Visual Simulation of Multiple Unmixable Fluids

Wen Zheng, Jun-Hai Yong, Jean-Claude Paul

Visual Simulation of Multiple Unmixable Fluids *

Wen Zheng[1, 2], Jun-Hai Yong[1], and Jean-Claude Paul[1, 3]

1. School of Software, Tsinghua University, Beijing 100084, P. R. China
2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China
3. CNRS, Paris 75794, France

E-mail: zhen-w@mails.tsinghua.edu.cn; yongjh@tsinghua.edu.cn; paul@tsinghua.edu.cn

Abstract We present a novel grid-based method for simulating multiple unmixable fluids moving and interacting. Unlike previous methods that can only represent the interface between two fluids (usually between liquid and gas), this method can handle an arbitrary number of fluids through multiple independent level sets coupled with a constrain condition. To capture the fluid surface more accurately, we extend the particle level set method to a multi-fluid version. It shares the advantages of the particle level set method, and has the ability to track the interfaces of multiple fluids. To handle the dynamic behavior of different fluids existing together, we use a multiphase fluid formulation based on a smooth weight function.

Keywords computational fluid dynamics, natural phenomena, physically based animation

1 Introduction

Water may be one of the most familiar substances in everyday life, and it plays a very important role in realistic animation, films, and computer games. In recent years, the highly realistic simulation of fluid becomes possible, and researchers have made great progress in obtaining impressive fluid animation. However, when different unmixable liquids meet, fluids will maintain a distinct interface during mixing, and this effect is not available through current methods.

The goal of our work is to simulate the dynamic behaviors and visual effects of multiple unmixable fluids interacting. To achieve this, we propose a multi-fluid particle level set method based on the particle level set method[1, 2] for arbitrary number of fluids. We define each fluid with an independent level set, and evolve them independently. To avoid unphysical overlap and void among level sets, we adopt a correcting method proposed by Merriman[3] to explicitly constrain level sets to a valid form. Furthermore, we have extended the particle correction scheme in the original method to a multi-fluid particle correction scheme.

To handle the dynamic behaviors of multi-fluid interaction, a multiphase fluid formula-

tion[4] is employed and modified to allow arbitrary number of fluids. In our method, animators can choose one or more specific regions (such as air region) to be ignored, and add some control in such regions.

2 Previous work

Recent years, great progress has been made in computer graphics field to produce realistic animation of liquid undergoing highly dynamic motion. Foster and Metaxas[5] first introduced the 3D Navier-Stokes equations and the Marker and Cell method proposed by Harlow and Welch[6]. Stam[7] proposed a semi-Lagrangian technique to greatly improve the efficiency. Fedkiw et al.[8] used a vorticity confinement method to preserve the small-scale flow structure. Foster and Fedkiw[1] and Enright et al.[2] proposed a hybrid particle level set method that combines massless marker particles with semi-Lagrangian level set method to accurately track the water surface. Instead of simulating the air effect, they used a velocity extrapolation method to fill the velocity field in air region. Losasso et al.[9] introduced octree-based data structure to provide levels of detail in fluid simulation. Song et al.[10] adopted CIP (constrained interpolation profile)-based advection method to reduce the numerical dissipation of semi-Lagrangian technique. They used a continuous two-fluid formulation to include the air effect.

Researchers have also put much attention on simulating many different phenomena of fluid. Carlson et al.[11] simulated fluid with variable viscosity. Hong and Kim[12] simulated bubbles by VOF (Volume of Fluid) method and introduced the effect of surface tension. Takahashi et al.[13] utilized particle system coupled with CIP(Constrained Interpolation Profile)-based VOF method to add splash and foam into water simulation. Stam[14] proposed a method to simulate 2D fluid on smooth surface with arbitrary topology. Goktekin et al.[15] brought in the elastic term into Navier-Stokes equations and simulated the viscoelastic fluid. Mihalef et al.[16] used CLSVOF (Coupled Level Set and Volume of Fluid) method and the slice method to control and simulate breaking ocean waves. Greenwood and House[17] simulated small bubbles of splashing fluid using escaped particles produced by a particle level set method without effort to simulate air. To simulate two-way coupling of fluid and solid, Carlson et al.[18] proposed a “Rigid Fluid” method that treats solid objects as if they are fluid.

