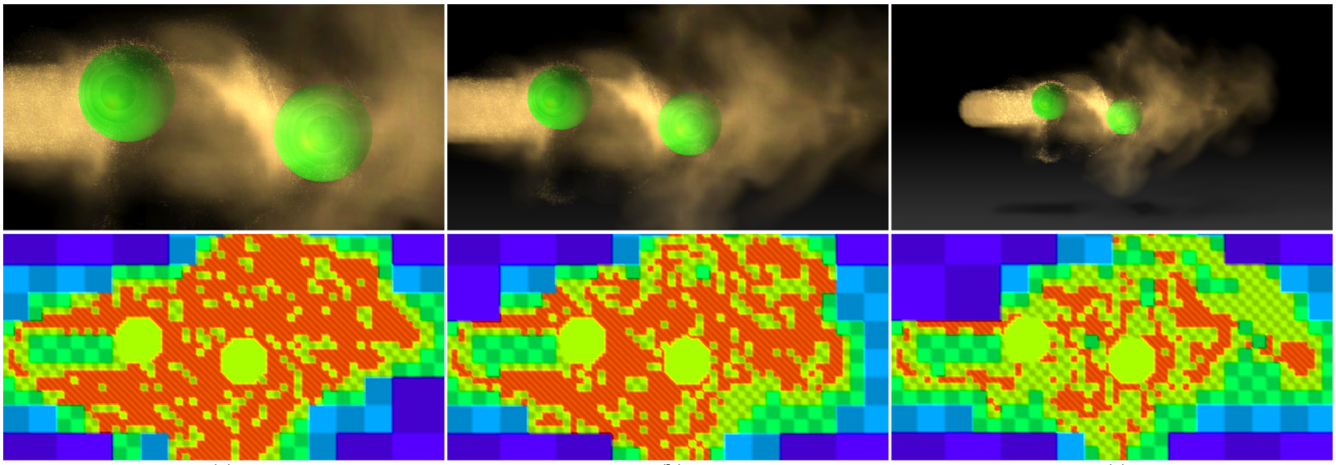
Enhanced View-Dependent Adaptive Grid Refinement for Animating Fluids

Rinchai Bunlutangtum*
Chulalongkorn University

Pizzanu Kanongchaiyos†
Chulalongkorn University



(a) (b) (c)

Figure 1: *Top: A flow of smoke through the oscillating spheres. Bottom: A cross-section of the simulation grid. Refinement is performed each frame based on viewing information. From left to right, the camera viewing angle (FOV) is* 20^o, 30^o *and* 50^o *respectively.*

Abstract

This paper presents a method for animating fluids on an octree grid. Our approach combines dynamic grid refinements with viewing information i.e., viewing angle, camera-to-fluid distance and resolution ratio of simulation grid over output image in order to optimize the grid for speed as well as preserve the animation quality. Our method reduces and removes fine details that are unnoticeable by means of level-of-detail. We determine whether to merge or divide the grid by using our proposed refinement conditions. In addition, particles, which are more flexible to conform to obstacle-fluid boundaries, are integrated in to our model to enhance animations and reduce the artefacts caused by dynamic refinements. Our method is fast and easily implemented, yet capable to be integrated as an extension of a current grid-based solver.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: Physically-based Animation, Natural Phenomena, Fluid Dynamics, Level of Detail, Adaptive Refinement

*e-mail: rinchai.b@chula.ac.th
†e-mail:pizzanu@cp.eng.chula.ac.th

VRCAI 2011, Hong Kong, China, December 11 – 12, 2011.
© 2011 ACM 978-1-4503-1060-4/11/0012 $10.00

1 Introduction

Computational cost is one of the topics of most concern in fluid animation. Although fluid animations are performed faster with current hardware, they are not fast enough as people always expect animations with higher details. We have realised that a large simulation domain contains lots of distant fluids which usually have less visual attention. Therefore, details can be reduced by distance for faster simulation.

In this paper, we propose an enhanced method for animating fluid on an octree grid. A view-dependent adaptive grid refinement constructed by incorporating viewing information with refinement conditions is integrated as a core of our model. We further adjust the grid resolution to properly suit the output resolution with a resolution ratio. In addition, a particle system is coupled to produce smooth animations and reduce artefacts.

Overall, the simulation gains speed as well as preserves details based on viewing information. The method can be applied for variety of simulation environments as we have implemented and tested our method in various scenarios such as one shown in Figure 1.

