Target-Driven Liquids Animation with Interfacial Discontinuities

Seung-Ho Shin
Korea University
+82-2-3290-3574
winstorm@korea.ac.kr

Chang-Hun Kim
Korea University
+82-2-3290-3199
chkim@korea.ac.kr

ABSTRACT

We propose a novel method of controlling a multi-phase fluid so that it flows into a target shape in a natural way. To preserve the sharp detail of the target shape, we represent it as an implicit function and construct the level-set of that function. Previous approaches add the target-driven control force as an external term, which then becomes attenuated during the velocity projection step, making the convergence process unstable and causing sharp detail to be lost from the target shape. But we calculate the force on the fluid from the pressure discontinuity at the interface between phases, and integrate the control force into the projection step so as to preserve its effect. The control force is calculated using an enhanced version of the ghost fluid method, which guarantees that the fluid flows from the source shape and converges into the target shape, while achieving a more natural animation then other approaches. Our control force is merged during the projection step avoiding the need for a post-optimization process to eliminate divergence at the liquid interface. This makes our method easy to implement using existing fluid engines, and it incurs little computational overhead. Experimental results show the accuracy and robustness of this technique.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Language Computational Geometry and Object Modeling—Physically based modeling; I.3.5 [**Computer Graphics**]: Three-Dimensional Graphics and Realism

—Animation

Keywords

Multi-phase fluid, controlling fluid, target-driven fluid, surface tension, pressure jump, bubbly water

1. INTRODUCTION

Recently, more realistic ways of controlling fluid animation have been proposed. Unlike morphing techniques, these methods produce a natural-looking fluid flow between a source shape and a target shape. Some of these new techniques also use a multiphase fluid simulation so that liquids can be included in the animation. Control is achieved by adding external forces to the Navier-Stokes equation so that the fluid flows from the source to the target configuration. But existing techniques do not take account of the interfacial discontinuity of pressure and density that occurs when two different fluids are adjacent to each other. Some important properties of fluids, such as surface tension and capillary phenomena, occur at such a discontinuity. In addition, flow around objects is greatly influenced by the discontinuity in properties at the interface. We therefore propose a new method that improves the control of multiphase fluid simulation by taking account of interfacial discontinuities. To allow a user to change the direction and shape of a fluid flow, we add control forces at the projection step of the modified ghost fluid method (GFM), which is able to take account of the discontinuities which occur at the interface between different fluids. The user may provide an image, 3D mesh data or sketches. Based on this input, the level-set of the target shape is constructed as a signed distance function, from which pressure jump values can be determined.

Controlling a liquid animation by adding forces at the interface between two immiscible fluids has several advantages:

- Because we are adding the control force at the projection step, the force is accurately preserved.
- The fluid simulation is still divergence-free and robust, despite the control force, and no optimization is required.
- Additional natural forces such as gravity and buoyancy can be added to the environment as external forces. Control and external forces are added independently at different stages.
- The control force can easily be added within an existing fluid simulation pipeline.

We propose an effective fluid animation method that makes a fluid flow from a source shape to a target shape. The resulting animations show realistic and smooth fluid motion as well as achieving the required target shape.

In Section 3 we introduce the equation for multiphase fluid simulation. Section 4 explains our modified ghost fluid method, which allow for pressure differences. Section 5.1 then describes how the target-driven force is calculated and how the signed distance level-set is constructed from a given image and mesh data. The addition of control forces at the projection step of a modified GFM is described in Section 5.2. In Section 6 we present our experimental results and Section 7 concludes this paper.

2. Related Work

Numerical simulation of the Navier-Stokes equation has become a standard technique for the realistic animation of fluids. Foster and Metaxas [7] used a fully three-dimensional Navier-Stokes solver in computer graphics, and an effective and robust solution to the Navier-Stokes equation that includes semi-Lagrangian advection has also been reported [19]. Foster and Fedkiw [5] used a conjugate gradient method to solve the linear equation and an implicit level-set surface to represent the interface area effectively. Their method smoothes the fluid interface, and changes of topology are represented robustly. This method has been extended to particle level-set method [2] with conservation of mass. Losasso et al. [14] used an adaptive octree data structure to show detailed fluid effects such as the crown phenomenon, while additional effects can be simulated by adding surface tension as an external force [10, 14], although this approach does not take account of the discontinuities in multiphase fluid interfaces.

