ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, Supplement, December 2007, pp.1−7

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

保证时间一致性的风格化液体动画*

景 昊+, 周秉锋

(北京大学 计算机科学技术系,北京 100871)

Stylized Liquid Animation with Temporal Coherence

JING Hao+, ZHOU Bing-Feng

(Department of Computer Science and Technology, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-82529690, E-mail: jinghao@icst.pku.edu.cn, http://www.icst.pku.edu.cn

Jing H, Zhou BF. Stylized liquid animation with temporal coherence. *Journal of Software*, 2007,18(Suppl.): 1−7. http://www.jos.org.cn/1000-9825/18/s1.htm

Abstract: Stylized liquid animations are produced in this paper. The liquid simulator is mainly based on octree-staggered grid. This paper uses the compact difference scheme which is often used in non-staggered grid to achieve fourth order accuracy. The compact scheme is modified suitable to adaptive octree-staggered grid. In stylized rendering procedure, a 2D texture mapping method is used to generate still images. Laplacian scheme smoothing in time axes is introduced to maintain temporal coherence in animation. This algorithm can reduce the stroke “twinkle” phenomenon among frames.
Key words: fluid simulation; stylized rendering; compact difference scheme; temporal coherence

摘 要: 采用基于八叉树的交错格存储策略进行液体流动现象模拟,并生成非真实感风格化的动画.提出了将紧致差分格式应用于标准交错格存储策略中,在流体模拟计算时可以达到四阶精度.另外,还将该差分格式扩展应用到自适应的八叉树交错格存储格式中.在风格化绘制过程中,采用二维纹理映射方法生成每帧的静态图像,采用时间轴上的拉普拉斯变化保持帧间的时间一致性.该方法可以有效地改善风格化液体动画中笔画的“闪烁”现象.
关键词: 流体模拟;风格化绘制;紧致差分格式;时间一致性

1 Introduction

Earlier works on physical based fluid simulation mainly focused on photorealistic simulation, which can provide complex motions and rich visual detail, including water[1−3] or smoke[4]. The fluid simulation algorithms have developed rapidly in recent years. They can even deal with complex boundary conditions including deforming boundaries and moving obstacles. Ray-Tracing techniques are often used to generate final photorealistic animation.

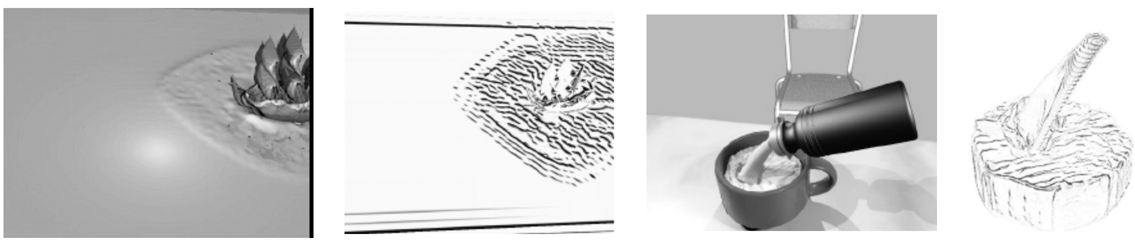
Although fluid animations in photorealistic rendering style are quite common, stylized fluid animations are less

* Supported by the National Natural Science Foundation of China under Grant No.60573149 (国家自然科学基金); the Natural Science Foundation of Beijing, China under Grant No.4072013 (北京市自然科学基金)

Received 2007-04-30; Accepted 2007-11-23

common, since the temporal coherence among frames is hard to maintain. It's a key challenge for stylized rendering so that strokes adapt smoothly over time to changes in the animation. Existing algorithms aimed at coherence when changing viewpoint to view same model. However, in our stylized fluid animation, we must deal with coherence of viewing different continuous 3D models. Compared with the time-consuming ray-tracing procedure, stylized rendering technique render frames much faster. It can be used for some special purpose, e.g. the cartoon movie, the fluid simulation on mobile devices on which the rendering speed is limited strictly. Recently, researchers have rendered cartoon effects of smoke and clouds[5].

