Reconstructing Surfaces of Particle-Based Fluids Using Anisotropic Kernel

Jihun Yu and Greg Turk

Presented by Po-Ram Kim

10 March 2011

Korea University
Computer Graphics Lab.
Abstract (1/1)

- Surface reconstruction method for particle-based fluid simulators
- 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)
1. Introduction (1/2)

- Fluid simulation
  - mesh-based methods
  - mesh-free methods
    - Smoothed Particle Hydrodynamics (SPH) is the most popular approach
      - simulating fluid is computationally simple and efficient
      - extracting high quality fluid surfaces is not straightforward

- Classical surface reconstruction methods have difficulties
  - due to irregularly placed 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
1. Introduction (2/2)

- Stretched, anisotropic smoothing kernel
  - each particle in the simulation
  - determined by capturing each particle’s neighborhood spatial distribution
    - Principle Component Analysis (PCA)

- Adjusted the centers of these kernels
  - variant of Laplacian smoothing
    - to counteract the irregular placement of particles
2. Related Work (1/3)

- Surface reconstruction method for particle-based fluid simulators
  - Particle-based fluid simulation for interactive applications.
    - Muller M. et al.
    - SCA 2003
  - Animating sand as a fluid.
    - Zhu Y. et al.
    - ACM SIGGRAPH 2005 Papers
  - Adaptively sampled particle fluids.
    - Adams B. et al.
    - ACM Trans. 2007

Particle position data → Compute density → Implicit function → Marching cube → Triangular mesh
2. Related Work (2/3)

• Particle-based fluid simulation for interactive applications.

• Animating sand as a fluid.
2. Related Work (3/3)

• Adaptively sampled particle fluids.
3. SPH Framework (1/4)

- Conservation of mass

\[
\frac{\text{d}\rho}{\text{d}t} + \rho \nabla \cdot \mathbf{v} = 0
\]

- Navier – Stokes equation

\[
\frac{\text{d}\mathbf{v}}{\text{d}t} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g}
\]

\( \mathbf{v} \): velocity vector, \( p \): pressure, \( \rho \): fluid density, \( \mathbf{g} \): gravitational acceleration vector,
\( \nu \): kinetic viscosity
3. SPH Framework (2/4)

- To evaluate the value $f$ at an arbitrary position $x$
  - an interpolation is applied with the neighboring particles
    : Particle approximation

$$f(x) = \sum_{j=1}^{N} f_j W(x - x_j) \frac{m_j}{\rho_j}$$

- $f_j$: the value of $f$ at the position of particle $j$ , $W$: smoothing kernel function
- $m$: mass , $\rho$: density
By applying the SPH particle approximation to the momentum equation

\[
\frac{d\mathbf{v}_i}{dt} = -\sum_{j=1}^{N} m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij} + \nu \sum_{j=1}^{N} (\mathbf{v}_j - \mathbf{v}_i) \nabla_i^2 W_{ij} + \mathbf{g}
\]

\( p_j \): the pressure of particle \( i \), \( \nabla_i \): direction gradient to particle \( i \)
3. SPH Framework (4/4)

- The density is interpolated by a sum of the weighted contributions
  \[ \rho_i = \sum_j m_j W(x_j - x_i, h_j) \]

- The pressure is described as a function of the density of the fluid
  such as given by the Tait equation
  \[ p_i = k \rho_0 \left( \left( \frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right) \]

- ref) Weakly Compressible SPH (WCSPH) framework [BT07]

- Our method can be used with any SPH framework
  - Particle-in-Cell (PIC)
  - Fluid-Implicit Particle (FLIP)
4. Surface Reconstruction

4.1. Surface Definition (1/6)

- The surface is defined as an isosurface of a scalar field
  - \( \varphi(x) \) gives a surface representation that coats the particles

\[
\varphi(x) = \sum_j^{m_j} \frac{\rho_j}{\varphi_j} W(x - x_j, h_j)
\]

- \( W \) is an isotropic smoothing kernel of the form

\[
W(r, h) = \frac{\sigma}{h^d} P\left(\frac{||r||}{h}\right)
\]

\( \sigma \) : scaling factor
\( d \) : dimension
\( r \) : radial vector
\( P \) : symmetric decaying spline