Hong and Kim [12] deal with interface discontinuities using the ghost fluid method (GFM) [4, 13]. They consider surface tension at the interface between two immiscible fluids at the projection step and introduce a discontinuous viscosity condition. This approach allows small-scale phenomena, such as capillary instability, the breakup of sheets of liquid, and bubbling water to be animated successfully [12]. We add a target-driven control force to the GFM approach, and exploit its ability to care multiphase interfaces.

Several methods of fluid control have been proposed to augment the simulation techniques mentioned above. Treuille et al. [22] proposed a way of controlling smoke by systematic optimization of control forces. However, the derivatives of each control parameter need to be updated during the entire simulation, making this approach very expensive. Using the adjoint method [15] can significantly improve the effectiveness of each iteration in a linear optimization, but a lot of time is still required, and the resulting motion is not natural because the optimization process requires the fluid density to be fixed.

Fattal and Lischinski [6] introduced an efficient method to match smoke density against user-specified distributions, using a model with a driving force term and a smoke-gathering term. This technique is much faster than nonlinear optimization. Shi and Yu [20] also used level-sets to animate smoke, placing velocity constraints on the smoke boundary to make it match the shape of the target. This method controls the shape of the smoke more efficiently and effectively than forcing and gathering terms [6]. However they are not appropriate for liquid simulation because the compressible fluid model is used.

Hong and Kim [11] have suggested another efficient and simple method to control a fluid animation, using a potential field based on the shape of the target. An external force obtained from the negative gradient of this potential field pushes the smoke towards the target shape. But oscillation occurs around the shape boundary due to inertia, and details of the target shape are not accurately modeled by the fluid. On the other hand, Shi and Yu [21] control a liquid animation, using a feedback force with a velocity and a shape component. Because the feedback force is reduced by the projection step that makes the velocity field divergence-free, they perform an additional optimization process to make the shape component divergence-free, which avoids reducing the force. We use the same concept of shape feedback, but we do not need to worry about a reduced force because the force is added at the projection step, rather than as an external term.

3. Fluid Simulation

The Navier-Stokes equation for fluid simulation is

$$\mathbf{u}_t = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla\cdot(\nu\nabla\mathbf{u}) - \frac{\nabla p}{\rho} + \mathbf{f} \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$ is the velocity, ν is the kinematic viscosity, and ρ is the density. The term $\mathbf{f}$ can be used to add external forces such as gravity, buoyancy [5], surface tension [10] or control forces [6, 11, 15, 22].

The numerical simulation of Equations (1) and (2) requires the value of $\mathbf{u}$ to be updated from $\mathbf{u}^n$ to $\mathbf{u}^{n+1}$ at the n^{th} time step. We discretize Equation (1) by splitting it into two equations by introducing an intermediate velocity $\mathbf{u}^*$:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n\cdot\nabla)\mathbf{u}^n + \nabla\cdot(\nu\nabla\mathbf{u}^n) + \mathbf{f} \tag{3}$$

$$\frac{\mathbf{u}^n - \mathbf{u}^*}{\Delta t} = -\frac{\nabla p}{\rho}. \tag{4}$$

The variable $\mathbf{u}^*$ can be used to compute the advection term using a semi-Lagrangian method [19].

We can write the divergence of Equation (4) as a form of Poisson's equation:

$$\nabla^2 p = \frac{\rho}{\Delta t}\nabla\cdot\mathbf{u}^*. \tag{5}$$

Once the pressure profile has been determined by solving this equation, we can obtain the final velocity profile:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p. \tag{6}$$

We use a signed distance function ϕ to represent implicitly the interface between two immiscible fluids, such as a liquid and a gas. The advection of ϕ can be described by the level-set equation,

$$\phi_t + \mathbf{u}\cdot\nabla\phi = 0. \tag{7}$$

To solve this equation numerically, we use the semi-Lagrangian particle level-set method [3]. Our simulation space is modeled as on octree [14].

4. Interfacial Dynamics

There is a discontinuous pressure profile at the interface Γ between two different fluids. Figure 1 shows the discontinuous pressure at the interface, Γ. The pressure of the right and left sides

are different across Γ. This makes it difficult to differentiate the pressure across Γ using standard finite differencing.

