Vis Comput
DOI 10.1007/s00371-012-0770-4

ORIGINAL ARTICLE

Visual simulation of turbulent fluids using MLS interpolation profiles

Sun-Tae Kim · Jeong-Mo Hong

Abstract A detailed description of turbulent fluids based on numerical simulation is an important research topic required by many visual effects. We propose a novel method to simulate fluids with turbulent small-scale details. By inserting diffusive derivatives and divergence-free constraints to moving least-squares (MLS) fitting, we upgrade the velocity interpolation method for existing fluid solvers to enhance the subgrid accuracy. The time-step restriction of asymptotic property of diffusive derivatives is resolved by means of coupling to the constrained interpolation profile (CIP) advection framework. The proposed constrained moving least-squares interpolation profile (CMIP) method provides intuitive control over turbulence through the adjustment of one parameter as though controlling the Reynolds number with an inviscid model. The proposed method generates improved visuals of the highly turbulent fluid and is complementary to existing techniques that are currently being used.

Keywords Moving least-squares · Turbulent smoke · Simulation control · Fluid simulation

1 Introduction

Creating more detail is one of the first priorities of the visual effect supervisors who are fascinated by the realistic image sequences that can be generated by fluid simulations. This quest for detail is becoming more intense due to advances in digital projectors and other displaying devices. Consequently, computer graphics researchers are constantly trying to develop fluid simulation techniques that can produce more visual detail within the limits imposed by practical computing resources.

Turbulence is crucial to the small-scale dynamic details of fluid motion required] for many visual effects. The main obstacle to improved simulations of turbulence is the chaotic nature of fluids [21]. Despite its highly random appearance, turbulence is deterministic [7], which makes the realistic appearance of turbulent motion difficult to capture simply by increasing the spatial or temporal resolution of a grid within a numerical simulation as an example in Fig. 1 shows.

Furthermore, many graphically oriented models of fluid simulation ignore viscosity due to computational efficiency and numerical dissipation, even though the most common measure of turbulence is the ratio of velocity to viscosity, which is the basis of the Reynolds number. Excluding viscosity forces us to rely on antidissipative methods and makes it difficult to achieve physically intuitive control over turbulence.

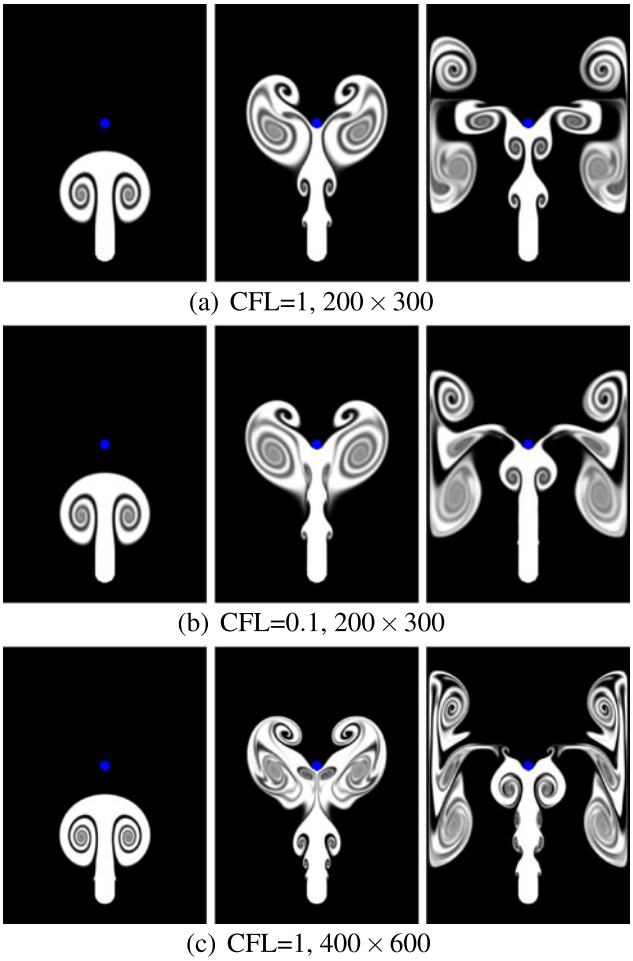
We present a new method of simulating fluids with turbulent details, which is focused on improving interpolation of the velocity field in the advection step. Our jumping-off point was the divergence constrained moving least-squares (DCMLS) method [13], in which the diffusive derivatives [14] allow the moving least squares (MLS) to include incompressibility or divergence-free constraints, and this achieves more physically plausible type of interpolation methods. However, a naïve coupling of DCMLS to fluid solvers by replacing trilinear interpolation of semi-Lagrangian method [33] appears unsuitable for computer graphics because the asymptotic property of the diffusive

Electronic supplementary material The online version of this article (doi:10.1007/s00371-012-0770-4) contains supplementary material, which is available to authorized users.

