Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012), pp. 1–000
P. Kry and J. Lee (Editors)

Wavelet based Adaptive Particle Fluid

Dongli Zhang Hong Qin

Stony Brook University, USA

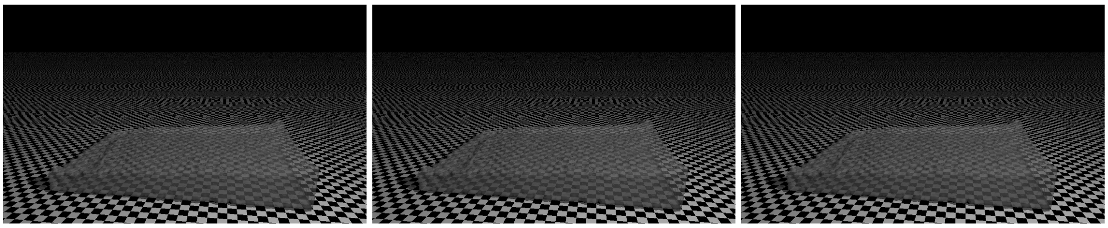

Figure 1: *New EG Logo*

Abstract

Simulation of SPH fluid with adaptive multiple size particles could improve the performance of simulation while preserving visual effects. One key point in adaptive SPH is the criteria of particle splitting and merging. In this paper, we propose a new criterion for adaptive sampled particle fluid. Our method is based on both the energy and surface particle. By the analysis of the energy of each particle in the wavelet frequency domain, we could decide the splitting and merging of the particle. Our algorithm could effectively sample regions where more details of flow are required and it is strictly physically based.

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

1. Introduction

Fluid simulation requires very high discretization resolution to preserve details and appealing visual effect. To improve the visual effect, a high resolution grid is required in Eulerian grid method, and a very large number of particles are demanded in Lagrangian Smoothed Particle Hydrodynamics (SPH) method. However, increasing the resolution may lead to heavy computational cost. Therefore, it is better to only allocate computing resources to regions where complex flow behavior emerges. With the help of octree data structure, [**?**] makes highly realistic water behavior, by adaptively solving the full three-dimensional Navier-Stokes equations in a grid.

There are also some works addressing this problem in Lagrangian method, especially, the SPH method. By dynamically and adaptively splitting large particles into small particles and merging small ones into large ones, we use large particles to simulate the general outline of fluid but small particles to animate turbulent features like splash and wave. We could sample the turbulent areas according to both geometry [**?**] and physics [**?**].

In this paper, we propose a new algorithm to simulate fluid with two scale particles. Since the merging and splitting of particles in different level of scales may introduce heavy computational cost, we simulate the fluid with particles only in two scales. Like [**?**], the refinement and simplification are performed for compressible flows based on SPH approach. But instead of geometry, out method is only physics based. We use wavelet to analyze the energy of each particle in both the frequency and spatial domain to sample turbulent regions. Our algorithm can also be used to simulate incompressible flow by modifying the constant gas equation.

2. Related Works

In this part, we will talk about the previous works. Since our method is a particle based method, we should primarily discuss the particle based methods that have been developed previously.

Generally, there are two physically based ways to simulate fluids, Eulerian method and Lagrangian method. They are all based on Navier-Stokes equation. Eulerian method solves the Navier-Stokes equation over a grid. [?] were the first to apply the Navier-Stokes equation to 3D liquid simulation in computer graphics community for incompressible fluid. But this algorithm is not stable. Later, [?] designed 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 grid in real-time. [?] proposed a new Eulerian fluid simulation method in real-time by utilizing a hybrid grid representation composed of regular cubic cells on top of a layer of tall cells.

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

Besides fluids, people are also simulating smoke by the combination of particle system and grid system method, like FLIP [?]. By generating some votex particles in the grid and couple the particles with grid based on vorticity confinement, [?] could improve the turbulent effects for water, smoke and explosion. [?] could simulate multiple speed smoke by simulating the low speed smoke in grid and high speed smoke in particle system. [?] proposed an interactive system featuring fluid-driven animation that responds to moving objects. It would generate a grid with the moving object and simulate the smoke near the object in the grid system. For particles out of the gird, the smoke would be animated in pure particle system, obscuring the point at which the fluid simulation ends.

Energy based fluid and smoke simulation is also a hot topic. We could compensate the lost energy back to the simulation system to preserve the details. [?] analyzes the low resolution grid system in frequency domain with wavelet analysis and then synthesizes the wavelet turbulence energy to the high resolution grid to preserve turbulence features. Instead of grid system, [?] applied the algorithm in [?] to the particle system to conserve the wavelet turbulence energy in SPH. In [?], we do not need to transfer the particle attribute to the auxiliary grid. Instead of isotropic noise, [?] added the procedural 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.

