Modeling Highly-Deformable Liquid

Chih-Wei Chiu Jung-Hong Chuang* Cheng-Chung Lin Jin-Bey Yu
Department of Computer Science and Information Engineering
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu, Taiwan 30050, ROC.

Abstract

Modeling liquid with complex surfaces is a great challenge in computer graphics. We proposed a hybrid approach combining the volume tracking method and smoothed particle hydrodynamics (SPH) to model liquids with highly deformable surfaces. The 3-D Navier-Stokes equations are solved by the marker-and-cell (MAC) method and a density volume representing the liquid is evolved over time and space by the volume-of-fluid (VOF) method. The VOF method uses a finite-volume discretization and maintaining a value for the liquid volume in each grid cell. The advection of the volumes is calculated by using cell-face fluxes, in which the liquid volume that leaves one cell is exactly the same as the liquid volume that enters the adjacent cells. Only a scalar convective equation needs to be solved to evolve the liquid surface forward in time and space. Potentially under-resolved effects are captured by SPH particles generated in that high-deformable regions near the liquid surface. Dynamics of these particles provides a better fieldity than that of other particle methods due to their capability of approximating the equations of fluid mechanics. Particle volumes are incorporated into the interpolated volume fractions before an unified iso-surface is extracted. As a result the liquid surface is more smooth than those of previous researches which rendered the splash by coating the particles with a field functions or as hard spheres.

Keywords: Liquid simulation, Fluid dynamics, Splashing, Particle dynamics.

*Contact author. E-mail: jhchuang@csie.nctu.edu.tw
Phone: 886-3-5731829. Fax: 886-3-5724176

1 Introduction

Producing realistic liquid animation is a great challenge in compute graphics due to the complex dynamics of flows. Various approaches have been attempted. Early graphics work concentrated on modeling just the surface of a body of water as a parametric function that could be animated over time to simulate wave transport. Recently, many researches have simulated various natural objects and phenomena by using a set of particles each of which has simple behavioural rules. Construction of the behavioural model is based on empirical and intuitive knowledge wihtout taking a standard apparoch to numerical simulation.

While a large volume could potentially be modeled with particles, the number of particles needed to fill a volume will grow with the cube of the resolution of the scene. Moreover, despite the simplified equations for the dynamics this model can be computationally expensive for a large volume of particles.

On the other end of the spectrum lies the Eulerian volume tracking method, which has already enjoyed great success in the computational fluid dynamics, but not yet widely used in the computer graphics community. An Eulerian method is characterized by a stationary grid partitioning the whole computational domain, whileas using particles is a Lagrangian method characterizing by a "moving" grid, which uses the particles to be the grid nodes. In the volume tracking method, the scene is partitioned by a Cartesian grid (regular or irregular). The initial fluid configuration is converted to a scalar field in the grid. A scalar value between zero and unity, known as the "volume fraction," indicates the proportion of liquid and air in the cell. Exact interface information is then discarded in favor of the discrete volume fraction data. Interfaces are subsequently tracked by evolving fluid volumes in time with the solution

of a standard convection equation.

The major negative aspect of particle methods is the cost in terms of CPU time and storage. To update a particle's position, the intergration of the sum of the forces acting on this particle by others needs be to calculated. Using volume fractions is more economical than particles in both space and time, as only one value (the volume fraction) needs to be stored for each grid cell. Only a scalar convective equation needs to be solved to propagate the volume fractions through the computational domain.

Another drawback of particles method is that there is no straightforward way to extract a smooth surface from the particles, since deducing the connectivity of particles is difficult or impossible. The volume data results from the volume tracking method can be easily visualized by employing an iso-surface extraction algorithm.

In complex flows, a liquid surface can be stretched and torn in a dynamics fashion, the use of only an initial seeding of particles will not capture these effects well, since regions will form that lack a sufficient number of particles to represent the fluid bodies. This would require adding extra particles when the surface becomes too sparsely resolved, and removing them as the surface folds, or splashes back over itself. Another advantage of an Eulerian volume tracking method over a Lagrangian particles method is the ability to withstand severe topological change robustly, since no topological or connectivity information is retained and the arbitrary complex interface can be inferred from the volume fractions. This enables the modeling of interfaces which are exposed to large deformations. The merging or breakup of the interface is handled in a natural way.

Vorticity driven stretching, tearing and splitting near the surface typifies the topological changes that must be captured. These surface features may be under-resolved for volume tracking methods with a coarse computational grid. While volume tracking is a fully Eulerian approach, the ability of volume tracking is restricted by the resolution computational grid. Particles evolution is a fully Lagrangian approach that has strength of resolving very fine effects, such as small droplets and water splash. Chief among the strengths of particles method is the ability to continue to advect bodies faithfully even when the body can not be supported on a grid.

Since they tend to have complementary strengths and weakness, a combined approach gives superior results under a wider variety of situations [18] [16] [5] [4]. We propose a method incorporating the volume tracking and smoothed particle dynamics to model complex fluid motion. Fluid surface is tracked by volume fractions and evolves over space and time. Particles are only generated in the area where the fluid motion is drastic, for example, when an object impacts the fluid surface. These particles move freely until they run into the volume fractions and are absorbed. The motions of particles are governed by smoothed particle hydrodynamics (SPH) [12], which are used by physicists for cosmological fluid simulation. Smoothed particles represent sample points that enable the approximation of the values and derivatives of local physical quantities inside a medium. SPH is favored than particle systems [20], which do not model inter-particle forces, and molecular dynamics [6], which does not model the coupling of pressure and velocity characterizing a real fluid.

