EUROGRAPHICS 2013 /
(Guest Editors)

Volume 32 (2013), Number 2

A Hybrid Method for Adaptive SPH Fluid Simulation

1158

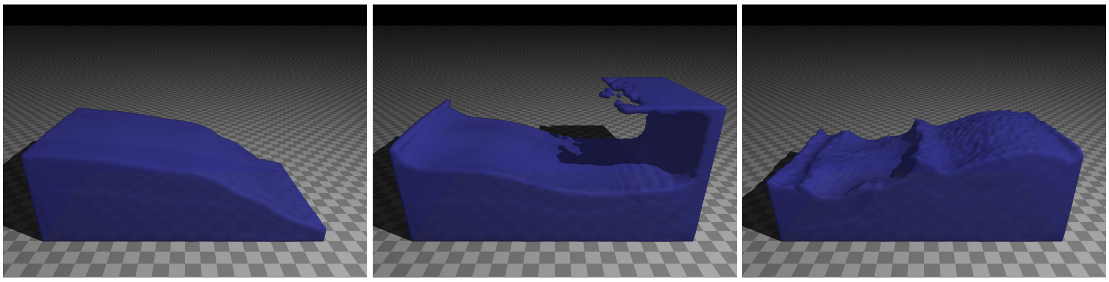

Figure 1: *Fluid simulation with our approach.*

Abstract

This paper proposes a new hybrid approach to adaptive SPH fluid simulation that better handles particle splitting and merging. Simulation of SPH fluid with adaptive, multiple-sized particles could enhance the animation performance while still retaining dynamic details with high-fidelity visual effects. However, conventional algorithms primarily refine simulation near fluid surface or in highly deformable region in spatial domain. We advocate a novel criterion that is based on the hybrid of both physics and geometry, and it is operated in both space domain and frequency domain. In particular, we use distance field as geometry criterion to localize region near fluid surface and we use wavelet-transformed energy in frequency domain as physics criterion to accurately detect inter-fluid turbulent region. Our adaptive SPH method based on space and frequency domain analysis is more accurate compared to the previous approaches. By coupling physics and geometry together, we could effectively detect regions that require particle splitting or merging. Moreover, our hybrid method is especially necessary to detect inner-fluid turbulent region and identify fine-detailed interface between different materials anywhere within fluid. We also propose a new surface tension model by integrating the air resistance into simulation, making the simulation more physically accurate. Through various animation scenarios, we demonstrate the efficacy and effectiveness of our new approach for high-fidelity fluid simulation.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Computer Graphics—Three-Dimensional Graphics and Realism

1. Introduction

Fluid simulation necessarily requires very high matter discretization and spatial resolution to preserve dynamic and surface details and to generate appealing visual effects for graphics applications. Towards these goals, a high resolution grid is required in Eulerian grid method, and a very large number of particles are demanded in Smoothed Particle Hydrodynamics (SPH) method. Therefore, it is much more desirable to only allocate computing resources to regions where complex flow behavior emerges. With the help of octree data structure, [LGF04] generates highly realistic water behavior, by adaptively solving the full three-dimensional Navier-Stokes equations in a grid-based system.

There are also some works addressing this problem in Lagrangian method, for example, the SPH method. By dynamically and adaptively splitting large particles into small particles and merging small particles into large ones, we could use large particles to simulate the gross shape of fluid but small particles to animate turbulent features (e.g., splash, vortex) and surface details (e.g., wave). We could sample the turbulent areas according to geometry [APKG07], physics [HHK08] or both [YWH*09].

In this paper, we propose a new criterion governing particle splitting and merging. Our method is both physically and geometrically based. It may be noted that, [HHK08] is also based on physics. In [HHK08], Hong et al. used the concept of Reynolds number as part of criterion. But this Reynolds number is only computed in spatial domain. Reynolds number is the value sampled at isolated points. It contains no information about its neighboring region. In our method, we use the wavelet-transformed energy as criterion. Wavelet analysis is an efficient and accurate mathematical tool to analyze data at different scales. In both [KTJG08] and [MGT*11], the wavelet-inspired technique is used as a tool to detect and compensate energy lost in order to preserve details. To the best of our knowledge, the work proposed in this paper is the first attempt to utilize wavelet analysis as a tool to help guide particle splitting and merging. Since this analysis is in frequency domain, we have a more global knowledge about the overall energy distribution.

