Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2008) *Posters and Demos*
R. McDonnell (Editor)

Fluid Animation with Multi-layer Grids

Jie Tan, Xubo Yang, Xin Zhao and Zhanxin Yang

Digital Art Laboratory, Shanghai Jiaotong University

Abstract

The most popular way to animate fluids is to solve the underlying physical equations. In this poster we introduce a general framework for physically-based fluid animation: the multi-layer Navier-Stokes solver. The Navier-Stokes equations are solved on a series of nesting grids in successive passes. Then the velocity and pressure fields of different layers are synchronized through interpolations. With the multi-layer framework, it is capable of combining the respective advantages of various grid types (discretizations), catching the multi-scale behavior of fluids and optimizing the computational resources.

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

1. Introduction

Real fluid phenomena often present multi-scale behavior, such as tiny splashes in the ocean and small vortices near the bank of a wide river. It requires sufficiently fine grids and long computational time in the simulation to get adequately resolved solution. We present a new method to address this issue by solving Navier-Stokes equations on multiple layers of grids with different resolutions or discretizations. This approach provides a flexible framework for fluid animation that opens a door for many new features and applications. We have implemented this multi-layer system and tested it in various cases. Although using multi-layer grids does not generally increase the numerical accuracy, it shortens the simulation time and achieves promising results in animating fluids in dynamic and complex scenes.

2. Navier-Stokes Equations on Multi-layer Grids

The basic idea behind the multi-layer approach is to start with a background layer of grid, usually with large grid spacing or low numerical accuracy (coarse MAC, height field), covering the entire domain. Then another layer of grid is superimposed that only covers the visually important and numerical difficult areas. If desired, more layers can be added where necessary to get adequately resolved solution. In general, we have a number of layers with different levels: $0 \le l \le l_{max}$ where the layer of level 0 denotes the background layer. Ω^l denotes the sub-domain covered by the layer of level l and $\partial\Omega^l$ denotes its boundary.

In our implementation, we identify the visually important areas based on following factors: the distance to the water surface, to the solid boundary, the magnitude of vorticity and the density of smoke. We calculate a score field for the whole domain by taking the weighted sum of each factor. A subdomain with a higher score means a higher level of grid is desired there.

When we integrate the convection term, semi-Lagrangian paths [Sta99] are traced back by one time step to find the velocities advected by the flowing fluid particles. For the layer of level l ($l \ge 1$), if a semi-Lagrangian path reaches a position outside Ω^l but still inside the fluid, the velocity is interpolated using the values stored in the layer of next lower level $(l-1)$. To solve the Poisson equation on the layer of level l inside Ω^l where $l \ge 1$, it is necessary to get the missing boundary conditions at the $\partial\Omega^l$. The missing boundary condition is acquired by interpolating the value at the boundary from the layer of next lower level $(l-1)$. And the interpolated values are fed to the layer of level l as Dirichlet boundary conditions to close the equations.

Finally, to make the velocity fields in all layers consistent and to revise the solution in layers of lower level, we interpolate the velocity field from a layer of higher level to its adjacent layer of lower level inside its covering region.

3. Results

Our simulator was run on an older 2.8GHz Pentium-D desktop with 2GB of memory. We start with the paddle-tank case (Figure 1) presented by [KFCO06]. A regular grid with resolution $32 \times 16 \times 16$ works as the background layer. The tetrahedral layer with approximately 7000 tetrahedra surrounding the paddle is created prior to the simulation. Thanks to the easy boundary conformance of our method, no remeshing is needed during the simulation. It takes about 2 seconds per time step on average on our unoptimized simulator. Although the resolution of the grid is quite low, 15000 variables in total ($\frac{1}{6}$ of the number in [KFCO06]), are used in solving the Navier-Stokes equations, the animation still catches the interesting vortices and achieves a comparable visual effect.

Figure 2 shows a series of comparisons in simulating the breaking dam scenario with single- and multi-layer grids. The left-upper image is animated using $24 \times 24 \times 16$ single-layer grids. The resultant animation looks sticky because of low sampling rates and high numerical dissipations. The right-upper and left-lower images are animated with a $24 \times 24 \times 16 - 48 \times 48 \times 32$ coarse-fine bi-layer grid and $48 \times 48 \times 32$ single-layer grid respectively. It is difficult to distinguish these two animation sequences by quality but the simulation time of the bi-layer grid method is only about one quarter of its single-layer counterpart. The last image is animated with a $48 \times 48 \times 32 - 96 \times 96 \times 64$ grid which shows turbulent flow and detailed water surface. Adding thin layers of refined resolution near the water surface improves the visual quality dramatically at far less cost than applying a fully refined single-layer grid, especially when tackling large bodies of water.

More test cases of the multi-layer framework are shown in Figure 3.

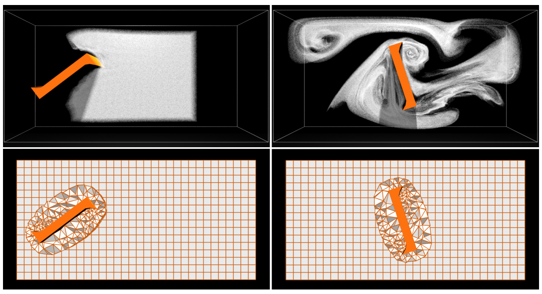

Figure 1: *Top: A paddle mixes smoke in a tank. Bottom: A cross-section of the regular-tetrahedral bi-layer grid used.*

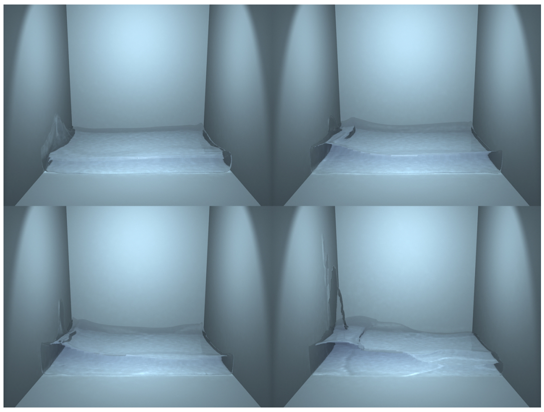

Figure 2: *A comparison of fluid animation with single- and multi-layer grids.*

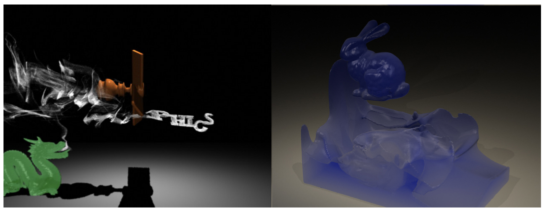

Figure 3: *More test cases for multi-layer framework.*

4. Conclusion

We have introduced a multi-layer framework for fluid animation, which enables easy coupling of two or more layers of grids with different resolutions and types. The multi-layer approach makes it possible to combine the advantages of different types of grids, to capture the multi-scale behavior and to optimize the computational resources. Although we only take regular-tetrahedral and coarse-fine grids as examples in this poster, the variety of combinations is not restricted to these. In addition to fluid animation, a number of physically-based methods use grids to solve the governing partial differential equations. Due to the simple concept and easy implementation of our multi-layer framework, we believe that this approach is straightforward enough to be applied in a broader class of physically-based animations.

5. Acknowledgement

This work is supported by 863 National High Technology R&D Program of China (No. 2006AA01Z307).

References

[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.

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