The paper is organized as follows. In Section 2, we review the previous work on modeling fluid from three aspects: the dynamics model, the geometry model, and the illumination model. Our method is outlined in Section 3, with detailed description about a numerical method to solve the 3-D Navier stokes equations; the surface kinematic condition; the necessary modification of SPH to model incompressible flow; and the rendering of our geometry model. Section 4 gives the result and discussion. A conclusion and future work are presented in Section 5.

2 Modeling Highly-Deformable Liquid

2.1 Overview

Figure 1 illustrates the computation flow in a timestep. Initially the physical domain is partitioned by a computation grid, and this partition is only performed once before the simulation begins. The fluid dynamic solver then calculates the velocities and pressure for the next time step. The volume fractions in the current time step and new obtained velocities are used to convect the fluid volume fractions. In the volume tracking method, the accuracy of fluid interface depends upon the resolution of the computational grid. Some surface detailed features may be under-resolved if the grid is too coarse. Particles are introduced in the area where the fluid surface is potentially exposed to large deformations.

Dynamics of particles is governed by smoothed particle hydrodynamics (SPH).

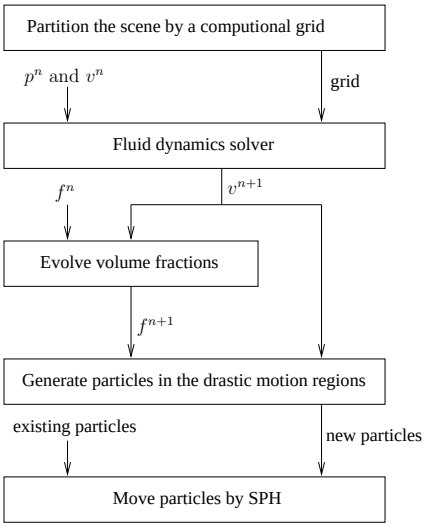


Figure 1: Simulation flowchart.

2.2 Solving Fluid Dynamics

In general, we would be interested in fluid properties that arise in the expression of the laws of conservation of mass, momentum, energy, etc. The transportation equation for the conservation of mass is

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{v} = 0. \tag{1}$$

This relationship, also known as the continuity equation, states that the mass of a fluid in a close domain can only be changed by flow across the boundaries.

The transport equation for the conservation of momentum is

$$\frac{D\mathbf{v}}{Dt} + \frac{1}{\rho}\nabla p - \frac{1}{\mathfrak{Re}}\nabla^2\mathbf{v} - \mathbf{g} = 0. \tag{2}$$

The only external force acting on fluid volume considered here is $\mathbf{g}$, the force due to gravity.

For a constant density and incompressible fluid, without loss of generality, ρ is assumed to be unity, and (1) and (2) are rewritten in the following non-conservation form

$$\nabla \cdot \mathbf{v} = 0, \tag{3}$$

$$\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} + \nabla p - \frac{1}{\mathfrak{Re}}\nabla^2\mathbf{v} - \mathbf{g} = 0. \tag{4}$$

The restriction to incompressible flow introduces the computational difficulty that (3) contains only velocity components, and there is no explicit link with the pressure as there is for compressible flow through density. One of the earliest and most widely used methods for solving (3) and (4) is the marker and cell (MAC) method [8]. The method is characterized by the use of a staggered grid and the solution of a Poisson equation for the pressure at every time step. Although the original form of the MAC method has certain weakness, the use of a staggered grid and a Poisson equation for the pressure has been refined in many modern methods derived from MAC.

To track the free surface, the initial (known) fluid interface geometry is used to compute fluid volume fractions in each computational cell. A scalar indicator function between zero and one, known as the volume fraction, f, is used to distinguish between two different fluids, liquid and air. A value of zero indicates the absence of liquid and a value of unity corresponds to a cell full of liquid. On a computational grid, volume fraction values between these two limits indicates the presence of the interface and the value itself gives an indication of the relative proportion of liquid and air occupying the cell volume. Exact interface information is then discarded in favor of the discrete volume fraction data. Interfaces are subsequently tracked by evolving fluid volumes in time with the solution of a standard convection equation.

Special care must be taken to approximate the convection equation to keep the interface compact, since using even the most well designed finite difference methods to advect interface results in an interface thickness of at least two to three cells in width. We employ the improved donor-acceptor scheme [9] that satisfies local boundedness while keeping the transitional area between the liquid and air restricted to one cell width. The consistent discretization of the continuity and momentum equations with respect to the discretization of the volume fraction equation is used to ensure a solution algorithm which satisfies the conservation of the flow properties at all times.

2.3 Modeling Splash

To compensate the motion under-resolved by the resolution of the Eulerian grid, we place Lagrangian particles near the region in the flow field having drastic motion. Section 2.3.1 describes how these regions are identified and how particles are positioned inside the cell's "cut-volume", defined

by the volume cut by a 3-D plane approximating the liquid interface in the cell.

Once generated, the particles may escape the liquid surface and their dynamics are governed by SPH. SPH is favored than ordinary particle systems since the former can approximate the Navier-Stokes equations and enables particles to interact with one another in the range of the kernel, thus providing a more realistic simulation. Section 2.3.2 describes some extensions that are necessary for SPH to be used in the computer grapihcs.

2.3.1 Particle Seeding

An obvious way to place particles is to put them in the surface cells, whose volume fractions are great than zero but less than unity. These cells are where the grid potentially under-resolve the interface. But not all surface cells have violent changes, particles in these cells do not resolve the interface much better than the volume fractions. To save computation time, particles are generated only in the surface cells meeting any one of following conditions:

$$u\delta t > \gamma\delta x, \quad v\delta t > \gamma\delta y, \quad w\delta t > \gamma\delta z. \qquad (5)$$

where δx, δy, and δz are cell sizes, δt is the time step, u, u, w are velocity components, and $\gamma \in (0, 1)$ is the convection threshold. Inequality (5) can be related to the CFL (Courant-Friedrichs-Lewy) condition, which physically states a particle of fluid should not travel more than one spatial step-size δx in one time step to keep the numerical method stable. A high γ values tends to produce less particles than a lower one.