Since the merging and splitting of particles at different level of scales may overburden any desktop animation system by introducing heavy computational cost, we simulate the fluid with particles in only two scales in this paper. Therefore, attributes including kernel length and particle mass are fixed values for particles in each layer. This way, we could precompute and cache them in the interest of time performance.

Our method has the following contributions:

- We propose a new adaptive SPH fluid simulation algorithm based on the hybrid of physics and geometry. When guiding SPH particle splitting and merging, we conduct rigorous analysis in both space and frequency domains. This is the first attempt to use wavelet analysis to guide particle splitting and merging in adaptive SPH fluid simulation. Our method is founded upon a solid foundation in both mathematics and physics.
- We also propose a new surface tension model by integrating the air resistance into simulation, making the simulation more physically accurate. We integrate the air into our simulation by applying the air resistance. Traditional SPH fluid simulations do not consider the interaction between fluid surface and air. In [SB12], air particles are periodically sampled near the fluid surface. But this sampling process is complex. By involving the air resistance, we could approximate the interaction between fluid and air while avoiding the complex particle sampling process.

2. Related Works

In this section, we shall briefly review the previous works relevant to the central theme of this paper. Since our method is a particle based method, we should primarily concentrate on the discussion of the particle based methods that have been developed previously.

Generally speaking, there are two physically based ways to simulate fluids, Eulerian method and Lagrangian method. They are all based on the Navier-Stokes equation. Eulerian method solves the Navier-Stokes equation over a grid tessellation. [FM96] is perhaps the first to apply the Navier-Stokes equation to 3D liquid simulation in computer graphics field for incompressible fluid. But this algorithm is not stable enough. Later on, [Sta99] designs an algorithm to guarantee the stability of fluid simulation by utilizing a semi-Lagrangian scheme to handle velocity advection. This algorithm is always stable for arbitrary time steps. Because of the computational cost of Eulerian method, it is hard to simulate fluid with a fine grid in real-time. [CM11] proposes a new Eulerian fluid simulation method in real-time by utilizing a hybrid grid representation consisting of regular cubic cells on top of a layer of tall cells.

Because of the numerical dissipation introduced in the Eulerian method, the particle based Lagrangian methods play a more and more important role in recent research and the most promising approach among them is arguably the SPH method. [MCG03] is the first to introduce the SPH method to the simulation of water. Later on, the SPH method is extended to the simulation of interaction between different fluids [MSKG05] and animation of viscoelastic fluid [CBP05]. There is also research about parallel implementation of SPH algorithm [HKK07].

Besides fluids, people also simulate smoke by the combination of particle system and grid system method, like FLIP [ZB05]. By generating some vortex particles in the grid and coupling the particles with grid based on vorticity confinement, [SRF05] can improve the turbulent effects for water, smoke, and explosion. [GLHB09] can simulate multiple speed smoke by simulating the low speed smoke in grid and high speed smoke in particle system. [CTG10] proposes an interactive system featuring fluid-driven animation that could respond to moving objects. It would generate a grid coupled with the moving object and simulate the smoke near the object in the grid system. For particles moving out of the surrounding grid, the smoke would be animated in a pure particle system.

Because the energy driven approach is always strictly physics based in computer graphics, energy based fluid and smoke simulation is also an active research topic. We could compensate the energy lost back to the simulation system in order to preserve the details. [KTJG08] analyzes the wavelet-transformed energy in the low resolution grid system in frequency domain and then synthesizes the wavelet turbulence energy to the high resolution grid to preserve turbulence features. Instead of the grid system, [MGT*11] applies the algorithm in [KTJG08] to the particle system to conserve the wavelet turbulence energy in SPH. In [KTJG08], we do not need to transfer the particle attribute to the auxiliary grid but directly work in the point cloud. Instead of isotropic noise, [PTC*10] adds the procedural approach based turbulence by generating and advecting turbulence particles based on anisotropic noise. Simulation based on anisotropic noise could make a more appealing visual effect than simulation based on isotropic noise alone.

