Anisotropic Particle Level-Set Method for Multiphase Fluid

Po-Ram Kim, Ho-Young Lee, Jung Lee and Chang-Hun Kim
Department of Computer Science and Engineering
Korea University
Seoul, Korea
Email: {poramkim, flymist, airjung, chkim}@korea.ac.kr

This paper presents how to track the surface of a multiphase fluid more accurately by using the particle level set method with anisotropic instead of spherical particles. While we use the weighted version of principal component analysis (WPCA) to construct the anisotropic particles, its computational cost is high. We adopt the distribution of particles from the directional derivative to generate the anisotropic particles. Compared to particle level set method, our approach provides more details of surface, corrects numerical dissipation, and preserves the volume of the fluid. Furthermore, we present particle-based fluid simulations with surface reconstruction that uses anisotropic particles.

Keywords: *Fluid simulation, Anisotropic particle, Particle level-set method*

ACM Classifications: *I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism*

1. Introduction

Accurately capturing the interfaces of a multiphase fluid is a challenging problem in computer graphics. The level set method within an Eulerian simulation can track the free surface of a liquid (Foster and Fedkiw, 2001). However, the standard grid-based semi-Lagrangian advection method only has first-order accuracy, so there is a large amount of numerical dissipation. Attempts to track the fluid interface more accurately have included the use of various triangle meshes or marker particles. Mesh-based surface tracking combines existing resampling methods with the use of convex hulls to connect surface features during topological changes (Treuille *et al*, 2003; Wojtan *et al*, 2009). The particle level-set method corrects the level-set near the surface. Enright *et al* (2002) added Lagrangian particles to the level-set in order to represent surface details more accurately. To implement this method, marker particles are seeded near the surface and advected by a velocity which is tri-linearly interpolated at the particle position. Then the particles can be used to correct errors caused by numerical dissipation. To calculate the level-set on both sides of the fluid interface, the particles are considered as spheres, with radii determined by their distance from the surface. This method is widely used because it is very simple and resulting surface preserves the volume of the fluid.

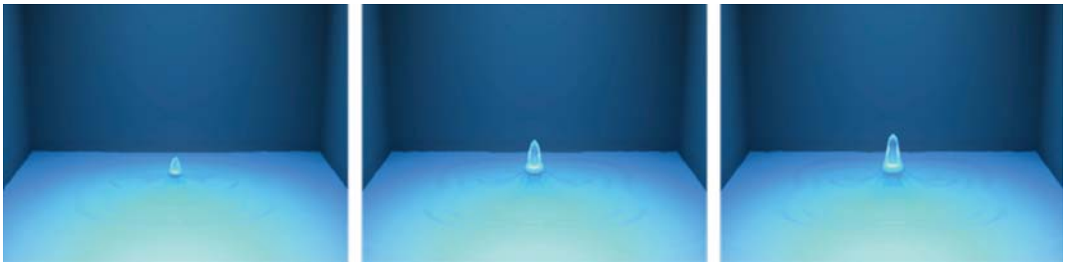
In this paper, we propose an anisotropic particle level-set method to achieve a more accurate fluid interface than the spherical particles. We create an anisotropic particle by considering distribution of particles. Initially we used a weighted version of principal component analysis (WPCA). This

Manuscript received: 14 August 2012
Communicating Editor: Seok-Hun Kim

represents the fluid surface accurately, but the search for neighbouring particles required to calculate singular value decomposition involves a high computational cost. So we devise a new method in which we use directional derivative to generate the anisotropic particles. This reduces the computational costs but still allows the fluid interface to be tracked more accurately.