It is possible [12] to take the discontinuous pressure at the interface into account, using ghost fluid method [4, 13]. The pressure at node i, which is p_i, and the pressure at node $i+1$, which is p_{i+1} are extrapolated across Γ to determine the ghost value, p_{i+1}^G and p_i^G:

$$p_i^G = p_i + J \tag{8}$$

$$p_{i+1}^G = p_{i+1} - J\,. \tag{9}$$

Using these equations, Equation (5) can be expended Equation (10) and (11) as follows [12]:

$$\frac{p_{i+1} + p_{i-1} - 2p_i}{\Delta x^2} = D(x_i) + \frac{J}{\Delta x^2} \tag{10}$$

$$\frac{p_{i+2} + p_i - 2p_{i+1}}{\Delta x^2} = D(x_{i+1}) - \frac{J}{\Delta x^2} \tag{11}$$

where D represents the right-hand side of Equation (5) in one dimension. Note that the left-hand side terms of Equations (10) and (11) are identical to those used by Foster and Fedkiw [5]. These equations can be solved using the same linear system that has been used to solve Poisson's equation.

Equations (8) and (9) are based on the assumption that the differential values of pressure on both sides of the interface are always equal. So the Equation (6) also gives us same velocity values on both sides. To accommodate a pressure differential, Equations (8) and (9) must be modified as follows:

$$p_i^{NewG} = p_i + J + (p_{i+1}{}^* - p_{i+2}{}^*) \tag{12}$$

$$p_{i+1}^{NewG} = p_{i+1} - J + (p_i{}^* - p_{i-1}{}^*)\,, \tag{13}$$

where $(p_{i+1}{}^* - p_{i+2}{}^*)$ and $(p_i{}^* - p_{i-1}{}^*)$ are obtained using the pressure value p^* from the time step before projection.

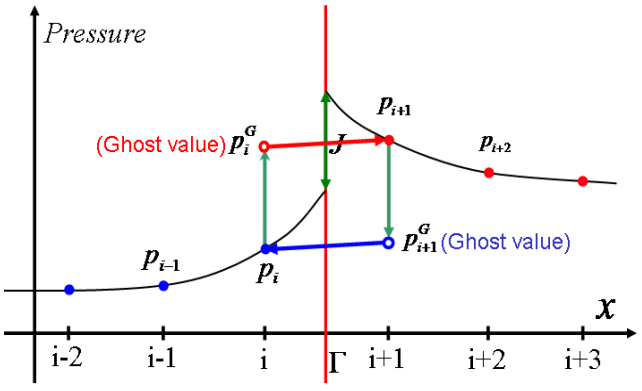


Figure 1. The discontinuous pressure field and the Ghost value.

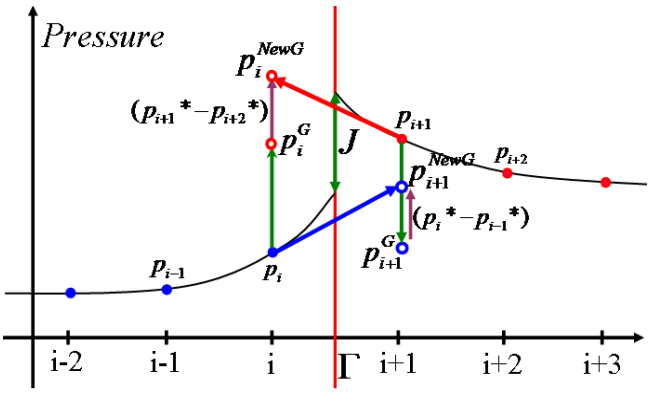


Figure 2. The discontinuous pressure differences and the new Ghost value.

To make a liquid flow into a target shape, we impose control pressure using the pressure jump described in Section 5.2. This causes pressure values to change rapidly, which affects the pressure differential across the interface. The modified GFM deals with this situation efficiently.

5. Achieving the Target shape

5.1 Shape Feedback Force

To make a fluid converge to a detailed target shape, that shape needs to be described clearly. We represent the target shape as a level-set with a signed distance function modeled by an octree. From this shape information we can calculate the shape feedback force.