- However, the resulting surfaces often have bumps
  - there are two reasons for this
4. Surface Reconstruction

4.1. Surface Definition (2/6)

- 1. The irregular placement of particles
  - It is difficult to represent an absolutely flat surface
4. Surface Reconstruction

4.1. Surface Definition (3/6)

- 2. The spherical shape of the smoothing kernels
  - It is not suitable to describe the density distribution near a surface
4. Surface Reconstruction

4.1. Surface Definition (4/6)

- To resolve the problem (The irregular placement of particles)
  - We apply one step of diffusion smoothing to the location of the kernel centers
    - 3D variant of Laplacian smoothing
    - It has an effect of denoising point clouds

\[ \bar{x}_i = (1 - \lambda)x_i + \lambda \sum_{j} w_{ij}x_j / \sum_{j} w_{ij} \]

\[ 0 < \lambda < 1 \]

- Smoothing process is used only for surface reconstruction
  - The averaged positions are not carried back into the simulation

- Our approach also shrinks the fluid volume by moving the kernels
4. Surface Reconstruction

4.1. Surface Definition (5/6)

- To cope with the problem of density distributions near the surface (The spherical shape of the smoothing kernels)
  - We use the smoothing kernels to be anisotropic
  - By replacing $h$ with a $d \times d$ real positive definite matrix $G$

\[
W(r, G) = \sigma \|G\| P(\|Gr\|)
\]

$G$ : rotates and stretches the radial vector $r$

- The key idea of our new method is to associate an anisotropy matrix $G$
  - with each particle so that for particle $j$
4. Surface Reconstruction

4.1. Surface Definition (6/6)

- We extract an isosurface from a redefined scalar field

\[ \phi_{new}(x) = \sum_{j} \frac{m_j}{\rho_j} W(x - \bar{x}_j, G_j) \]

- For our examples, we use a B-cubic spline kernel
4. Surface Reconstruction

4.2. Determining the Anisotropy (1/8)

- Our new approach determines an anisotropy matrix $G$ for each particle
  - Inside the fluid volume
    - The smoothing kernel $W$ is isotropic.
  - around a particle that is near a flat surface
    - The particle density will decay faster along the normal axis than along the tangential axes.
    - $G$ 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
    - $G$ should shrink $W$ in order to capture the sharp feature
4. Surface Reconstruction

4.2. Determining the Anisotropy (2/8)

• In order to determine G
  ▪ we apply the **weighted version of Principal Component Analysis (WPCA)**

• PCA vs WPCA
  ▪ A drawback of the conventional PCA
    • when the number of samples is small and the sample positions are noisy
      = sensitivity to outliers
      = it often produces inaccurate information
  ▪ WPCA achieves significant robustness towards outliers and noisy data
4. Surface Reconstruction

4.2. Determining the Anisotropy (3/8)

- **WPCA**
  - begins by computing a weighted mean of the data points
  - constructs a weighted covariance matrix $C$ with a zero empirical mean
  - performs an eigendecomposition on $C$
    - The resulting eigenvectors give the principal axes
    - The eigenvalues indicates the variance of points along the corresponding eigenvectors

- We then construct an anisotropy matrix $G$ to match the smoothing kernel $W$ with the output of WPCA
4. Surface Reconstruction

4.2. Determining the Anisotropy (4/8)

- Covariance matrix $C_i$
  \[ C_i = \frac{\sum_j w_{ij} (x_j - x_i^w)(x_j - x_i^w)^T}{\sum_j w_{ij}} \]

- The weighted mean $x_i^w$
  \[ x_i^w = \frac{\sum_j w_{ij} x_j}{\sum_j w_{ij}} \]

- The function $w_{ij}$ is an isotropic weighting function
  \[ w_{ij} = \begin{cases} 
    1 - (\|x_i - x_j\|/r_i)^3 & \text{if } \|x_i - x_j\| < r_i \\
    0 & \text{otherwise}
  \end{cases} \]
4. Surface Reconstruction

4.2. Determining the Anisotropy (5/8)

- The singular value decomposition (SVD)

\[ C = R\Sigma R^T \]

\[ \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_d) \]