In a particle system, the computing efficiency could be improved by constructing the particle system adaptively. It can make the distribution of computing resources more reasonable and reduce the complexity of calculation. For regions that do not need to preserve details, fewer particles would be enough; while for physically and visually important regions, a larger number of particles are required instead. [DG96] first applies the adaptive particle system to the simulation of highly deformable models. [APKG07] analyzes the particle system based on the extended local feature size to decide when to split or merge particles. This concept of extended local feature size is purely geometry based, but the turbulent regions computed with this method obey the rules in physics. Unlike [APKG07], which simulates compressible fluid, [HHK08] can simulate incompressible fluid by the application of the FLIP [ZB05] algorithm. This method computes the advection with SPH but analyzes the deformability of fluid and conserves incompressibility by transferring the particle attribute value to the auxiliary grid system. [YWH*09] samples the turbulent region based on both geometry and physics, and it introduces the non-uniform particle system and proposes a generalized distance field function. Then, the authors utilize the pressure to evaluate the turbulence of fluid based on physics. Thus, the auxiliary grid is not required. [SG11] presents a two-scale particle method based on the idea of simulating distinct particle sizes in individual yet tightly coupled simulations. Since traditional adaptive SPH method is not hardware based, [ZSP08] presents a new GPU-friendly algorithm for weakly compressible adaptive SPH.

Compared to other, above-mentioned topics, research works on adaptive particle fluid simulation are still in their infancy. Strongly motivated by this observation, we propose a hybrid adaptive SPH method in this paper. Our method is based on both physics and geometry. Unlike [HHK08], the analysis is not only in spatial domain but also in frequency domain. Figure 1 illustrates the examples that are generated with our approach.

3. SPH Model

In this section, we first introduce the basic knowledge of SPH fluid simulation. SPH is an interpolation method for particle system. In SPH, the fluid is composed of a set of particles with inter-particle forces such as pressure and viscosity computed at the position of a particle within a smoothing kernel. We define field quantities, such as velocity and density, at discreted particle locations anywhere in space. Instead of original attribute value, we use the smoothed attribute value to guide particle behavior. We can interpolate a scalar quantity A at location r by a weighted sum of contributions from all neighboring particles with equation (1):

$$A_S(r) = \sum_j m \frac{A_j}{\rho_j} W(r - r_j, h) \qquad (1),$$

where j is the index of all neighboring particles, m is the mass of particle, r is its position, ρ the density, A the field quantity at position r and h is the kernel length of the particle.

Function $W(r,h)$ is the radial kernel function which is used to compute the smoothed field quantity according to the contribution from neighbors. For more details about SPH method and its kernel functions, please refer to [MCG03].

Generally speaking, the governing force equation of SPH is as follows:

$$F_{Total} = F_{Pressure} + F_{Viscosity} + F_{External} \qquad (2).$$

The total force applied on each particle is the sum of forces from pressure, viscosity and other external forces like gravity and surface tension. The acceleration of each particle is from the total force with the following equation:

$$a_i = \frac{F_{Total}}{\rho_i} \qquad (3).$$

According to [MCG03], the equation to compute density is:

$$\rho_i(r) = \sum_j m_j W(r - r_j, h) \qquad (4).$$

Based on the constant gas equation, we could yield the pressure for each particle:

$$p = \kappa(\rho - \rho_0) \qquad (5),$$

where ρ_0 is the rest density of fluid. The equations for pressure force and viscosity force are:

$$F_i^{Pressure} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(r - r_j, h) \qquad (6),$$

$$F_i^{Viscosity} = \mu \sum_j m_j \frac{v_j - v_i}{\rho_j} \nabla^2 W(r - r_j, h) \qquad (7).$$

4. Adaptive Particle Fluid Simulation

In this section, we will describe our algorithm simulating fluid with adaptive two scale particles. Our algorithm is both physically and geometrically based.

4.1. Physically based Wavelet Energy

In [APKG07], the splitting and merging of particles are based on geometry. In [HHK08], this process is based on both physics and geometry. In previous adaptive SPH research, people primarily subdivide the region with high deformability. The highly deformable regions are always near the surface of fluid. This would not be accurate when the goal is to detect highly turbulent region not located just over fluid surface but at inner fluid, because the deformability is obtained at only one isolated point.

In such scenarios, details may be spreaded out everywhere. Therefore, we propose a new hybrid adaptive SPH method not only in space domain but also in frequency domain. By coupling particle energy and distance field together, we propose new criteria for particle splitting and merging. Our method is also a hybrid approach based on physics and geometry.