As seen in Figure 3, the force on a point P_1 which is outside the target shape is calculated in the direction $-\frac{\nabla\phi_{target}}{\|\nabla\phi_{target}\|}$, while the force on P_2 inside the target shape is calculated in the direction $\frac{\nabla\phi_{liquid}}{\|\nabla\phi_{liquid}\|}$. The term $\phi_{target}(x,t)$ is a signed distance function of the target shape at time t, and $\phi_{liquid}(x,t)$ is a signed distance function of the liquid boundary, In Sections 5.2, we shall explain how these forces are integrated at the projection step.

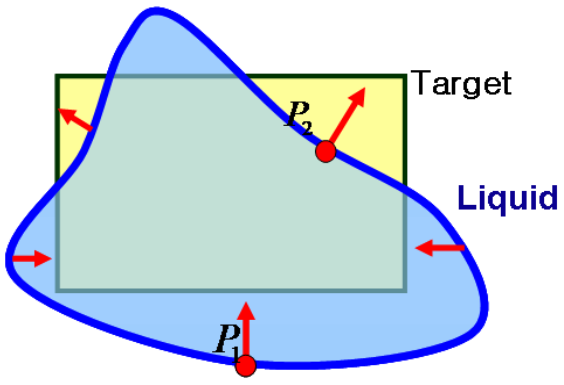


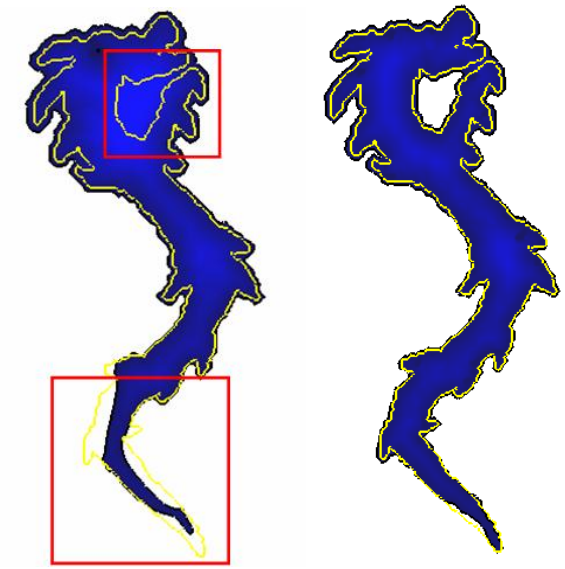
Figure 3. The shape feed back force on the liquid interface.

5.2 Using Pressure Jump

We now add the control force which we designed in Section 5.1, during the projection step. This force is determined from the magnitude of J, the pressure jump in Equations (12) and (13). In Hong and Kim's method [12], surface tension causes a jump J in pressure across Γ, and the magnitude of J is $\sigma\kappa_\Gamma$. The surface tension coefficient is σ and κ_Γ is the curvature, which can be determined by interpolating between the curvatures $\kappa = \nabla \cdot (\frac{\nabla \phi_{liquid}}{\| \nabla \phi_{liquid} \|})$ of near nodes.

Conversely, we determine the magnitude of J from the signed distance function of the target shape, and this pressure jump provides the force that makes the fluid assume the target shape. As shown in Figure 3, when a point P_1 on the liquid surface is outside the target shape, the magnitude of J is $-\sigma\phi_{target\_\Gamma}$, where ϕ_{target_Γ} is the value of the signed distance function of the target shape at the liquid interface, which is determined by interpolating between values of the signed distance function, ϕ_{target} at near nodes. But in the case of a point P_2 on the liquid interface, but inside the target shape, the control force is added in the direction normal to the interface. The magnitude of J may be $-\sigma\kappa$, which signifies a negative direction of the surface tension. The variable J is used in the modified ghost fluid method described in Section 4, to update the velocity field. Finally, when this projection step takes place, the effect of Shape feed back force appears.

Figure 4 shows the difference between controlling a liquid using an external force and applying fluid control during the projection step.



(a) **(b)**

Figure 4. Controlling a liquid with (a) an external force and with (b) a pressure jump. The yellow line is the boundary of the target shape.

Neither of these simulations executes force optimization process for divergence-free on the liquid boundary. The simulations were run until the liquid stopped moving because the forces were in equilibrium. Even though the target shape has concavities and sharp corners, the liquid is controlled satisfactorily using our method.

