points. Next, WPCA constructs a weighted covariance matrix \mathbf{C} with a zero empirical mean and performs an eigendecomposition on \mathbf{C} . The resulting eigenvectors give the principal axes, and the eigenvalues indicates the variance of points along the corresponding eigenvalues. We then construct an anisotropy matrix \mathbf{G} to match the smoothing kernel W with the output of WPCA.

In our approach, the weighted mean \mathbf{x}_i^w and the covariance matrix \mathbf{C}_i of particle i are formulated as follows:

$$\mathbf{C}_i = \sum_j w_{ij} (\mathbf{x}_j - \mathbf{x}_i^w) (\mathbf{x}_j - \mathbf{x}_i^w)^T / \sum_j w_{ij}, \quad (9)$$

$$\mathbf{x}_i^w = \sum_j w_{ij} \mathbf{x}_j / \sum_j w_{ij}. \quad (10)$$

The function w_{ij} is an isotropic weighting function with respect to particle i and j with support r_i .

$$w_{ij} = \begin{cases} 1 - (\|\mathbf{x}_i - \mathbf{x}_j\|/r_i)^3 & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < r_i, \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

With the finite support of w_{ij} , the computation is confined to the neighborhood particles within the radius r_i . In our examples, we choose r_i to be $2h_i$ in order to include enough neighborhood particles and obtain reasonable anisotropy information. This kernel is also used to compute the averaged position of the particles in Equ. 6.

With each particle, the singular value decomposition (SVD) of the associated \mathbf{C} gives the directions of stretch or compression for deforming the smoothing kernel W in terms of eigenvectors and eigenvalues. The SVD yields

$$\mathbf{C} = \mathbf{R}\mathbf{\Sigma}\mathbf{R}^T, \quad (12)$$

$$\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_d). \quad (13)$$

where \mathbf{R} is a rotation matrix with principal axes as column vectors, and $\mathbf{\Sigma}$ is a diagonal matrix with eigenvalues $\sigma_1 \geq \dots \geq \sigma_d$. In order to deal with singular matrices and prevent extreme deformations, we check if $\sigma_1 \geq k_r \sigma_d$ with a suitable positive constant $k_r > 1$. The condition is true when the largest variance in one principal axis is much bigger than the smallest variance in another axis. In this case, we modify \mathbf{C} such that the ratio between any two eigenvalues are within k_r . Also, when the number of particles in the neighborhood is small, we reset W to a spherical shape by setting $\mathbf{G} = k_n \mathbf{I}$ in order to prevent poor particle deformations for nearly isolated particles. In addition, we multiply \mathbf{C} by scaling factor k_s such that $\|\mathbf{C}\| \approx 1$ for the associated particle inside fluid volume, in order to keep the volume of W constant for particles with the full neighborhood. The aforementioned processes are formulated as follows to obtain a modified covariance matrix $\tilde{\mathbf{C}}$.

$$\tilde{\mathbf{C}} = \mathbf{R}\tilde{\mathbf{\Sigma}}\mathbf{R}^T \quad (14)$$

$$\tilde{\mathbf{\Sigma}} = \begin{cases} k_s \text{diag}(\sigma_1, \tilde{\sigma}_2, \dots, \tilde{\sigma}_d) & \text{if } N > N_\epsilon, \\ k_n \mathbf{I} & \text{otherwise} \end{cases} \quad (15)$$

where $\tilde{\sigma}_k = \max(\sigma_k, \sigma_1/k_r)$, N the number of neighboring particles and N_ϵ is a threshold constant. In our examples, we use $k_r = 4$, $k_s = 1400$, $k_n = 0.5$ and $N_\epsilon = 25$.