In particle system, the computing efficiency could be improved by constructing the particle system adaptively. It can make the distribution computing resources 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 region, a larger number of particles are required. [?] first applied the adaptive particle system to the simulation of highly deformable models. [?] analyzed 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 geometry based, but the turbulent regions computed with this method obey the rules in physics. Unlike [?], which simulates compressible fluid, [?] could simulate incompressible fluid by applying the FLIP [?] 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. [error] sampled the turbulent region based on both geometry and physics. They introduced the non-uniform particle system and proposed a generalized distance field function. They use the pressure to evaluate the turbulence of fluid based on physics. Thus, the auxiliary grid is not required. [?] presented a two-scale particle method based on the idea to simulate distinct particle sizes in individual but coupled simulations. Since traditional adaptive SPH method is not hardware based, [?] presented a new GPU-friendly algorithm for weakly compressible adaptive SPH.

Compared to other topic, research works on adaptive particle fluid simulation is not many. Therefore, we propose energy based adaptive SPH method in this paper. Our method is physically based. Unlike [?], it is not only in spatial domain but also frequency domain.

3. SPH Model

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 interparticle forces such as pressure and viscosity computed at the position of a particle by a smoothing kernel. We define field quantities, such as velocity and density, at discrete particle locations anywhere in space. We can interpolate a scalar quantity A at location r by a weighted sum of contributions from all neighboring particles with equation (1):

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

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

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

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

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

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

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

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

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

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

$$p = \kappa(\rho - \rho_0) \quad (4)$$

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

$$F_i^{Pressure} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(r - r_j, h) \quad (5)$$

$$F_i^{Viscosity} = \mu \sum_j m_j \frac{v_j - v_i}{\rho_j} \nabla^2 W(r - r_j, h) \quad (5)$$

4. Adaptive Particle Fluid Simulation

In this section, we will describe our algorithm to simulate fluid with adaptive size particles.

4.1. Wavelet Energy

In [?], the splitting and merging of particles are based on geometry and in [?] this process is based on physics, that is, the deformability of fluid. Our method is also based on physics, which is energy. In previous adaptive SPH paper, people primarily use their algorithm to simulate fluid in scenarios of lake, pool or canyon. They use small particles to preserve features like splashes and surface waves. These features always appear on the surface region of fluids.

However, when we want to simulate fluids flowing out of the water pipe or the hole on the dam and even the spring, the algorithm in previous paper may not work well. In these scenarios, details are everywhere. In these scenarios, the details always happen on high energy regions, such as the hole on the dam. By computing the boundary condition, we may adaptively split the particles near the object boundary to improve the visual effect. However, in different scenes, the objects and the boundary conditions are different, there are no fixed rules to guide particle splitting and merging. We should adapt our program to different boundary condition.

Therefore, we propose a new adaptive SPH method based on energy in our paper to simulate fluids flowing out of the hole on the dam or the water pipe. By the combination of particle energy and surface particle, we propose new criteria for particle splitting and merging.

In physics, we use the following equation to compute the energy of particle:

$$E_i = \frac{1}{2} m v^2 \quad (8)$$

However, this equation is only in spatial domain. To detect the region where the turbulence will occur, we analyze the energy in both spatial and frequency domain. Fourier transform is one solution to obtain information in both the spatial and frequency domains. [?] solved this problem by utilizing a wavelet transform and then calculating the energy of each grid cell. [?] analyzed the energy distribution in a particle system in both spatial and frequency domain with wavelet. In the particle system, a grid does not exist that can explicitly define the neighborhood value. They directly determine the wavelet transform of the velocity field u and the energy spectrum, 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 [?]

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

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

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

Unlike [?], we have particles in two scales. But the equation to compute wavelet transformed velocity is the same as in [?]. After generating the transformed velocity, we compute the energy in frequency domain:

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

Figure 2(a) is the energy distribution generated with the equation. The read particle is in high energy level and blue one is in low energy level.

Fluid surface is the place where details always occur. We also need to take the surface into consideration. We directly apply the method in [?] to our algorithm. In [error], people first compute the surface particle and then calculate the depth value for all particles with marching method. Since this is time consuming and we already could detect turbulent regions with energy analysis, we only split particles on the surface into small ones.

As in [?], the surface of fluid can be founded by using an additional field quantity which is 1 at particle location

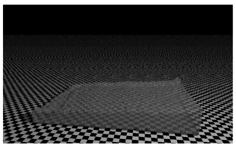

Figure 2: *Figure 2: Energy distribution in 2D*

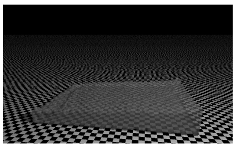

Figure 3: *Figure 3: Surface particles in 2D.*

and 0 the everywhere else. We call this the color field. The equation for the smoothed color field c is:

$$c_i = \sum_j \frac{m_j}{\rho_j} W(r_i - r_j, h) \quad (12)$$

The gradient of the smoothed color field is:

$$n = \nabla c_i \quad (13)$$

Its normalized scalar field is $|n|$. Given a threshold l, if $|n|$ is larger than the threshold l, the particle is on the fluid surface.