As our knowledge, all previous grid-based methods[1, 2, 7, 9-20] focused on fluid system with one single liquid or with one liquid and one air. The dynamic motion of arbitrary number of fluids together, especially more than two fluids, is unseen at present.

3 Method Outline

Our system consists of three modules: geometry, dynamics, and rendering. And the main program flow of our system is a cyclic process as shown in Fig.1. At every step, we first input the old velocity field into the dynamics module to calculate the new velocity in the next step. Then in the geometry module, we use this new velocity to drive the old surface to a new position. Finally, rendering module visualizes the new surface to get

one frame of the animation.

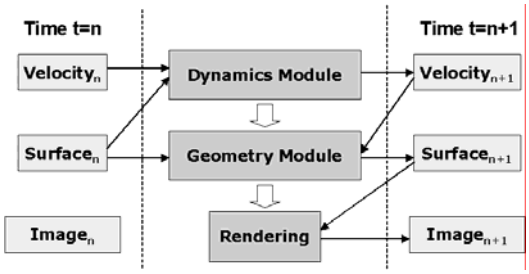


Fig.1. One loop of the program flow of our system.

In Section 4, we will describe our new surface tracking method for multi-fluid used in the geometry module. In Section 5, we will describe the formulation and numerical methods we used in the dynamics module to handle the multi-fluid dynamics in our system. In Section 6, we will give two animation results, and in Section 7, we will conclude our work.

4 Surface Tracking

The particle level set method proposed in [1, 2] provides a smooth and accurate way for two-fluid surface tracking. However, when the number of fluids is larger than two, due to the existence of triple-junctions, this method becomes insufficient. We propose here a novel surface tracking technique to solve the surface tracking problem in multi-fluid animation based on the particle level set method.

4.1 Overview of Particle Level Set Method

Particle level set method uses level set function ϕ to represent the free surface of fluid. In our implementation, we take ϕ=0 as free surface isovalue, ϕ>0 region as inside region and ϕ<0 region as outside region. Then the implicit surface is evolved by solving the advection equation,

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0, \tag{1}$$

where $\boldsymbol{u}$ is the undergoing velocity. This equation can be solved by a semi-Lagrangian method proposed by Stam[7]. After advection, a reinitialization procedure will be executed to maintain the distance function property of level set function by solving the equation,

$$\frac{\partial \phi}{\partial \tau} = -S(\phi_{\tau=0})(|\nabla \phi| - 1), \tag{2}$$

where τ is a fictitious time and $S(\phi)$ is signed function given by

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + (\Delta x)^2}}. \tag{3}$$

Meanwhile, two sets of Lagrangian particles, which are initially placed around fluid surface on both sides, will be driven by the undergoing velocity and move forward. After level set advection and reinitialization, particles on the wrong side will be marked as *escaped particle*. Then a level set value ϕ_p defined by *escaped particle* will be computed at each of the eight corners of the cell containing this *escaped particle* and correct the cell-defined level set value ϕ there.

4.2 Multi-fluid Particle Level Set Method

Now we propose the novel method suitable for arbitrary number of fluids called multi-fluid particle level set method consisting of a new representation of *N*-fluid surface and a new particle correction scheme which can maintain *N*-fluid regions. Here, *N* stands for the number of different fluids.

4.2.1 Interface Representation

To handle arbitrary number of fluids, we choose to represent each fluid with an independent

level set, say ϕ_i for fluid i, and we evolve each level set independently using Equation (1) and (2). But independent advection produces unphysical overlap (the regions where more than one fluid satisfy ϕ>0) and void (the regions where no fluids satisfy ϕ>0) as seen in Fig.2. Several constrains should be satisfied by level sets at any position $\boldsymbol{x}$:

$$\begin{cases} \phi_i(\boldsymbol{x}) \geq 0, \\ \phi_j(\boldsymbol{x}) = -\phi_i(\boldsymbol{x}), & \exists j \neq i, \\ \phi_k(\boldsymbol{x}) \leq \phi_j(\boldsymbol{x}), & \forall k \neq i, j. \end{cases} \tag{4}$$

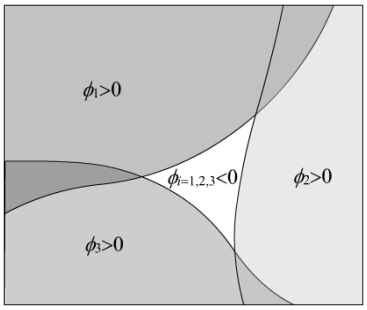


Fig.2. Overlap and void among independent level sets.