In this paper, we generate stylized liquid animation by physical based fluid simulation method. We place two main requirements in our liquid animation system. First, detailed 3D liquid movement near the interface should be computed. Second, strokes in stylized animation should maintain frame-to-frame coherence. The simulator we used is based on the simulation method on adaptive octree staggered grid[6]. We modified the compact difference scheme suitable for octree grid. It can achieve fourth order accurate computation without adding more computation time. Our simulator ensures that the basic fluid motion is plausible. We modified the adaptive octree grid method to simulate the liquid movement more steadily and accurately. Particle level set method is used to build triangle mesh each frame for rendering. Then we render stylized frames from the triangle meshes. We present a texture mapping stroke rendering technique to maintain the frame-to-frame coherence. Figure 1 show two scenes we rendering using our simulator and stylized algorithm.



(a) (b) (c) (d)

Fig.1 Two liquid scenes with ray-tracing technique and stylized rendering

2 Previous Work

The physical based fluid simulation has become an important tool in the entertainment industry. Solving the 3D Navier-Stokes equations was popularized within the past decade[4,7,8]. The most commonly used method is the "staggered grid" storage scheme[9]. But the complex boundary conditions are hard to describe in this methods. Losasso, *et al*.[6] presented an octree-based method. It retains many of the advantages of regular grids while allowing computational effort to be focused in particular parts of the simulation domain. It also has advantage on detailed tracking of free surface issue. The liquid surface simulation combining with Navier-Stokes equations was also present in recent years[1,2,9]. Besides above structured grid method, some researchers introduced the unstructured tetrahedral mesh method from Computational Fluid Dynamics (CFD) to computer graphics domain[10]. These algorithms created can complex fluid animation, even including deforming boundary conditions. Since our main purpose is not dealing with complex fluid environment, we didn't choose those unstructured grid method. Our liquid simulation method is based on octree-based method[6]. In existing staggered grid method, researchers often used the simple difference scheme, e.g. FDS (forward difference scheme), BDS (backward difference scheme), which is second order accurate naturally because of the grid storage scheme. We modify compact difference scheme which is high order accurate and often used in non-staggered grid on octree staggered grid to achieve fourth order accurate calculation.

For stylized rendering, most algorithms render strokes following the feature lines on 3D models as the base paths of strokes[11−13]. They mainly focus on the silhouette and crease edges on models. Kalnins, *et al*.[12] parameterize the silhouette edge to a Catmull-Rom spline to representing the base path. They synthesize strokes by exampling offsets along each stroke path. Arc-Length parameterization is used in their work. Isenberg, *et al*.[13] make good work on producing silhouette strokes with artifacts filtering. Researchers have done excellent works on extracting feature lines which can convey shape more efficiently than silhouette, including ridge-valley lines[14] and suggestive lines[15,16].

Some researchers make stylized strokes rendering by simulating the brush artist used to draw a picture. Chan, *et al*.[17] proposed a method for generating Chinese Painting style rendering from 3D models. They separate a model to several parts following the geometry characteristic, then use texture mapping to produce Chinese brush style painting. Their method has limitations. The separating procedure has to be done by user interactively, and their method can only handle model with simple surface features.

Existing stylized liquid animation methods have been presented in Ref.[18,19]. Those methods, however, are not physics based methods, they can not simulate fluid motion as in real word. Physics based method is used to generate cartoon effects of smoke and clouds[5], But they didn't handle the liquid animation in non-photorealistic style.

Little attention has been given to the temporal coherence for stylized animation. Temporal coherence is especially challenging for stylized rendering based on stroke, like silhouette and suggestive lines. Since the inter-frame correspondence as frames evolve over time may not exist. Masuch, *et al*.[20] describe a solution for the setting when natural silhouette arc-length parameterizations remains consistent, but it only applied under simple conditions (e.g. rotating about a cylinder). Bourdev[21] gave a more general approach based on parameter samples from nearby strokes in the previous frame. Kalnins, *et al*.[22] presented a more robust sample propagation algorithm. All above algorithms aimed at coherence when changing viewpoint to view same model. In fluid simulation, models of each frame are different. Existing algorithms may fail at this condition.