Since we now have both the energy and surface, we could establish our criterion to guide particle splitting and merging. By combining the energy and surface and equation together, we get the following equation:

$$D_i = \alpha \cdot \hat{e} + \beta \cdot |n_i| \quad (14)$$

α and β are coefficients. For each particle i, if D is larger than a threshold k, we may split the particle into small particles. If D is smaller than another threshold h, we may merge small particles into one large particle. We use different threshold to guide particle splitting and merging because we hope to avoid the unstable oscillation between merging and splitting.

4.2. Two Scale SPH Equation

Since we now have particles in two scales in our particle system, we could not use the original SPH interpolation equation in [**?**]. We extend the interpolation equation in [**?**] to our algorithm. We use the same equation in [**?**] to compute the density for the particle. However, we modified the constant gas equation as follows to calculate pressure:

$$\rho = \kappa((\frac{\rho}{\rho_0})_3 - 1) \quad (15)$$

Since two particles in different scale have different kernel length and mass, we extend equations in [**?**] as follows:

$$F_i^{Pres} = -\sum_j m_j m_i (\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2})(\frac{\nabla W(x_{ij}, h_i) + \nabla W(x_{ij}, h_j)}{2}) \quad (16)$$

and

$$F_i^{Visc} = \mu \sum_j V_i V_j (v_j - v_i) \frac{\nabla^2 W(x_i j, h_i) + \nabla^2 W(x_i j, h_j)}{2} \quad (17)$$

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

4.3. Merging and Splitting

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

$$m_i = \sum_j m_j \qquad r_i = \sqrt[3]{\sum_j r_j^3}$$

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

Unlike merging, the splitting of a large particle is performed by creating a new set of n particles. Usually, n is 2, 4 or 8. In our demo, we use $n = 2$. With a larger n we could generate better visual effect. The mass, kernel length and velocity of each new particle are computed by the following equations:

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

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 check reference [**?**].

We summarize the algorithm in Algorithm 1.

[htb] Wavelet based Adaptive SPH [1] Compute the density for each particle with equation (4). Compute the pressure for each particle based on constant gas equation (15). Compute the smoothed pressure for each particle based on equation (16). Compute the smoothed viscosity force for each particle based on equation (17). Compute the surface value and surface tension for each particle [**?**]. Compute velocity based on the total force applied to particle. Compute the energy for each particle based on wavelet analysis. Adaptively split and merge particles. Advect the particle to the new position.

5. Implementation and Result

We have implemented our SPH simulator with CUDA in GPU. All examples in this paper are implemented on a laptop with two 2.67 GHZ Core i5 CPU, 2.92 GB memory and NVIDIA NVS 3100m graphics card. Particles are rendered directly as points. We use Mexican hat wavelet as the mother wavelet.

The examples in Figure4, Figure 5 and Figure 6 are simulated by 8125 particles. For particles fulfill the criteria of splitting, we may split each particle into two small particles. We may also split each one into four or eight. In Figure 4, we simulate the fluid with the original SPH method [**?**]. In Figure 5, we simulate the fluid with our adaptive algorithm. And in Figure 6, we simulate the fluid with original SPH method directly in small size particles (16250).

6. Discussion and Future Work

We have presented a novel technique for particle based fluid simulation that uses refinement and simplification based on energy. By analyzing the energy in wavelet frequency domain, we could detect the turbulent region in the fluid. In the future, we would like apply our technique to multiple fluid interaction and smoke simulation.

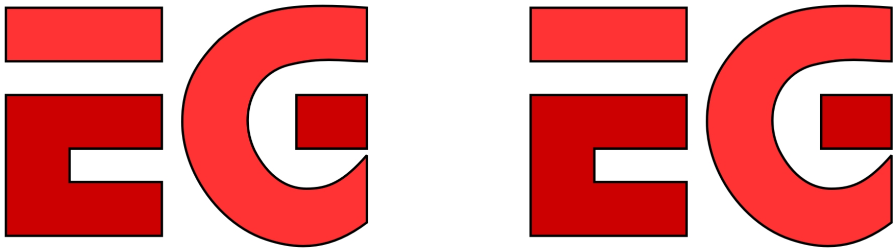

Figure 4: *For publications with color tables (i.e., publications not offering color throughout the paper) please* ***observe****: for the printed version – and ONLY for the printed version – color figures have to be placed in the last page.*
For the electronic version, which will be converted to PDF before making it available electronically, the color images should be embedded within the document. Optionally, other multimedia material may be attached to the electronic version.