In previous work, fluid control has been achieved by a composite external force which aggregates many different forces. In that method, the fluid control force is counterbalance by other forces, and its magnitude is also reduced at the projection step. But in our method, control and external forces are added independently at different steps, and both control and external forces survive to influence the simulation.

6. Results

We have successfully applied our method to control liquids with an interfacial pressure discontinuity. We implemented our method in both 2D and 3D using C++ and tested it on a Windows PC with an Intel Pentium IV processor running at 3.0GHz and 1GB of RAM. 2D simulations were performed a 512^2 grid, and each a single time step took about 1 ~ 3 seconds. And 3D simulations were performed a 128^3 grid and each time step took about 5 seconds. Because no optimization is required, computing the control force only costs less than 3 percent of the total simulation time.

Figure 5 shows two-dimensional fluid animation in which the source shape '2007' flows into the target shape 'casa'.

Figure 5. Liquids making the word '2007' flow into the target shape, which is a title image of CASA.

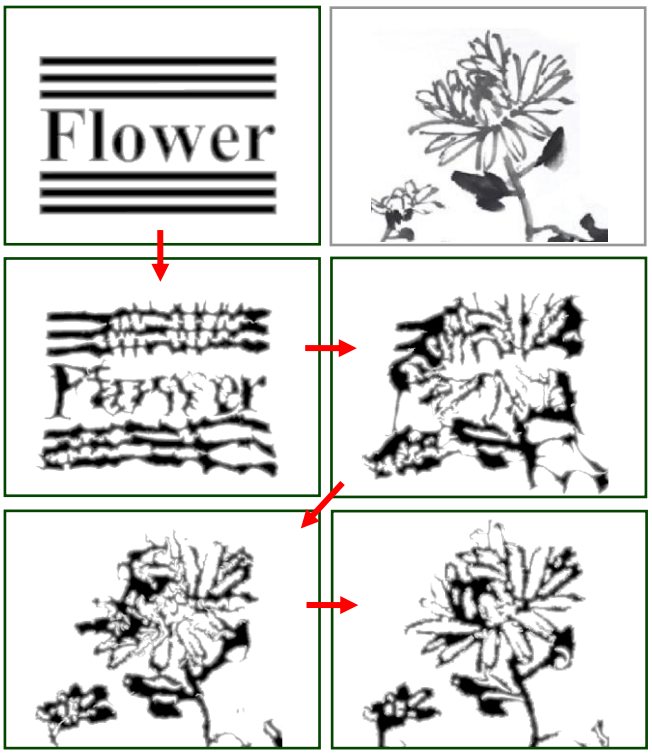
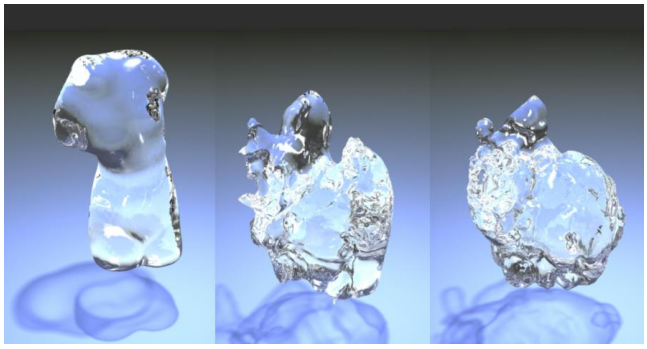
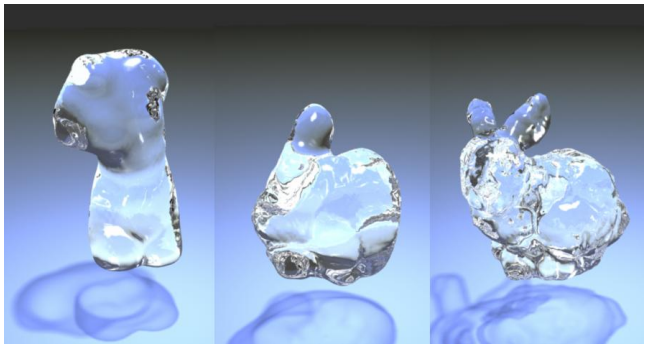


Figure 6. Liquids making the word 'Flower' flow into the target image, which is an oriental painting (top right).