3 Liquid Simulation

We use the incompressible Navier-Stokes equations to simulate the movement of fluid for the conservation of mass and momentum as follows.

$$\nabla \cdot V = 0 \tag{1}$$

$$\frac{\partial V}{\partial t} + (V \cdot \nabla)V = -\nabla p + \nu\nabla^2 V + F \tag{2}$$

where $V=(u,v,w)$ is the velocity field, F is the external forces (e.g. gravity), and p is pressure filed. In existing method, Eq.(2) is often solved in two steps. First, a guess velocity field V^* is introduced ignoring the pressure term with forward difference scheme. V can be computed as Eq.(3).

$$V = V^* - \Delta t \nabla p \tag{3}$$

Then the pressure is defined as the solution to the Poisson Eq.(4).

$$\nabla^2 p = \frac{\nabla \cdot V^*}{\Delta t} \tag{4}$$

In our simulation method, we used a compact difference scheme to compute the guess velocity field V^* first. It is fourth order accurate in space (original second order difference scheme can be found in Ref.[9]). The second order Runge-Kutta algorithm is applied to achieve high order accurate in time difference.

3.1 Compact difference scheme in regular staggered grid

We write Navier-Stokes Eq.(2) in discrete form of Eq.(5).

$$\frac{V^{t+\Delta t}-V_{temp}}{\Delta t}+\nabla_d p^{t+\Delta t}-F=0 \tag{5}$$

where $V^t=\{u,v,w\}$, $V_{temp}=V^t-\Delta t G_d(V^t)$

$$G(V^t)=\begin{Bmatrix} uu_x+uv_y+uw_z-\nu(u_{xx}+u_{yy}+u_{zz}) \\ vu_x+vv_y+vw_z-\nu(v_{xx}+v_{yy}+v_{zz}) \\ wu_x+wv_y+ww_z-\nu(w_{xx}+w_{yy}+w_{zz}) \end{Bmatrix} \tag{6}$$

G_d,∇_d are discrete forms of G,∇. First, we define the compact difference scheme in standard staggered grid as Fig.2 shows. The vector field (velocity) store at cell faces, and the scalar filed (pressure, density, etc.) store at cell center. For compact scheme of uu_x:

$$uu_x=\frac{(uu)'_{i+1,j+1/2,k+1/2}+4(uu)'_{i,j+1/2,k+1/2}+(uu)'_{i-1,j+1/2,k+1/2}}{6}=\frac{(uu)_{i+1,j+1/2,k+1/2}-(uu)_{i-1,j+1/2,k+1/2}}{2\Delta x}.$$

We can also get compact scheme of uv_y and uw_z.

$$uv_y=\frac{(uv)_{i,j+3/2,k}-(uv)_{i,j-1/2,k}}{2\Delta y},\quad uw_z=\frac{(uw)_{i,j,k+2/3}-(uw)_{i,j,k-1/2}}{2\Delta z}.$$

For compact scheme of u_{xx},

$$u_{xx}=\frac{u''_{i+1,j+1/2,k+1/2}+10u''_{i,j+1/2,k+1/2}+u''_{i-1,j+1/2,k+1/2}}{12}=\frac{u_{i+1,j+1/2,k+1/2}-2u_{i,j+1/2,k+1/2}+u_{i-1,j+1/2,k+1/2}}{\Delta x^2}.$$

We can also get compact scheme of u_{yy} and u_{zz}.

$$u_{yy}=\frac{u_{i,j+3/2,k}-2u_{i,j+1/2,k}+u_{i,j-1/2,k}}{\Delta y^2},\quad u_{zz}=\frac{u_{i,j,k+3/2}-2u_{i,j,k+1/2}+u_{i,j,k-1/2}}{\Delta z^2}.$$

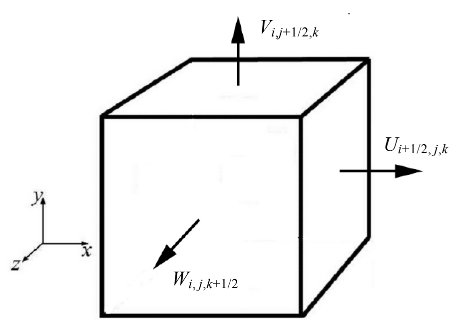


Fig.2 A standard staggered grid cell

The other derivative term in Eq.(6) can be treated similarly. Following above equations, we can compute the guess velocity field V^* and update velocity. The main difference of the compact difference formula and the simple difference formula in Ref.[9] is the subscript of each term.