2 Background

Grid-based simulation was introduced by [Harlow and Welch 1965]. It is a common approach for animating smoke [Stam 1999; Fedkiw et al. 2001; Stam 2003], liquid [Foster 1996], fire [Nealen 2001], explosion [Feldman et al. 2003], viscoelastic materials [Goktekin et al. 2004] and bubbles [Zheng et al. 2009]. A fixed uniform grid is simple and straightforward to implement compared to others. It works well for coarse grids but for higher grid resolutions, it consumes high computational cost due to its uniformity.

[Losasso et al. 2004; Shi and Yu 2004] introduced octree structures for animating fluid on adaptive grid. Cells are subdivided when detail is needed and are merged when detail are no longer necessary. With this approach, the grid is optimized for detail. In addition,

simulation using octree grid can be performed faster with parallel computing such as in [Ament and Straß er 2009].

[Barran 2006] proposed a view-dependent grid constructed on a transformed polar coordinate that is more suitable for viewing angle. However, the grid is not optimized for details since refinements may breaks the cell's symmetry and causes simulation errors.

[Bunlutangtum and Kanongchaiyos 2011] incorporated viewing information with refinements on octree grid. The grid is optimized for both viewing and details. However, the simulation still contains artefacts caused by dynamic grid refinement.

An adaptive grid using tetrahedral meshes such as [Chentanez et al. 2007] is another refinement technique. Tetrahedral meshes easily conform to curved and complex boundaries. Their size can be graded to optimize the simulation but they need a specific scheme to deal with free surfaces, moving obstacles and artificial damping.

3 Methods

The simulation domain is constructed on octree structure as described in [Losasso et al. 2004]. The number of cells in a domain is proportional to the computational cost. The key contribution of this paper is to reduce the number of cells as much as possible while visual results are still preserved.

At each time step, the grid is optimized based on grid refinement methods described in [Shi and Yu 2004] and [Bunlutangtum and Kanongchaiyos 2011]. We have made a few modifications to the refinement methods to reduce the number of cells throughout domain by level-of-detail.

3.1 Grid Refinements

Refinement conditions are used to decide for each cell whether to subdivide or merge the grid. In general, refinement conditions are defined by comparing a constant threshold with a fluid variation such as [Shi and Yu 2004]. We have modified these conditions with additional parameters for addressing viewing angle; thus, not only is fluid detail optimized but also the viewing angle as well.

3.1.1 Measuring Fluid Variation

Fluid variation measures the difference of fluid values among adjacent cells, which roughly indicates the amount of detail to be preserved. High fluid variation implies sharp edges or boundaries where details are needed, while low fluid variation is vice versa.

$$C(x,y,z) = max(|\nabla_x^2 u|, |\nabla_y^2 v|, |\nabla_z^2 w|) \tag{1}$$

Equation 1 measures the variation of velocity, where u, v, w denote scalar components of velocity vector: $\mathbf{u} = (u, v, w)$. ∇_x^2 is a Laplacian on x direction: $\nabla_x^2 u = \partial^2 u / \partial x^2$. ∇_y^2 and ∇_z^2 are defined likewise.

3.1.2 View-Dependent Weighting Factor

Once the fluid variation is obtained, we then refine the grid by comparing the fluid variation with two thresholds, one for subdivision and another one for merging. However, not only is fluid variation considered for the refinement, but also the viewing angle is needed to perform a view-dependent grid refinement. Thus, we factor these thresholds with a *view-dependent weighting factor* as stated below.

$$\Psi = \alpha \left(\frac{r}{R} \right) \tag{2}$$

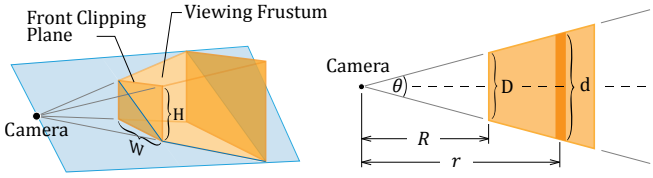


Figure 2: *Three-dimensional viewing frustum (left) and its diagonal cross-section (right). Visible volume is shaded in orange.*

Ψ denotes a view-dependent weighting factor. r and R are the Euclidean distances measured from the camera to an arbitrary cell's node and from the camera to a front clipping plane respectively (see Figure 2). α is a camera's perspective ratio defined as follow.

$$\alpha = \frac{D}{2R} = \tan(\frac{\theta}{2}) \tag{3}$$

Equation 3 refers to the viewing frustum (figure 2). D is the diagonal length of the front clipping plane: $D = \sqrt{H^2 + W^2}$, where H and W are the height and weight of front clipping plane respectively. R is the Euclidean distance from camera to the center of front clipping plane, θ is the camera's field of view (FOV) measured diagonally.

