Unified Spray, Foam and Bubbles for Particle-Based Fluids

Markus Ihmsen et al.
CGI 2012

Presented by MyungJin Choi
2013.06.06

Copyright of figures and other materials in the paper belongs original authors.

Computer Graphics @ Korea University
Introduction

• Basic Knowledge

• Previous Works

• Contribution
Introduction: Basic Knowledge

Three Types of Approaches for Fluid

• Eulerian
  ▪ Focus on Space(Grid)

• Lagrangian
  ▪ Focus on Particle

• Hybrid approach
  ▪ Eulerian + Lagrangian
Introduction: Three Types of Approaches for Fluid Eulerian Approach

- Eulerian approach
  - Focus on change of each spaces(grid)
- Grid-Based

- Simulating Water and Smoke with an Octree Data Structure
  Losasso, F et al. / SIGGRAPH 2004

- Fluid Animation with Dynamic Meshes
  Klingner, B et al. / SIGGRAPH 2006

- Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques
  Irving, G et al. / SIGGRAPH 2006

- Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid
  Chentanez, N et al. / SIGGRAPH 2011
Introduction: Three Types of Approaches for Fluid Lagrangian Approach

- Lagrangian approach
  - Focus on change of each particles
- Particle-Based

**Two-Scale Particle Simulation**
Solenthaler, B *et al.* / SIGGRAPH 2011
Introduction: Three Types of Approaches for Fluid

Hybrid Approach

- Hybrid approach
  - Eulerian(Grid) + Lagrangian(Particle)
    - Pressure value can be modified by Eulerian methods

Hybrid Smoothed Particle Hydrodynamics
Raveendran, K et al./ SIGGRAPH 2011
Introduction: Previous Works

Grid-Based

- Using two types of grid
- Not real-time simulation

- Using SPH and Particles Level-Set method
- Unwanted noise with FLIP

- Real-time simulation
- Not focus on detail feature

**Animation of Open Water Phenomena with Coupled Shallow Water and Free Surface Simulations**
Thurey, N. *et al.* / SIGGRAPH 2006

**Two-way Coupled SPH and Particle Level Set Fluid Simulation**
Losasso, F. *et al.* / TVCG 2008

**Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid**
Chentanez, N *et al.* / SIGGRAPH 2011
Introduction: Previous Works

Grid-Based

Realistic Animation of Fluid with Splash and Foam
Takahashi, T. et al. / EUROGRAPHICS 2003

- Numerical problems
- Restriction of time step
- Hypothetical model
Introduction: Previous Works

Particle-Based

- Numerical problems
- Restriction of time step
- Hypothetical model

A Layered Particle-Based Fluid Model for Real-Time Rendering of Water
Bagar, F. et al. / EGSR 2010
Introduction: Previous Works

Weak Points of Previous Works

• Grid-Based
  - Be restricted to their grid-based method
  - Can be lost more detail feature
  - Not considered other feature

• Particle-Based
  - Does not distinguish between foam and fluid particles
  - Does not generate additional particles
Introduction:

Contribution

• Author propose a technique for adding diffuse material to **Particle-Based** fluid simulations
  ▪ Not grid-based
    • Difference from other works
    • Can be used all particle methods
      ▪ E.g. : WC SPH, PIC SPH, etc
    ▪ Just relies on same type of material
      • Only water particle
Algorithm

• Diffuse Material
  ▪ Overview
  ▪ Formation of Diffuse Material

• Advection and Dissolution
  ▪ Overview
  ▪ Advection
  ▪ Dissolution
Algorithm: Diffuse Material

Overview

• How to classify diffuse material
  ▪ Potential

• How to determine the amount of diffuse material
  ▪ Energy

• How to determine property of diffuse particles
  ▪ Sampling
Algorithm: Diffuse Material

How to classify diffuse materials

- **Diffuse material** means **Foam, spray and bubble**
  - It develops when the surface tension of water molecules is reduced and water mixes with air
Algorithm: Diffuse Material