- \( R \) is a rotation matrix with principal axes as column vectors
- \( \Sigma \) is a diagonal matrix with eigenvalues \( \sigma_1 \geq \ldots \geq \sigma_d \)

- 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 \( k_r > 1 \)
  - we modify \( C \) such that the ratio between any two eigenvalues are within \( k_r \)
4. Surface Reconstruction

4.2. Determining the Anisotropy (6/8)

- When the number of particles in the neighborhood is small
  - We reset \( W \) to a spherical shape
  - by setting \( G = k_n I \)
    - in order to prevent poor particle deformations for nearly isolated particles

- We multiply \( C \) by scaling factor \( k_s \) for the associated particle inside fluid volume
  - \( \| k_s C \| \approx 1 \)
    - In order to keep the volume of \( W \) constant for particles with the full neighborhood
4. Surface Reconstruction

4.2. Determining the Anisotropy (7/8)

- Covariance matrix $\tilde{C}$.

\[
\tilde{C} = R\tilde{\Sigma}R^T
\]

\[
\tilde{\Sigma} = \begin{cases} 
  k_s \text{diag}(\sigma_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_d) & \text{if } N > N_\varepsilon, \\
  k_n I & \text{otherwise}
\end{cases}
\]

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

4.2. Determining the Anisotropy (8/8)

- In order to make the kernel $W$ of particle $i$
- Deform must be an inversion of $\mathbf{C}_i$
- Then our approach produces $G_i$ as a symmetric matrix of the form

$$G_i = \frac{1}{h_i} \mathbf{R} \tilde{\mathbf{\Sigma}}^{-1} \mathbf{R}^T$$
5. Implementation (1/1)

- For the neighborhood search, we use a variation of a hash grid

- At every reconstruction step
  - compute an axis aligned bounding box
  - store an index of the particle in the hash grid cells
  - examine each particle that is stored in the cell and tag as a neighbor

- Obstacles are represented as tetrahedral volume meshes for collision detection with particles

- We use the Marching Cubes algorithm
  - to create a mesh that represents the fluid surface
6. Results (1/3)

• Figure 1: water crown (24k particles)
6. Results (2/3)

- Figure 3: 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.
6. Results (3/3)

- Figure 4: double dam break
6. Results

6.1. Comparison and Limitations (1/3)

- The simulation is the double dam break simulation with 140K particles
  - 1. Particle-based fluid simulation for interactive applications.
    - The isotropic reconstruction method produces unacceptably bumpy surfaces
  - 2. Animating sand as a fluid.
    - Some surface bumps are still apparent
  - 3. Adaptively sampled particle fluids.
    - The method produces a still smoother surface
    - this method creates the highest quality surfaces from among the prior methods
  - 4. Ours
    - Our method produces surfaces that are even smoother than the method of Adams
    - 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.
Isotropic Kernel Method [MCG03]
Method of Zhu & Bridson [ZB05]
Method of Adams et al. [APKG07]
Our Anisotropic Kernel Method
6. Results

6.1. Comparison and Limitations (3/4)

- Table 1 shows timings for the double dam break example.

- The dimensions of the marching cubes grid: 230*190*350.

- Timings(ours)*2 = timings(Zhu).

- When we dropped this re-calculation to a lower rate:
  - the surface results from the Adams method became significantly lower in quality.

- The method of Adams et al. is the most time consuming of the four methods.

<table>
<thead>
<tr>
<th>Reconstruction method</th>
<th>Surface reconstruction</th>
<th>Simulation</th>
<th>Opaque rendering</th>
<th>Transparent rendering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isotropic</td>
<td>0.39</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhu and Bridson</td>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adams</td>
<td>1.76</td>
<td>2.19</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>Anisotropic</td>
<td>0.96</td>
<td></td>
<td></td>
<td>20.08</td>
</tr>
</tbody>
</table>

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

6.1. Comparison and Limitations (4/4)

- There are several limitations to our approach for surface reconstruction
  - The surfaces that are created using this method contain less volume than prior approaches
    - This is due to the averaging of particle centers

- Unlike mesh-based smoothing approaches
  - Our method does not shrink the surface near thin sheets of fluids

- 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
7. Conclusion and Future Work (1/1)

- 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
  - 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