2. Related Work

The simulation of liquid using the Navier-Stokes equation has been researched for a long time. Foster and Metaxas (1996) developed a three-dimensional Navier-Stokes method for fluid simulation. Stam (1999) proposed a semi-Lagrangian integration scheme to simulate unconditionally stable fluids using a three-dimensional grid. Losasso *et al* (2004) utilized an adaptive octree structure to obtain a high-resolution surface. Hong and Kim (2005) considered surface tension between multi-phase fluids using the ghost fluid method (GFM) to deal with discontinuities at the fluid interface. To represent the fluid surface more accurately, Enright *et al* (2002) proposed the particle level set method. Then, they improved this method by introducing a semi-Lagrangian approach to track the surface more accurately and rapidly. Mihalef *et al* (2007) handled the dynamics of a liquid and its surface colour. Ianniello and Mascio (2010) tracked the interfaces by Lagrangian oriented particles in conjunction with a level-set. Advection of simulation is also for greater accuracy, while CIP (Song *et al*, 2005) ensure high order accuracy. Furthermore there are several hybrid approaches for bubble (Greenwood and House, 2004) and splash (Hong *et al*, 2008; Kim *et al*, 2006). Losasso *et al* two-way coupled particle level-set and SPH (Lee *et al*, 2009; Losasso *et al*, 2008) in order to simulate diffuse regions such as splashing. Kim *et al* (2012) simulate an ellipsoidal shape of an air bubble by a drag force on its upper surface. Wojtan *et al* (2010) were able to represent detailed fluid surfaces with thinner features using triangle meshes. Our anisotropic particle level-set approach has been inspired by anisotropic particle methods. Liu *et al* (2006) employed anisotropic kernels in the SPH simulation. Yu *et al* (2010) formulated anisotropic smoothing-kernels using the WPCA method. Jo *et al* (2011) simulated SPH-based fluids using anisotropic kernels formed by the particle velocities. We refer to the works of Tatar *et al* (2009), Zheng *et al* (2006) and Rahman and Murshed (2008) for creation of anisotropic kernel. Rahman and Murshed propose a texture generation method by computing their movement of a motion distribution followed by the generation of image frames. Zheng *et al* detected a pattern by judging the eclipse and Support Vector Machines (SVM). Other interesting works include viscoelastic fluids (Goktekin *et al*, 2004), control methodology (Treuille *et al*, 2003), vortex particle (Selle *et al*, 2005) and sub-grid turbulence model (Schechter and Bridson, 2008).



(a) (b) (c)

Figure 1: Water drop tower – (a) PLS (b) APLS with WPCA (c) APLS with directional derivative

3. Particle Level-set Method (PLS)

This paper uses the Navier-Stokes equation to simulate large volumes of liquid. The momentum conservation equation is

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

and the mass conservation equation is

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

where **u**= (u,v,w) is velocity, *p* is pressure and **f** is the sum of the external forces, including gravity and control forces. We use octree structures (Losasso and Fedkiew, 2004) for fast grid simulation, back and forth error compensation and correction (BFECC) (Kim and Rossignac, 2005) to achieve 2nd-order accuracy in fluid volume preservation, and the particle level-set method (Enright and Fedkiw, 2002) to represent complicated fluid surfaces.

First, marker particles are seeded in the surface region. The radius r_p of a particle is as follows:

$$r_p = \begin{cases} r_{\max} & \text{if} \quad s_p\phi(\mathbf{x}_p) > r_{\max} \\ s_p\phi(\mathbf{x}_p) & \text{if} \quad r_{\min} \le s_p\phi(\mathbf{x}_p) \le r_{\max} \\ r_{\min} & \text{if} \quad s_p\phi(\mathbf{x}_p) < r_{\min} \end{cases} \tag{3}$$

where s_p is the sign of the particle, $ø(\mathbf{x}_p)$ is the implicit function and $\mathbf{x}_p$ is the particle position. The minimum radius is 0.1·min(*dx*,*dy*,*dz*) and the maximum radius is 0.5·min(*dx*,*dy*,*dz*). In most of the examples presented later in this paper, 32 marker particles are used in each cell. Then the level-set is integrated using Equation (1), while the particles are advected with the interpolated velocity at their positions. The error correction scheme proposed by Enright *et al* (2002) uses spherical particles. A spherical implicit function $ø_p(\mathbf{x})$ is determined by the marker particle radius, as follows:

$$\phi_p(\mathbf{x}) = s_p(r_p - |\mathbf{x} - \mathbf{x}_p|) \tag{4}$$

where r_p is the particle radius and **x** is the position of the node. Then a new value of the level-set is determined by comparing the spherical implicit function $ø_p(\mathbf{x})$ with the current level-set value. The positive and negative level-set values are then obtained as follows:

