Interactive Fluid Simulation Using Augmented Reality Interface

Makoto Fujisawa and Hirokazu Kato

Graduate School of Information Science, Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara, Japan
{fujis,kato}@is.naist.jp

Abstract. This paper presents an interactive fluid simulation system using augmented reality interface. The presented system uses Smoothed Particle Hydrodynamics to simulate the behavior of liquid and adopts a particle-particle interaction approach to calculate the surface tension that becomes important in a small-scale liquid. Fluid-solid interaction can be calculated effectively by representing a solid as a distance function. Therefore, the shape of the solid can be represented precisely without increasing the number of the particles. Moreover, The system can directly operate the solid by augmented reality interface.

Keywords: real-time fluid simulation, surface tension, augmented reality interface.

1 Introduction

Fluid simulation has been mainly used in an industrial field to design products. In these fields, the accuracy is very important and then the computational time tends to be quite long. On the other hand, in computer graphics field, several methods for real-time fluid simulation had been developed in recent years. Especially, particle method, which is a technique of representing liquid by many particles, is used mainly for real-time application, game, and interactive system.

There are many researches to generate fluid animations such as ocean wave, explosion, and dam breaking. However, the many of fluid phenomena familiar to us are small-scale, for example, pouring water to the cup, flowing droplets. In such small-scale fluid phenomena, surface tension becomes a dominant factor. To realize the interaction with small-scale liquid such as water droplets, the fluid simulation must be stable and fast even though the surface tension is very large. We adopt Smoothed Particle Hydrodynamics (SPH) method with a particle-particle interaction approach for the surface tension calculation. In addition, to simulate high-viscosity fluid such as cream and incompressibility fluid, we use a double density relaxation method [3].

Interface is important part of the interactive fluid simulation system. Generally, movements of mouse in 2D are converted into limited movements in 3D space. However, this technique is not intuitive operation, and it is difficult to deal for inexperienced users. It is also possible to use 3D user interface such as PHANToM. But the cost is high. On the other hand, augmented reality (AR) interface is used as three dimensional interface that contains virtual and real objects simultaneously. AR

R. Shumaker (Ed.): Virtual and Mixed Reality, LNCS 5622, pp. 431–438, 2009.

interface only requires to a cheap camera and marker patterns which is printed on a paper, and provide more intuitive operations. The AR interface is used to operate a solid in our fluid simulator. The liquid represented by particles can control freely by using the solid. As a result, it is possible to use it as assistance for making CG animations.

The rest of the paper is organized as follows: Section 2 describes previous work and Section 3 details our fluid simulation techniques. Section 4 and 5 address our rendering method of liquid surface and AR interface used in the simulation. Section 6 shows the result of our method. Finally, we conclude the paper in section 7 with a discussion of future work.

2 Related Work

As a physical-based simulation, fluid dynamics have been used to animate complicated behaviors of water [4], gas [15], flames [14] etc. To capture a smooth interface of liquid, structured or unstructured grid has been used in computer graphics. The grid-based method, which known as a type of Eulerian approach, discretize the computational space by finite cells. Each cell contains a discriminant function (e.g. Level set function) to identify liquid region or not. This function represents the smooth surface of liquid. However, it is difficult to represent exact boundary of solid because of axis aligned grid. Losasso et al. [12] used an unstructured grid. The size of which is finer around free fluid and solid surfaces. However, a boundary region between the objects and fluid is represented by hexahedrons and good results are not always obtained. Klingner et al. [7] proposed a method in which tetrahedral grids are generated according to a boundary shape at each time step for interactions with moving solid objects. However, the generation of adaptive grids requires additional processing time.

