Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008) *ISSN: 1907-5022*
Yogyakarta, 21 Juni 2008

A 3D FLUID DYNAMIC RENDERING FOR REAL-TIME SURGICAL CUTTING SIMULATION

Sugeng Rianto
Physics Department Brawijaya University
Jl. Veteran 2 Malang Indonesia
priantos@brawijaya.ac.id

ABSTRACT

In this paper, we introduce a 3D fluid dynamics solver for real-time rendering in virtual environment. We approach the solution of differential equations based on the constrained interpolation propagation (CIP) technique on a GPU. Since the CIP combine the solution for fluid equations and their interactions with the environment together, the Navier-Stokes equation can be solved efficiently. Furthermore, to achieve high performance results for real time rendering without involving a supercomputer, we take advantages of the parallelism and programmability on the GPU. The rendering is performed on pixels that can be considered to be a grid of cells; thus, processing on multiple vertices and pixels can be done simultaneously in parallel. This strategy is effective enough to render fluid dynamic model for real-time virtual cutting in 3D computer generated object. Experimental results demonstrate that the rendering of skin cutting followed by blood flowing over the anatomical surface run smoothly in a real-time for virtual reality interaction.

Keywords: *Cutting simulation, fluid dynamic, real time rendering, virtual surgery.*

1. INTRODUCTION

Virtual reality (VR) is widely used as a training tool from science to engineering or social sciences [1, 2]. The VR techniques provide the potential for a realistic, safe, controllable environment for novice doctors to practice surgical operations, allowing them to make mistakes without serious consequences[3, 4]. Virtual reality is also widely known as virtual (or synthetic) environments (VEs). These are three dimensional computer generated environments that the user is not only able to experience interactively but also able to manipulate in real time[5]. The way humans interact with their physical environments are artificially imitated in VEs.

VEs provide natural interface between humans and computers in virtual reality. A VE consists of an interfacing system with humans through output of sensory information and input of commands. Several examples of such input and output devices are 3D space mouse, data gloves, 3D navigation devices, and haptic feedback.

The major goal of computer graphics and virtual environment simulation is to provide "realistic" methods to allow an easy creation of digital equivalents for natural phenomena. One of the important natural phenomena in surgical simulation is fluid dynamic behavior.

Although fluid dynamic phenomenon should not be neglected in simulation, mostly surgical simulation developers avoid including fluid dynamic during simulation. Beside to reduce the computational load of the system, to solve the underlying laws that govern fluids motions is a common problem. The fluid dynamic equation is known to be difficult to solve in an efficient way, sometimes even too difficult to be solved at all. In real-time simulation, fluid modeling even is expensive to be realized in standard rasterization [6].

Furthermore, a real time rendering in immersive virtual environment with haptic interaction demand high frequency processing (1000 Hz) and the visual aspects need frame rates in the order of 30 fps. On the other hand, a surface deformation or volume rendering that is usually included in the simulation also need high performance computations. This reason also outweighs people to neglect fluid dynamic inclusion.

This study introduces novel framework utilizing the immersive virtual environment for fluid interaction in surgical cutting procedures and haptic rendering. The fluid dynamic computations rely on the Navier-Stokes equation in three-dimension (3D) is solved using constrained interpolation profile (CIP) based on GPU programming. The framework is expected to provide efficient computation which is applicable for haptic rendering in immersive virtual environment.

2. PREVIOUS WORK

Several methods have been developed to simulate fluids and its interaction to objects based on the Navier-Stokes (NS) equations to exhibit physically realistic fluid behaviors [7, 8]. In consideration to computation cost some studies attempted to solve the NS equation with simplifications [9, 10]. Several researchers use two-dimensional (2D) solution approach to obtain interactive performance on animating height field fluid on surface [11, 12]. Interactive animation on fluid that allow a large simulation time step in stable condition has been pioneered by [9]

The method in solving NS equation itself has been developed in many ways. Recently, studies using cubic interpolated propagation or constrained interpolation profile (CIP) provide better and efficient computations [13, 14] for fluid dynamic modeling. This method is particularly useful for non dissipative liquids and for reducing numerical diffusion [15].

The cubic interpolated propagation CIP methods was firstly introduced in [16] as a universal solver for hyperbolic equation by cubic interpolation. It is then used for solving fluid dynamic equation as a conservative semi-Lagrangian solver for solid, liquid and gas [17, 18]. Some studies prove that this method is efficient enough to solve dynamic fluid equation in all states with motions [13, 19].