$$\phi^+ = \max_{\forall p \in \mathbf{E}^+}(\phi_p, \phi^+) \tag{5}$$

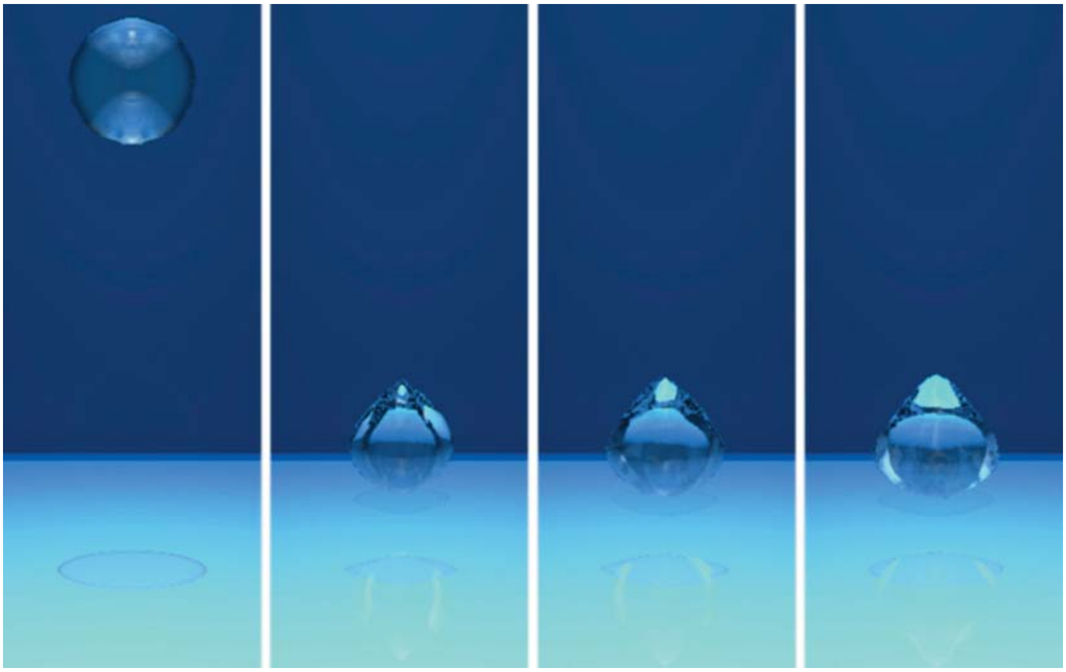
$$\phi^- = \min_{\forall p \in \mathbf{E}^-}(\phi_p, \phi^-) \tag{6}$$

where $ø^+$ is the level-set value in the $ø>0$ region, $ø^-$ is the level-set value in the $ø<0$ region, $\mathbf{E}^+$ is the set of escaped positive particles, and $\mathbf{E}^-$ is the set of escaped negative particles. More details on this are given elsewhere (Enright and Fedkiw, 2002; Enright and Fedkiw, 2005).

4. Anisotropic Particle Level-set Method (APLS)

We use anisotropic, instead of spherical, particles in the particle level-set method. We specify an anisotropic particle by its position, its three axes and its magnitudes along three axes.

4.1 Determining Particle Axes and Magnitudes Using WPCA



(a) (b) (c) (d)

Figure 2: Water-ball drop
(a) The initial water-ball drop (b) PLS (c) APLS with WPCA (d) APLS with directional derivative

We first determine the axes of a particle from the distribution of neighbouring particles by employing the WPCA Method (Yu and Turk, 2010). We compute weighted mean $\mathbf{x}_i^w$ from the neighbouring particle to construct the covariance matrix, as follows:

$$\mathbf{x}_i^w = \frac{\sum_j w_{ij}\mathbf{x}_j}{\sum_j w_{ij}} \min_{\forall p \in \mathbf{E}^-}(\phi_p, \phi^-) \tag{7}$$