In our method, the first factor is physically based wavelet-transformed energy. In the field of physics, we use the following equation to compute the energy of particle:

$$E_i = \frac{1}{2}mv^2 \qquad (8).$$

However, this equation is only applicable in spatial domain. The energy is a value computed at only one isolated point. To detect the region where the turbulence will occur, we need to analyze the energy in both spatial and frequency domain. Fourier transform is one feasible solution to obtain information in both the spatial and frequency domains. [KTJG08] solves this problem by utilizing a wavelet transform to compute the wavelet-transformed energy for each grid cell. [MGT*11] analyzes the energy distribution in a particle system in both spatial and frequency domain with wavelet-transformed energy. In the particle system, a grid does not exist that can explicitly define the neighborhood value. They directly determine the wavelet transform $\hat{u}$ of the velocity field u and the energy spectrum $\hat{e}$ of e, at a scale s by taking a weighted sum of neighboring particles and using the SPH method. We could use the following equation in [MGT*11] to compute the transformed velocity in frequency domain:

$$\hat{u}_i = \frac{1}{\sqrt{s}\psi_{sum}} \sum_j u_j \psi\left(\frac{x_i - x_j}{s}, \frac{y_i - y_j}{s}, \frac{x_i - z_j}{s}\right) \qquad (9),$$

where $\hat{u}$ is the wavelet transform of velocity u, s is the wavelet scale, ψ is the mother wavelet function and

$$\psi_{sum} = \sum_j \psi \qquad (10).$$

We use Mexican hat wavelet as the mother wavelet. Figure 2 is the graph of Mexican Hat wavelet function. To our best knowledge, this approach represents the first attempt to apply this equation to guide adaptive particle fluid simulation. The reason we use wavelet is that it has very solid foundation in mathematics and physics. It is proved that wavelet analysis can provide us with accurate global knowledge of data in frequency domain.

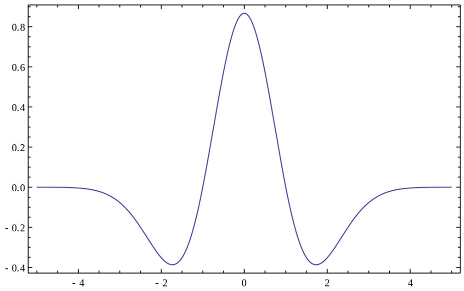

Figure 2: *Mexican Hat Wavelet.*

In [MGT*11], the computation of wavelet-transformed velocity is only for particles in one size. But in our work, we have particles in two scales. The equation to compute wavelet-transformed velocity in this paper is the same as that in [MGT*11]. We use different scale s for particles in different size, and s is usually the same as the kernel length h. After generating the transformed velocity, we compute the energy in frequency domain with:

$$\hat{e}_i = \frac{1}{2}m_i\hat{u}_i^2 \qquad (11).$$

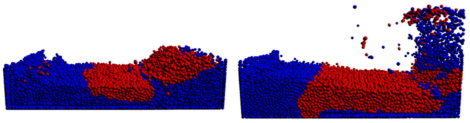

Figure 3: *Energy distribution in 3D.*

Figure 3 is the energy distribution generated with the equation (11). The red particle indicates the high energy level and blue one indicates the low energy level. The wavelet-transformed energy in frequency domain could indicate the turbulent region.

4.2. Geometrically based Distance Value

Geometry is another factor indicating where fluid details actually exist. The visual importance of region always increases while it moves approaching the fluid surface. Therefore, one naive way is that we could split particles near fluid surface and merge particles deep inside fluid. To find all surface particles, we first find the fluid surface. We define the fluid surface as an implicit function:

$$\phi(x) = |x - p| - h, (12)$$

where x is the sampled position, p is the nearest particle to position x and h is the kernel length of particle p. $\phi > 0$ represents the air region and $\phi < 0$ represents the fluid region. $\phi = 0$ is the fluid surface.

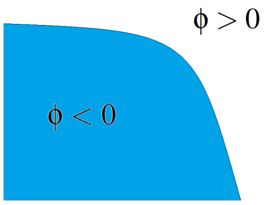

Figure 4: *Representation of surface with implicit function. $\phi < 0$ is the fluid region and $\phi > 0$ is the air region.*

We define all particles near fluid surface $\phi = 0$ as surface particles. Unlike [APKG07] which accurately generated the distance field for particles, we only approximate distance field in our algorithms. In [APKG07], only one particle's distance value would be computed for each iteration. In our method, we compute distance values for a number of particles in each iteration. For 20k particles, seven iterations are usually enough as in Algorithm 1. Figure 5 is the example demonstrating the approximate distance field of particles. Red particles are surface particles. The darker the blue particle is, the closer it is to the fluid surface.