Each smoothed particle initially carries a fixed mass and will never split or merge with other particles. The total volume of particles in a cell should not excced the volume of liquid in the cell indicated by the volume fraction. The mass carried by a particle can be arbitrarily defined. The less the mass, the more particles are necessary to approximate a same volume of liquid in the cell.

There is another restriction on the maximum number of particles in a cell. If the number of particles in the cell have exceeded a predefined maximum, particles will not be seeded, since the existing particles should provide accurate approximation of the interface and in this way we prevent from potentially introducing excessive particles to the SPH simultion.

To seed particles in the a surface cell, the interface in the cell must be reconstructed, a process similar to what we do in volume fraction convection. Particles will then be randomly placed below the interface. Initial velocities of particles are linearly interpolated from the velocities defined on the cell faces according to their positions, as shown in Figure 2. Conversely, particles entering the area below the interface are removed from the system.

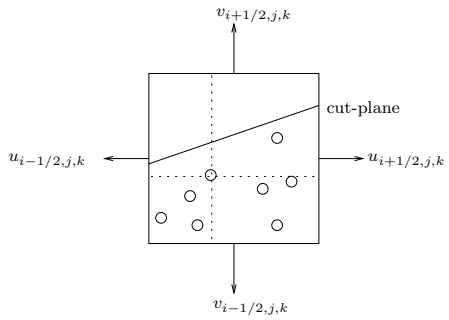


Figure 2: Interpolate particle velocity from the cell velocity.

When a piecewise constant/"stair-stepped" approximation is used in convection volume fractions, the interface is inadequate for seeding particles. Because reconstructed piecewise constant interface is parallel to one of the axes, particles positioned below these axis-aligned planes will form a regular pattern, which is not natural in a realistic animation. We approximate the interface by a Piecewise Linear Interface Calculation (PLIC) method [22]. Note that the PLIC method is not used for convecting volume fractions for efficiency, while a piecewise constant/stair-stepped approximation is accurate enough for simulation.

In a PLIC method, the interface between materials is approximated in each computational cell with a linear interface defined by the equation

$$\mathbf{n} \cdot \mathbf{x} = \lambda, \qquad (6)$$

where $\mathbf{n}$ is the normal of the interface plane and λ is a scalar. Any cell having volume fractions f between zero and one will possess an interface defined by (6). We choose $\mathbf{n}$ to point outward the fluid, hence (6) will be positive for any point $\mathbf{x}$ lying within the fluid, zero for any point $\mathbf{x}$ lying on the line, and negative for any point $\mathbf{x}$ lying outside of the fluid.

Determining λ is the most difficult reconstruction task because the value of λ is constrained by mass conservation. In other words, the value of λ is constrained such that the resulting plane passes through the cell with a truncation volume equal

to the cell material volume $f_{i,j,k}$. This determination requires inverting a $V(\lambda)$ relation. However, the determination of $\mathbf{n}$ is not constrained by volume conservation consideration. We follow the approach in [22] by using finite difference stencils, yielding an explicit expression of $\mathbf{n}$. The three components of the normal in the cell (i,j,k) is given by,

$$\begin{aligned} n_x &= f_{i+1/2,j,k} - f_{i-1/2,j,k} \\ n_y &= f_{i,j+1/2,k} - f_{i,j-1/2,k} \\ n_z &= f_{i,j,k+1/2} - f_{i,j,k-1/2} \end{aligned}$$

This method is shown to be first-order accurate, as measured with the least square criteria.

We distinguish the "forward problem", that is, to find the volume fraction $V_f = f_{i,j,k}$ occupied by one material given λ, from the "inverse problem", which consists of finding λ given the volume fraction. The problem is essentially geometric in nature. The relation $\lambda = V^{-1}(V_f)$ is nonlinear and varies in each mixed cell, due to its dependence upon local data. A case-by-case implementation results that is not efficient, general, concise, or easily to maintained and understood. Therefore we employ an analytical expression connecting the volume fraction f and λ in this thesis.

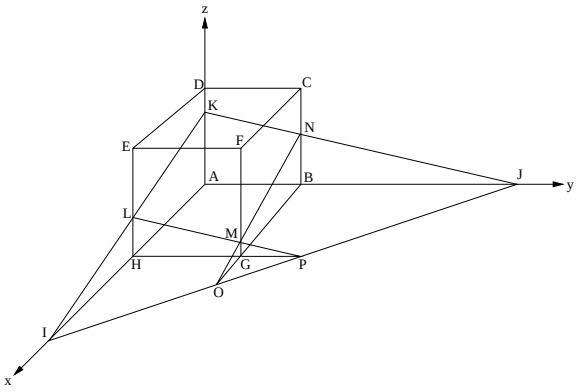


Figure 3: The "cut volume" is the region inside the parallelepiped ABCEFGH and below the plane IJK