3.2 Compact difference scheme in octree staggered grid

We extend the above compact scheme from regular staggered grid to octree grid. Figure 3 shows a standard octree grid. Data access is not as convenient as for regular staggered grid. Using the Green's theorem, the x direction difference of u^* can be computed as Eq.(7).

$$\frac{\partial u^*}{\partial x}=\frac{[(u_6^*+u_7^*+u_8^*+u_9^*)/4+u_3^*+u_4^*+u_5^*]/4-u_1^*}{\Delta x} \tag{7}$$

Similarly, we can compute y and z direction difference. Following Eq.(7), we can compute the compact scheme difference on octree staggered grid. Although it is more difficult than in regular grid, there is strong benefit in designing a discretization that leads to a symmetric linear system when computing pressure p as the solution of Poisson Eq.(4). Losasso, *et al*.[6] have declared this situation in their paper.

The compact difference scheme on octree staggered grid we described above is fourth order accurate. Since the only difference of the compact difference formula and the simple difference formula in Ref.[9] is the subscript of each term, the computation time is the same as simple second order difference scheme.

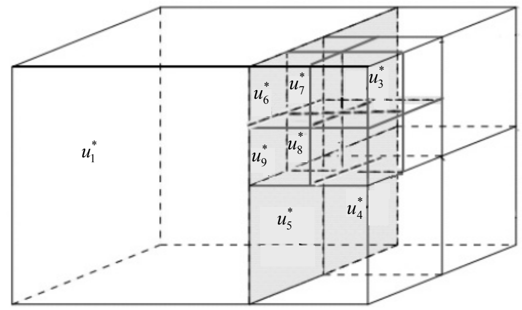


Fig.3 An octree grid cell

4 Stylized Rendering with Temporal Coherence

After update velocity field on octree, we move particles with an explict second order Runge-Kutta technique. Then particle level set method[2] is used to build liquid surface at each time step. We generate each stylized frames from surface meshes. We use 2D texture mapping method to generate stylized frames with strokes. Radial curvature κ is used to control the texture coordinates. But if we rendered frames with our method directly to produce the liquid animation, there exist serious “twinkle” of strokes among frames. Figure 4 shows the “twinkle” phenomenon.

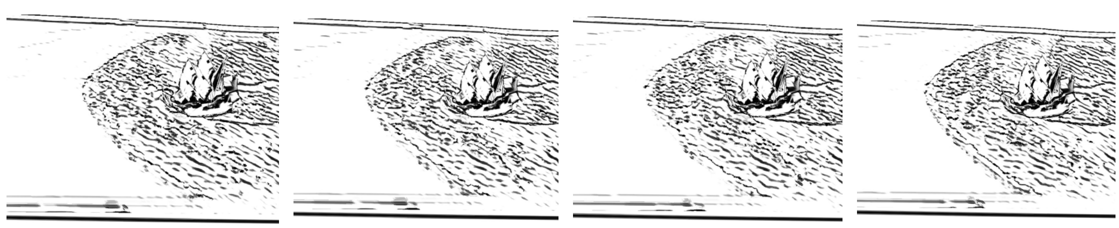

(a) (b) (c) (d)

Fig.4 Strokes “twinkle” phenomenon in stylized animation

In our method, We use 2D texture as Fig.5 shows with GL_LINEAR option in OpenGL. This mode can smoothes the color transition boundary. With this texture whose u and v texture coordinates are controlled by radial curvature, we can produce stylized still images from 3D models.