The weight function w_{ij} is

$$w_{ij} = \begin{cases} 1-(\|\mathbf{x}_i - \mathbf{x}_j\|)^3 & \textit{if} \qquad \|\mathbf{x}_i - \mathbf{x}_j\| < r_i \\ 0 & \textit{otherwise} \end{cases} \tag{8}$$

The weighted w_{ij} function is calculated by using neighbour particles which have the same sign within radius r_i. Then, the covariance matrix $\mathbf{C}_i$ is obtained as follows:

$$\mathbf{C}_i = \frac{\sum_j w_{ij}(\mathbf{x}_j - \mathbf{x}_i^w)(\mathbf{x}_j - \mathbf{x}_i^w)^T}{\sum_j w_{ij}} \tag{9}$$

For each marker particle, we perform a singular value decomposition of the covariance matrix $\mathbf{C}_i$ to obtain the eigenvectors and eigenvalues, which become the axes and magnitudes ($\sigma_1 < \sigma_2 < \sigma_3$) of the anisotropic particle.

4.2 Error Correction using Anisotropic Particles

Because we use anisotropic particles, we need to modify the error corrections scheme of the particle level-set method. We can obtain the radius of an anisotropic particle r_p in any direction by solving Equation (10). The local coordinate system of each particle transforms the position of the node. The local coordinate system is determined by three axes in Section 4.1. We can calculate r_p for an arbitrary direction as follows:

$$r_p = \sqrt{\frac{1}{\frac{x_{lx}^2}{\sigma_1'^2} + \frac{x_{ly}^2}{\sigma_2'^2} + \frac{x_{lz}^2}{\sigma_3'^2}}} \tag{10}$$

where $\mathbf{x}_l = (x_{lx}, x_{ly}, x_{lz})$ is the position of node in the local coordinate system, σ'_1 is the shortest distance ($\o_p$) to the surface, $\sigma'_2 = (\sigma_2 / \sigma_1)\o_p$ and $\sigma'_3 = (\sigma_3 / \sigma_1)\o_p$. So we apply this r_p to Equation (4) and obtain the new $\o_p(\mathbf{x})$ using anisotropic particle. This costly approach improves the representation of the level-set surface, but it takes about 21 times longer than particle level-set method. This is because the data-structures used in the anisotropic particle level-set method using WPCA are inappropriate for neighbour searching. We could use fewer anisotropic particles but then the resulting surface would be worse. We therefore introduce a new method to avoid this problem in section 4.3.

Algorithm 1: APLS Method

```
for all leaf nodes do
    (re)seed marker particles and set their radius
for all leaf nodes do
    for all particles i do
        Time integration of marker particles
for all leaf nodes do
    Time integration of the implicit function
for all leaf nodes do
    for all particles i do
        delete and add marker particles
        compute updated $\phi_p(\mathbf{x})$ using PLS
for all leaf nodes do
    for all particles i do
        compute normal at the its position ($\nabla\phi$)
        set minor axis ($\mathbf{e}_1$)
        set major axis $\mathbf{e}_2$ using the directional derivative
        set last axis $\mathbf{e}_3$ = cross-product ($\mathbf{e}_1$ ,$\mathbf{e}_2$)
for all leaf nodes do
    for all particles i do
        compute new $\phi_p(\mathbf{x})$ using the anisotropic particles
```

4.3 Reducing of Computational Cost using the Directional Derivative

If the number of anisotropic particles in APLS using WPCA decreases, the computational cost also decreases but the results would be of lower quality. Therefore, we propose that the new method has good performance in time consumption as well as improves representation of the surface. We generate an anisotropic particle using directional derivative instead of WPCA method. First, we discard the WPCA calculation. Second, we update the level-set using PLS. Third, we determine three axes of the anisotropic particle using directional derivative; the error correction module is computed in the anisotropic particle level-set method. More details are described in Algorithm 1.