Although the CIP has been accepted as an alternative method for fluid solver that fits the need of computer animation, there has been no study that uses this method in immersed virtual environment with haptic interaction. [20, 21] attempted to conduct fluid dynamic computation on GPU to get parallelism for more efficiency.

This paper introduced a framework which combines CIP and parallelism in GPU, trying to achieve most efficient solution of fluid dynamic equations for real time rendering in virtual environment with haptic interaction.

The rest of the paper is organized as follows: CIP as the fluid dynamic solvers is described in Section 3; computational fluid dynamics implemented in GPU is described in Section 4; Section 5 described the experimental set-ups and the results; with Section 6 concludes the paper.

3. FLUID DYNAMIC SOLVER

3.1 Cubic interpolated propagation (CIP)

Although we make simplifying assumptions to model a complex fluid motion in surgical cutting, producing realistic fluid movement in the presence of air, scalpel, and skin is formulated as close as possible to the driven physical laws. The formulation assumes that liquid movement is an incompressible and homogenous fluid. It is incompressible if the volume of the fluid is constant during simulation period and homogenous fluid if its density is constant over space and time.

The Navier-Stokes equation for incompressible flow for multiphase fluid in Cartesian grid can be described as follow:

$$\frac{\partial u}{\partial t} = -(u \bullet \nabla)u - \frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2 u + F_c \qquad (1)$$

$$\nabla \bullet .u = 0$$

(2
)

where u is a velocity field of fluid, p is a pressure, □is the density of the fluid, □is a viscosity coefficient and F_c is external forces respectively. The behaviors of fluid can be predicted by solving equations (1) and (2). Flow dynamics simulation can be done using staggered grid that defines velocity component at cell faces and scalar variable in cell centers. The fraction of volume of fluid in each cell is transported by advection equation as formulated below.

$$\frac{\partial f}{\partial t} = -(u \bullet \nabla)f \qquad (3)$$

where f is the function of fluid fraction (VOF). By using the CIP method [18, 22, 23] and the advection form ($-(u \bullet \nabla)u$), the equation (3) can be solved.

The interface between liquid and solid is traced by distributing Equation (4) into advection phase shown in Eqn. (5) and non-advection in Eqn.(6) as follows:

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = G \qquad (4)$$

$$\frac{\partial f}{\partial t} = G \qquad (5)$$

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0 \qquad (6)$$

The CIP scheme calculates the advection phase by shifting a cubic interpolated profile into space according to the total derivative equation. Since the interpolation profile uses the cubic polynomial and the spatial derivative value at each grid point, it can be derived that:

$$\frac{\partial f'}{\partial t} + u\frac{\partial f'}{\partial x} = G' - f'\frac{\partial u}{\partial x} \qquad (7)$$

$$\frac{\partial f'}{\partial t} = G' - f'\frac{\partial u}{\partial x} \qquad (8)$$

$$\frac{\partial f'}{\partial t} + u\frac{\partial f'}{\partial x} = 0 \qquad (9)$$

The exact solution of Equation (6) is

$$f(x,t) = f(x - ut, 0) \qquad (10)$$

If the velocity is assumed to have a constant value within short time, we get

$$f(x, t + \Delta t) \cong f(x - u\Delta t, t) \qquad (11)$$

However, since *x*-□□*t* is not always located on grid points, this point is interpolated locally by Hermit polynomials within the calculation grid point as the value for the next time step.

Equation (1) can be used to approximate the other states as follow: The pressure equation is solved using the Poisson's equation, the liquid is assumed to have no viscosity, and the external force is determined by the force released from the haptic interaction.

3.2 Fluid interaction and simulation

Although CIP can combine the solution of fluid dynamics and the fluid-solid or fluid-air interaction together in one algorithm, the dynamic interaction between fluid and solid, and between haptic and solid, need to be studied in order to achieve realistic visual result. Such interaction can be described as collision between two objects. For haptic rendering, collision detection and the impulse response is employed using the method introduced in [24]. The interface between fluid and solid are simulated with a volume of solid (VOS) value. The VOS defines the fraction of a solid contained in a fluid cell and approximate the shape of rigid body in the fluid solver. The cells with VOS values less than 0.5 are considered as fluid otherwise a solid.