(a) With an external force.



(b) With a pressure jump.

Figure 7. Liquid flowing from a Venus to a bunny. These figures show comparison of our method (b) with previous method (b) which uses an external force. The resolution is 128^3.

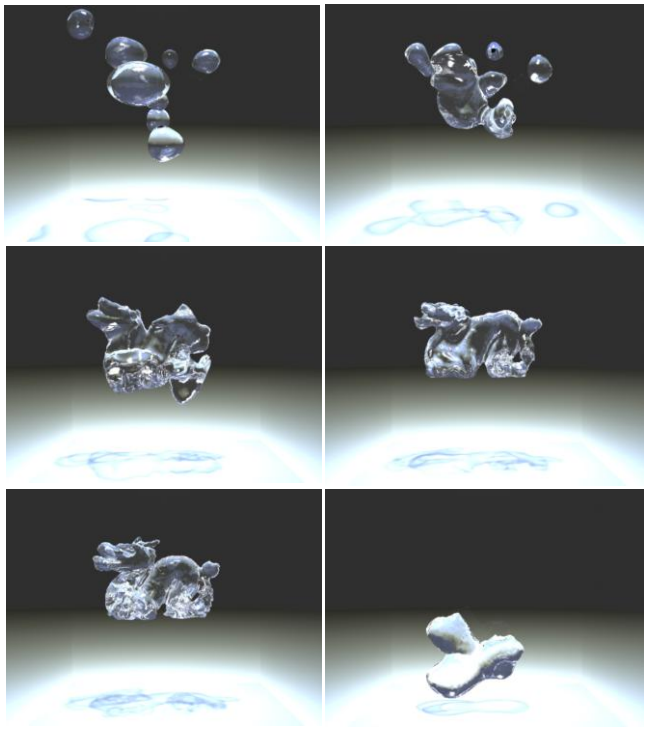


Figure 8. Floating bubbles coalesce to become a Chinese dragon while falling under gravity. The resolution is 128^3.

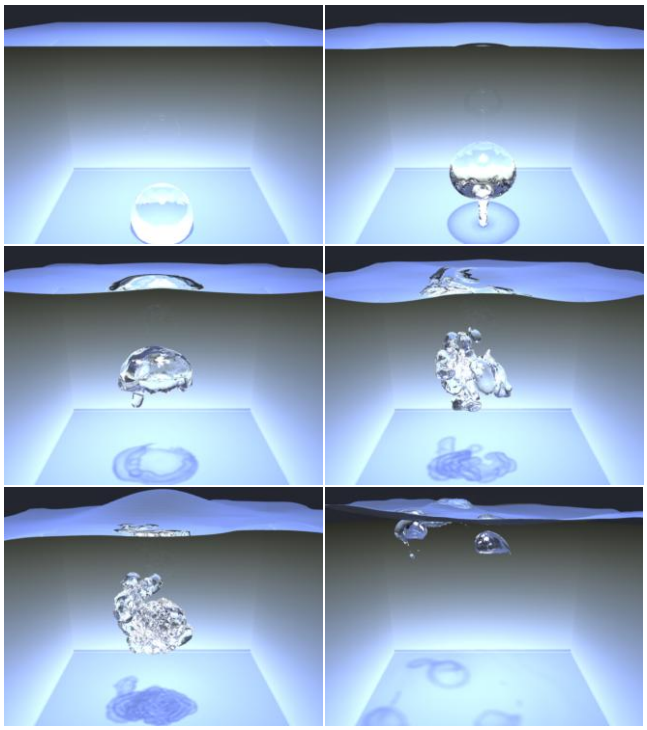


Figure 9. Air bubbles change into the target shape, which is a bunny. The resolution is 128^3.

Figure 6 shows interesting result of controlling liquids. A liquid flow is rendered by the effect of oriental painting. Although the

target shape is complicated and containing many gaps, the liquid assumes the details of the target shape successfully.

Figure 7 shows a comparison between the results achieved by our method and previous method in which an external force is used drive the fluid to its target. These simulations were run until the liquid stopped moving because the potential energy was in equilibrium. Using our method, small-scale details on the target shape, such as the ears of the bunny, but the external force fails to persuade the liquid to fill all the details and the bunny is malformed. Even if we perform this simulation at sparse resolution and without user interaction, the result is still plausible.

