Pacific Graphics 2010
P. Alliez, K. Bala, and K. Zhou
(Guest Editors)

Volume 29 (2010), Number 7

Creating and Preserving Vortical Details in SPH Fluid

Bo Zhu, Xubo Yang† and Ye Fan

Digital Art Lab, MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems,
Shanghai Jiao Tong University, China

Abstract

We present a new method to create and preserve the turbulent details generated around moving objects in SPH fluid. In our approach, a high-resolution overlapping grid is bounded to each object and translates with the object. The turbulence formation is modeled by resolving the local flow around objects using a hybrid SPH-FLIP method. Then these vortical details are carried on SPH particles flowing through the local region and preserved in the global field in a synthetic way. Our method provides a physically plausible way to model the turbulent details around both rigid and deformable objects in SPH fluid, and can efficiently produce animations of complex gaseous phenomena with rich visual details.

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

1. Introduction

Simulating the turbulent details around the moving objects immersed in a flow is a hot topic in computer graphics. Small-scale detail creation and preservation are two important issues in turbulence animation. To model the process of turbulence creation around objects, adaptive [KFCO06], [LGF04] and multi-grid [DMYN08] methods are widely used. By solving the Navier-Stokes equations with higher resolutions in a local space, the temporal generation and evolution of small-scale vortices can be modeled accurately. But when these vortices move out of the high-resolution region, they are difficult to be preserved in the coarse grid. Vortex particle method [SRF05, YKH*09, PTS*09] provides an efficient way to preserve the small-scale details in fluid. Some previous effects [PTS*09] have been made to seed vortex particles into the flow in a physically plausible way. But since the vortex particles are seeded into the flow in a discrete time point, it is difficult to model the temporal *creatio ex nihilo* process of the vortices formation as in adaptive and multi-grid methods.

To our knowledge, it is still a challenge to both model the realistic turbulence formation and to preserve the turbulent vortices in the global flow in one method. In this paper, we propose a new method to reach these two goals in SPH fluid. The small-scale vortices are formed in a local region by solving the high-resolution Navier-Stokes equations, and are smoothly transferred to the global flow field. The local turbulence creation and the global vortices preservation are bridged in a novel way: each SPH particle flowing through the local region is regarded as a vortex particle and

Figure 1: *An octopus expels ink in a SPH fluid. Small scale vortical details are captured around it and are preserved in the global flow field.*

† corresponding author email: yangxubo@sjtu.edu.cn

carries the vorticity. The vortical details around objects are gradually formed in a natural way instead of being seeded. And carrying vorticities on SPH particles can help to prevent the particle distortion problem in the previous vortex particle methods [YKH*09, RK08].

Our method can be regarded as a complementary for a standard SPH simulator to simulate turbulent gaseous phenomena. Most previous SPH methods in computer graphics are focused on capturing the free surface liquid details(e.g. [APKG07], [LTKF08]), while the work focusing on detailed gas animation is rare. In this paper, we provide a fast and physically plausible way to animate the visually interesting gaseous turbulence formation around moving objects in SPH fluid. We design a new SPH-FLIP solver to enforce non-slip solid boundary conditions and resolve the high-resolution incompressible flow around the object in a translating grid. Then the created small-scale vortices are preserved out of the local grid in a synthetic way. The main contributions of our method are:

- A new translating grid method to model small-scale turbulence formation around moving objects in SPH fluid and a hybrid SPH-FLIP algorithm to resolve the high-resolution details in a non-inertial reference frame.
- A fast and stable SPH vortex particle method to preserve the sub-scale vortical details in SPH fluid generated from the local regions.
- A unified method to create and preserve the vortical details around both rigid and soft objects.

2. Previous Work

Four kinds of previous work are closely related to our method: SPH, FLIP, vortex particle and translating grid. As a pure mesh-free method, smoothed particle hydrodynamics (SPH) has been widely used in computer graphics to simulate various kinds of liquid phenomena, and we refer readers to [KCR08, AW09] for details. Though SPH is fast and suitable for graphics applications, it has a hard time enforcing incompressibility. In [BT07], a weakly compressible SPH method (WCSPH) was proposed to solve this problem. Besides the purely mesh-free methods, structured or unstructured grids often help to solve the incompressible Navier-Stokes equations in a particle-based way. In [PTB*03] and [SBH09], Eulerian grid and Voronoi diagram were respectively used to make the computational domain divergence free. In [ZB05], FLIP-PIC was proposed to resolve incompressible fluids using a hybrid particle-mesh method. In [LTKF08] and [LHK09], the FLIP method was coupled with SPH to produce animations with rich liquid details. In their method, FLIP was used to compute the large-bulk water motion and SPH was used to capture the sub-scale splash details. In comparison, we hybrids the two methods in a contrary way: SPH is used to simulate large-scale fluid motions and FLIP to capture small-scale vortices. In [GLHB09], SPH was coupled with the standard Euler method to simulate the scenes with both high and low speed gases.