In order to make the kernel W of particle i deform according to $\tilde{\mathbf{C}}_i$, \mathbf{G}_i must be an inversion of $\tilde{\mathbf{C}}_i$ and scaled by $1/h_i$ to reflect the original radius of particle i . Then our approach produces \mathbf{G}_i as a symmetric matrix of the form:

$$\mathbf{G}_i = \frac{1}{h_i} \mathbf{R}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{R}^T. \quad (16)$$

5. Implementation

For the neighborhood search, we use a variation of a hash grid described in [AW09] to deal with the ellipsoidal support of the smoothing kernels. At every reconstruction step, we

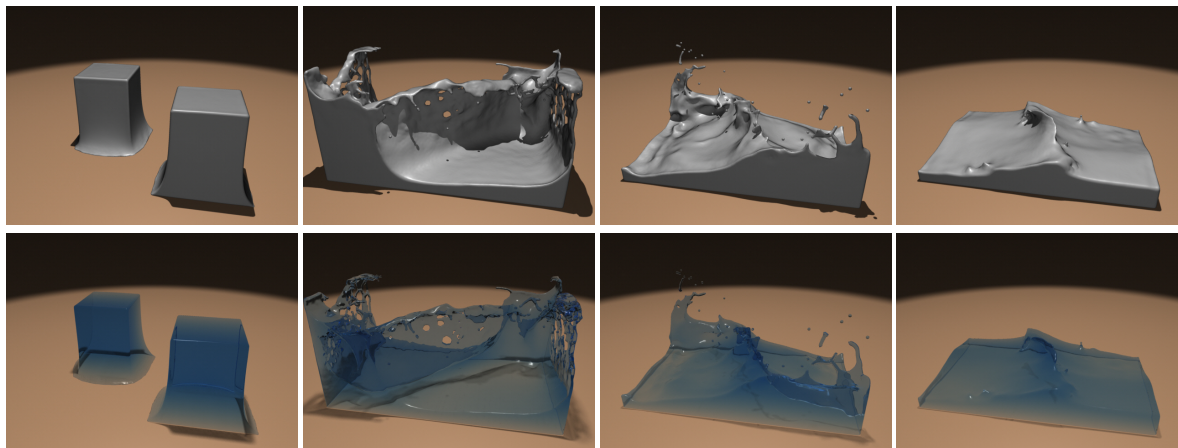


Figure 4: Double dam break simulation.

first compute an axis aligned bounding box for the ellipsoid associated with each particle. We then store an index of the particle in the hash grid cells that overlap with the bounding box. In order to find the neighbors for a particle in a grid cell, we simply examine each particle that is stored in the cell and tag as a neighbor the ones that are inside the ellipsoid.

Obstacles are represented as tetrahedral volume meshes for collision detection with particles. We use the Marching Cubes algorithm [LC87] to create a mesh that represents the fluid surface from the scalar field of Equation 8.

All of our simulations and surface reconstruction algorithms were run on a 2.4 GHz Intel Core2 Duo CPU with 1.72GB of memory. We use NVIDIA Gelato for rendering the resulting animations. All of our results were simulated using the Weekly Compressible SPH (WCSPH) approach of Becker et al. [BT07]. We used a fixed time step of 0.0001s for running our simulations.

6. Results

In this section, we describe three simulations that were used to evaluate our surface reconstruction method: water crown, flow on bunny, and double dam break.

Figure 1 is an animation of a small drop of water that splashes into a larger body of water, causing a water crown. Only 24k particles were used to create this simulation. Note that even at this low particle count, the particle-based nature of the simulation is difficult to discern from the images. Our anisotropic kernel reconstruction of the surface creates a water crown that is smooth and unbroken near its base, and produces plausible pinch-off at the top. When the fluid rebounds in the center, a thin spike of water is maintained due to the stretching of the smoothing kernels along the spike's axis. When the water settles, the surface is smooth.

Figure 3 shows a viscous fluid that is poured over the Stanford Bunny. The fluid sheet that falls runs off the bunny is thin, usually just one particle thick, and yet the sheet is flat and smooth. In this sheet, the kernels are stretched in the two dimensions that run parallel to the sheet, and are compressed perpendicular to the sheet.

Our last simulation is a double dam break (Figure 4), in which two blocks of water at opposite sides of a tank are suddenly released. The two parcels of water rush towards each other, collide in the center of the tank, and this throws up a thin sheet of water that runs diagonally across the tank. Similar to the bunny simulation, this thin sheet is often just one particle thick.