Figure 8 shows interesting dynamic effects that occur when several bubbles of liquid coalesce, assuming the shape of a Chinese dragon while falling under gravity.

Figure 9 shows target-driven air bubble animation. We controlled the air bubbles using negative direction of shape feedback force.

7. Conclusions and Future Work

We have described a technique for better control of multiphase fluid flow by improved handling of the interfacial discontinuity. A more effective control force is achieved by the existing ghost fluid method. Our fluid simulation is divergence-free and robust, despite the addition of the control force and the avoidance of any optimization process. We demonstrate realistic and smooth fluid motion within animation in which the fluid accurately assumes a target shape. This technique can be easily implemented on existing fluid simulation pipelines.

A possible extension of this work would be to control the interaction of different kinds of liquid such as water and oil.

8. ACKNOWLEDGMENTS

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

9. REFERENCES

[1] Adalsteinsson, D., Sethian, J. A. The fast construction of extension velocities in level set methods. *Journal of Computational Physics, 148*, 1, 2–22, 1999.

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

[3] Enright, D., Losasso, F., Fedkiw, R. A fast and accurate semi-lagrangian particle level set method. *Computers and Structures, 83*, 479–490, 2005.

[4] Fedkiw, R, Aslam, T., Merriman, B., and Osher, S. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics, 152*, 457–492, 1999.

[5] Foster, N., Fedkiw, R. Practical animation of liquids. In *Proceeding of SIGGRAPH 01*, pp. 23–30, 2001.

[6] Fattal, R., and Lischinski, D. Target-driven smoke animation. *ACM Transactions on Graphics, 23*, 3, 439-446, 2004.

[7] Foster, N., Metaxas, D. Realistic animation of liquids. *Graphical Models and Image Processing, 58*, 5, 471–483, 1996.

[8] Foster, N., Metaxas, D. Controlling fluid animation. In *Proceedings of Computer Graphics International*, pp. 178–188, 1997.

[9] Fedkiw, R., Stam, J., and Jensen, H. W. Visual simulation of smoke. *ACM Transactions on Graphics, 20*, 3, 15-22, 2001.

[10] Hong, J.-M., Kim, C.-H. Animation of bubbles in liquid. *Computer Graphics Forum, 22*, 3, 253–262, 2003.

[11] Hong, J.-M., Kim, C.-H. Controlling fluid animation with geometric potential. *Computer Animation and Virtual Worlds*, 15, 3-4, 147–157, 2004.

[12] Hong, J.-M., Kim, C.-H. Discontinuous fluids. *ACM Transactions on Graphics, 24*, 3, 915-920, 2005.

[13] Kang, M., Fedkiw, R., and Liu, X-.D. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing, 15*, 323–360, 2000.

[14] Losasso, F., Gibou, F., Fedkiw, R. Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics, 23*, 3, 455–460, 2004.

[15] McNamara, A., Treuille, A., Popovi, Z., Stam, J. Fluid control using the adjoint method. *ACM Transactions on Graphics, 23*, 3, 447–454, 2004.

[16] Rasmussen, N., Enright, D., Nguyen, D., Marino, S., Sumner, N., Geiger, W., Hoon S., and Fedkiw, R. Directable photorealistic liquids. *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, p. 193-202, Aug. 2004.

[17] Osher, S., and Fedkiw, R. Level set methods: an overview and some recent results. *Computational Physics, 169*, 2, 2001.

[18] Sethian, J. Level Set Methods and Fast Marching Methods. *Cambridge University Press*, 1999.

[19] Stam, J. Stable fluids. *ACM Transactions on Graphics, 18*, 3, 121-128, 1999.

[20] Shi, L., and Yu Y. Controllable smoke animation with guiding objects. *ACM Transactions on Graphics, 24*, 1, 140-164, 2005.

[21] Shi, L., and Yu Y. Taming liquids for rapidly changing targets. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 229-236, 2005.

[22] Treuille, A., McNamara, A., Popovic, Z., and Stam, J. Keyframe control of smoke simulations. *ACM Transactions on Graphics, 22*, 3, 716-723, 2003.