We assume that the three components of $\mathbf{n}$ are all positive and we need to determine the cut volume ABGHKNML of the rectangular cell which is also below the given plane, as shown in Figure 3. [7] have shown that the volume ABGHKNML is given by the expression

$$\begin{aligned} V = \frac{1}{6n_xn_yn_z}\,[\,\lambda^3 &- \sum_{i\in\{x,y,z\}} \Psi(\lambda - n_i\delta i) \\ &+ \sum_{i\in\{x,y,z\}} \Psi(\lambda - \lambda_{max} + n_i\delta i)\,], \end{aligned} \tag{7}$$

where the function $\Psi(y)$ is defined as

$$\Psi(y) = \begin{cases} y^3, & \text{for } y > 0, \\ 0, & \text{otherwise,} \end{cases}$$

and $\lambda_{max} = n_x + n_y + n_z$

For the moment, we restrict our analysis to a unity cube whose $\delta x = \delta y = \delta z = 1$; then volume V and volume fraction V_f coincide. We also normalize the plane equation (6) by dividing it by $(n_x + n_y + n_z)$; then $\lambda_{max} = 1$. Note that expression for V is invariant with respect to a permutation of the indices, so we need to consider only one case, say $n_x \le n_y \le n_z$. The graph of V has odd symmetry with respect to the point $(V,\lambda) = (1/2, 1/2)$, so we restrict the analysis to the range $0 \le \lambda \le 1/2$.

Let $n_{xy} = n_x + n_y$, and $n = min(n_{xy}, n_z)$. The range of λ, $[0, 1/2]$, is partitioned into four subintervals $[0, V_1), [V_1, V_2), [V_2, V_3)$, and $[V_3, 1/2])$, where

$$\begin{aligned} V_1 &= \tfrac{n_x^2}{\max(6, n_y n_z, \epsilon)}, \\ V_2 &= V_1 + \tfrac{n_y - n_x}{2n_z}, \\ V_3 &= \min(V_{31}, V_{32}), \end{aligned}$$

where ϵ is an arbitrary small number and

$$\begin{aligned} V_{31} &= \tfrac{n_z^2(3n_{xy} - n_z) + n_x^2(n_x - 3n_z) + n_y^2(n_y - 3n_z)}{6n_xn_yn_z}, && \text{if } n = n_z, \\ V_{32} &= \tfrac{n_{xy}}{2n_z}, && \text{if } n = n_{xy}. \end{aligned}$$

The inverse problem is given by [21] as follows:

$$\begin{aligned} \lambda &= \sqrt[3]{6n_xn_yn_zV}, && \text{for } 0 \le V < V_1, \\ \lambda &= \tfrac{1}{2}\left(n_x + \sqrt{n_x^2 + 8n_yn_z(V - V_1)}\right), && \text{for } V_1 \le V < V_2, \\ P_1(\lambda) &= a_3'\lambda^3 + a_2'\lambda^2 + a_1'\lambda + a_0' = 0, && \text{for } V_2 \le V < V_3, \end{aligned}$$

and for $V_3 \le V \le 1/2$, we have

$$\begin{aligned} P_2(\lambda) &= a_3''\lambda^3 + a_2''\lambda^2 + a_1''\lambda + a_0'' = 0, && V_{31} \le V_{32} \\ \lambda &= n_zV + \tfrac{n_{xy}}{2}, && V_{32} \le V_{31} \end{aligned}$$

For coefficients of the two cubic polynomials we have

$$\begin{aligned} &a_3' = -1,\ a_2' = 3n_{xy},\ a_1' = -3(n_x^2 + n_y^2), \\ &a_0' = n_x^3 + n_y^3 - 6n_xn_yn_zV, \\ &a_3'' = -2,\ a_2'' = 3,\ a_1'' = -3(n_x^2 + n_y^2 + n_z^2), \\ &a_0'' = n_x^3 + n_y^3 + n_z^3 - 6n_xn_yn_zV. \end{aligned}$$

In previous relations V_1 is an approximation of the value $n_x^2/6n_yn_z$. This approximation is needed because the limit for V_1, as $n_x, n_y \to 0$, is zero; however numerically we must prevent the denominator of V_1 from becoming zero.

2.3.2 Particles Dynamics

After the initial positions and velocities of smoothed particles are specified, their trajectories and acceleration are governed by SPH. Next we briefly describe SPH.

Smoothed Particle Hydrodynamics One reason that the use of smoothed particle hydrodynamics (SPH) [12] over other numerical methods such as finite difference method (FDM) is that SPH does not depend on the boundary conditions. The FDM is based on breaking up the computational region into a grid. The reason for creating a grid is that in order to calculate the spatial derivative of the field variables, one must divide the difference of the filed variables at two points by the distance between the two points.

In SPH, the fluid is sampled by a set of elements called *particles* (which may also be regarded as interpolation points). A particle j has a fixed mass m_j, a position $\mathbf{r}_j$, a velocity $\mathbf{v}_j$, and a density ρ_j depending on the local density of particles. As a sample point, it can also carry physical field values like pressure or temperature. Then in a way very similar to Monte-Carlo techniques, these fields and their derivatives can be approximated by a discrete sum. At the heart of SPH is an interpolation method which allows any function to be expressed in terms of its values at a set of disordered points.

The integral interpolant of any function $\varphi(\mathbf{r})$ is defined by

$$\varphi(\mathbf{r}) = \int_\Omega \varphi(\mathbf{r}')W_h(\mathbf{r}-\mathbf{r}',h)\mathbf{dr}',$$

and W_h is an interpolating kernel which has the two properties

$$\int_\Omega W_h(\mathbf{r}-\mathbf{r}',h)\mathbf{dr}' = 1,$$

and

$$\lim_{h\to 0} W_h(\mathbf{r}-\mathbf{r}',h) = \delta(\mathbf{r}-\mathbf{r}'),$$

where the limit is to be interpreted as the limit of the corresponding integral interpolants.