To preserve the vortical details in a flow, Vortex particle [SRF05, PTS*09] and procedure synthesis methods [NSCL08] are two prevailing categories. [YKH*09] combines these two categories to create high-resolution velocity fields from the vortices carried on vortex particles in a procedure synthesis way. Similar ideas can be seen in [YNBH09], in which the details are carried on particles advected in a flow to provide visually interesting results of rivers. However, advecting particles in a flow may induce the particles gathering together in an unnatural way. [RK08] proposed a remesh method to solve this problem. Properly seeding vortex particles is also important issue in turbulence simulation. In [PTS*09], a pre-computing model was proposed to simulate the waked turbulence around objects based on wall boundary layer theory. Their method provides a fast and physically plausible way to create vortical details in Euler fluid.

To model the small-scale turbulence formation around objects immersed in a flow, many adaptive methods have been proposed to solve the flow near the object accurately with multi-resolution grids [LGF04], tetrahedron meshes [KFCO06] or particles [APKG07]. Another kind of methods uses extra local grids with fixed high resolutions to track the visually important regions without generating new meshes. [DMYN08] uses overlapping grids with different resolutions to simulate the interactions between rigid objects and smoke. Small-scale vortices can be created within the local grids, but they are difficult to be preserved in the global coarse grid due to numerical diffusions. This method is physically accurate, but both the global and the local regions need to be resolved by using Euler method. In comparison, translating grid methods [SCP*04, REN*04, CTG10] provide more flexibility by sacrificing some accuracy. In [CTG10], an Euler grid solver and a particle system is coupled to provide interactive fluid simulations in a large unbounded space. This method is extremely fast, but it is difficult to simulate the smoke motion out of the local grid since no global flow is resolved.

3. Method

3.1. Method Overview

As illustrated in Fig. 2, the basic idea of our method is to capture the vortical details around a moving object with an attached overlapping grid and to preserve these details in the global flow field with SPH vortex particles. There are three key components in our method: a SPH solver to simulate global flow field, a hybrid SPH-FLIP solver to model the temporal turbulence formation and a SPH vortex particle method to bridge the global and the local regions. Each of them will be discussed in details in the following.

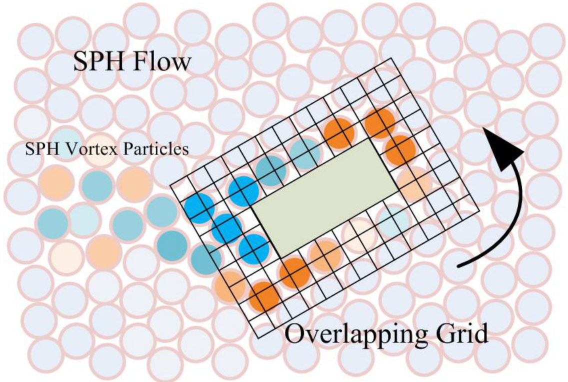

Figure 2: *Method overview. Vortical details are created in the overlapping grid and preserved in SPH fluid. Different color represents the direction and magnitude of vorticity on each SPH vortex particle.*

3.2. Modeling Global Flow using WCSPH

The global flow field is modeled by using a conventional weakly compressible SPH solver [BT07]. The fluid follows the Navier-Stokes equation which is given as:

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{f}. \tag{1}$$

The computation domain is represented by a set of discrete particles, and for each particle, the SPH approximation of the Navier-Stokes momentum equation is:

$$\frac{d\mathbf{u}_i}{dt} = -\sum_j m_j\left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2}\right)\nabla_i W(x_i - x_j, h) + \sum_j m_j \frac{(\mu_i + \mu_j)(\mathbf{v}_i - \mathbf{v}_j)}{\rho_i\rho_j(x_i - x_j)^2}(x_i - x_j)\nabla_i W(x_i - x_j, h) + \mathbf{f}. \tag{2}$$

The three terms on the right side of Eq. 2 are the discrete formulation of pressure force, viscous force and body force respectively. The pressure at each particle is obtained using the Tait equation:

$$p = \frac{\rho_0 c_0^2}{\gamma}\left(\left(\frac{\rho}{\rho_0}\right)^2 - 1\right), \tag{3}$$

in which $\gamma = 7$ and c_0 is the sound speed. We use the fourth-order weighting function mentioned in [TM05] as the kernel function for SPH integration approximation. Objects in the fluid are also represented by SPH particles. With the global SPH solver, the large-scale vortices can be simulated around the moving object easily as in Fig. 5.