To enhance constrain (4), we adopt a simple correction method proposed in [3]. After level set advection, we replace level set value of each fluid, say fluid i, by a new level set value defined by

$$\phi_{i,new} = \frac{1}{2}\left(\phi_i - \max_{j \neq i}(\phi_j)\right). \tag{5}$$

It is noticeable that this method will slightly change the zero-level set position near triple junctions, but it is sufficient to work well in our results. If the motion of triple junctions is important, a more reasonable method is available in [19] using a symmetric projection from N level sets to an N-1 dimensional manifold without any correction.

4.2.2 Multi-fluid Particle Correction Scheme

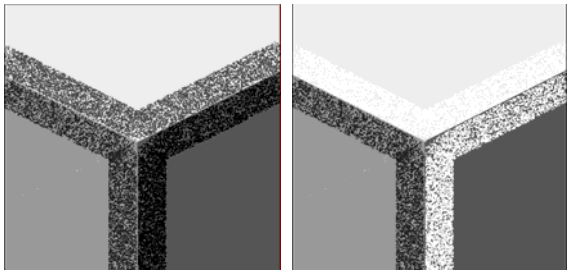
With our new interface representation, N level sets divide space into N regions, and each region represents the inside of exact one fluid. Thus it is natural to extend two sets of particles to N sets of particles. Each set of particles maintains the inside region of one corresponding fluid. We describe the extended particle level set method as follows. Only differences from the original method are presented.

Initialization of Particles. We randomly place N sets of particles near the interface. We notate the particle sets of fluid i as P_i. Then P_i will be put to the region where $3\min(\Delta x, \Delta y, \Delta z) < \phi_i < 0.1\min(\Delta x, \Delta y, \Delta z)$, and the radius $r_{p,i}$ of particle p in P_i is defined as

$$r_{p,i} = \begin{cases} r_{\max}, & if \quad \phi_i(\boldsymbol{x}_{p,i}) > r_{\max}. \\ \phi_i(\boldsymbol{x}_{p,i}), & if \quad r_{\min} \leq \phi_i(\boldsymbol{x}_{p,i}) \leq r_{\max}. \\ r_{\min}, & if \quad \phi_i(\boldsymbol{x}_{p,i}) < r_{\min}. \end{cases} \tag{6}$$

where $r_{\max}=0.5\min(\Delta x, \Delta y, \Delta z)$ and $r_{\min}=0.1\min(\Delta x, \Delta y, \Delta z)$.

Error Correction. After level set advection and reinitialization, we perform error correction procedure for each fluid independently. For fluid i, we treat P_i as *positive particles* and all other particles as *negative particles*, and then follow the original particle level set method to correct the interface of fluid i.



(a) (b)

Fig.3. Multi-fluid Particle Level Set Method. Three kinds of fluids together: light color, grey color, and dark color. (a) Particle positions. The particles lies inside the corresponding fluid. (b) Positive (dark) and Negative (white) particles for grey color fluid.

Triple Junctions Correction. This extended

method acts exactly the same as the original method in regions far from triple junctions. But it is possible to produce some overlap and void near triple junctions, so we enhance correction (5) again after particle correction.