3.1.3 Adaptive Thresholds

Thresholds factored with a view-dependent weighting factor are called *adaptive thresholds*. Let T be a threshold for grid refinement specified on a viewing frustum's front clipping plane with FOV 90^o. Then an adaptive threshold for an arbitrary viewing frustum at distance r and FOV θ is defined as follows.

$$T^* = (\tau \Psi \phi) T \tag{4}$$

Where T^* is the adaptive threshold, Ψ is the view-dependent weighting factor and τ is a view-dependent coefficient for specifying the weight that the viewing should affect the grid refinement.

Image resolution, which is a multiplication of height and weight of screen, is another factor that should be considered. If we lower the image resolution but keep everything else fixed, the grid can be allowed to be coarser (since details are negligible in a low-resolution image). To address this, we define a *resolution ratio* (ϕ) as a square root of grid resolution over image resolution:

$$\phi = \sqrt{\frac{res_{grid}}{res_{image}}} \tag{5}$$

3.1.4 Refinement Conditions

Once both fluid variation and adaptive thresholds are obtained, cells are refined using the following refinement conditions.

$$C(x,y,z) > T_s^* = (\tau \Psi \phi) T_s \tag{6}$$

$$C(x,y,z) < T_m^* = (\tau \Psi \phi) T_m \tag{7}$$

Equation 6 and 7 are refinement conditions for subdivision and merging respectively. $C(x,y,z)$ is the measuring of fluid variation. T_s^* is an adaptive threshold for subdivision and T_m^* is an adaptive threshold for merging.

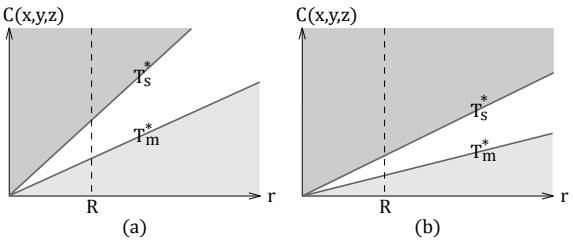


Figure 3: *Plots of the refinement stages on fluid variation $C(x, y, z)$ versus cell-to-camera distance r. Stages i.e., subdivision (dark grey area), idle (white area), merging(light grey area) are separated by subdivision condition (upper line) and merging condition (lower line). Steepness of both lines are defined by view-dependent coefficient (τ), FOV (α) and a resolution ratio (ϕ).*

If Equation 6 is satisfied then subdivision is performed. Likewise, if Equation 7 is satisfied then merging is performed. If neither Equation 6 nor Equation 7 is satisfied, then an idle state is assigned.

Figure 3 shows the relation of refinement stages, determined by fluid variation versus cell-to-camera distance. The grid may coarsened faster for steep slope (a) and vice versa for flat slope (b). There are many parameters that affect the steepness of the slope. For instance, the steepness of the slope may be increased by applying a greater VD coefficient (increase τ), using a wider FOV camera (increase α) or rendering with a lower resolution result (increase ϕ).

3.2 Discretization

The Navier-Stokes equations are solved term by term with a stable scheme [Stam 1999] and with a vorticity confinement [Fedkiw et al. 2001]. Then we follow a discretization scheme [Losasso et al. 2004] to solve the Poisson equations on the octree grid.

3.3 Coupling with Particle System

To model the animation from the computed grid, billboarding and particle system are coupled. The computed density values are used for constructing billboards, which is a fast and an effective way to represent the continuously existence of fluid. Meanwhile, we use the computed velocity vector fields to model the motion of particles, since particles can move freely from cell to cell throughout the domain, yielding a better representation of the fluid motions. In addition, they conform well to curved and complex boundaries, which can be used to reduce artefacts caused by coarse grids.

Particle motion is modelled by the Euler method. Other methods such as Runge-Kutta methods, BFECC method [Kim et al. 2007] and the MacCormack method [Selle et al. 2007] can be applied as well. However, the Euler method is the fastest and its accuracy is sufficient for most situations. The method is described as follows.

$$P_i(\mathbf{x}_i, t + \Delta t) = P_i(\mathbf{x}_i, t) + \mathbf{u} \tag{8}$$

Particle P_i is defined by its position $\mathbf{x}_i$. At each time step, the particle is moved forward by a velocity $\mathbf{u}$, where $\mathbf{u}$ is obtained by a linear interpolation of velocity stored in adjacent neighbouring cells.

4 Results and Discussion

All experimental results reported in this paper are performed on an octree grid with approximately 100,000 nodes. Typical simulation times were about 10 seconds per frame on a machine with 2.40 GHz dual core CPU and 2 GB of memory. Table 1 gives a detailed overview of the computation time of our method and the comparison with the corresponding simulation without VD refinement.