How to classify diffuse materials

• Remember below clamping function
  ▪ It returns the value between 0 and 1

\[
\Phi(I, \tau_{\text{min}}, \tau_{\text{max}}) = \frac{\min(I, \tau_{\text{max}}) - \min(I, \tau_{\text{min}})}{\tau_{\text{max}} - \tau_{\text{min}}}
\]

I : input data
\(\tau_{\text{max}}\) : user defined value
\(\tau_{\text{min}}\) : user defined value
• Trapped Air
  ▪ Air is trapped by impacts
    • E.g. : when the lip of a wave hits shallow water
      ▪ In this case air is dragged under water
  ▪ In this regions have high turbulences
    • To classify this regions the curl operator might be good choice
      ▪ Because of turbulence
    • But, author proposes to use relative velocities
      ▪ Because of unintended result of curl operator
Algorithm: Diffuse Material

How to classify Trapped Air

- The amount of trapped air is larger when fluid particles move towards each other
  - It can be measured by below function
    - It has the value between 0 and 2

\[
1 - \hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{x}}_{ij}
\]

\[
\hat{\mathbf{v}}_{ij} = \frac{\mathbf{v}_i - \mathbf{v}_j}{\| \mathbf{v}_i - \mathbf{v}_j \|}, \quad \hat{\mathbf{x}}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\| \mathbf{x}_i - \mathbf{x}_j \|}
\]
Algorithm: Diffuse Material

How to classify Trapped Air

Move towards from each other

\[ \hat{\mathbf{x}}_{ij} \quad \hat{\mathbf{v}}_{ij} \]

\[ \mathbf{v}_i \quad \mathbf{v}_j \]

\[ \theta = 2\pi \]

\[ \hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{x}}_{ij} = -1 \]

\[ 1 - \hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{x}}_{ij} = 2 \]

Move away from each other

\[ \hat{\mathbf{x}}_{ij} \quad \hat{\mathbf{v}}_{ij} \]

\[ \mathbf{v}_i \quad \mathbf{v}_j \]

\[ \theta = 0 \]

\[ \hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{x}}_{ij} = 1 \]

\[ 1 - \hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{x}}_{ij} = 0 \]
Algorithm: Diffuse Material

How to classify Trapped Air

• The scaled velocity difference

\[ v_i^{diff} = \sum_j |v_{ij}| (1 - \hat{v}_{ij} \cdot \hat{x}_{ij}) W(x_{ij}, h), \]

\[ W(x_{ij}, h) = \left\{ \begin{array}{ll}
1 - \frac{|x_{ij}|}{h}, & \text{if } |x_{ij}| \leq h \\
0, & \text{otherwise}
\end{array} \right\} \]

• The Potential of Trapped Air \( I_{ta} \)

\[ I_{ta} = \Phi(v_{i}^{diff}, \tau_{min}, \tau_{max}) \cdot W(x_{ij}, h) \]
• Wave Crest
  ▪ At the crest of a wave, whitewater is created
    • By strong wind or when the wave gets unstable

▪ In this regions have high curvature
  • Curvature is used to classify this regions
    ▪ The surface is locally convex
Algorithm: Diffuse Material

How to classify Wave Crest

- The surface curvature
  - It can be approximated with below function
    - It has the value between 0 and 2

\[
k_i = \sum_j k_{ij} = \sum_j (1 - \hat{n}_i \cdot \hat{n}_j) \cdot W(x_{ij}, h)
\]

\(\hat{n}\): normalized surface normal

- But it can’t distinguish convex from concave regions
  - The angles between \(\hat{n}_i\) and \(\hat{x}_{ji}\) (relative position)
Algorithm: Diffuse Material

How to classify Wave Crest

\[ \tilde{k}_{ij} = \begin{cases} 0, & \hat{x}_{ji} \cdot \hat{n}_i \geq 0 \\ k_{ij}, & \hat{x}_{ji} \cdot \hat{n}_i < 0 \end{cases} \]