The dynamics of rigid bodies are considered based on force *Fc* acting on the centre of mass and the torque *Tc* from the centre of mass as follow:

$$F_C = Mg + \iint_s (p.n) n ds \qquad (12)$$

$$T_C = \iint_s (r - r_C) n ds \qquad (13)$$

where *g* is the gravity, *M* is mass of the object, *p* is the pressure of boundary between a solid cell and a fluid cell, *n* is the normal vector to the boundary, *r* is a point on the boundary, r_c is the object's centre of gravity, and *s* is the boundary between fluid and the solid object or the area of object submerged in the cell.

4. MODELLING AND COMPUTATION OF THE FLUID DYNAMIC ON GPU

Our proposed model for the fluid solver that is developed here, is adapted from [25] with novelties on the numerical methods. As explained above the CIP numerical method to solve the NS-equations is used. The solver is run on GPU and is divided into four fragment programs: advection, cubic-Hermit, divergence and gradient, and boundary conditions.

The fluid motion model is initialized by creating an octree around the mesh surface. The algorithm for the texture mesh is developed based on the octree texture from the fragment program using the tree lookup as suggested in [26]. Then, the density values are updated using a render-to-texture operation during the simulation and stored in 2D texture map. The octree's leafs store the index of a pixel in the form of the three 8-bit values (in RGB Channels). The simulation will also access the density of the neighbors to recode the neighboring information.

The fragment programs are written in Cg with the call *tex2D(Dmap, I)*, where *I* is the index stored within a leaf *L* of the octree and *Dmap* is the density map. This call returns the density value associated with leaf *L*. The call *tex2D(N ,I)* yields the index within the density map of a 3D space neighbour. To get all neighbour information, we require 26 textures in 3D or 9 textures in 2D. When all textures have been created, the density map can be render on the texture mesh and the location of each density can be retrieve from the octree.

5. EXPERIMENTAL SETUP AND RESULTS

5.1 Experimental setup

The proposed framework is implemented on a personal computer (PIV Dual Core 3.2 GHz, 2 GB RAM, 80GB HD), the PHANTOM Haptic premium 1.5 (from Sensable) in Reachin display with stereo LCD shutter glasses. The visual and haptic rendering is controlled using Reachin API 4.2 (Figure 1) in combination with Visual Studio.net and Python. We use the data set that is downloaded from [27] and is cropped into the anatomical section of interests. These data then is reconstructed to generate 3D volume and 3D iso-surfaces (mesh). The volume is overlapped with surface for cutting purposes.

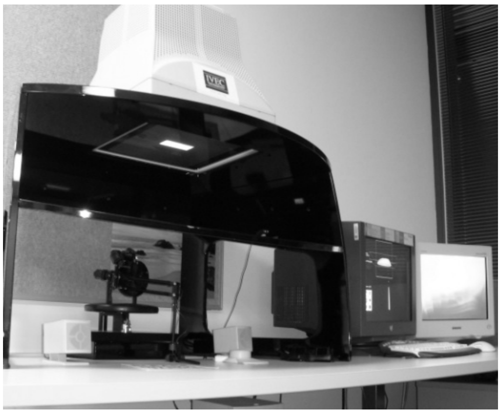
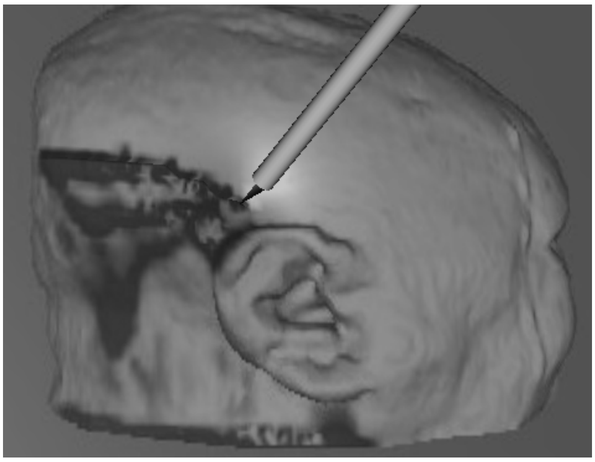


Figure 1. Haptic workbench with Reachin display and work station.

In this study we employ mesh cutting techniques by combining adaptive re-meshing and mesh sub-division for creating surface cutting paths. The cutting paths are created when the following conditions are met: the scalpel collides with a surface, the stylus button is pressed, and the force is applied greater than the surface stiffness. The force here represents pressure force and the sharpness constant of the blade. The cutting process will be followed by flowing of the fluid. At the same time, voxel removals in the cutting areas are undertaken. The fluid, representing blood, flows over the surface toward lower parts on the surface or toward the centre of the gravity force.