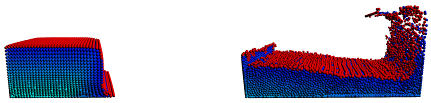

Figure 5: *Distance field in 3D.*

Algorithm 1 Distance Field Approximation.

```
1: Set the tag for each particle as 0.
2: Set the tag for each surface particle as 1. Set their distance value as 0.
3: For all particles whose tag is 1, propagate the distance value to their neighbors within kernel length $h$ (in one iteration).
4: If tags for all particles are 1, Stop. Otherwise, go to Step 3.
```

4.3. Two Scale SPH Equation

Since we are able to detect both high energy particles and particles close to fluid surface, we could establish our criterion to guide particle splitting and merging. Since our algorithm is both physically and geometrically based, we define two sets of thresholds for both physics and geometry. We denote that, *dist_merge* and *dist_split* are two geometry thresholds, and *energy_merge* and *energy_split* are two physical thresholds.

For a particle, if the energy $\hat{u} > energy\_split$ **or** distance field $d < dist\_split$, we split the large particle into small particles. If the energy $\hat{u} < energy\_merge$ **and** distance field $d > dist\_merge$, we merge small particles into one large particle. We couple physics and geometry together.

We use the same equation in [MCG03] to compute the density for the particle. However, we modify the constant gas equation as follows to guarantee the incompressibility of fluid:

$$\rho = \kappa((\frac{\rho}{\rho_0})^7 - 1) \qquad (13)$$

Since two particles in different scales have different kernel length and mass, unlike [APKG07] we design a new set of equations for particle pressure and viscosity computation:

$$F_i^{Pres} = -\sum_j \frac{m_j + m_i}{2}(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2})(\frac{\nabla W(x_{ij}, h_i) + \nabla W(x_{ij}, h_j)}{2})(14),$$

and

$$F_i^{Visc} = \mu \sum_j \frac{V_i + V_j}{2}(v_j - v_i)\frac{\nabla^2 W(x_i j, h_i) + \nabla^2 W(x_i j, h_j)}{2}(15),$$

with $V = \frac{m}{\rho}$ the particle volume.

4.4. Merging and Splitting

We use a similar method as in [HHK08] to merge and split particles. When merging a group of particles, the new particle is placed at the center of gravity of all original particles. We can compute the mass, kernel length, position and velocity of the newly created particle i from all particle members forming the new particle j by:

$$m_i = \sum_j m_j \qquad h_i = \sqrt[3]{\sum_j h_j^3}$$

$$x_i = \frac{\sum_j x_j V_j}{\sum_j V_j} \qquad u_i = \frac{\sum_j u_j V_j}{\sum_j V_j}$$

where h is the particle kernel length.

Unlike merging, the splitting of a large particle is performed by creating a new set of n particles. In our video demonstration, we set n as 3. With a larger n we could generate better visual effects. The mass, kernel length and velocity of each new particle are computed by the following equations:

$$m_j = \frac{m_i}{n} \qquad h_j = \sqrt[3]{\frac{h_i^3}{n}} \qquad u_j = u_i$$

There are multiple ways to determine the position of the new generated sub particles. We choose to place them randomly at the position within the radius of the original large particle. For more details on particle splitting and merging, please refer to [HHK08].

We summarize the algorithm in Algorithm 2.

Algorithm 2 Wavelet based Adaptive SPH.

```
1: Find the fluid surface according to the implicit function.
2: Generate the distance field for particle system.
3: Compute particle wavelet energy.
4: Compute density and pressure for each particle.
5: Compute the velocity of particle.
6: Split and merge the particle.
7: Reconcile the velocity of particle based on air resistance.
8: Advect the particle to the new position.
```

5. Surface Tension

We also design a new surface tension model. In previous work [MCG03], the surface tension is only based on the interaction between surface particles. The concept of color field [MCG03] has been proposed before. The surface tension is derived from both gradient and Laplacian of gradient field. Surface particles are always particles whose normalization of gradient of color field is larger than the predefined threshold value. In our work, we regard particles near implicit function as surface particles. But we use the same method to compute surface tension for surface particles. To make the simulation more physically accurate, we integrate the air resistance into simulation.