- Wave Crest identifier \( \tilde{k}_i \)

\[ \tilde{k}_i = \sum_j \tilde{k}_{ij} \]
Algorithm: Diffuse Material

How to classify Wave Crest

- But, all convex regions doesn’t generate diffuse materials
  - Only the fluid particle moves in normal
    - Check this Using Additional function

\[
\delta_{vn} = \begin{cases} 
0, & \hat{v}_i \cdot \hat{n}_i < 0.6 \\
1, & \hat{v}_i \cdot \hat{n}_i \geq 0.6
\end{cases}
\]
Algorithm: Diffuse Material

How to classify Wave Crest

- We finally gets the Potential of Wave Crest $I_{wc}$

$$I_{wc} = \Phi(\tilde{k}_i \cdot \delta^v_n, \tau^{min}, \tau^{max})$$

$\tilde{k}_i$: value for checking convex
$\delta^v_n$: value for checking move direction
Algorithm: Diffuse Material

How to determine the amount of diffuse material

- When the surface tension is decrease, diffuse material is generated
  - Using Weber number
    - But its exact computation requires to correctly model the change in surface tension
  - Because of this reason, kinetic energy is used
    - The kinetic energy $I_k$

\[
I_k = \Phi(E_{k,i}, \tau^{min}, \tau^{max}),
\]
\[
E_{k,i} = 0.5m_i v_i^2
\]
Algorithm: Diffuse Material

How to determine the amount of diffuse material

• The number of diffuse particles $n_d$ is

\[ n_d = I_k (k_{ta} I_{ta} + k_{wc} I_{wc}) \Delta t \]

$I_k$: kinetic energy  
$k_{ta}$: maximum number of diffuse particles of trapped air  
$I_{ta}$: potential of trapped air  
$k_{wc}$: maximum number of diffuse particles of wave crest  
$I_{wc}$: potential of wave crest  
$\Delta t$: time step
Algorithm: Diffuse Material

How to determine property of diffuse particles

- Property has velocity and position
  - Generated diffuse particles have cylinder shape
Algorithm: Diffuse Material

How to determine property of diffuse particles

- Position of diffuse particle $x_d$ is
  
  $$x_d = x + r \cos \theta e'_1 + r \sin \theta e'_2 + h \hat{v}_f$$

  $$r = r_v \sqrt{X_r}, \quad \theta = X_\theta 2\pi, \quad h = X_h \cdot \|\Delta t v_f\|$$

  $X_r, X_\theta, X_h \in [0..1]$ and they are random variables

- Velocity of diffuse particle $v_d$ is
  
  $$v_d = r \cos \theta e'_1 + r \sin \theta e'_2 + \hat{v}_f$$
Algorithm: Advection and Dissolution

Overview

• Advection
  ▪ Advection is about how to determine move of diffuse material

• Dissolution
  ▪ It is about when diffuse particles are destroyed
Algorithm: Advection

Classify type of diffuse materials

• Diffuse material is influenced from water by their environment
  ▪ In air, small influenced
  ▪ On surface, medium influenced
  ▪ In water, highly influenced

• Three types of diffuse materials
  ▪ Spray, foam and bubble
    • It is classified by their position
Algorithm: Advection

Classify type of diffuse materials

- The type of diffuse materials can be determined by two methods
  - The gradient of the density field
  - The number of neighbors

- In author’s all experiments
  - Less than 6 neighbors are considered as spray
  - More than 20 neighbors are considered as bubbles
  - All other cases are considered as foam
Algorithm: Advection

The motion of diffuse materials

- Spray
  - Assume that the motion of spray is influenced by external force \( F_{ext} \) and gravity \( g \)
  - Using Euler-Cromer method, the velocity of a spray particle is updated as
    \[
    v_{\text{spray}}(t + \Delta t) = v_{\text{spray}}(t) + \Delta t \left( \frac{F_{ext}(t)}{m} + g \right)
    \]
  - The position is updated as
    \[
    x_{\text{spray}}(t + \Delta t) = x_{\text{spray}}(t) + \Delta t v_{\text{spray}}(t + \Delta t)
    \]
Algorithm: Advection