5.2 Results

Several examples of snapshot surgical cutting simulations are shown in figure 2. These examples describe the cutting simulation followed by blood flowing over the surface in the cutting path in a head (figure 2a) and in a heart (figure 2b). The fluids start flowing just after initial cutting and continue to flow from the cutting path. The volume of fluid in these stages is controlled by the size of the cutting hole and their relative position and is added continuously from each cutting path.



a) A surgical cutting simulation on the head's skin

The simulations show the movement of the fluid at different surface conditions. The first cutting example (figure 2a) is applied over the head anatomy on a resolution of 512 x 512 x 220 on the skin surface with 37096 triangles ran at 26 fps. This figure drops down into 18 fps when the cutting process is done. However, in the second example (figure 2b) on the resolution of 128 x 128 x 128 with 1619 triangles show that the visualization ran about 50fps and cutting process not reduce the speed. We also can interact with fluid even without making a cutting.

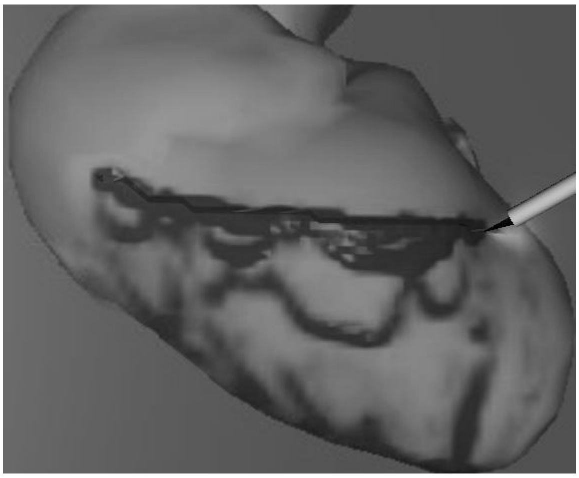


Figure 2. Interactive cutting over the mesh surface with liquid flowing along the surface paths

b) Surgical cutting simulation is applied in heart

It is clearly seen that the modeling technique in this study can simulate surgical cutting with flowing fluid in stable good interactive control. Although, there is no problem in flowing fluid model, the cutting procedure sometime miss the cut area when the blade move to fast. In this case, the cutting process must be repeated from the most end of the last cutting path to connect with new path.

6. CONCLUSION AND FUTURE WORK

The computation and modeling of fluid for haptic rendering and surgical cutting simulation can be done at reasonable efficiency using CIP fluid solver that combines the solution of fluid dynamics and its interaction with solid object together. By that using CIP fluid solver and GPU programming, the computation for solving Navier-Stokes equation can be done efficiently without involving supercomputer.

Our results confirm that by combining the CIP solver and GPU programming, the haptic interaction for skin cutting followed by blood flowing model over the anatomical surface can be simulated in a real-time interaction with visually realistic display.

More detail results such as involving two or more fluid and their interactions will be gathered in the next study. This may investigate the fluid dynamic interaction during surgical simulation, the performance details, and the validity of the model in the simulation.

REFERENCES

[1] L. Rosenblum and M. Macedonia, "Virtual Orthopedic Surgery Training," *IEEE Computer Graphics and Applications,* pp. 6-9, 2000.

[2] R. J. Stone, "Haptic feedback: A potted hystory, from telepresence to virtual reality," *Lecturer notes in computer science,* vol. 2058/2001, pp. 1-7, 2000.

[3] Anonimous, "The cutting edge of virtual reality; REPORT: MEDICAL TECHNOLOGY; Cutting edge of virtual reality," in *The Economist*, 2001, p. 17.

[4] A. Higgins and S. Koucky, "Mission impossible?," *Machine Design,* vol. 74, p. 28, Dec 12 2002.

[5] T. M. Wasfy and A. K. Noor, "Visualization of CFD results in immersive virtual environments," *Advances in Engineering Software,* vol. 32, pp. 717-730, 2001/9 2001.

[6] K. Crane, I. Llamas, and S. Tariq, "Real-Time Simulation and Rendering of 3D Fluids," in *GPU GEM 3*, H. Nguyen, Ed.: Addison-Wesley Professional, 2007.

[7] M. T. Carlson, "RIGID, MELTING, AND FLOWING FLUID," in *College of Computing*: Georgia Institute of Technology, 2004.