3.3. Modeling Turbulence Formation in Local Grids

A hybrid SPH-FLIP solver is used to model the turbulence formation in local regions around objects. A high-resolution local grid is generated for each object and translates according to its motion. For each grid, FLIP is used to resolve the local Navier-Stokes equations in a non-inertial reference frame, with updated boundary and inner flow conditions transferred from the overlapping SPH particles. The Galiean Invariance of the local flow is ensured by adding inertial forces on the grid.

In each time step, computations on a local grid contain the following steps: 1. Physical data is transferred from SPH particles to the local region. 2. Inertial forces are added on the local grid. 3. FLIP solver compute the local velocity field with the updated boundary and inner flow conditions. 4. The local velocities are interpolated back onto the SPH flow field.

3.3.1. Physical Data Transfer from SPH to FLIP

Three types of physical data transfer are used to keep the consistency between the global and the local regions: 1. grid boundary interpolation; 2. SPH particle splitting; 3. solid boundary condition setting. First, as in conventional zonal methods [Fuj95, DMYN08], the outer most part of a local grid with fixed width is defined as the boundary region. As in Fig. 3, physical values are interpolated from SPH particles to the boundary grid points based on:

$$I_{p\_to\_grid} : \mathbf{u}_{grid} = \frac{1}{n_d}\sum_p \mathbf{u}_p W\left(\frac{1}{d}(\mathbf{x}_{grid} - \mathbf{x}_p)\right), \tag{4}$$

where n_d is the normalization value calculated as $\sum_p W(\frac{1}{d}(\mathbf{x}_{grid} - \mathbf{x}_p))$, and W is a trilinear interpolation kernel. Reference velocities of the grid attached to the object need to be considered in interpolations. The relationships between the global, the local and the reference velocities are:

$$\mathbf{u}_{global} = \mathbf{u}_{local} + \mathbf{u}_{ref}, \tag{5}$$

where $\mathbf{u}_{ref}$ is the velocity of the grid translating with the object, and $\mathbf{u}_{local}$ is the flow velocity on the local grid.

When SPH particles enter into the inner region of the local grid, a set of massless FLIP particles are seeded with the SPH velocities (as in Fig. 4). These FLIP particles are then added to the local solver to correct the local flow field according to the global SPH fluid. At last, the solid boundary conditions are updated in a gird-based way explicitly. For rigid objects, since the object and the local grid are always relative static, no solid boundary update is needed. For deformable objects, solid cells in the Euler grid need to be updated in each time step according to the relative positions of the object.

3.3.2. Galiean Invariance of Local Grids

The local grid attached to an object is an non-inertial reference frame with linear and angular accelerations. Inertial forces are added on the local grids to ensure the Galiean Invariance, as in [SCP*04, DMYN08]. The inertial force on each grid cell is calculated as:

$$\mathbf{f}_{inertial} = \mathbf{f}_a + \mathbf{f}_{col} + \mathbf{f}_{cen}. \tag{6}$$

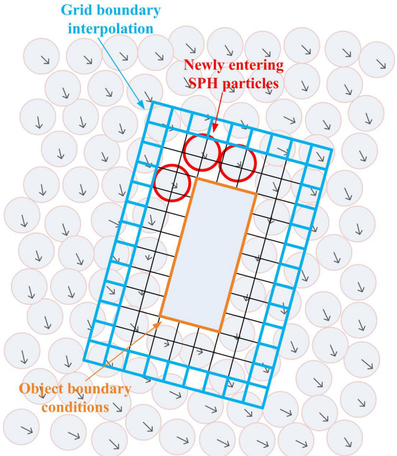

Figure 3: *Three types of data transfer from SPH particles to the local FLIP solver.*

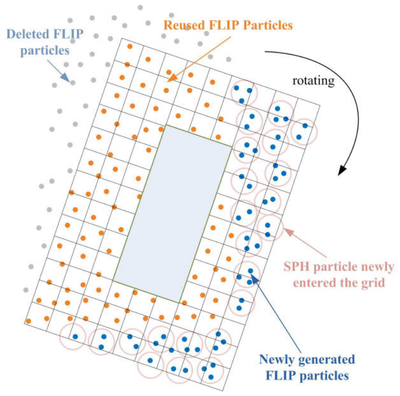

Figure 4: *Seeding FLIP particles from SPH particles in the inner region.*

The three force terms are acceleration force, Coriolis force and centrifugal force respectively:

$$\mathbf{f}_a = -\rho[\frac{d\mathbf{v}}{dt} + \frac{d\omega}{dt} \times (\mathbf{x}_i - \mathbf{x}_c)], \quad (7)$$

$$\mathbf{f}_{col} = -2\rho \cdot \omega \times \mathbf{u}, \quad (8)$$

$$\mathbf{f}_{cen} = -\rho \cdot \|\omega\|^2 \cdot (\mathbf{x}_i - \mathbf{x}_c). \quad (9)$$