The motion of diffuse materials

- Foam
  - Foam is purely advected according to the averaged local fluid velocity

\[
\tilde{V}_f(X_d, t + \Delta t) = \frac{\sum_f V_f(t + \Delta t)K(X_d(t) - X_f(t), h)}{\sum_f K(X_d(t) - X_f(t), h)},
\]

\[
V_f(t + \Delta t) = \frac{X_f(t + \Delta t) - X_f(t)}{\Delta t}
\]

f: fluid particles
d: diffuse particle

- The position is updated as

\[
x_{foam}(t + \Delta t) = x_{foam}(t) + \Delta t\tilde{V}_f(X_d, t + \Delta t)
\]
Algorithm: Advection

The motion of diffuse materials

- Bubble
  - Bubble is advected according to the averaged local fluid velocity
  - Bubble is governed by buoyancy
    - Due to high density contrast of water and air
    - Buoyancy counteracts gravity

$$v_{bub}(t + \Delta t) = v_{bub}(t) + \Delta t(-k_b g + k_d \frac{\hat{V}_f(x_d, t + \Delta t) - v_{bub}(t)}{\Delta t})$$

- $k_b$: user defined value
- $k_d$: user defined value

- The position is updated as

$$x_{bub}(t + \Delta t) = x_{bub}(t) + \Delta t v_{bub}(t + \Delta t)$$
Algorithm: Dissolution

When diffuse particles are destroyed?

• Bubbles and spray doesn’t have life time
  ▪ But if they becomes foam, have life time
    • Life time is predetermined value
    • Subtract time step form life time

• Large clusters of foam are more stable
  ▪ Set the life time in relation to the generation potentials
Rendering

• The color of diffuse particles is determined by two algorithm
  ▪ Ray Casting
  ▪ Alpha Blending
Ray Casting

- Ray Casting
  - Eye ray is cast through the diffuse volume with direction $\omega$
    - Bounded by an axis aligned bounding box
      - With start point $x_s$ and end point $x_e$
      - With equally spaced intervals $\Delta x$
    - When ray reached to diffuse particle at point $x$, local volume density $\rho(x)$ is computed
      - Finally, compute the radiance $L$
Rendering:
Ray Casting

• The radiance $L$
  ▪ It is in the range between zero and one

$$L(x, \omega) = \prod_{i=0}^{s} e^{-\rho(X_s+i\Delta X)\tau\Delta X},$$

$$s = \left| \frac{x - x_s}{\Delta x} \right|, \tau \text{ is user defined scale factor}$$
Rendering:

Alpha Blending

• Final color $C_p$ is

$$C_p = (1 - L(x_e, \omega))C_d + L(x_e, \omega)C_b$$

$C_d$: color of diffuse material
$C_b$: color of background
• Author simulates 4 scenarios
  ▪ Wave, Tower, Ship, Lighthouse
  ▪ In all scenarios, use follow user defined values
    • $\tau_{wc}^{min} = 2, \tau_{wc}^{max} = 8$
    • $\tau_{ta}^{min} = 5, \tau_{ta}^{max} = 20$
    • $\tau_{k}^{min} = 5, \tau_{k}^{max} = 50$
Results

• Wave
Results

- Tower
Results

- Ship
Results

- Light house
Results

- Timings and particle counts for the presented show cases
  - Wave: 5s per frame
  - Tower: 92s per frame
  - Ship: 31s per frame
  - Light house: 27s per frame

<table>
<thead>
<tr>
<th></th>
<th># particles</th>
<th>computation times</th>
<th>frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fluid</td>
<td>diffuse</td>
<td>fluid</td>
</tr>
<tr>
<td>Wave</td>
<td>220k</td>
<td>2.7m</td>
<td>53 min</td>
</tr>
<tr>
<td>Tower</td>
<td>3.5m</td>
<td>7m</td>
<td>995 min</td>
</tr>
<tr>
<td>Ship</td>
<td>2m</td>
<td>2m</td>
<td>185 min</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>640k</td>
<td>3.3m</td>
<td>420 min</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>640k</td>
<td>15m</td>
<td>420 min</td>
</tr>
</tbody>
</table>
Conclusion

• Resulting in visually plausible, highly detailed flow patterns

• The realism of fluid simulations is significantly improved especially for low resolution fluids

• But this does not conserve mass