This normalized kernel gives the spatial mass distribution profile over a smoothing length h. Then the smoothed values and derivatives of a continuous field φ known only at particle locations can be approximated by

$$< \varphi(\mathbf{r}) > \quad = \quad \sum_j m_j \frac{\varphi_j}{\rho_j} W_h(\mathbf{r}-\mathbf{r}_j), \tag{8}$$

$$< \nabla_\mathbf{r}\varphi(\mathbf{r}) > \quad = \quad \sum_j m_j \frac{\varphi_j}{\rho_j} \nabla W_h(\mathbf{r}-\mathbf{r}_j). \tag{9}$$

The essential point is that we can construct a differentiable interpolant of a function from its values at particles (interpolation points) by using a kernel which is differentiable. Derivatives of this interpolant can be obtained by ordinary differentiation; there is no need to use finite difference and no need for a computational grid. This is one of the main reasons SPH is so popular. It removes the need for a mesh to calculate spatial derivatives. With equation (8) and (9), we are ready to write down our numerical equations for the smoothed values of each point. From here on, we will remove the brackets and place a subscript i on the value. This implies that the value being calculated is the smoothed value for particle i. The rewritten version of our two SPH equations are then:

$$< \varphi(\mathbf{r}) > \quad \equiv \quad \varphi_i = \sum_j m_j \frac{\varphi_j}{\rho_j} W_{ij},$$

$$< \nabla_\mathbf{r}\varphi(\mathbf{r}) > \quad \equiv \quad \nabla_i\varphi_i = \sum_j m_j \frac{\varphi_j}{\rho_j} \nabla_i W_{ij},$$

where $W_{ij} = W_h(\mathbf{r}_i - \mathbf{r}_j, h)$ and ∇_i implies the spatial derivative with respect to particle i's coordinates. For example the density at particle i may be evaluated using

$$\rho_i = \sum_{j\neq i} m_j W_{ij}. \tag{10}$$

Writing the continuity (1) in the form

$$\frac{D\rho}{Dt} = -\nabla\cdot(\rho\mathbf{v}) + \mathbf{v}\cdot\nabla\rho,$$

and using SPH particle interpolants for the RHS, the rate of change of the density of particle i becomes

$$\frac{d\rho_i}{dt} = \sum_{j\neq i} m_j \mathbf{v}_{ij}\cdot\nabla_{ij} W_{ij},$$

where $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$.

The momentum equation for particle i becomes

$$\frac{d\mathbf{v}_i}{dt} = -\sum_{j\neq i} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij}\right)\nabla_{ij} W_{ij} + \mathbf{g}, \tag{11}$$

where Π_{ij} is the viscous term given by

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c\mu_{ij} + \beta\mu_{ij}^2}{\overline{\rho}_{ij}}, & \mathbf{v}_{ij}\cdot\mathbf{r}_{ij} < 0; \\ 0, & \text{otherwise} \end{cases}$$

and

$$\mu_{ij} = \frac{h\mathbf{v}_{ij}\cdot\mathbf{r}_{ij}}{\mathbf{r}_{ij} + (0.01h)^2}.$$

In these expressions the notation $\overline{\rho}_{ij} = (\rho_i + \rho_j)/2$ and $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ has been used, and c is the speed of sound. Because of its symmetry the viscous term conserves linear and angular momentum. The first term involving α is analogous to a shear and bulk viscosity. The second one, comparable to the von Neumann-Richtmyer artificial viscosity used in grid-based method, prevents particle interpenetration at high speed.

Extension to SPH If (10) is used for incompressible fluids like water, where the density falls discontinuously to zero at the surface, the density, however, will be smoothed over the length $2h$ and surface particles will have a low density [13]. The equation of state will then introduce incorrect pressures and degrade the calculation. It is therefore preferable to depart from normal practice and approximate the rate of change of the density. All particles are then assigned the same initial density which only changes when particles are in relative motion.

There are at least two ways that SPH might be extended to incompressible flow. The first is to work directly with the constraint of constant density. It's possible to include these constraints easily in the SPH formulation by using the Gibbs–Appel equation [19] which are generalized versions of Gauss' principle of least constraint. Unfortunately, the resulting equations are cumbersome, and it has not been possible to solve them efficiently without further approximation.

The second approach [2] is based on the observation that real fluid such as water are compressible, but with a speed of sound which is very much greater than the speed of bulk flow. The equation of state has the form

$$p = p_0 \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right] \tag{12}$$

with $\gamma = 7$, and a zero subscript denotes reference quantities. The choice of $\gamma = 7$ in (12) causes pressure to respond strongly to variations in density. Thus, perturbations to the density field remain small. However, as the density fluctuations increase, small errors in density correspond to increasingly larger errors in pressure [15].

In previous work involving incompressible fluids, the subtraction of 1 in (12) was introduced to remove spurious boundary effects at a free surface. It is well established that SPH is unstable when attractive forces act between particles [1]. Consequently, for the simulation presented in thesis, we use

$$p = c^2 \rho, \tag{13}$$

as suggested by [15].

The speed of sound must be chosen carefully to ensure an efficient and accurate solution of a given problem. The value of c must be large enough that the behavior of the corresponding quasi-incompressible fluid is sufficiently close to that of the real fluid, yet it should not be so large as to make the time step prohibitively small. One way to find appropriate c is to balance the force in Navier-Stokes momentum equation (2) to come up with several relations to which the the speed of sound is in proportion[15]. The speed of sound should be comparable with the largest of

$$\frac{\mathrm{V}^{*2}}{\delta}, \frac{\mathrm{V}^{*}}{\mathfrak{Re}\mathrm{L}^{*}\delta}, \frac{\mathbf{g}\mathrm{L}^{*}}{\delta}, \tag{14}$$

where $\delta = \Delta\rho/\rho_0$. The first term in (14) corresponds to that derived by [13]. The second and third terms ensure that pressure forces are comparable with viscous and body forces, respectively.