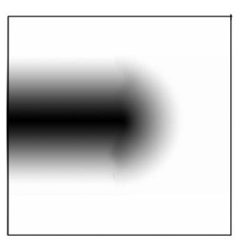


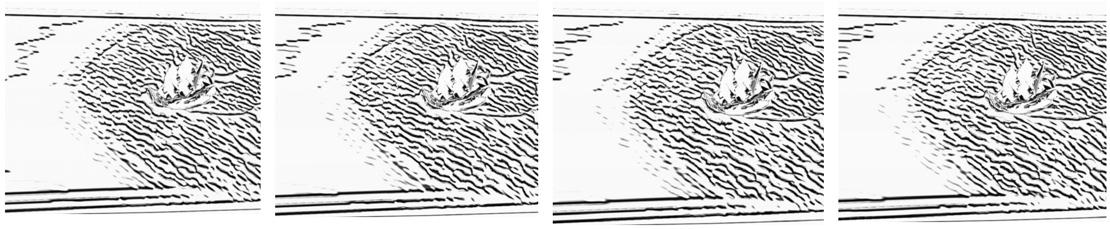
Fig.5 Texture used in stylized rendering

To maintain temporal coherence, we define the u and v texture coordinates with consideration of relationship among frames. Because we use particle level set to build triangle mesh, we record the relationship between vertices of mesh and particles. Assuming at current time step t, a vertex v_p^t in frame f^t corresponds to particle p_i when marching liquid surface. The radial curvature of v_p^t is κ_p^t. At previous time step $t-\Delta t$, particle p_i corresponds vertex $v_p^{t-\Delta t}$ with radial curvature $\kappa_p^{t-\Delta t}$. At next time step $t+\Delta t$, particle p_i corresponds vertex $v_p^{t+\Delta t}$ with radial curvature $\kappa_p^{t+\Delta t}$. We make coordinates in u and v direction equal. An iterative procedure to compute the texture coordinate $\kappa_p^{t(N)}$ is applied following Eq.(8) and Eq.(9).

$$\Delta\kappa = \kappa_p^{t+\Delta t} + \kappa_p^{t-\Delta t} - 2\kappa_p^t \tag{8}$$

$$\kappa_p^{t(N)} = \kappa_p^{t(N-1)} + \lambda\Delta\kappa \tag{9}$$

where $N\geq1$ is iteration times, $0<\lambda<1$. We let $\lambda=0.2$ and $N=4$ in our paper. Eq.(8) and Eq.(9) can be understood as a laplacian smoothing in time axes. Using $\kappa_p^{t(N)}$ as u and v coordinates of vertex v_p^t, we can generate results as Fig.6 shows. In animation, the "twinkle" phenomenon among frames is reduced.



(a) (b) (c) (d)

Fig.6 Stylized animation after applying temporal coherence maintaining algorithm

5 Conclusions

In our paper, we generate stylized liquid animation with temporal coherence. The compact difference scheme which is often used in non-staggered grid is modified to use in octree-staggered grid which our liquid simulator is based on. Our compact difference scheme improves the accuracy to fourth order while not increasing the computation time.

In stylized rendering process, we use 2D texture mapping method to generate stylized frames. Laplacian smoothing in time axes is introduced to maintain frame-to-frame coherence in animation. The algorithm we used can reduce the stroke "twinkle" among frames and improve the animation quality.

References:

[1] Foster N, Fedkiw R. Practical animation of liquids. In: SIGGRAPH 2001: Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 2001. 23−30.

[2] Enright DP, Marschner SR, Fedkiw RP. Animation and rendering of complex water surfaces. ACM Trans. on Graphics, 2002,22(3): 736−744.

[3] Losasso F, Shinar T, Selle A, Fedkiw R. Multiple interacting liquids. ACM Trans. on Graphics, 2006,25(3):812−819.

[4] Fedkiw R, Stam J, Jensen HW. Visual simulation of smoke. In: SIGGRAPH 2001: Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 2001. 15−22.

[5] McGuire M, Fein A. Real-Time rendering of cartoon smoke and clouds. In: NPAR 2006: Proc. of the 4th Int'l Symp. on Non-Photorealistic Animation and Rendering. New York, ACM Press, 2006. 21−26.