The minor axis $\mathbf{e}_1$ is determined from the gradient of ϕ. The major axis is the axis with the minimum variation of level-set value. The direction of minimum variation is calculated using the directional derivative, as follows:

$$\nabla_{\mathbf{e}}\phi(\mathbf{x}) = \frac{\phi(\mathbf{x}+h\mathbf{e}) - \phi(\mathbf{x}-h\mathbf{e})}{2h} \tag{11}$$

where $\phi(\mathbf{x})$ is the level-set value at the position $\mathbf{x}$ and $\mathbf{e}$ is the direction that we seek, $\|\mathbf{e}\| = 1$. In our example, h is the particle radius. To find the minimum variation, we calculate the directional derivative iteratively using Equation (11) on the plane, normal to $\mathbf{e}_1$. The direction that has the minimum variation is the major axis $\mathbf{e}_2$. The $\mathbf{e}_3$ axis is calculated by the cross product of the minor axis $\mathbf{e}_1$ and the major axis $\mathbf{e}_2$. We can introduce into Equation (10) since we can obtain σ_1, σ_2, and σ_3 by calculating the directional derivative along the axes $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ using Equation (11). The values of σ_1, σ_2, and σ_3, are inversely proportional to the variation of the level-set in the corresponding direction. This simple calculation improves the performance of the anisotropic particle level-set method using directional derivative.

4.4 Additional Trials

We applied our new method to the surface reconstruction module of a particle-based fluid simulation using SPH (Muller and Gross, 2003; Muller and Gross, 2005; Adams and Guibas, 2007). Our simulations and surface reconstruction algorithms largely follow Yu and Turk (2010). We only modify their method of setting the anisotropic kernels using WPCA.

The normal vector at each particle that is used to calculating the surface tension force is the minor axis of the anisotropic particle. We calculate the rate of density change for all neighbouring particles. If a particle j has the smallest rate of density change, we temporarily align the vector $\mathbf{x}_j$-$\mathbf{x}_i$ with the major axis. The direction of the third axis is the cross product of the minor and major

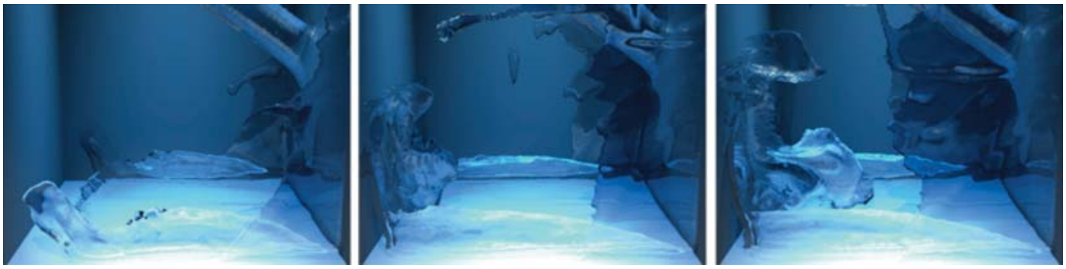

(a) **(b)** **(c)**

Figure 3: Water pouring into a tank (a) 26th frame (b) 43rd frame (c) 60th frame

Algorithm 2-1: Particle based Fluid Simulation

```
1   for all particles $i \in \mathbf{S}$ do
2       find neighbors $\mathbf{N}_i$
3   for all particles $i \in \mathbf{S}$ do
4       compute $\rho_i, p_i$
5   for all particles $i \in \mathbf{S}$ do
6       compute forces
        (gravity, pressure, viscosity, surface tension)
7   for all particles $i \in \mathbf{S}$ do (for all about previous step) do
8       compute anisotropic kernel    (compute three axes)
        →Algorithm2-2
9       compute node density (for surface reconstruction)
10  for all particles $i \in \mathbf{S}$ do
11      compute new $\mathbf{v}_i, \mathbf{x}_i$
```

axes. However, the major and minor axes must be at right align to each other. So we have to change the direction of the major axis to become the cross-product of the direction of the minor axis and the third axis. The value of σ_1, σ_2, and σ_3 are inversely proportional to the rate of density change along the corresponding axes. This approach eliminates the computational time required to obtain three eigenvectors and three eigenvalues.