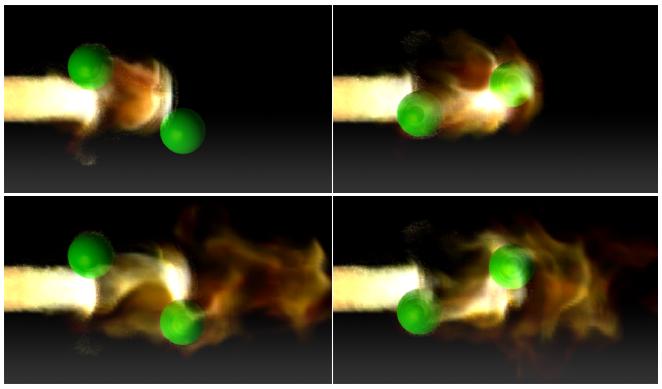


Figure 4: *Animation of fire (from left to right and top to bottom) using octree grid with view-dependent adaptive grid refinement. Rendered with approximately 340k particles.*

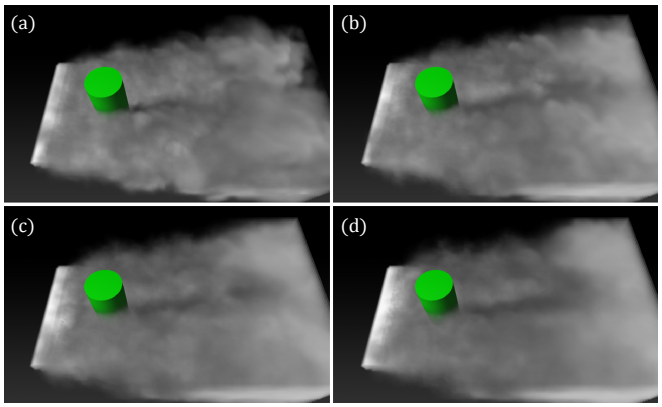


Figure 5: *A turbulence flow over a cylindrical rod. (a) no VD refinement. (b) VD refinement with $\tau = 1$ (+7.2% speed-up), (c) $\tau = 5$ (+26.2% speed-up) and (d) $\tau = 10$ (+41.1% speed-up).*

Figure 1 shows a flow of smoke moving through oscillating spheres. Different fields of view are applied to illustrate the benefits of VD refinement. For wide FOVs, the grid is coarsened to save computational time since details usually get smaller and can be neglected whereas for narrow FOVs, the grid is refined to obtain finer details.

Figure 6 demonstrates the two-way coupling of the particle system with billboarding. Using particles ensures a good conformance to curved surfaces and complex boundaries that produces interesting vortex effects, while using billboards generates shading effects, such as fire shown in Figure 4.

Figure 5 demonstrates the effect of VD refinement. Increasing the VD coefficient (τ) shifts up the refinement thresholds and reduces the number of nodes, corresponding to the simulation speed-up. When $\tau = 1$, the optimization is exactly proportional to the grid perspective. But in practice, τ can be up to 5 before detail loss become noticeable because foreground fluid, which usually has higher detail, usually occludes others those with lower detail behind.

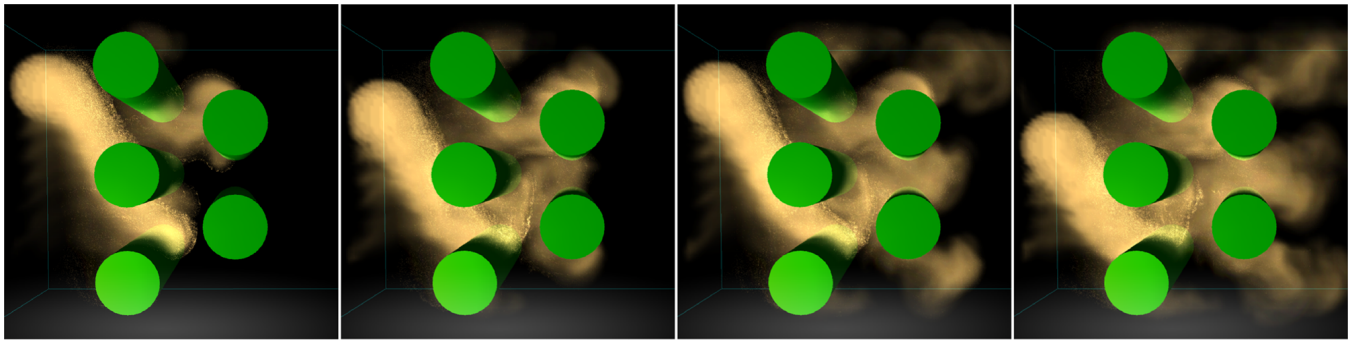


Figure 6: *Smoke moving through a set of obstacles. Simulated with a view-dependent coefficient $\tau = 1$ and $FOV = 45^o$.*