[6] Losasso F, Gibou F, Fedkiw R. Simulating water and smoke with an octree data structure. In: SIGGRAPH 2004: Proc. of the Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 2004. 457−462.

[7] Foster N, Metaxas D. Controlling fluid animation. In: CGI'97: Proc. of the 1997 Conf. on Computer Graphics Int'l. Washington: IEEE Computer Society, 1997. 178−188.

[8] Stam J. Stable fluids. In: SIGGRAPH'99: Proc. of the 26th Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 1999. 121−128.

[9] Foster N, Metaxas D. Realistic animation of liquids. In: Proc. of the Graphics Interface 1996. Orlando: Academic Press, 1996. 204−212.

[10] Klingner BM, Feldman BE, Chentanez N, O'Brien JF. Fluid animation with dynamic meshes. In: SIGGRAPH 2006: ACM SIGGRAPH 2006. New York: ACM Press, 2006. 820−825.

[11] Hertzmann A, Zorin D. Illustrating smooth surface. In: SIGGRAPH 2000: Proc. of the 27th Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 2000. 517−526.

[12] Kalnins RD, Markosian L, Meier BJ, Kowalski MA, Lee JC. WYSIWYG NPR: Drawing strokes directly on 3D models. In: SIGGRAPH 2002: Proc. of the 29th Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 2002. 755−762.

[13] Isenberg T, Halper N, Strothotte T. Stylizing silhouettes at interactive rates: From silhouette edges to silhouette strokes. Computer Graphics Forum, 2002,21(3):249−258.

[14] Ohtake Y, Belyaev A. Ridge-Valley lines on meshes via implicit surface fitting. In: SIGGRAPH 2004: Proc. of the 31st Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 2004. 609−612.

[15] DeCarlo D, Finkelstein A, Rusinkicwicz S, Santella A. Suggestive contours for conveying shape. ACM Trans. on Graphics, 2003, 22(3):848−855.

[16] Sousa MC, Prusinkiewicz P. A few good lines: Suggestive drawing of 3D models. Computer Graphics Forum, 2003,22(3):381−390.

[17] Chan C, Akleman E, Chen J. Two methods for creating Chinese painting. In: Proc. of Pacific Graphics 2002. Washington: IEEE Computer Society, 2002. 403−412.

[18] Yu JH, Xu XG, Peng QS. Cartoon water forms synthesis using random sinusoidal functions. Journal of Computer Research and Development, 2001,38(5):519−523 (in Chinese with English abstract).

[19] Peng QS, Yin XQ, YU JH. A rain model for cartoon animation. Journal of Software, 2002,13(9):1881−1886 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/1881.pdf

[20] Masuch M, Schumann L, Schlechtweg S. Animating frame-to-frame coherent line drawings for illustrative purpose. In: Proc. of Simulation and Visualisierung. SCS Europe, 1998. 101−112.

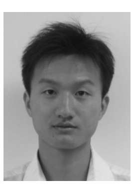
[21] Bourdev L. Rendering nonphotorealistic strokes with temporal and arc-length coherence [MS. Thesis]. Brown University, 1998.

[22] Kalnins RD, Davidson PL, Markosian L, Finkelstein A. Coherent stylized silhouettes. In: Proc. of SIGGRAPH 2003. New York: ACM Press, 2003. 856−861.

附中文参考文献:

[18] 于金辉,徐晓刚,彭群生.用随机正弦波拟合卡通流水.计算机研究与发展,2001,38(5):519−523.

[19] 彭群生,尹小勤,于金辉.卡通动画雨模型.软件学报,2002,13(9):1881−1886. http://www.jos.org.cn/1000-9825/13/1881.pdf



JING Hao was born in 1980. He is a Ph.D. candidate at the Institute of Computer Science and Technology, Peking University. His current research interests include non-photorealistic rendering and physics based fluid simulation.

ZHOU Bing-Feng was born in 1963. He is a professor and doctoral supervisor at the Institute of Computer Science and Technology, Peking University. His research areas are digital geometry processing, virtual reality, image based rendering and non-photorealistic rendering.