As in the standard liquid FLIP solver [ZB05], a FLIP-PIC scheme is used to handle the viscosity term. A weighted average between FLIP and PIC on a particle is introduced to damp outlying particle velocities:

$$\mathbf{u} = (1-\alpha)\mathbf{u}_{flip} + \alpha \mathbf{u}_{pic}, \quad (10)$$

in which $\mathbf{u}_{flip}$ is the velocity absence of numerical dissipation and $\mathbf{u}_{pic}$ is an averaged velocity of the surrounding fluid.

Similarly, the SPH viscosity is calculated as:

$$\mathbf{u} = (1-\alpha)\mathbf{u}_p + \alpha \mathbf{u}_{average}, \quad (11)$$

where $\mathbf{u}_{average}$ is the averaged velocity of the neighbor particles. As in [AW09], it is calculated as:

$$\mathbf{u}_{average} = \sum_j \mathbf{u}_j \frac{m_j}{\rho_j} W(\mathbf{r}_{ij}, h). \quad (12)$$

By using Eq. 11 as the viscous term in SPH integrations and tuning α in Eq. 10 and 11, the fluid in the local and the global regions are modeled with consistent viscosity. A velocity blend method similar to [CTG10] is used to interpolate the high-resolution velocity field back to the upsampled global flow field. The overall SPH-FLIP algorithm can be seen in Algorithm 1.

Algorithm 1 Hybrid SPH-FLIP Solver

//Data transfer from SPH to FLIP
for each SPH particles newly enter a local grid **do**
  Create n massless FLIP particles
  $\mathbf{u}_{flip} \leftarrow \mathbf{u}_{sph}$
end for
for each grid point in inner region **do**
  $\mathbf{u}_{grid} \leftarrow I_{p\_to\_grid}(\mathbf{u}\, of\, neighbor\, FLIP\, particles)$
end for
for each grid point in boundary region **do**
  $\mathbf{u}_{grid} \leftarrow I_{p\_to\_grid}(\mathbf{u}\, of\, neighbor\, SPH\, particles)$
end for
Enforce solid boundary conditions on grid

//FLIP solver
Add inertial forces and other body forces on grid
Solve the Poisson equation on the grid
Interpolate $\mathbf{u}_{grid}$ to FLIP particles in a FLIP-PIC scheme

Advect FLIP particles in the grid
Delete FLIP particles out of the grid

//Data transfer from FLIP to SPH
Interpolate the local velocities back onto the SPH flow field

3.4. Preserving Vortical Details

To preserve the small-scale vortices generated in local grids, we carry the vorticity information onto the SPH particles passing through the local grid, and preserve them in the global region with a vortex synthesis method. Each SPH particle passing through the local grid is regarded as a vortex particle: in each time step, when the particle is in the local grid, its vorticity is updated according to the velocities on the grid; when the particle moves out of the grid, its vorticity evolves according to the vorticity form of the Navier-Stokes equations in the global flow.

For each SPH particle in the local grid, the vorticity can be computed as the curl of sampled velocity with the same resolution as the global SPH particle. When a SPH particle moves out of the local grid, the vorticity on it is evolved following the vorticity form of the Navier-Stokes equations:

$$\frac{d\omega}{dt} = (\omega \cdot \nabla)\mathbf{u} + \mu \nabla^2 \omega, \tag{13}$$

in which the first term represents vortex stretching and the second term accounts for vortex diffusion. To control the vortex diffusion in the global region efficiently, we use an attenuation coefficient instead of solving the diffusion term in Eq. 13. Similar to [SRF05, YKH*09], a high-resolution velocity field is computed by vorticity confinement force $\mathbf{F}_p(\mathbf{x}) = \varepsilon_p(\mathbf{N}_p \times \omega_p)$. It is worth notice that the velocity field generated from vortex particles is not inherently divergence-free. To make the field divergence-free, similar methods to [MWGZ09] can be used.