6.1. Comparison and Limitations

Figure 5 shows a comparison between an isotropic surface reconstruction approach [MCG03], Zhu and Bridson's approach [ZB05], the method of Adams et al. [APKG07] and our anisotropic kernel approach. The simulation that is used for comparison is the double dam break simulation with 140K particles. As the figure shows, the isotropic reconstruction method produces unacceptably bumpy surfaces. Zhu and Bridson's approach creates noticeably smoother surfaces, but some surface bumps are still apparent. In fairness to their method, Zhu and Bridson also perform a small amount of additional grid-based smoothing that we have omitted. The method of Adams et al. produces a still smoother surface, and this method creates the highest quality surfaces from among the prior methods that we have tested. Our anisotropic kernel method produces surfaces that are even smoother than the method of Adams et al. Moreover, our method creates a thin sheet of water in the center of the image that is largely unbroken, where the method of Adams et al. creates a sheet with many holes in a lace-like pattern.

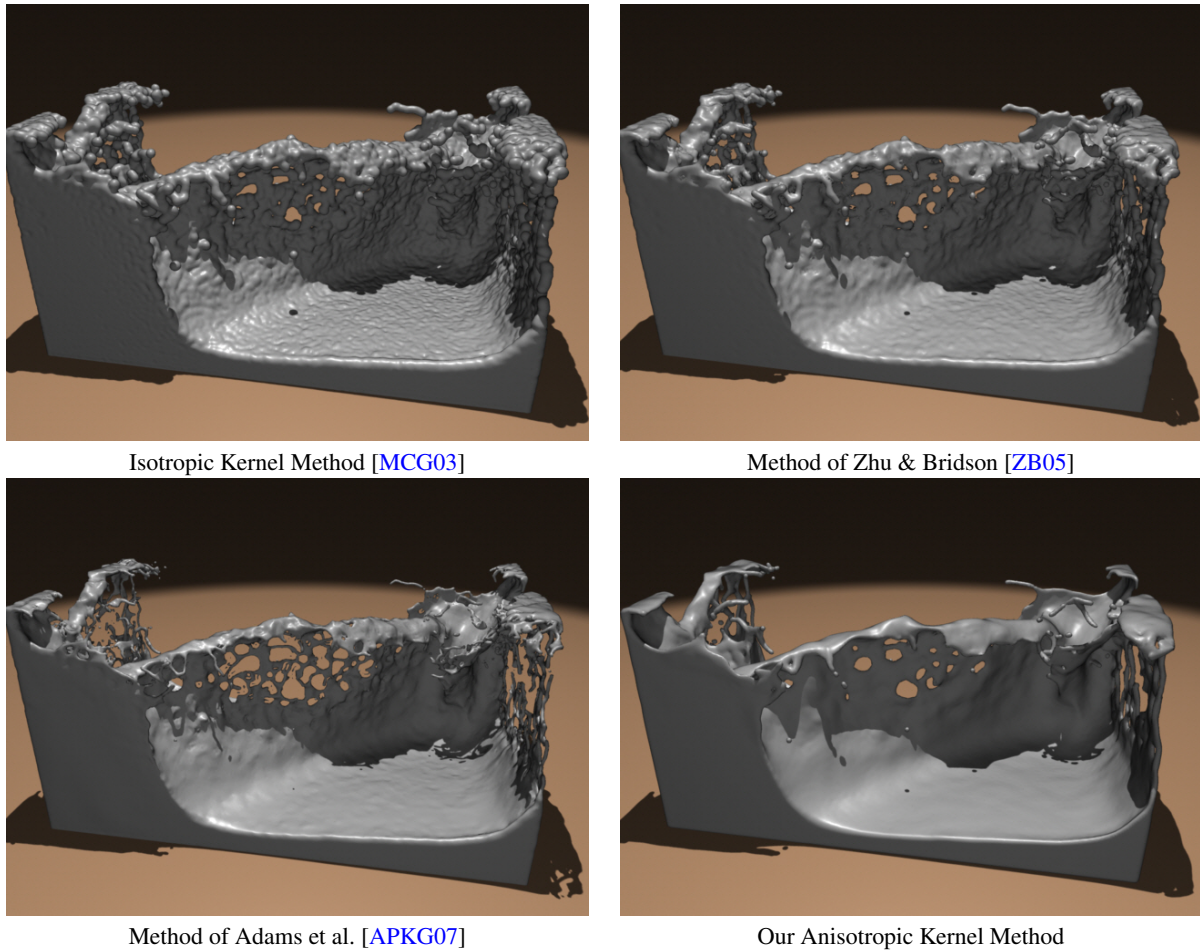


Figure 5: Comparison between different surface reconstruction approaches on the Double dam break animation.

Table 1 shows timings for the double dam break example. The dimensions of the marching cubes grid for these results is $230 \times 190 \times 350$. The timings are per-frame averages (in minutes) across all of the frames of the animation. Our surface reconstruction approach is roughly twice as expensive as the isotropic kernel method and Zhu and Bridson’s approach. The method of Adams et al. is the most time consuming of the four methods, due to the need to re-calculate the signed distance field at a rate of 300 frames-per-second. When we dropped this re-calculation to a lower rate, the surface results from the Adams method became significantly lower in quality.

There are several limitations to our approach for surface reconstruction. Perhaps the most important caveat is that the surfaces that are created using this method contain less volume than prior approaches. This is due to the averaging of particle centers. Appendix A gives an analysis of this volume difference, and demonstrates that a particle near a flat

surface will move a fraction of the smoothing kernel radius h_i . Unlike mesh-based smoothing approaches, however, our method does not shrink the surface near thin sheets of fluids. Also note that this smaller volume is only a side-effect of the surface reconstruction process, and it is not carried into the physics of the simulation.