[8] J. X. Chen and N. d. V. Lobo, "Toward Interactive-Rate Simulation of Fluids with Moving Obstacles Using Navier-Stokes Equations," *Graphical Models and Image Processing,* vol. 57, pp. 107-116, 1995/3 1995.

[9] J. Stam, "Stable Fluids," *Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques,* pp. 121-128, 1999.

[10] N. Foster and R. Fedkiw, "Practical animation of liquids," *Proc. of SIGGRAPH 2001, ACM Press/ ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series,* pp. 23-25, 2001.

[11] N. Foster and D. Metaxas, "Controlling Realistic Animation," 1996.

[12] J. Stam, "Flows on surfaces of arbitrary topology," *ACM Transactions on Graphics (TOG) archive,* vol. 22, pp. 724-731, July 2003.

[13] T. Yabe, H. Mizoe, K. Takizawa, H. Moriki, H.-N. Im, and Y. Ogata, "Higher-order schemes with CIP method and adaptive Soroban grid towards mesh-free scheme," *Journal of Computational Physics,* vol. 194, pp. 57-77, 2004/2/10 2004.

[14] K. Takizawa, T. Yabe, M. Chino, T. Kawai, K. Wataji, H. Hoshino, and T. Watanabe, "Simulation and experiment on swimming fish and skimmer by CIP method," *Computers & Structures Frontier of Multi-Phase Flow Analysis and Fluid-Structure,* vol. 83, pp. 397-408, 2005.

[15] O.-Y. Song, H. Shin, and H.-S. Ko, "Stable but nondissipative water," *ACM Trans. Graph.,* vol. 24, pp. 81-97, 2005.

[16] T. Yabe, T. Aoki, G. Sakaguchi, P.-Y. Wang, and T. Ishikawa, "The compact CIP (cubic-interpolated pseudo-particle) method as a general hyperbolic solver," *Computers & Fluids,* vol. 19, pp. 421-431, 1991.

[17] T. Yabe, T. Ishikawa, P. Y. Wang, T. Aoki, Y. Kadota, and F. Ikeda, "A universal solver for hyperbolic equations by cubic-polynomial interpolation II. Two- and three-dimensional solvers," *Computer Physics Communications,* vol. 66, pp. 233-242, 1991/0 1991.

[18] T. Yabe, Y. Ogata, K. Takizawa, T. Kawai, A. Segawa, and K. Sakurai, "The next generation CIP as a conservative semi-Lagrangian solver for solid, liquid and gas," *Journal of Computational and Applied Mathematics,* vol. 149, pp. 267-277, 2002/12/1 2002.

[19] T. Utsumi, T. Yabe, J. Koga, T. Aoki, and M. Sekine, "Accurate basis set by the CIP method for the solutions of the Schrodinger equation," *Computer Physics Communications,* vol. 157, pp. 121-138, 2004/2/15 2004.

[20] Y. Liu, X. Liu, and E. Wu, "Real-time 3D Fluid Simulation on GPU with Complex Obstacles," in *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, 2004, pp. 247-256.

[21] W.Li, Z.Fan, X.Wei, and A.Kaufman, "GPU-Based flow simulation with complex boundaries," *Technical Report 031105, Computer Science Department, SUNY at Stony Brook,* 2003.

[22] T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki, "Realistic Animation of Fluid with Splash and Foam," *Computer Graphics Forum,* vol. 22, pp. 391-400, 2003/09/01 2003.

[23] K. Yokoi and F. Xiao, "Mechanism of structure formation in circular hydraulic jumps: numerical studies of strongly deformed free-surface shallow flows," *Physica D: Nonlinear Phenomena,* vol. 161, pp. 202-219, 2002/1/15 2002.

[24] A. Gregory, M. C. Lin, S. Gottschalk, and R. Taylor, "Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device," *Computational Geometry,* vol. 15, pp. 69-89, 2000/2 2000.

[25] M. J. Harris, "Fast Fluid Dynamics Simulation on the GPU," in *GPU Gems - Triks for Real-Time Graphics*, R. Fernando, Ed.: Addison Wesley, 2004, pp. 637-665.

[26] S. Lefebvre, S. Hornus, and N. Fabrice, "Octree Textures on the GPU," in *GPU Gems 2 - Programming Techniques for High-Performance Graphics and General-Purpose Computation*, M. Pharr, Ed.: Addison Wesley, 2005, pp. 595-613.

[27] Anonimous, "The Visible Human Project," U. S. N. L. o. Medicine, Ed.: http://www.nlm.nih.gov/research/visible/visible_human.html, 2003.