In [SB12], people apply ghost air particles to integrate the surrounding air into simulation. By generating a set of ghost air particles near the fluid surface, we could have air participating into simulation. However, [SB12] requires the sampling of air particles near the fluid surface, which is a complex process. To balance the simulation performance of physical accuracy, we substitute the ghost air particles with air resistance force. We consider this force from the pure geometry's perspective instead.

We first find the surface normal vector $\vec{d}$ from surface particle pointing outside to air region according the gradient of color field. Then we project the velocity of particle $\vec{v}$ onto the surface normal vector $\vec{d}$. The result length l is the result of dot product$\vec{d} \cdot \vec{v}$. If l is larger than 0, the particle is moving outward the fluid. Then we need to apply the air resistance force on the particle with the following equation: $AirForce = -(\vec{d} \cdot \vec{v}) \times c$, where c is a user defined coefficient to control the air resistance magnitude.

In Figure 6, the red particle is moving outward the fluid. Green arrow is the surface normal generated from the gradient of color field and the black arrow is the current velocity of particle. We project the black arrow onto the green arrow to compute the resistance magnitude from air. The algorithmic details are documented in Algorithm 3.

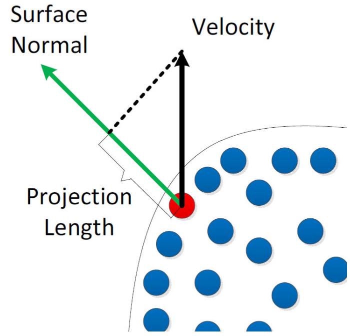

Figure 6: *Our surface model. Red particle is the surface particle moving outward. Green arrow is the surface normal and black arrow is the velocity of particle.*

Algorithm 3 Surface Tension

```
1: Apply surface tension as in [MCG03] to all surface particles.
2: Compute the gradient vector from surface particle to air
   (based on gradient of color field).
3: Project the velocity of particle onto its surface normal
   vector, and compute the length $l$.
4: $AirForce = -(\vec{d} \cdot \vec{v}) \times c$, $c$ is a user defined coefficient
   to control the air resistance magnitude.
```

6. Implementation and Experimental Results

We have implemented our SPH simulator with C++. All examples in this paper are implemented on a 64-bit machine with four 2.40 GHZ Core i7 CPU and 8GB memory. After extrapolating the surface triangular mesh with marching cube, we render the fluid with POV-Ray.

In Figure 7, we demonstrate with animation in our method. Pictures on the left side are simulation directly rendered as particles. Picture of the right side are corresponding fluid rendered with POV-Ray. Simulation begins with about 220k small size particles. Merging and splitting are dynamically happen during the simulation. Yellow particles are small size particles and red particles are large size particles. Each large particle is equivalent to three small particles. According to the demonstration, our method could detect the turbulent region and refine the simulation effectively. In Figure 7(c), fluid on top layer is moving toward left side. Because the left side is turbulent, we refine the simulation in region even deep inside fluid on left side. Table 1 is the set of parameters of simulation.

Table 2 is the experiment result. One purpose in adaptive

SPH is to improve the performance of simulation. Because our algorithm is not implemented on GPU, it would take more time for each frame compared to the traditional SPH algorithm. However, according to the experiment results in Table 2, the bottleneck of performance is surface tracking. 60% of simulation costs are for surface tracking, which is used to find all surface particles. We find all surface particles with the fast marching algorithm with an auxiliary grid of resolution $128 \times 64 \times 64$. The performance of fast marching algorithm on CPU is $O(N)$. Since this could be ported to GPU, the performance would be constance $O(1)$ time. And the bottleneck for performance could be avoided.

Simulation Parameters		
Parameter	Value	Unit
Large Kernel	0.04	m
Large Mass	0.02	kg
Gravity	3.2	m/s
Rest density	1000	kg/m^3
Gas Constant	1.0	N/A
Viscosity	10.5	$Pa \cdot s$
Time Step	0.001	s

Table 1. Simulation Parameters

7. Conclusion and Future Work

We have developed a novel technique for particle based fluid simulation that takes advantage of our new refinement and simplification mechanism. Such mechanism is founded up the hybrid of physics and geometry. In particular, we use both the wavelet-transformed energy and distance field to guide the particle splitting and merging. With our adaptive SPH algorithm, not only regions near fluid surface but also highly turbulent regions anywhere within fluid can be properly handled whenever and wherever particle splitting and merging are necessary. Consequently, high-fidelity SPH simulation supporting dramatic turbulence and small-scale details can be achieved effectively. The detection and refinement of microscopic material interface can also directly benefit from the utility of this algorithm, as abrupt material change can be more accurately analyzed in frequency domain. Through comprehensive experiments, we have demonstrated that our method could preserve high quality visual details. One limitation is that, the current animation system is not yet parallel, as the implementation is entirely in CPU (not GPU). We would like to upgrade our prototype system with CUDA which is our ongoing research efforts.