5 Multiphase Fluid Dynamics

5.1 Formulation and Numerical Solution

To correctly handle the dynamic behavior of fluid surface between multiple fluids, we introduce a two-phase fluid dynamics formulation, which treats all fluids in a unified way, and solves them as if they are the same fluid, so that we can easily handle the discontinuity problem. The incompressible Navier-Stokes equations are:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{7}$$

and

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nu \nabla^2 \boldsymbol{u} - \frac{1}{\rho} \nabla P + \boldsymbol{g}, \tag{8}$$

where $\boldsymbol{u}$ is the velocity vector, ρ is the density, ν is the kinematic viscosity, P is the pressure, and $\boldsymbol{g}$ is the gravity vector. Here density ρ and viscosity ν will change abruptly across the fluid interface. In two-fluid simulation, to smear out the discontinuities in density and viscosity, we usually use a smooth Heaviside function to thicken the interface

$$H(\phi) = \begin{cases} 0, & if \quad \phi < -\varepsilon. \\ \frac{1}{2} + \frac{\phi}{2\varepsilon} + \frac{1}{2\pi} \sin(\frac{\pi\phi}{\varepsilon}), & if \quad |\phi| \le \varepsilon. \\ 1, & if \quad \phi > \varepsilon. \end{cases} \tag{9}$$

where ε is the "thickness" of the thickened interface. We use ε=2min(Δx, Δy, Δz) in our implementation. Then we can get the weighted density and viscosity needed in equation (8) for two-fluid simulation:

$$\begin{cases} \rho = H(\phi_1)\rho_1 + (1 - H(\phi_1))\rho_2, \\ \nu = H(\phi_1)\nu_1 + (1 - H(\phi_1))\nu_2, \end{cases} \tag{10}$$

where ρ_1 is the density of fluid 1, ν_1 is the viscosity of fluid 1, ρ_2 is the density of fluid 2, ν_2 is the viscosity of fluid 2, and ϕ_1 is the level set function of fluid 1.

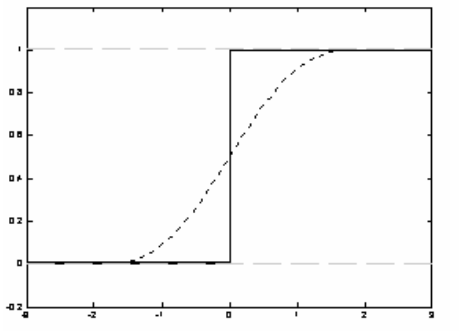


Fig.4. Thickened interface by Heaviside function. Discontinuity (solid line) is smoothed within a "thickness" of ε=2.

For N-fluid simulation, we exploit a weight function based on (9):

$$W_i = \frac{H(\phi_i)}{\sum_{j=1}^{N} H(\phi_j)}. \tag{11}$$

Then the weighted density and viscosity for N-fluid simulation is

$$\begin{cases} \rho = \sum_{i=1}^{N} W_i \rho_i, \\ \nu = \sum_{i=1}^{N} W_i \nu_i. \end{cases} \tag{12}$$

To get the numerical solution, we project Equation (8) into two steps: first directly add gravity term into velocity, compute diffusion term by an implicit iterative method and solve advection term by a semi-Lagrangian method introduced by Stam[7]; secondly, compute the pressure by solving equation (7) using PCG (Precondition Conjugate Gradient) method and add pressure term into velocity.

5.2 Air Effect

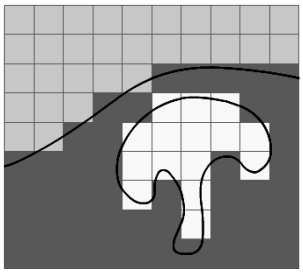


Fig.5. Inside air region and outside air region. Dark region: liquid; Grey region: outside air; Light region: inside air.

A full *N*-fluid simulation lacks controllability. Thus we provide a simple control scheme by ignoring dynamic effect from one or more specific fluids such as air, and add some user-defined wind field in that region. To achieve this, we first set the weight function of ignored fluid to zero: W_i=0 in (11). Then we extrapolate the velocity into that region using the method in [2].

Usually, the dynamic effect of the air surrounded by liquid has more importance than the outside atmosphere air. For example, the inside air can form large bubbles. Thus we further distinguish the inside and the outside region from the air region by a modified flood fill algorithm proposed by Greenwood et al.[17], and then we can apply some control to outside air only.