The particle-based "Lagrangian" method abandons the grid completely. In this method, fluid is represented by a finite number of moving particles. Movement of the particle is equivalent to the movement of the fluid. The particle method can compute effectively as compared to the grid-based Eulerian approach that discretizes the whole space with the grids, because it only has to arrange the particles in the liquid domain where we want to calculate the movement. Therefore, the particle method frequently used for real-time applications. The most common of which include Moving Particle Semi-implicit(MPS) [8] and Smoothed Particle Hydrodynamics(SPH) [13]. MPS employed a semi-implicit algorithm to treat incompressible flows. The semi-implicit algorithm increases the computational cost. On the other hand, SPH that computes compressible flows, is often used in computer graphics because the computation cost of SPH is typically lower than that of MPS. Tartakovsky et al. [16] introduced a particle-particle interaction model in the calculation of SPH to represent the surface tension of liquid drop. The particle-particle interaction model was also applied to MPS by Liu et al. [10]. Harada et al. [5] proposed an improved calculation model of wall boundary computation. They used an implicit representation for solid surface. Clavet et al. [3] proposed a robust and stable method for viscoelastic fluid simulation. They adopted a prediction-relaxation scheme in SPH for numerical stability. The incompressibility was accomplished by a technique called "double density relaxation". For liquid simulation, we use the method of [3].

There are some researches which uses physic simulation in augmented reality environment. Allard and Raffin [1] developed a virtual reality system for augmenting real worlds with physical simulation and Chae and Ko [2] introduced rigid collisions by using Open Dynamics Engine. Kwatra et al. [9] used a fluid simulation and a complete 3D geometry reconstructed from binocular video sequences to enable one-way coupling from the video to the fluid. Imura et al. [6] integrated the particle-based fluid simulation method into augmented reality environment. We use SPH and the interaction between fluid and solids for operating the fluid as well as [6]. Our method also can treat highly viscous fluid and small-scale liquid phenomena such as water drops.

3 Fluid Simulation

3.1 Physical Quantity

In SPH, a physical property $\phi(\mathbf{x})$ is approximated by summing weighted properties of neighbor particles.

$$\phi(\mathbf{x}) = \sum_{j \in N} m_j \frac{\phi_j}{\rho_j} W\left(\mathbf{x}_j - \mathbf{x}, h\right) . \tag{1}$$

where *N* is the set of neighboring particles that are closer than the effective radius *h*. m_j and ρ_j are the mass and density of particle respectively. *W* is the kernel function. The gradient of this property is approximated by using the derivative of the kernel function. The density ρ can be found from equation (1).

$$\rho(\mathbf{x}) = \sum_{j \in N} W\left(\mathbf{x}_j - \mathbf{x}, h\right) . \tag{2}$$

We assume all the particles have the unit mass (i.e. $m_j = 1$) for simplification. For density calculation, we employ $W = (1 - |\mathbf{x}_j - \mathbf{x}| / h)^2$ as a kernel function.

Given the density ρ, the pressure *p* is obtained from the equation of state.

$$p = k(\rho - \rho_0) . \tag{3}$$

where *k* is the stiffness parameter and ρ_0 is the rest density.

3.2 Incompressibility

SPH was developed for compressible flow in astrophysics, In order to simulate incompressibility and compute surface tension, we use a double density relaxation, and adopt a prediction relaxation approach for numerical stability [3].

The double density relaxation has two different particle densities: density and near-density. The near-density prevents particle clustering and provides surface tension effect without curvature evaluation.

A typical SPH computation updates the position and velocity of each particle by accumulating various forces. This explicit force integration tends to be unstable, particularly, when large time step is used and/or when large force is acted on particles (e.g. small water drop). The prediction relaxation avoids explicit force integration by using a technique more similar to an implicit scheme. At the start of the time step, predicted position of particle is computed from previous position and velocity. The predicted position is relaxed by relaxation displacements due to pressure, collision with solid, etc during the time step. The velocities are recomputed by subtracting previous positions from relaxed position at the end of the time step.

The density displacement between two particles is

$$D_{ij} = \Delta t^2 p_i \nabla W_p\left(\mathbf{x}_j - \mathbf{x}_i, h\right) . \tag{4}$$

where $W_p\left(\mathbf{x}_j - \mathbf{x}_i, h\right) = \left(1 - |\,\mathbf{x}_j - \mathbf{x}_i\,| / h\right)^2$ is the kernel function of the density relaxation. As described above, we use the near-density ρ^{near} in order to incompressibility and anti-clustering. The density displacement with the near-density is

$$D_{ij} = \Delta t^2 \left(p_i \nabla W_p\left(\mathbf{x}_j - \mathbf{x}_i, h\right) + p_i^{near} \nabla W_n\left(\mathbf{x}_j - \mathbf{x}_i, h\right)\right) . \tag{5}$$

where $W_n\left(\mathbf{x}_j - \mathbf{x}_i, h\right) = \left(1 - |\,\mathbf{x}_j - \mathbf{x}_i\,| / h\right)^3$ is the near-density kernel and the near-density is calculated by using this kernel as well as equation (2). $p^{near} = k^{near} \rho^{near}$ is the near-pressure. The near-density kernel acts as a repulsion force because its value becomes sharper near the center.