References

[APKG07] Adams B., Pauly M., Keiser R., Guibas L. J.: Adaptively sampled particle fluids. *ACM Trans. Graph. 26*, 3 (July 2007). 2, 3, 4, 5

[CBP05] Clavet S., Beaudoin P., Poulin P.: Particle-based viscoelastic fluid simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), pp. 219–228. 2

[CM11] Chentanez N., Müller M.: Real-time eulerian water simulation using a restricted tall cell grid. *ACM Trans. Graph. 30*, 4 (July 2011), 82:1–82:10. 2

[CTG10] Cohen J. M., Tariq S., Green S.: Interactive fluid-particle simulation using translating eulerian grids. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, ACM, pp. 15–22. 2

[DG96] Desbrun M., Gascuel M.-P.: Smoothed particles: a new paradigm for animating highly deformable bodies. In *Proceedings of the Eurographics workshop on Computer animation and simulation '96* (New York, NY, USA, 1996), Springer-Verlag New York, Inc., pp. 61–76. 3

[FM96] Foster N., Metaxas D.: Realistic animation of liquids. *Graph. Models Image Process. 58*, 5 (Sept. 1996), 471–483. 2

[GLHB09] Gao Y., Li C.-F., Hu S.-M., Barsky B. A.: Simulating gaseous fluids with low and high speeds. *Computer Graphics Forum 28*, 7 (2009), 1845–1852. 2

[HHK08] Hong W., House D. H., Keyser J.: Adaptive particles for incompressible fluid simulation. *Vis. Comput. 24*, 7 (July 2008), 535–543. 2, 3, 4, 5, 6

[HKK07] Harada T., Koshizuka S., Kawaguchi Y.: Smoothed particle hydrodynamics on gpus. *Structure* (2007), 1–8. 2

[KTJG08] Kim T., Thürey N., James D., Gross M.: Wavelet turbulence for fluid simulation. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 50:1–50:6. 2, 3, 4

[LGF04] Losasso F., Gibou F., Fedkiw R.: Simulating water and smoke with an octree data structure. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 457–462. 1

[MCG03] Müller M., Charypar D., Gross M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 154–159. 2, 3, 5, 6

[MGT*11] Makoto F., Go M., Toshiyuki A., Jun M., Hirokazu K.: A Fast Simulation Method Using SPH and Wavelet for Sub-Particle-Scale Turbulent Flow. pp. 67–72. 2, 3, 4

[MSKG05] Müller M., Solenthaler B., Keiser R., Gross M.: Particle-based fluid-fluid interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), SCA '05, ACM, pp. 237–244. 2

[PTC*10] Pfaff T., Thuerey N., Cohen J., Tariq S., Gross M.: Scalable fluid simulation using anisotropic turbulence particles. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 174:1–174:8. 3

[SB12] Schechter H., Bridson R.: Ghost sph for animating water. *ACM Trans. Graph. 31*, 4 (July 2012), 61:1–61:8. 2, 6

[SG11] Solenthaler B., Gross M.: Two-scale particle simulation. *ACM Trans. Graph. 30*, 4 (July 2011), 81:1–81:8. 3

[SRF05] Selle A., Rasmussen N., Fedkiw R.: A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. 24*, 3 (July 2005), 910–914. 2

[Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), SIGGRAPH 99, Addison-Wesley Publishing Co., pp. 121–128. 2

[YWH*09] YAN H., WANG Z., HE J., CHEN X., WANG C., PENG Q.: Real-time fluid simulation with adaptive sph. *Comput. Animat. Virtual Worlds 20* (June 2009), 417–426. 2, 3

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph. 24*, 3 (July 2005), 965–972. 2, 3

[ZSP08] ZHANG Y., SOLENTHALER B., PAJAROLA R.: Adaptive sampling and rendering of fluids on the gpu. In *Volume Graphics* (2008), Eurographics Association, pp. 137–146. 3