6 Results

We implemented our method on a PC with an Intel® Celeron®4 1.70GHz CPU, 512MB memory. The liquid surface is abstracted using Marching Cubes method[20], and rendered with the open-source renderer PIXIE[21]. The photon mapping method[22] is used to handle the indirect diffusion and caustics. The following examples (Fig.6 and Fig.7) show the ability of our method to make photorealistic multi-fluid animation.

In the example of Fig.6, we simulate two liquid walls collapse within air, and we ignore the air effect. We place a solid block in the middle to make the fluid motion more complex. The density ratio between the two liquids is 2, and the viscosity ratio is 0.2. The grid resolution used in this animation is 100×75×75, and the number of triangles generated by surface abstraction is about 30,000 to 70,000. It costs 15~30 minutes to get one frame simulated and rendered.

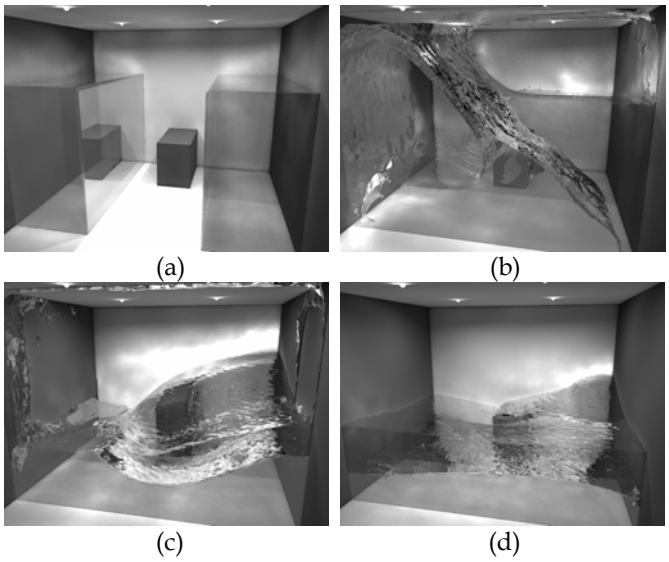


Fig.6. Animation snapshots of liquid walls collapse and crash. (a) heavy liquid and light liquid are initially set to two quiet walls; (b) (c) liquid walls crash and splash; (d) liquids settle down to form two horizontal layers.

In the example of Fig.7, we simulate solid ball smashing into two quiet liquid layers within air. The density and viscosity ratio between the two liquids is the same as in the previous example. The grid resolution is 80×75×60, and the number of triangles generated by surface abstraction is about 30,000 to 70,000. It costs 10~20 minutes to get one frame simulated and rendered.

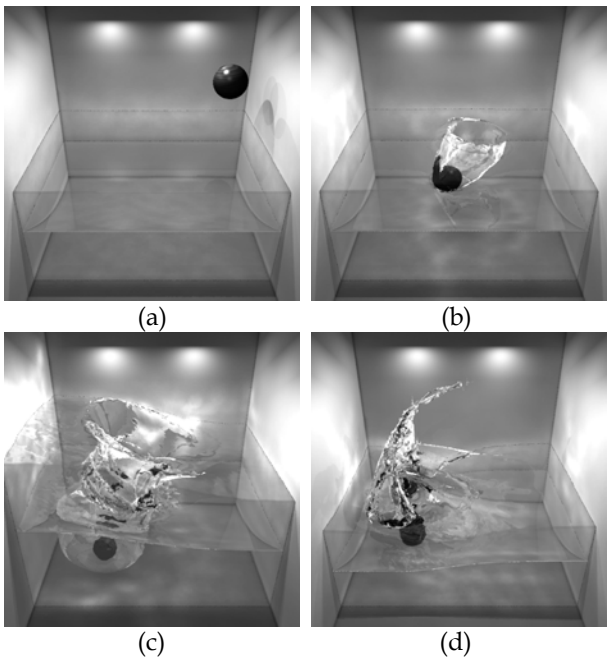


Fig.7. Animation snapshots of a solid ball falling into two layers of liquids. (a) Heavy liquid and light liquid are initially set to two quiet layers; (b) (c) solid ball falls into the liquid layers; (d) liquids splash.