The surface tension force can be calculated by equation (5) as inter-particle force. There is another method for calculating surface tension that defines a color function to distinguish between liquid and non-liquid. The curvature of the liquid surface can be computed from the color function. Although the method using the color function works well with large-scale simulation, it is difficult to apply it to small-scale liquid phenomena such as small drop of water, because the method adds the force explicitly. Surface tension is physically caused by the forces acting on molecules at the surface. The forces can be modeled by the particle-particle interactions. We can include the surface tension in our simulation without using the color function which may cause numerical instability.

3.3 Interaction with Solid

Generally, a solid boundary in particle methods is represented as static particles. These particles are treated just like the fluid particles during the density and pressure calculations but the relative positions doesn't change. If we use this method, several layers of particles along the boundary is required. Furthermore, the particles can not accurately represent the boundary.

In order to improve these issues, Harada et al. [5] had introduced a precomputed wall weight function to compute the contribution of a boundary. The computation of fluid density with the wall weight function is

$$\rho(\mathbf{x}) = \sum\nolimits_{j \in N_{fluid}} W\left(\mathbf{x}_j - \mathbf{x}, h\right) + \sum\nolimits_{j \in N_{wall}} W_{wall}\left(\mathbf{x}_j - \mathbf{x}, h\right)$$

$$= \sum\nolimits_{j \in N_{fluid}} W\left(\mathbf{x}_j - \mathbf{x}, h\right) + Z_{wall}\left(r_{iw}\right) . \tag{6}$$

where $Z_{wall}\left(r_{iw}\right)$ is the wall weight function, r_{iw} is the distance between a particle i and wall. The wall weight function can be computed before the simulation step starts and simply referred in the step.

4 Rendering

The positions of particles have been computed, and then we have to visualize the free surface of liquid. One way to visualize the free surface of liquid is to render an iso surface of particles by using an additional field quantity called color field which is 1 at particle locations and 0 otherwise. The color field c_s can identify the free surface of liquid.

$$c_s(\mathbf{x}) = \sum\nolimits_{j \in N} W_{cs}\left(\mathbf{x}_j - \mathbf{x}, h\right) . \tag{7}$$

We use Marching Cubes [11] to triangulate the iso-surface $c_s(\mathbf{x}) - T = 0$ where T is user defined threshold. We implemented Marching Cubes on GPU using CUDA to accomplish the real-time computation.

5 Augmented Reality Interface

AR interface is 3D input device that can be constructed at a low price, and we use this for the interaction with 3D flow simulation. The solid objects is moved by the AR interface, the user can operate the fluid through the solid. The solid objects can take any shape as described in section 0 because it is represented by a distance function.

6 Results

This section describes the results of our method. The results are performed on a PC with Core 2 Duo 3.16GHz CPU and 3.0GB RAM. The AR interface is implemented by AR Toolkit, and we make the two markers to indicate the environment and the user operation.

Fig. 1 shows the result of liquid in a tank and a solid ball augmented onto markers. The ball moves according to the movement of the marker and interacts with the liquid. The number of particles was about 800. Surface polygons were extracted from the particles using equation (7) and Marching Cubes and the process was accelerated