There are three velocity fields generated in each simulation step: $\mathbf{u}_{global}$ carried on SPH particles, $\mathbf{u}_{local}$ stored in local grids and $\mathbf{u}_{vor}$ stored in a fine global grid. To get the final flow field $\mathbf{u}_{final}$ for smoke advection, the three fields are synthesized in one grid with high resolution covering the entire fluid region. First, $\mathbf{u}_{global}$ is upsampled onto the global grid based on Eq. 4. Second, $\mathbf{u}_{local}$ is blended with the upsampled global grid velocities in local regions. Third, $\mathbf{u}_{vor}$ is added onto the grid to preserve the turbulence effects. The overall scheme can be seen in Algorithm 2

Algorithm 2 Overall Scheme

//Global SPH solver
Solve the global flow field using WCSPH

//Hybrid SPH-FLIP solver
for each translating grid in the SPH flow **do**
  Resolve the local flow by using Algorithm 1
  Interpolate the local field $\mathbf{u}_{grid}$ back to the upsampled global field $\mathbf{u}_{global}$
end for

//Vortex transfer from local grids to SPH flow
for each SPH particle inside a local grid **do**
  Mark it as SPH vortex particle
  Update its vorticity according to velocities on the grid
end for
for each SPH vortex particle outside the grids **do**
  Update its vorticity according to Eq. 13.
end for

//Velocity Synthesis
$\mathbf{u}_{final} \leftarrow \mathbf{u}_{global} + \mathbf{u}_{vor}$
Advect and render smoke particles

4. Results

We have implemented our method to create several test scenes. The physics simulation framework was implemented in C++ and the animations were rendered using Pixie. All simulations ran on a dual-core 2.93GHz CPU and 4GB of memory. In our demos, smoke is represented by millions of particles advected in the fluid velocity field. In 2D case, the density values on particles are interpolated to a high resolution density field for visual effects. In 3D demos, these particles are directly rendered as sprite points with transparency in Pixie. For each example, the SPH particle number, local grid size and the computation time per frame of different parts are given in Table 1.

First, we give an example of smoke in a tank mixed by a paddle with scripted motion in two dimensions, as in Fig. 5. 7k SPH particles were used for global flow simulation, and one translating grid (32×72, 4 times finer than global field) was used for turbulence generation. The results of pure SPH method and our hybrid method are compared: the global flow fields of the two methods are the same, while in our approach small-scale vortical details are gradually generated around the rotating paddle and are evolving in the global flow, as shown in Fig. 5.

In the left two images in Fig. 6, a stick mixes the smoke letters "PG" in three dimensions. The stick moves in an "∞" pattern and rotates with an angular velocity of $2\pi\, rad \cdot s^{-1}$ at the same time. We use the setup shown in the right two images in Fig. 6 to demonstrate the ability of our method to model turbulence formation around deformable objects. A soft worm swims in the SPH flow and produces interesting vortical details along its path. In Fig. 7, a rigid and a deformable propeller moves and mixes smoke in the same SPH flow. The visually interesting vortices waked around the rotating paddles flow out of the local grid and evolve in the global flow.

In Fig. 8, we test scenes including multiple overlapping grids in one SPH flow. Smoke rising from a source passes through five static sticks and produce interesting turbulent details around the obstacles. Five local grids are used in the simulation. No random vortex particle is generated in the source and all the vortical details are captured around the objects. The simulation results employing standard SPH are given in the top line of Fig. 8 for comparison. In Fig. 1, we build up a complex scene including an octopus expelling and mixing ink with its eight limbs. Small-scale vortical details are created around its gracile limbs swaying in fluid.

As in Table 1, we measure the particle number, grid size and the timing data of different parts of the five 3D examples. A fixed integration time step (0.4ms) is used in global SPH simulation for numerical stability. The global SPH timings given in the table are for one frame for the 50fps video, and in each frame there are 50 SPH simulation loops. Compared with the WCSPH simulation, extra time cost of turbulence