7 Conclusion

We present a novel method to produce fluid animations of arbitrary number of unmixable fluids interacting together. In geometry module, we propose a novel surface tracking technique called multi-fluid particle level set method which uses independent level sets with a constrain condition to represent the interfaces of arbitrary number of fluids, and uses a new particle correction method with multiple sets of particles to maintain multiple level set functions. In dynamics module, to handle the multi-fluid dynamics, we adopt a two-phase fluid dynamics formulation and extend it to multiphase case by constructing a weighted smooth function based on the Heaviside function, so that we can easily solve the multi-fluid system in a unified formulation. Additionally, we also provide the users with some flexibility to control.

The limitation of our method is that, because we independently represent and evolve the surface of each fluid, the storage and computation cost will increase linearly as the number of fluids grows. This brings difficulty when we need to simulate hundreds of fluids together, and a further improvement will be necessary for this case.

References

[1] Foster N, Fedkiw R. Practical animation of liquids. In *Computer Graphics (Proc. ACM SIGGRAPH 2001)*, Fiume E (ed.), ACM Press / ACM SIGGRAPH, 2001, pp. 23-30.

[2] Enright D, Marschner S, Fedkiw R. Animation and rendering of complex water surfaces. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2002)*, 2002, 21(3): 736-744.

[3] Merriman B, Bence J K, Osher S J. Motion of Multiple Junctions: A Level Set Approach. *Journal of Computational Physics*, 1994, 112(2): 334-363.

[4] Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 1994, 114(1): 146–159.

[5] Foster N, Metaxas R. Realistic animation of liquids. In *Graphics Interface'96*, 1996, pp. 204-212.

[6] Harlow F H, Welch J E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *Phys. Fluids*, 1965, 8(12):

2182–2189.

[7] Stam J. Stable fluids. In *Proc. of SIGGRAPH 99*, pp. 121-128.

[8] Fedkiw R, Stam J, Jensen H. Visual simulation of smoke. In *Proc. of ACM SIGGRAPH 2001*, pp. 15-22.

[9] Losasso F, Gibou F, Fedkiw R. Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics* (*Proc. of ACM SIGGRAPH 2004)*, 2004, 23(3): 457-462.

[10] Song O, Shin H, Ko H. Stable but Non-Dissipative Water. *ACM Transactions on Graphics*, 2005, 11(2): 1-16.

[11] Carlson M, Mucha P, Van Horn III R, Turk G.. Melting and flowing. In *ACM SIGGRAPH Symposium 2002 on Computer Animation*, pp. 167-174.

[12] Hong J M, Kim C H. Animation of bubbles in liquid. *Computer Graphics Forum (Proc. Of Eurographics 2003)*, 2003, 22(3): 253-262.

[13] Takahashi T, Fujii H, Kunimatsu A, et al. Realistic Animation of Fluid with Splash and Foam. In *Computer Graphics Forum (Proc. of EUROGRAPHICS 2003)*, 2003, 22(3): 391-400.

[14] Stam J. Flows on Surfaces of Arbitrary Topology. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2003)*, 2003, 22(3): 724-731.

[15] Goktekin T G, Bargteil A W, O'Brien J. A Method for Animating Viscoelastic Fluids. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)*, 2004, 23(3): 463-467.

[16] Mihalef V, Metaxas D, Sussman M. Animation and Control of Breaking Waves. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2004*, pp. 315-324.

[17] Greenwood S T, House D H. Better with Bubbles: Enhancing the Visual Realism of Simulated Fluid. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2004*, pp. 287-296.

[18] Carlson M, Mucha P J, Turk G.. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)*, 2004, 23(3): 377-384.

[19] Smith K, Solisy F, Chopp D. A Projection Method for Motion of Triple Junctions by Level Sets. *Interfaces and Free Boundaries*, 2002, 4(3): 263-276.

[20] Lorensen W E, Cline H E. Marching Cubes: A high resolution 3D surface construction algorithm. In *International Conference on Computer Graphics and Interactive Techniques*, 1987, pp. 163-169.

[21] PIXIE. http://sourceforge.net/projects/pixie.

[22] Jensen H W. A Practical Guide to Global Illumination using Ray Tracing and Photon Mapping. *SIGGRAPH 2002*, Course 19.