The expression of pressure in (13) resulted in positive, i.e., purely repulsive, forces expressing the natural expansion of the fluid. However, we would like to animate material with constant density at rest. Consequently, the material should exhibit some internal cohesion, resulting in attraction-repulsion forces as in the Lennard-Jones model. To keep density at ρ_0, we replace the equation of state by

$$p = k(\rho - \rho_0). \tag{15}$$

(15) designed to maintain density close to a constant value, has a double advantage [3]. First, if particles have the same mass they will tend to be evenly distributed inside the object. This is essential since we are using them as sample points for approximating continuous function. Moreover, constant density results in a constant volume. The material will then tend to naturally come back to is initial volume after a compression.

Replace p by its value in (11) leads to

$$\frac{d\mathbf{v}_i}{dt} = -\sum_{j \neq i} \left(k m_j \frac{\rho_i - \rho_0}{\rho_i^2} + m_j \frac{\rho_i - \rho_0}{\rho_j^2} + \Pi_{ij} \right) \nabla_{ij} W_{ij} + \mathbf{g}. \tag{16}$$

The first term in parentheses of (16) is a density gradient descent, that tends to minimize the difference between current and desired densities. The second is a symmetry term that ensures the action-reaction principle. The parameter k determines the strength of the density recovery. Its plays the same role as a stiffness parameter in a standard molecular dynamics. A larger k will simulate a stiff material while a smaller k models a soft one.

After the acceleration of the particles is obtained from (16), the velocities and positions are updated by the Leapforg scheme:

$$\begin{aligned} \mathbf{v}_i^{n} &= \mathbf{v}_i^{n-\frac{1}{2}} + \frac{\delta t}{2} \frac{d\mathbf{v}_i^{n}}{dt}, \\ \mathbf{v}_i^{n+\frac{1}{2}} &= \mathbf{v}_i^{n-\frac{1}{2}} + \delta t \frac{d\mathbf{v}_i^{n}}{dt}, \\ \mathbf{r}_i^{n+\frac{1}{2}} &= \mathbf{r}_i^{n} + \delta t \mathbf{v}_i^{n+\frac{1}{2}}. \end{aligned}$$

The Kernel The use of different kernels in SPH is analogue to the use of different difference schemes in finite difference method. The advantage of SPH is that the kernel can be calculated in a subroutine, or a table, and it is then trivial to change a code with one kernel into a code with another. The kernel based on spline functions [14]

$$W(\mathbf{r},h) = \frac{\sigma}{h^\nu}\begin{cases} 1-\frac{3}{2}s^2+\frac{3}{4}s^2, & \text{if} \quad 0 \le s \le 1, \\ \frac{1}{4}(2-s)^3, & \text{if} \quad 1 \le s \le 2, \\ 0, & \text{otherwise} \end{cases}$$

where $s = r/h$, ν is the number of dimensions and σ is a normalization constant with the values $\frac{2}{3}$, $\frac{10}{7\pi}$, $\frac{1}{\pi}$ for one, two and three dimensions respectively, has advantages. Thos kernel has compact support; the second derivative is continuous, and the dominant error term in the integral interpolant is $O(h^2)$.

2.4 Rendering

The liquid in the simulation is represented by volume fractions and smoothed particles. Although the volume fractions can be easily rendered by extracting an iso-surface from the density volume, the inability of representing density values less than the iso-value will prevent the animation sequence from preserving a constant volume [11]. Particles, however, when serving as the point skeletons of an implicit surface, can not form a smooth surface unless the number is very large and evenly distributed.

To produce the smooth surface of realistic liquid, another volume fractions of higher resolution (the "fine" density volume) are interpolated from the original volume fractions (the "coarse" density volume), and the volumes of smoothed particles are converted to volume fractions of the occupying cells in the fine density volume. In the cells where both volume fractions and smoothed particles exist, the converted volume fractions of smoothed particles always predominate under the assumption that particles should give accurate values in the regions with drastic fluid motion, since they tracks the fluid surface independent of the resolution of the computational grid. Smoothed particles are eliminated when they fall back into the liquid volume or outside the simulation domain. At last, an iso-surface extraction alogrithm (Marching Cubes here) is employed to extract a smooth surface. While the volume fractions are interpolated when rendering the liquid surface, they are not used in the future simulation process. Figure 4 illustrates the rendering procedure.

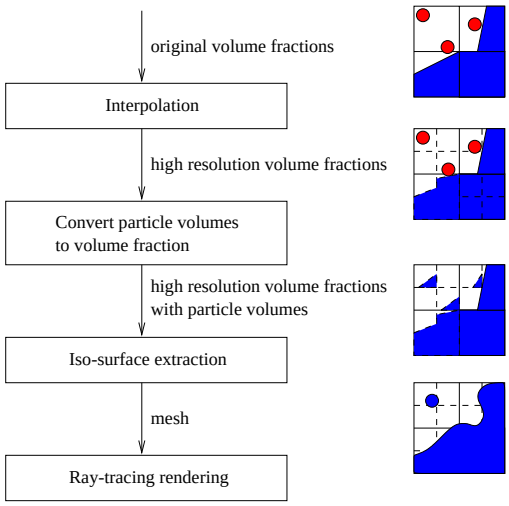


Figure 4: Rendering flowchart.

3 Results

All images in this thesis are rendered by a ray-tracer at the resolution of 640×480 pixels. All computation and rendering are performed on a PC armed with an AMD 1000 MHz processor and 512 MB RAM.