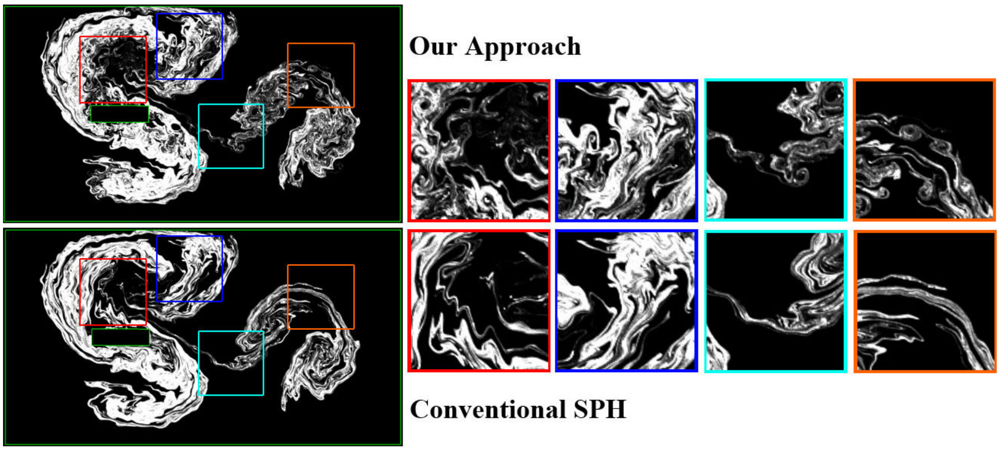

Figure 5: *A paddle mixes smoke in a 2D box. The simulation results of our approach (top) and conventional SPH (bottom)are given respectively.*

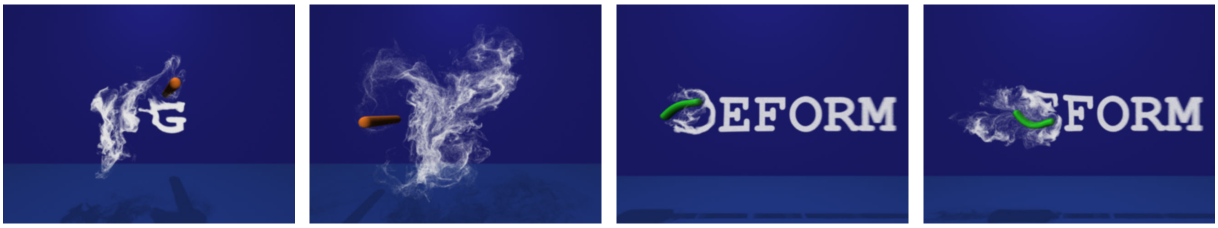

Figure 6: *Left two images: a rotating stick mixes smoke. Right two images: a deformable worm swims and perturbs the fluid.*

Figure 7: *Rigid (yellow) and soft (green) propellers mix smoke.*

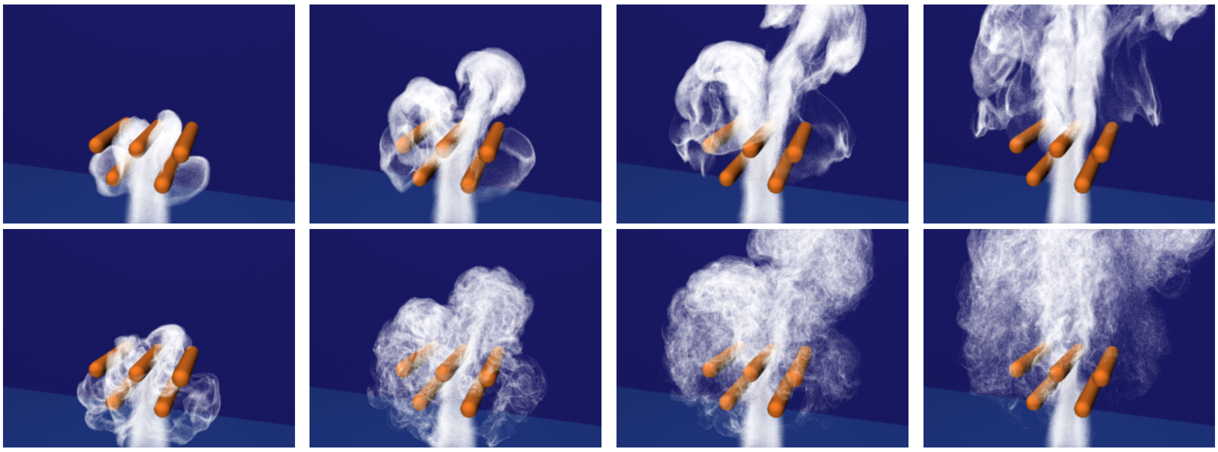

Figure 8: *Rising smoke passes obstacles. Standard SPH (top) and our approach (bottom) are compared.*

Table 1: *Particle numbers, grid size and timing data*

Scenes	SPH Particle Number	Global SPH Time(s)	Local Grid Size	Local Grid Number	Data Transfer(s)	Local Solver Time(s)	Vortex Time(s)	Total Extra Time(s)	Vortex/SPH Time Ratio
Stick	51k	47.1	$48\times48\times40$	1	0.10	1.4	0.02	1.5	3.1%
Worm	46k	45.3	$64\times64\times40$	1	0.12	2.9	0.02	3.0	6.7%
Propeller	126k	109.5	$40\times40\times40$	1	0.24	1.0	0.04	1.1	1.0%
Smoke	133k	147.5	$24\times24\times64$	5	1.1	2.4	0.04	3.5	2.4%
Octopus	194k	170.3	$96\times96\times96$	1	0.35	19.5	0.07	20	11%

modeling is within a small proportion (1%-10%). The timings of different parts including physical data transfer, local SPH-FLIP solver and vortex evolution are measured respectively. It is concluded that the most part (>90%) of the extra computation is spent on solving the local Navier-Stokes equations. Therefore our approach can work on an ordinary SPH simulator to produce animations with rich visual details, with only a small proportion of extra computation.