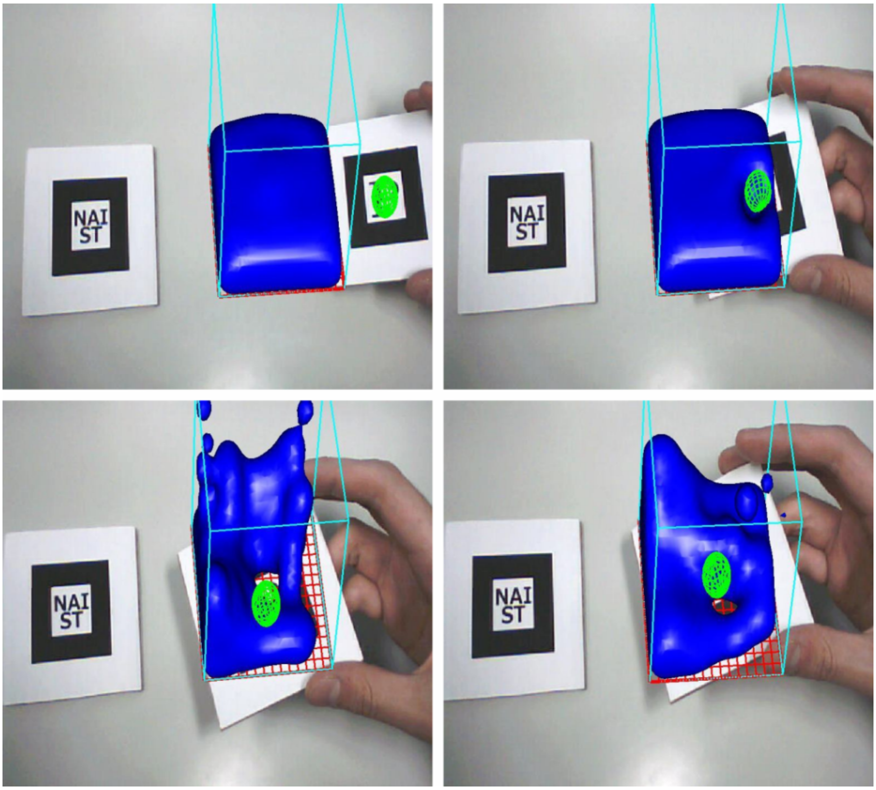

Fig. 1. Liquid in a tank interacting with a ball

by using GPU. The wall weight functions were used for the wall of the tank and the ball. Fig. 2 shows water drop simulation with a solid box rendered by wireframe. The shape of water drop is mainly governed by surface tension. Appropriate shape is reproduced by our stable surface tension calculation. The users can move the drop of water by using the solid. The drop of water spattered when the recognition of the marker becomes unstable depending on the environment and the marker moves too rapidly. It is necessary to improve it to become smooth movement by using Kalman filtering etc.

7 Conclusion and Future Work

We presented an interactive fluid simulation system using augmented reality interface. Our system adopted SPH to simulate the behavior of liquid and used the particle-particle interaction approach for the surface tension calculation. In addition, we used the double density relaxation method to simulate high-viscosity fluid and incompressibility fluid. In order to achieve intuitive operation, the augmented reality interface was used and fluid-solid interaction could be calculated effectively by representing the solid as a signed distance function.