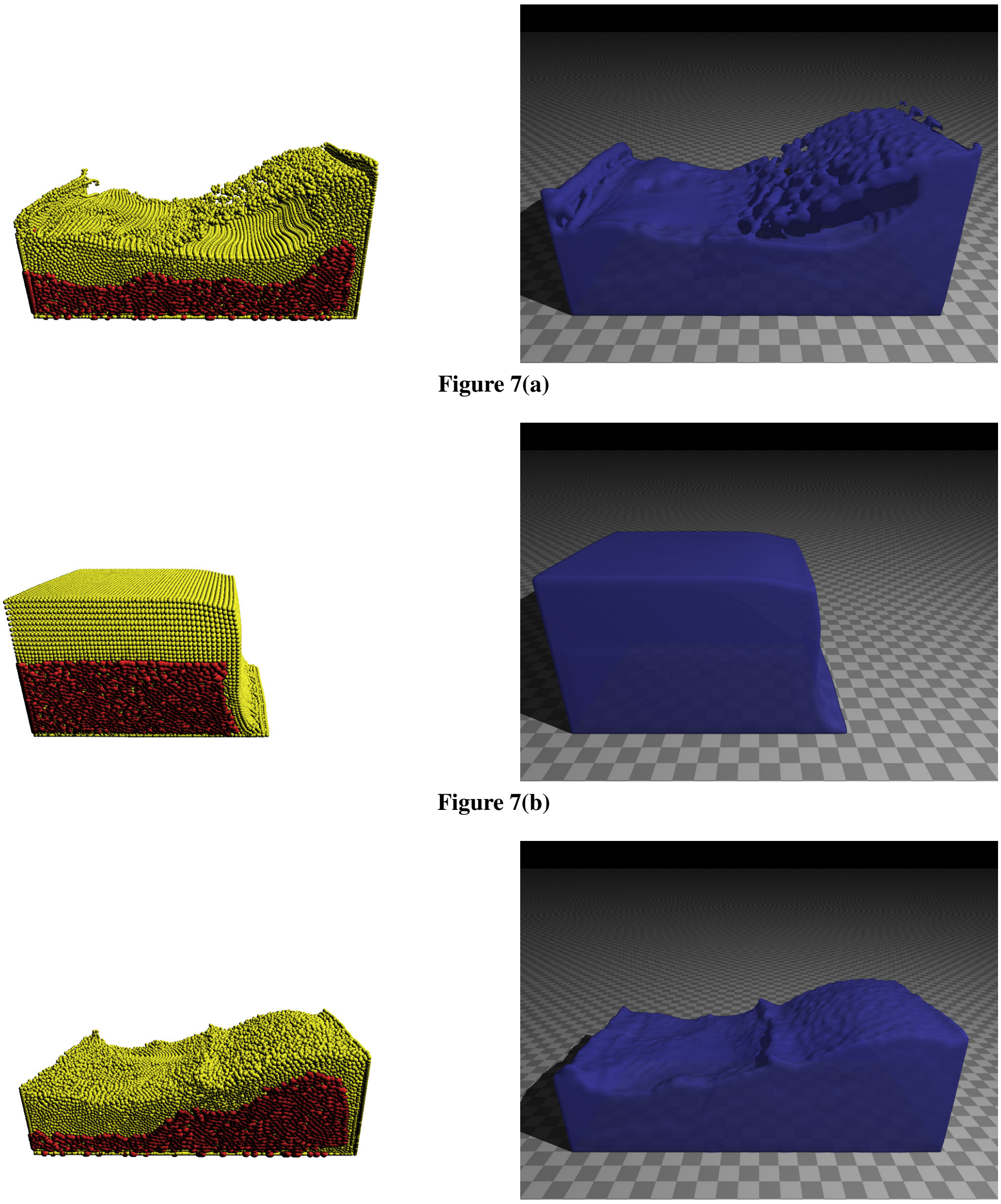

Figure 7(a)

Figure 7(b)

Figure 7(c)

Figure 7: *The example showing the hybrid approach with 220k particles.*

Particles	Normal SPH Time	Adaptive SPH Time	Surface Tracking	Split-Merge
16000	107(ms)	405(ms)	248(ms)	2(ms)
120k	1508(ms)	3123(ms)	1869(ms)	19(ms)
960k	12.5(s)	21.6(s)	12.9(s)	104(ms)

Table 2. Experiment Results