Scene	Refine	Discretize	Particle	Total	Gain
Fig.1a	.183s	9.47s	.078s	9.73s	5.90%
Fig.1b	.150s	8.14s	.065s	8.36s	19.13%
Fig.1c	.169s	7.40s	.064s	7.63s	26.24%
Fig.4	.131s	8.90s	.057s	9.09s	5.39%
Fig.5b	.166s	9.69s	.073s	9.93s	7.21%
Fig.5c	.100s	7.72s	.073s	7.89s	26.23%
Fig.5d	.087s	6.14s	.070s	6.30s	41.12%
Fig.6	.147s	9.72s	.063s	9.85s	5.98%

Table 1: *Performance comparison of our method on various scenes. Timings are given per frame. Gain measures the speed-up compared to the simulation without VD refinement.*

5 Conclusion

We have presented a view-dependent adaptive grid refinement for optimizing fluid simulation on octree structure. The method incorporates the refinement conditions with viewing information, optimizing the grid based on both viewing information and fluid variation. We have illustrated that the method can significantly reduce the number of cell nodes and improve the simulation time, while preserving the fluid details as well as its behavior. In addition, we have shown that the method can be integrated with a particle system to enhance visual results and reduce motion artefacts.

Acknowledgements

The authors would like to thank the VRCAI reviewers for their comment and suggestions. This research was partially funded by the TRF-Master Research Grants MRG-WI535E005, Yodia Multimedia, the Higher Education Research Promotion and National Research University Project of Thailand, Office of the Higher Education Commission, CUCP Academic Excellence Scholarship from Department of Computer Engineering, Chulalongkorn University.

References

AMENT, M., AND STRASS ER, W. 2009. Dynamic Grid Refinement for Fluid Simulations on Parallel Graphics Architectures. In *EUROGRAPHICS SYMPOSIUM ON PARALLEL GRAPHICS AND VISUALIZATION*, Citeseer.

BARRAN, B. 2006. *View dependent fluid dynamics*. PhD thesis.

BUNLUTANGTUM, R., AND KANONGCHAIYOS, P. 2011. Adaptive grid refinement using view-dependent octree for grid-based smoke simulation. In *Proceedings of the 4th International Conference on Motion in Games*, vol. 7060, 204–215.

CHENTANEZ, N., FELDMAN, B., LABELLE, F., O'BRIEN, J., AND SHEWCHUK, J. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, vol. 1, 219–228.

FEDKIW, R., STAM, J., AND JENSEN, H. 2001. Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, Citeseer, 15–22.

FELDMAN, B. E., O'BRIEN, J. F., AND ARIKAN, O. 2003. Animating suspended particle explosions. *ACM Transactions on Graphics 22*, 3 (July), 708.

FOSTER, N. 1996. Realistic Animation of Liquids. *Graphical Models and Image Processing 58*, 5 (Sept.), 471–483.

GOKTEKIN, T. G., BARGTEIL, A. W., AND O'BRIEN, J. F. 2004. A method for animating viscoelastic fluids. *ACM Transactions on Graphics 23*, 3 (Aug.), 463.

HARLOW, F. H., AND WELCH, J. E. 1965. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids 8*, 12, 2182.

KIM, B., LIU, Y., LLAMAS, I., AND ROSSIGNAC, J. 2007. Advections with significantly reduced dissipation and diffusion. *IEEE transactions on visualization and computer graphics 13*, 1, 135–44.

LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics 23*, 3 (Aug.), 457.

NEALEN, A. 2001. Physically Based Simulation and Animation of Gaseous Phenomena in a Periodic Domain. *I Can*.

SELLE, A., FEDKIW, R., KIM, B., LIU, Y., AND ROSSIGNAC, J. 2007. An Unconditionally Stable MacCormack Method. *Journal of Scientific Computing 35*, 2-3 (Nov.), 350–371.

SHI, L., AND YU, Y. 2004. Visual smoke simulation with adaptive octree refinement. In *Computer Graphics and Imaging*, Citeseer, 13–19.

STAM, J. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 121–128.

STAM, J. 2003. Real-time fluid dynamics for games. In *Proceedings of the Game Developer Conference*, Citeseer, vol. 18.

ZHENG, W., YONG, J.-H., AND PAUL, J.-C. 2009. Simulation of bubbles. *Graphical Models 71*, 6 (Nov.), 229–239.