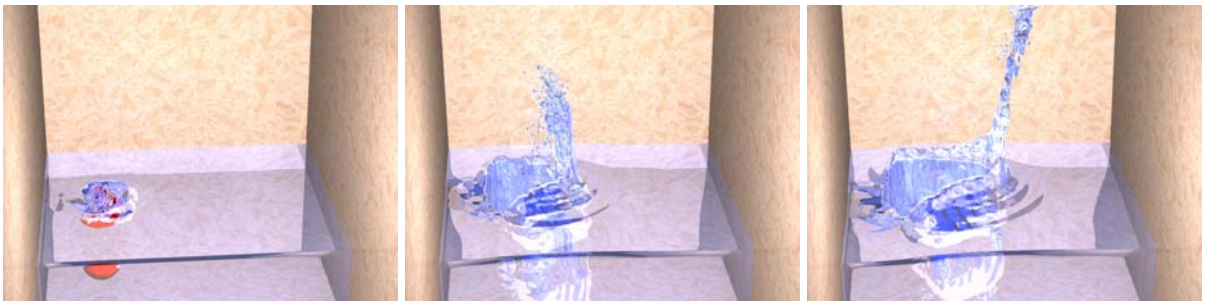
Figure 5 shows an orange liquid is discharge from a pipe at the speed of 1.8 into a container of dimension $6 \times 5 \times 5$ and of resolution $30 \times 24 \times 24$. Figure (a)-(c) shows the frame at roughly the 42th, 92th and 124th minute of the simulation, and takes 12, 10, and 8 minutes to render, respectively.

Figure 6 (a)-(c) show a sequence of selected frames of throwing a sphere into a tank of water at the 79th, 118th, and 138th minute of the simulation, respectively. The sphere has a initial horizontal 1.1 and vertical velocity of -0.8. The dimension of the computational grid is 8 in width and height and 12 in depth. The computation time depends upon the complexity of the scene and varies substantially throughout the simulation. It takes a long time to reach the second frame, when the sphere just hits the water surface, due to many iterations need to solve the pressure Poisson equation at each cycle. The computation is quick after the water surface is getting calm. For example, only 20 minutes are need to evolve from 12th frame to 24th frame. It costs about 10 minutes to render a frame.

Figure 7 shows a sequence of selected frames of throwing a cube. The cube has a initial horizontal

(a) Frame 16 (b) Frame 32 (c) Frame 48

Figure 5: Liquid discharged from a pipe ($30 \times 24 \times 24$ gridl cells)



(a) Frame 2 (b) Frame 12 (c) Frame 24

Figure 6: Selected frames of throwing a shpere into a tank of water ($32 \times 32 \times 48$ grid cells)

and vertical velocity of -2.5. The dimension of the computational grid is 7 in width and height and 6 in depth. The computation time for each frame is 30 seconds in average. The average rendering time for a frame is three minutes.

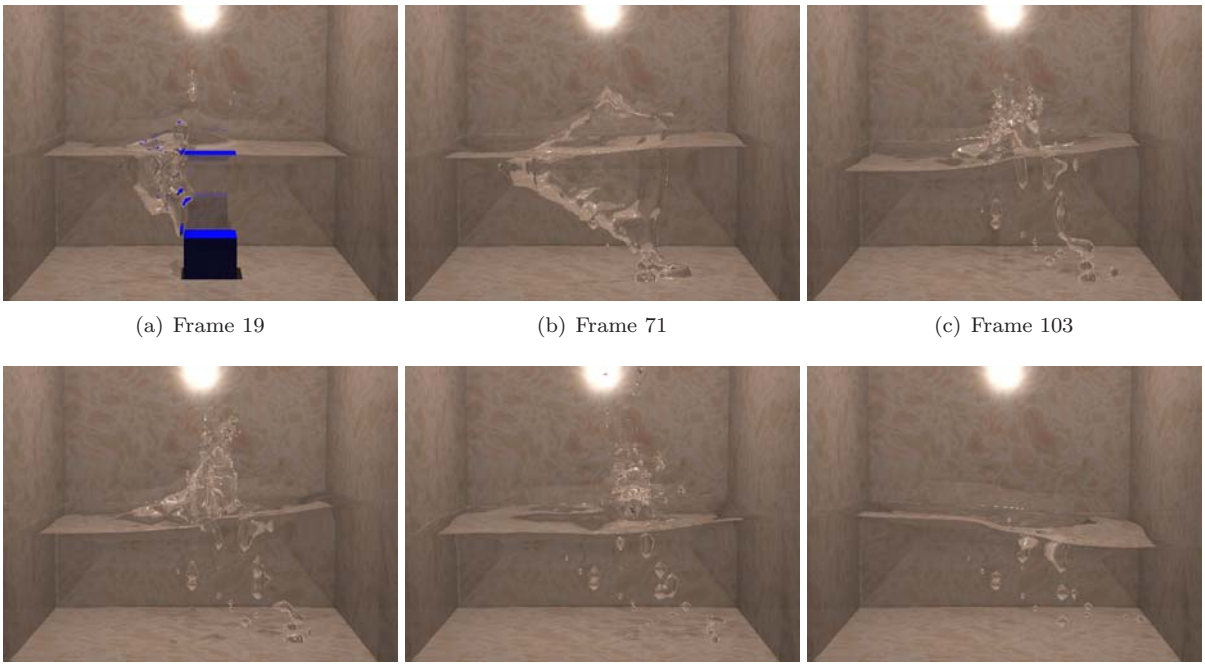
Similar hybrid models in [18] and [16] permits a specific error when a solid object strike the surface at some angle to the veritical. Because they modeled the liquid volume by vertical columns, the force form the solid object is applied in a strictly vertical and the droplets escaping from the liquid surface can only be given an initial vertial velocity. Consequently, the resulting wave pattern and splash drops have no directional preference, which is incorrect. In this thesis, since the liquid volume is simulated by 3-D Navier-Stokes equations, the resulting waves naturally travel in the biased direction of the impact and the initial velocities of the droplets are interpolated near the surface.