Even though our method produces surfaces that have less noise than the other methods that we tested, it is still possible to see small bumps when the surface is magnified. These slight variations in the surface can be seen in the pattern of the caustics of the water crown animation when the water settles. We think that these slight ripples could easily be smoothed away using mesh-based smoothing, but we left our meshes un-altered in order to clarify what can be achieved using anisotropic kernels alone.

Reconstruction method	Surface reconstruction	Simulation	Opaque rendering	Transparent rendering
Isotropic	0.39			
Zhu and Bridson	0.50	2.19	0.64	20.08
Adams	1.76			
Anisotropic	0.96			

Table 1: Average per frame timings (in minutes) for four surface reconstruction methods on the double dam break simulation.

7. Conclusion and Future Work

We have presented a new method of reconstructing surfaces from particle-based fluid simulations. This method relies on repositioning and stretching the kernels for each particle according to the local distribution of particles in the surrounding area. Our method preserves thin fluid sheets, maintains sharp features, and produces smooth surfaces when the simulated fluid settles. This method is also competitive in speed as compared to other recent techniques for SPH surface reconstruction.

There are several avenues for future work using this method. One possibility is to only create stretched particles near the surface of the fluid, and not in the fluid bulk, in order to accelerate the process of surface reconstruction. In addition, since the approach only depends on local information about particle information, it would be possible to parallelize the surface creation process. This approach to surface creation should be applicable to particle-based simulations other than SPH, and in particular this approach can be tried for PIC or FLIP simulations [ZB05]. Finally, it would be interesting to investigate whether there is a way to carry texture information along with the surface, as is possible using Semi-Lagrangian contouring [BGOS06].

Acknowledgements

This work is supported in part by NSF Grants CCF-0917093, CCF-0728977 and CCF-0811485. We would like to thank the members of the Computer Graphics Lab in Georgia Institute of Technology and Chris Wojtan and Karthik Raveendran for their help in simulation and rendering.