5. Discussion

The local SPH-FLIP solver is used to model the temporal turbulence formation and to shed small-scale vortices into the global flow. With the extra local solver, the new generated vortices around objects can be captured in a "vortex sensitive" way. It is easy to enforce non-slip solid boundary conditions rigorously on the local grid, without adding mirror particles as standard SPH methods [CL03, YRS08] did. On the other hand, since much larger time step (typically every 50 SPH simulation steps) can be used for visually turbulence generation in local space, and the global turbulence evolution is based on a synthesis method, it is much faster than directly resolving the local details in each simulation step by using adaptive SPH methods [APKG07].

SPH vortex particles are used in our approach to avoid the vortex distortion problem as mentioned in [RK08]. In standard vortex particle method [YKH*09], the passively advected vortex particles may gather together to produce strong and unnatural vorticity force. In SPH method, the pressure force on each particle will prevent it from overlapping with each other, and help to distribute the vorticity smoothly in the flow.

Similar to [DMYN08], we use extra overlapping grids to capture the flow details. Our approach can preserve these details out of the local grids, while in [DMYN08] these vortices smear out quickly in the global region. But our approach is only a physically plausible method for fluid animation, and is not as accurate as [DMYN08] in the CFD context. Compared to [PTS*09], in our approach the small-scale vortices are gradually generated instead of being seeded into the flow, so the turbulence formation can be modeled more naturally. But this process relies on solving extra Navier-Stokes equations and is less efficient than [PTS*09].

The main limitation of our approach is that the turbulence generation process is not rigorously physically accurate. Blending the FLIP field and the SPH field in local space might cause some physical inconsistency. The local field is resolved based on the global flow, but with higher resolutions and more rigorous non-slip solid conditions. We give two examples to show that the difference between the two fields is visually trivial in the boundary regions of the local grid. As in Fig. 9, the local flow represented by FLIP particles and the global flow represented by SPH particles are resolved in different solvers at the same time. In the boundary region, the new FLIP particles keep relative static with the SPH particles in the local space.

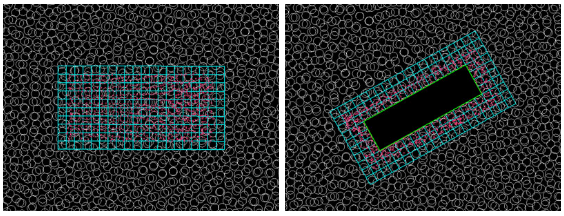

Figure 9: *The consistency of SPH particles and FLIP particles in different reference frames. Left picture is a static reference frame without obstacle, right picture is the rotating reference frame with a rigid object.*

Another limitation is that solving extra local Navier-Stokes equations might become computational expensive when the number of local grids increases. This will limit the ability of our approach in simulating scenes including a large number of moving objects, e.g. simulating a school of fish swimming in fluid environment. A method with higher efficiency to model the temporal turbulence generation in a local space, such as the reduced fluid model [WST09], can be considered.

As discussed above, developing a fast method instead of resolving local Navier-Stokes equations for turbulence modeling is our future work. By considering other features of the objects moving in fluid, e.g. the periodic actions of the swimming fish, it is interesting to introduce some pre-computed model in our framework.

6. Conclusion

We present a new method to create the votical details around the moving objects and preserve these details in SPH fluid. By binding a high-resolution overlapping grid around the object and solving the Navier-Stokes equations in the local region with a hybrid particle-mesh method, the temporal formation of the small-scale vortices are modeled realistically. Then these details are carried on SPH particles and preserved in the global field in a synthetic way. Our method provides an efficient way to create and preserve the sub-particle vortical details in SPH fluid, and can handle both rigid and deformable objects.

7. Acknowledge

This work was supported by the National Basic Research Program of China (Grant No. 2009CB320804) and the National Natural Science Foundation of China (Grant No.60970051).