Algorithm 2-2: Compute Anisotropic Particle

```
1   minor axis = normal vector of particles i
2   for all particles $i \in \mathbf{N}_i$ do
3       compute rate of density change i,j   ($\Delta\rho_{ij}$)
4       if ($\Delta\rho_{\min} > \Delta\rho_{ij}$)
5           $\Delta\rho_{\min} = \Delta\rho_{ij}$
6           major axis = $\mathbf{x}_j - \mathbf{x}_i$
7       end if
8   the other axis = cross-product (minor axis , major axis)
9   major axis = cross-product (minor axis , the other axis)
```

5. Results

We performed simulations on an Intel Core i7 CPU running at 2.93GHz, and rendered the fluid models by ray-tracing. We used BFECC (back and forth error compensation and correction) for the advection, with an octree grid. The maximum depth of the octree is 7, so yielding a 128*128*128 grid. We used Monte-Carlo integration to measure the volume of the fluid.

Figure 1 shows that the water-ball bounced back from the surface. Figure 1 (a) is the 21st frame with the PLS. It takes 0.7226 sec to run the advection module with 32 particles, and the volume of fluid decreases by 0.00129. Figure 1 (b) is the 21st frame with the APLS using WPCA. It takes 15.8472 sec to run the advection module with 32 particles, and the volume of fluid decreases by 0.00123. Figure 1 (c) is the 21st frame with the APLS using directional derivative. It takes 0.7477 sec to run the advection module with 32 particles, and the volume of fluid decreases by 0.00107. Figure 1 (c) shows complex water surface by adding trivial time.

Figure 2 shows that the water-ball was dropped onto the fluid surface. Figure 2 (a) is initial water-ball drop and Figure 2 (b) is the 7th frame with the PLS. It takes 0.6172 sec to run the advection module with 32 particles, and the volume of fluid decreases by 0.00027. Figure 2 (c) is the 7th frame with the APLS using WPCA. It takes 14.3663 sec to run the advection module with 32 particles, and the volume of fluid decreases by 0.00011. Figure 2 (d) is the 7th frame with the APLS using directional derivative. It takes 0.6226 sec to run the advection module with 32 particles, and the volume of fluid decreases by 0.00005. Figure 2 (d) is similar to Figure 2 (c) for the volume, although the computational time of Figure 2 (d) is about 0.04333 times lower than the APLS with WPCA.

Figure 3 shows water pouring into a tank. The stretching feature of the liquid's surface is observed in this physical behaviour and visual result. Figure 3 (a) shows a water source and we added the source in every frame. Figure 3 (b) shows water colliding with a wall. The anisotropic particle level-set helps the water surface maintain sharp features. Next, Figure 3 (c) shows the complex surface of the water.

Table 1 shows how the volume of fluid in Figure 1 and Figure 2 is conserved. Compared to the first frame, the large amount of volume with the PLS in the 70th frame and the 100th frame is lost. But the volume of the fluid is conserved in the third and fourth column. In the 100th frame, it takes 1.0160 sec to run the advection module with PLS and 1.0378 sec to run the advection module with APLS using directional derivative. As a result, the PLS and the APLS with directional derivative show similar performance, but the volume of fluid with APLS using directional derivative is more conserved.

Frame	**PLS**	**APLS with WPCA**	**APLS with directional derivative**
1st	1.15089 (0.6010 sec)	1.15089 (14.0895 sec)	1.15089 (0.6073 sec)
7th	1.15062 (0.6172 sec)	1.15078 (14.3663 sec)	1.15084 (0.6226 sec)
21st	1.1496 (0.7226 sec)	1.14966 (15.8472 sec)	1.14982 (0.7477 sec)
70th	1.13582 (0.8449 sec)	1.15089 (18.5929 sec)	1.14146 (0.8896 sec)
100th	1.12717 (0.0160 sec)	1.15089 (22.6637 sec)	1.13641 (1.0378 sec)

Table 1: Volume and time-cost for water ball drop

6. Conclusion and Future Work

In this paper, we present the anisotropic particle level-set method that captures the interface of the multiphase fluid accurately. The anisotropic particle is created by particles' distribution. To get the three axes, we use the gradient of level-set value and directional derivative. As a result, our anisotropic particle level-set method provides more accurate simulation than spherical particle level-set method and is faster than APLS with WPCA. However there is still numerical dissipation in too thin feature. So we will include anisotropic escaped particles. They act as splash where numerical dissipation occurred.