A feature characteristic of piecewise constant volume tracking methods is the unphysical and numerical creation of *floatsam* ("floating wreckage") and *jetsam* ("jettisoned goods") [17], as found in Figure 7 (c)-(f). These terms are appropriate for isolated, sub-mesh size material bodies that separate from the main material body because of errors induced by the volume tracking algorithm. These material remnant tend to be ejected from interfaces in piecewise constant volume tracking methods when the flow has significant vorticity and shear near the interface [10].

The calculation time grows more than cubicly with the resolution of the grid. When the cell size decreases, the simulation time-step also need to decrease to satisfy the CFL condition and therefore more cycles are needed for a frame than in the grid of a lower resolution.

4 Conclusion

To model liquid with dramatic deformation, a hybrid approach combining the volume tracking method and smoothed particle hydrodynamics (SPH) has been proposed. The 3-D Navier-Stokes equations are solved by the marker-and-cell (MAC) method and a density volume representing the liquid is evolved over time and space by the volume-of-fluid (VOF) method. The VOF method uses a finite-volume discretization and maintaining a value for the liquid volume in each grid cell. The advection of the volumes is calculated by us-



(a) Frame 19 (b) Frame 71 (c) Frame 103

(d) Frame 127 (e) Frame 159 (f) Frame 197

Figure 7: Selected frames of throwing a cube into a tank of water ($28 \times 24 \times 28$ grid cells)

ing cell-face fluxes, in which the liquid volume that leaves one cell is exactly the same as the liquid volume that enters the adjacent cells.

The proposed water splash modeling is a major improvement over previous hybrid volume-particle methods. Rules are proposed to identify regions with drastic motion and to position particles adequately. To add fidelity of the particle trajectory, the dynamics of the particles are governed by SPH, while previous methods treated splash as interaction-less particles. A smooth implicit surface is extracted from the liquid density volume which interpolates the volume fractions in a higher resolution and converts particle volumes to volume fractions.

Modeling liquid by volume fractions is much more efficient and economical than traditional particle systems. Besides, the high realism of the animation is enabled by the coupling of pressure and velocity in 3-D Navier-Stokes equations. Potentially under-resolved subtle features are captured by smoothed particles that are adequately placed in the neighborhood of the highly-deformable regions. Close scrutiny reveals that SPH can provide more physically-correct particle motions because it approximates the Navier-Stokes equations. Consequently, it can continue to take into account the coupling between pressure and velocity terms, which is lacked in other particle simulation methods. Particle volumes are incorporated into the interpolated volume fractions before an unified iso-surface is extracted. As a result the liquid surface is more smooth than those of previous researches which rendered the splash by coating the particles with a field functions or as hard spheres.

References

[1] D.S. Balsara. Von-Neumann stability analysis of Smoothed Particle Hydrodynamics–suggestions for optimal algorithms. *Journal of Computational Physics*, 121(2):357, 1995.

[2] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, 1967.

[3] M. Desbrun and M.-P. Gascuel. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Proceedings of The 6th Eurographics Workshop on Animation and Simulation*, pages 61–76, 1996.

[4] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water sur-

faces. In *Proceedings of ACM SIGGRAPH '02*, 2002. To appear.

[5] N. Foster and R. Fedkiw. Practical animation of liquids. In *Proceedings of ACM SIGGRAPH '01*, pages 23–30, 2001.

[6] G.Miller. Globular dynamics: A connected particle system for animating viscous fluids. *Computer & Graphics*, 13(3):305–309, 1989.

[7] D. Gueyffier, A. Nadim, J. Li, R. Scardovelli, and S. Zaleski. Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152:423–456, 1999.

[8] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965.

[9] C.W. Hirt and B.D. Nichols. Volume of Fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[10] D.B. Kothe and W.J. Rider. Comments on modeling interfacial flows with Volume-of-Fluid methods. Technical Report LA-UR-94-3384, Los Alamos National Laborator, 1995.

[11] A. Kunimatsu, Y. Watanabe, H. Fujii, T. Saito, K. Hiwada, and H. Ueki. Fast simulation and rendering techniques for fluid objects. *Computer Graphics Forum*, 20(3):57–67, 2001.

[12] J.J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.

[13] J.J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110:399–406, 1994.

[14] J.J. Monaghan and J.C. Lattanzio. A refined method for astrophysical problems. *Astron. Astrophys*, 149:135–143, 1985.

[15] J.P. Morris, P.J. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136(1):214–226, 1997.

[16] D. Mould and Y.H Yang. Modeling water for computer graphics. *Computers & Graphics*, 21(6):801–814, 1997.

[17] W.F. Noh and P.R. Woodward. Slic (simple line interface method). In A.I. van de Vooren and P.J. Zandbergen, editors, *Lecture Notes in Physics*, volume 59, pages 330–340. Springer Verlag, Berlin, 1976.

[18] J.F. O'Brien and J.K. Hodgins. Dynamic simulation of splashing fluids. In *Computer Animation '95*, pages 198–205, 1995.

[19] L.A. Pars. *A treatise on analytical dynamics*. Wiley, New York, 1965.

[20] W.T. Reeves. Particle systems–a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2):91–108, 1983.

[21] R. Scardovelli and S. Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *Journal of Computational Physics*, 164(1):228–237, 2000.

[22] D.L. Youngs. Time dependent multi-material flow with large fluid distortion. In K.W. Morton and M.J. Baines, editors, *Numerical methods for fluid dunamics*, pages 273–285. Academic Press, 1982.