References

[APKG07] ADAMS B., PAULY M., KEISER R., GUIBAS L. J.: Adaptively sampled particle fluids. In *ACM SIGGRAPH 2007 papers* (2007), p. 48. 2, 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. 2, 4

[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* (2007), pp. 209–217. 2, 3

[CL03] COLAGROSSI A., LANDRINI M.: Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *J. Comput. Phys. 191*, 2 (2003), 448–475. 7

[CTG10] COHEN J. M., TARIQ S., GREEN S.: Interactive fluid-particle simulation using translating eulerian grids. In *I3D '10: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), pp. 15–22. 2, 4

[DMYN08] DOBASHI Y., MATSUDA Y., YAMAMOTO T., NISHITA T.: A fast simulation method using overlapping grids for interactions between smoke and rigid objects. *Comput. Graph. Forum 27*, 2 (2008), 477–486. 1, 2, 3, 7

[Fuj95] FUJII K.: Unified zonal method based on the fortified solution algorithm. *J. Comput. Phys. 118*, 1 (1995), 92–108. 3

[GLHB09] GAO Y., LI C.-F., HU S.-M., BARSKY B. A.: Simulating gaseous fluids with low and high speeds. *Computer Graphics Forum 28* (2009), 1845–1852(8). 2

[KCR08] KOUMOUTSAKOS P., COTTET G.-H., ROSSINELLI D.: Siggraph core: Flow simulations using particles: bridging computer graphics and cfd. 2

[KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. *ACM Trans. Graph. 25*, 3 (2006), 820–825. 1, 2

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), pp. 457–462. 1, 2

[LHK09] LEE H.-Y., HONG J.-M., KIM C.-H.: Interchangeable sph and level set method in multiphase fluids. *Vis. Comput. 25*, 5-7 (2009), 713–718. 2

[LTKF08] LOSASSO F., TALTON J., KWATRA N., FEDKIW R.: Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 797–804. 2

[MWGZ09] MA C., WEI L.-Y., GUO B., ZHOU K.: Motion field texture synthesis. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers* (2009), pp. 1–8. 5

[NSCL08] NARAIN R., SEWALL J., CARLSON M., LIN M. C.: Fast animation of turbulence using energy transport and procedural synthesis. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers* (2008), pp. 1–8. 2

[PTB*03] PREMOZE S., TASDIZEN T., BIGLER J., LEFOHN A. E., WHITAKER R. T.: Particle-based simulation of fluids. *Comput. Graph. Forum 22*, 3 (2003), 401–410. 2

[PTS*09] PFAFF T., THÜREY N., SELLE A., , GROSS M.: Synthetic turbulence using artificial boundary layers. *ACM SIGGRAPH Asia 2009 Papers* (December 2009), 10. 1, 2, 7

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 193–202. 2

[RK08] ROSSINELLI D., KOUMOUTSAKOS P.: Vortex methods for incompressible flow simulations on the gpu. *The Visual Computer 24* (2008), 699–708. 2, 7

[SBH09] SIN F., BARGTEIL A. W., HODGINS J. K.: A point-based method for animating incompressible flow. In *SCA '09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), pp. 247–255. 2

[SCP*04] SHAH M., COHEN J. M., PATEL S., LEE P., PIGHIN F.: Extended galilean invariance for adaptive fluid simulation. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 213–221. 2, 3

[SRF05] SELLE A., RASMUSSEN N., FEDKIW R.: A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. 24*, 3 (2005), 910–914. 1, 2, 5

[TM05] TARTAKOVSKY A. M., MEAKIN P.: A smoothed particle hydrodynamics model for miscible flow in three-dimensional fractures and the two-dimensional rayleigh-taylor instability. *Journal of Computational Physics 207*, 2 (2005), 610 – 624. 3

[WST09] WICKE M., STANTON M., TREUILLE A.: Modular bases for fluid dynamics. *ACM Trans. Graph. 28*, 3 (2009), 1–8. 7

[YKH*09] YOON J.-C., KAM H. R., HONG J.-M., KANG S.-J., KIM C.-H.: Procedural synthesis using vortex particle method for fluid simulation. *Comput. Graph. Forum 28*, 7 (2009), 1853–1859. 1, 2, 5, 7

[YNBH09] YU Q., NEYRET F., BRUNETON E., HOLZSCHUCH N.: Scalable real-time animation of rivers. *Computer Graphics Forum (Proceedings of Eurographics 2009) 28*, 2 (mar 2009). 2

[YRS08] YILDIZ M., ROOK R. A., SULEMAN A.: Sph with the multiple boundary tangent method. *International Journal for Numerical Methods in Engineering* (2008). 7

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (2005), pp. 965–972. 2, 4