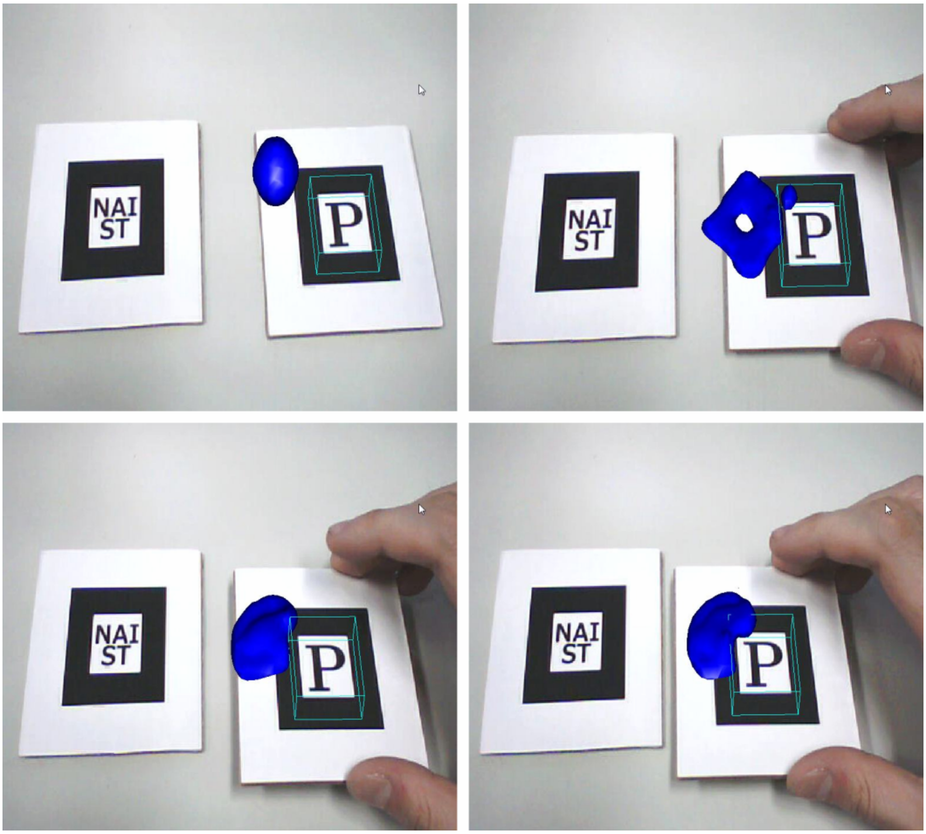

Fig. 2. Water drop interacting with a solid box

For future work, as mentioned in section 0, we may be able to estimate the solid motion using Kalman filtering. Other future research topics include a combination of another simulation and AR Interface, for example, elastic body deformation, air flow over the whole space etc.

References

1. Allard, J., Raffin, B.: Distributed Physical Based Simulations for Large VR Applications. In: VR 2006: Proceedings of the IEEE conference on Virtual Reality, pp. 89–96 (2006)
2. Chae, C., Ko, K.: Introduction of Physics Simulation in Augmented Reality. In: International Symposium on Ubiquitous Virtual Reality, pp. 37–40 (2008)
3. Clavet, S., Beaudoin, P., Poulin, P.: Particle-based Viscoelastic Fluid Simulation. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 219–228 (2005)
4. Enright, D., Marschner, S., Fedkiw, R.: Animation and rendering of complex water surfaces. In: Proceedings of ACM SIGGRAPH 2002, pp. 736–744 (2002)
5. Harada, T., Koshizuka, S., Kawaguchi, Y.: Improvement of the Boundary Conditions in Smoothed Particle Hydrodynamics. Computer Graphics & Geometry 9(3), 2–15 (2007)

6. Imura, M., Amada, T., Yasumuro, Y., Manabe, Y., Chihara, K.: Synthetic Representation of Virtual Fluid for Mixed Reality. In: Proceedings of 8th International Conference on Virtual Reality, pp. 135–142 (2006)
7. Klingner, B.M., Feldman, B.E., Chentanez, N., O'Brien, J.F.: Fluid animation with dynamic meshes. In: Proceedings of ACM SIGGRAPH 2006, pp. 820–825 (2006)
8. Koshizuka, S., Tamako, H., Oka, Y.: A particle method for incompressible viscous flow with fluid fragmentation. Computational Fluid Dynamics Journal 4(4), 29–46 (1995)
9. Liu, J., Koshizuka, S., Oka, Y.: A hybrid particle-mesh method for viscous, incompressible, multiphase flows. Journal of Computational Physics 202, 65–93 (2005)
10. Kwatra, V., Mordohai, P., Narain, R., Penta, S.K., Carlson, M., Pollefeys, M., Lin, M.: Fluid in Video: Augmenting Real Video with Simulated Fluids. Computer Graphics Forum (Proc. Eurographics) 27(2), 487–496 (2008)
11. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. Computer Graphics (SIGGRAPH 1987 Proceedings) 21(4), 163–169 (1987)
12. Losasso, F., Gibou, F., Fedkiw, R.: Simulating water and smoke with an octree data structure. In: Proceedings of SIGGRAPH 2004, pp. 457–462 (2004)
13. Monaghan, J.J.: An Introduction to SPH. Computer physics communications 48, 89–96 (1988)
14. Nguyen, D., Fedkiw, R., Jensen, H.: Physically based modeling and animation of fire. In: Proceedings of SIGGRAPH 2002, pp. 721–728 (2002)
15. Stam, J.: Stable fluids. In: Proceedings of SIGGRAPH 1999, pp. 121–128 (1999)
16. Tartakovsky, A., Meakin, P.: Modeling of surface tension and contact angles with smoothed particle hydrodynamics. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) 72(2), 026301 (2005)