Acknowledgements

This work was supported by the Industrial Strategic Technology Development Program (No.10041784, Techniques for reproducing the background and lighting environments of filming locations and scene compositing in a virtual studio) funded by the Ministry of Knowledge Economy (MKE, Korea)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0005713)

References

ADAMS, B., PAULY, M., KEISER, R. and GUIBAS, L. J. (2007): Adaptively sampled particle fluids. *ACM Trans. Graph* 26.

ENRIGHT, D., LOSASSO, F. and FEDKIW, R. (2005): A fast and accurate semi-Lagrangian particle level set method, *Comput. Struct.* 83 (February), 479–490.

ENRIGHT, D., MARSCHNER, S. and FEDKIW, R. (2002): Animation and rendering of complex water surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques (New York, NY, USA), SIGGRAPH'02, ACM*: 736–744.

FOSTER, N. and FEDKIW, R. (2001): Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA), SIGGRAPH '01: 23–30.

FOSTER, N. and METAXAS, D. (1996): Realistic animation of liquids. Graph. Models Image Process. 58: 471– 483.

GOKTEKIN, T.G., BARGTEIL, A.W. and O'BRIEN, J.F. (2004): A method for animating viscoelastic fluids. *ACM Trans. Graph*. 23: 463–468.

GREENWOOD, S.T. and HOUSE, D.H. (2004): Better with bubbles, enhancing the visual realism of simulated fluid. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation SCA '04*, Eurographics Association, 287–296.

IANNIELLO, S. and DI MASCIO, A. (2010): A self-adaptive oriented particles Level-Set method for tracking interfaces. *J. Comput. Physics* 229(4): 1353–1380.

LEE, H-Y., HONG, J-M. and KIM, C-H. (2009): Interchangeable SPH and level set method in multiphase fluids *Vis. Comput.* 25: 713–718.

HONG, J-M. and KIM, C-H. (2005): Discontinuous fluids. *ACM Trans. Graph.* 24: 915–920.

HONG, J-M., LEE, H-Y., YOON, J-C. and KIM, C-H. (2008): Bubbles alive. In *ACM SIGGRAPH 2008 papers, SIGGRAPH '08* 48: 1–4.

JO, E-C., KIM, D-Y. and SONG, O-Y. (2011): A new SPH fluid simulation method using ellipsoidal kernels. In *Journal of Visualization* 14(4): 371–379.

KOREN, Y. and CARMEL, L. (2003): Visualization of labeled data using linear transformations. In *Proceedings of the Ninth annual IEEE conference on Information visualization*, 121–128.

KIM, J., CHA, D., CHANG, B., KOO, B. and IHM, I. (2006): Practical animation of turbulent splashing water. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '06, Eurographics Association*, 335–344.

KIM, B., LIU, Y., LLAMAS, I. and ROSSIGNAC, J. (2005): Flowfixer: Using BFECC for fluid simulation. In *Proceedings of Eurographics Workshop on Natural Phenomena*, 51-56.

KIM, P-R., LEE, H-Y., KIM, J-H. and KIM, C-H. (2012): Controlling Shapes of Air Bubbles in a Multi-phase Fluid Simulation. *Vis. Comput.* 28(6–8): 597–602.

LIU, M.B., LIU, G.R. and LAM, K.Y. (2006): Adaptive smoothed particle hydrodynamics for high strain hydrodynamics with material strength. *Shock Waves* 15: 21–29.

LOSASSO, F., GIBOU, F. and FEDKIW, R. (2004): Simulating water and smoke with an octree data structure. *ACM Trans. Graph.* 23: 457–462.

LOSASSO, F., TALTON, J., KWATRA, N. and FEDKIW, R. (2008): Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics* 14 : 797–804.

MIHALEF, V., METAXAS, D. and SUSSMAN, M. (2007): Textured Liquids based on the Marker Level Set. *EUROGRAPHICS 2007*, 26(3).