References

- [APKG07] ADAMS B., PAULY M., KEISER R., GUIBAS L. J.: Adaptively sampled particle fluids. *ACM Trans. Graph.* 26, 3 (2007), 48. 2, 6, 7
- [AW09] ADAMS B., WICKE M.: Meshless approximation methods and applications in physics based modeling and animation. In *Eurographics 2009 Tutorials* (2009), pp. 213–239. 3, 5
- [BB06] BROCHU T., BRIDSON R.: Fluid animation with explicit surface meshes. In *Symposium on Computer Animation 2006, poster session* (2006). 2
- [BB09] BROCHU T., BRIDSON R.: Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing* 31, 4 (2009), 2472–2493. 2
- [BGOS06] BARGTEIL A., GOKTEKIN T., O'BRIEN J., STRAIN J.: A semi-Lagrangian contouring method for fluid simulation. *ACM Transactions on Graphics (TOG)* 25, 1 (2006), 38. 2, 8
- [BIT09] BECKER M., IHMSEN M., TESCHNER M.: Corrotated sph for deformable solids. In *Eurographics Workshop on Natural Phenomena* (2009). 1
- [Bli82] BLINN J.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics (TOG)* 1, 3 (1982), 235–256. 2
- [BT07] BECKER M., TESCHNER M.: Weakly compressible sph for free surface flows. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 209–217. 3, 4, 6
- [BTT09] BECKER M., TESSENDORF H., TESCHNER M.: Direct Forcing for Lagrangian Rigid-Fluid Coupling. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (2009), 493–503. 1
- [DCG98] DESBRUN M., CANI-GASCUEL M.: Active implicit surface for animation. In *Graphics Interface* (1998), pp. 143–150. 3
- [DTS01] DINH H., TURK G., SLABAUGH G.: Reconstructing surfaces using anisotropic basis functions. In *International Conference on Computer Vision (ICCV)* (2001), vol. 2, pp. 606–613. 3
- [EFFM02] ENRIGHT D., FEDKIW R., FERZIGER J., MITCHELL I.: A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics* 183, 1 (2002), 83–116. 2
- [ELF05] ENRIGHT D., LOSASSO F., FEDKIW R.: A fast and accurate semi-Lagrangian particle level set method. *Computers and Structures* 83, 6-7 (2005), 479–490. 2
- [EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 736–744. 2
- [HN81] HIRT C., NICHOLS B.: Volume of fluid/VOF/ method for the dynamics of free boundaries. *Journal of computational physics* 39, 1 (1981), 201–225. 2
- [KAG*05] KEISER R., ADAMS B., GASSER D., BAZZI P., DUTRE P., GROSS M.: A unified Lagrangian approach to solid-fluid animation. *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics 0* (2005), 125–148. 1
- [KC03] KOREN Y., CARMEL L.: Visualization of labeled data using linear transformations. In *Proceedings of IEEE Information Visualization* (2003), vol. 2003. 4
- [KV03] KALAIHAH A., VARSHNEY A.: Statistical point geometry. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), Eurographics Association, p. 115. 3
- [LAD08] LENAERTS T., ADAMS B., DUTRÉ P.: Porous flow in particle-based fluid simulations. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–8. 1

- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), ACM, pp. 163–169. 6
- [LLL06] LIU M. B., LIU G. R., LAM K. Y.: Adaptive smoothed particle hydrodynamics for high strain hydrodynamics with material strength. *Shock Waves 15* (Mar. 2006), 21–29. 3
- [MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 154–159. 1, 3, 6, 7
- [MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 141–151. 1
- [Mon94] MONAGHAN J.: Simulating free surface flows with sph. *J. Comput. Phys. 110*, 2 (1994), 399–406. 3
- [MSKG05] MÜLLER M., SOLENTHALER B., KEISER R., GROSS M.: Particle-based fluid-fluid interaction. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM, pp. 237–244. 1
- [MST*04] MÜLLER M., SCHIRM S., TESCHNER M., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids. In *Journal of Computer Animation and Virtual Worlds (CAVW)* (2004), pp. 159–171. 1
- [Mül09] MÜLLER M.: Fast and robust tracking of fluid surfaces. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), ACM, pp. 237–245. 2
- [OF02] OSHER S., FEDKIW R.: *Level set methods and dynamic implicit surfaces*. Springer Verlag, 2002. 2
- [OVSM95] OWEN J. M., VILLUMSEN J. V., SHAPIRO P. R., MARTEL H.: Adaptive Smoothed Particle Hydrodynamics: Methodology II. 3
- [PTB*03] PREMOZE S., TASDIZEN T., BIGLER J., LEFOHN A., WHITAKER R. T.: Particle-based simulation of fluids. In *Proceedings of Eurographics 2003* (2003), pp. 401–410. 3
- [SBH09] SIN F., BARGTEIL A., HODGINS J.: A point-based method for animating incompressible flow. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009). 3
- [SP08] SOLENTHALER B., PAJAROLA R.: Density contrast sph interfaces. In *Proceedings of ACM SIGGRAPH / EG Symposium on Computer Animation* (2008), pp. 211–218. 1
- [SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid–solid interactions: Research articles. *Comput. Animat. Virtual Worlds 18*, 1 (2007), 69–82. 1
- [Str01] STRAIN J.: A fast semi-Lagrangian contouring method for moving interfaces. *Journal of Computational Physics 170*, 1 (2001), 373–394. 2
- [Tau00] TAUBIN G.: Geometric signal processing on polygonal meshes. *Eurographics State of the Art Reports* (2000). 4
- [TKPR06] THÜREY N., KEISER R., PAULY M., RÜDE U.: Detail-preserving fluid control. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 7–12. 1
- [Wil08] WILLIAMS B. W.: *Fluid Surface Reconstruction from Particles*. Master's thesis, The University of British Columbia, Canada, February 2008. 3
- [WTGT09] WOJTAN C., THÜREY N., GROSS M., TURK G.: Deforming meshes that split and merge. In *ACM SIGGRAPH 2009 papers* (2009), ACM, p. 76. 2
- [ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. In *ACM SIGGRAPH 2005 Papers* (2005), ACM, p. 972. 2, 3, 6, 7, 8

Appendix A: Analysis of Volume Shrinkage

Our surface reconstruction approach applies a smoothing step to the positions of the kernel centers, and this pull the kernels on the boundary towards the bulk of the fluid. Typically, this results in a slight amount of volume shrinkage when the fluid surface is reconstructed. In this appendix, we estimate the maximum distance between the original positions of the kernel centers and the smoothed positions of the kernels for particles that are on flat surface. Suppose that a particle i is located at the origin of an Eulerian coordinate system and the neighboring particles are continuously located in a hemisphere of radius r_i above the xy plane. Due to the spatial symmetry, the weighted mean is on the positive z -axis. We compute the length of the weighted mean $\|\mathbf{x}_i\|$ from Equ. 10 and Equ. 11 using spherical coordinates. The numerator and the denominator of $\|\mathbf{x}_i^w\|$ are formulated as

$$\left\| \sum_j w_{ij} \mathbf{x}_j \right\| = \pi \int_0^{r_i} \int_0^{\frac{\pi}{2}} \sin 2\phi (r^3 - r^6 r_i^{-3}) dr d\phi, \quad (17)$$

and

$$\sum_j w_{ij} = 2\pi \int_0^{r_i} \int_0^{\frac{\pi}{2}} \sin \phi (r^2 - r^5 r_i^{-3}) dr d\phi. \quad (18)$$

A simple algebraic manipulation yields

$$\|\mathbf{x}_i^w\| = 9r_i/28. \quad (19)$$

Using Equ. 6, we estimate a bound between the kernel center position \mathbf{x}_i and the updated center position $\bar{\mathbf{x}}_i$ after the volume smoothing by

$$\|\bar{\mathbf{x}}_i - \mathbf{x}_i\| = \|\lambda(\mathbf{x}_i^w - \mathbf{x}_i)\| \leq 9\lambda r_i/28. \quad (20)$$

With the values $\lambda = 0.9$, $r_i = 2h_i$ that we used in our examples, we obtain $\|\bar{\mathbf{x}}_i - \mathbf{x}_i\| \leq 0.58h_i$. Since the extracted isosurface encloses $\bar{\mathbf{x}}_i$'s, the maximal distance from the reconstructed surface to simulation particles on the surface is less than $0.58h_i$. This analysis shows that the volume shrinkage effect will not cause significant visual artifacts, since h_i is small compared to the size of simulation domain.