MULLER, M., CHARYPAR, D. and GROSS, M. (2003): Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland), SCA '03, Eurographics Association, 154–159.

MULLER, M., SOLENTHALER, B., KEISER R. and GROSS, M. (2005): Particle-based fluid-fluid interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer Animation* (New York, NY, USA), SCA '05, ACM, 237–244.

RAHMAN, A. and MURSHED, M. (2008): Dynamic Texture Synthesis Using Motion Distribution Statistics. *Journal of Research and Practice in Information Technology* 40 (2).

SCHECHTER, H. and BRIDSON, R. (2008): Evolving sub-grid turbulence for smoke animation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland), SCA '08, Eurographics Association. :1–7.

SELLE, A., RASMUSSEN, N. and FEDKIW, R. (2005): A vortex particle method for smoke, water and explosions. In *ACM SIGGRAPH 2005* Papers (New York, NY, USA, 2005), SIGGRAPH'05, ACM, 910–914.

SONG, O-Y., SHIN, H. and KO, H-S. (2005): Stable but non-dissipative water. *ACM Trans. Graph.* 24: 81–97.

STAM, J. (1999): Stable fluids. In *Proceedings of the 26th annual conference on Computer Graphics and Interactive Techniques* (New York, NY, USA), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., 121–128.

TĂTAR, D., ŞERBAN, G., MIHIŞ, A. and MIHALDEA, R. (2009): Textual Entailment as a Directional Relation. *Journal of Research and Practice in Information Technology* 41 (1).

TREUILLE, A., McNAMARA, A., POPOVI´C, Z. and STAM, J. (2003): Key-frame control of smoke simulations. In *ACM SIGGRAPH 2003* Papers (New York, NY, USA), SIGGRAPH '03, ACM, 716–723.

WOJTAN, C., THÜREY, N., GROSS, M. and TURK, G. (2009): Deforming meshes that split and merge. *ACM Trans. Graph*. 28 76 : 1-10.

WOJTAN, C., THÜREY, N., GROSS, M. and TURK, G. (2010): Physics-inspired topology changes for thin fluid features. *ACM Trans. Graph*. 29(50): 1-8.

YU, J. and TURK, G. (2010): Reconstructing surfaces of particle-based fluids using anisotropic kernels. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland), SCA'10, Eurographics Association, 217–225.

ZHENG, Z., YANG, J. and ZHU, Y. (2006): Face Detection and Recognition using Colour Sequential Images. *Journal of Research and Practice in Information Technology* 38(2).

Biographical Notes

***Po-Ram Kim** is presently a doctoral student in the Department of Visual Information Processing at Korea University. She received an MSc degree from the Department of Visual Information Processing in 2011 at Korea University and a BSc degree from the Department of Information Communication at the Sejong University of Korea in 2009. Her research interests include physically-based fluid simulation.*

Po-Ram Kim

***Ho-Young Lee** is currently a PhD student in the Department of Computer and Radio Communications Engineering at Korea University. He received his MSc degree from the Department of Computer Science in 2008 at Korea University and a BSc degree from the Department of Computer Engineering at the Korea University of Technology and Education in 2006. His current research interest is physically-based fluid simulation.*

Ho-Young Lee

***Jung Lee** is a research professor in the Department of Information & Communication in Korea University. His research interests include computer graphics, image processing, and image-based modeling/rendering.*

Jung Lee

***Chang-Hun Kim** is a professor in the Department of Brain and Cognitive Engineering at Korea University. He received his BA degree in economics from Korea University in 1979. After graduation, he joined the Korea Advanced Institute of Science and Technology (KAIST) as a research scientist, where he was involved in many national research projects in the area of computer aided design and geometric modeling. He received his PhD from the Department of Electronics and Information Science, Tsukuba University, Japan, in 1993. During 1993–1995, he headed the Human Interface and Graphics Laboratory at the System Engineering Research Institute (SERI). His current research interests include fluid animation and mesh processing. He is also a member of IEEE Computer Society and ACM.*

Change-Hun Kim