S.-T. Kim · J.-M. Hong (✉)
Dongguk University—Seoul, 30 Pildong-ro 1-gil (26 Pildong-3), Jung-gu, Seoul, 100-715, Korea
e-mail: jmhong@atelierj.pro

S.-T. Kim
e-mail: stkim@atelierj.pro

Published online: 30 November 2012



(a) CFL=1, 200×300

(b) CFL=0.1, 200×300

(c) CFL=1, 400×600

Fig. 1 The simulations in (**b**) and (**c**) do not show any significant changes in terms of turbulence when they are compared to (**a**). This means that increasing temporal and spatial resolution is not a straightforward method to control turbulence

derivatives requires small time-steps, leading to computational inefficiency. But we have made a dramatic breakthrough by coupling DCMLS to the constrained interpolation profile (CIP) advection scheme [35, 36]. By devising appropriate derivative constraints for MLS fitting and taking advantage of the derivative advection of the CIP method, we have developed a constrained moving least-squares interpolation profile (CMIP) method which is able to generate more realistic simulations of highly turbulent fluids.

A significant benefit of the CMIP method is that it offers intuitive way of controlling turbulence. The strength of turbulence can be adjusted by a single parameter; which is roughly equivalent to tweaking the Reynolds number, but without introducing the burden of a viscosity computation. CMIP is an upgrade of the interpolation modules of existing fluid solvers and can be produced by modifying well-known processes within the MLS method. It is therefore easy to implement, and remains compatible with existing techniques, even though it takes full advantage of the flexibility and approximating property of MLS. The CMIP method can supplement vortex-based methods [3, 29], which are robust generators of medium and large-scale eddies. An immediate advantage of a better background simulation would be to establish a stronger coupling to procedural methods [18, 23, 28], Eulerian–Lagrangian hybrid method [27], and subgrid projection method [20] of producing turbulent effects.

2 Related work

Since the pioneering work on direct numerical simulation of fluids by [5, 6] and [33], producing the look of a turbulent fluid has been an important topic in computer graphics. The numerical dissipation problem of the semi-Lagrangian method employed for the large steps in flow advection has motivated many studies. The vorticity confinement method of [3] was an early antidiffusion approach that utilized vorticity estimation; it was improved by employing Lagrangian particles as vorticity carriers, resulting in the vortex particle method [29]. Other authors have also improved the accuracy of the semi-Lagrangian method by introducing the back and forth error compensation and correction (BFECC) [16], MacCormack [30] and the constrained interpolation profile (CIP) method [17, 32].

Many authors have sought to reinforce subgrid details by particle simulation. [15] and [22] used SPH models to resolve subgrid splashes. [9], [12], and [34] took similar approaches to give a more detailed appearance to multiphase fluids. [8] used Lagrangian particles to capture high-speed features of gases. The introduction of physical models into subgrid dynamics has also been successful in modeling multiphase interfaces [10], flames [11, 26], and liquids [19]. A major trend is toward the generation of subgrid visuals within fluid simulations, through the introduction of procedural tools such as curl-noise [1]. Additionally, [23], [18], and [28] have used Kolmogorov turbulence theory to couple a grid-based simulation to subgrid turbulence. [20] performs subgrid projections by placing coarse-grid solutions as boundary conditions. [27] solves detailed buoyant turbulence on Lagrangian vortex sheets with coarse Eulerian background simulation. Our approach is complementary to these because the quality of base simulation always remains important, and we are enhancing a fundamental part of the numerical algorithm used for fluid simulation.

3 Outline of the Eulerian simulation of fluids

We will now briefly review the Eulerian method of fluid simulation and look at potential improvements to it. In essence, our method endows existing fluid solvers with controllable subscale turbulence by enhancing the interpolation module of the advection method.

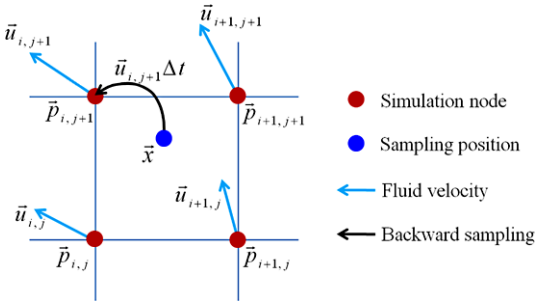


Fig. 2 The semi-Lagrangian method traces the velocity field backward by $\mathbf{u}\triangle t$ and determines the sampling position $\mathbf{x}$. The value of a variable stored and advected at sample nodes for an example at $\mathbf{p}_{i,j}$ is updated by calculating it at $\mathbf{x}$ by building an interpolation profile from neighboring node values at $\mathbf{p}_{i,j}$, $\mathbf{p}_{i+1,j}$, $\mathbf{p}_{i,j+1}$, and $\mathbf{p}_{i+1,j+1}$

The Navier–Stokes equations describing inviscid incompressible fluid motion are as follows:

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

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

where $\mathbf{u}$ is the velocity, p is the pressure, ρ is the density, and $\mathbf{f}$ is an aggregate of external forces such as gravity and control forces. A numerical simulation based on these equations advances by updating the value of $\mathbf{u}$ so that $\mathbf{u}^n$ becomes $\mathbf{u}^{n+1}$ during the nth timestep Δt. Following the spirit of Chorin's projection method [2], we discretize Eq. (1) by splitting it into two equations by introducing an intermediate status $\mathbf{u}^*$:

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

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{\nabla p}{\rho}. \tag{4}$$

To update $\mathbf{u}^n$ to $\mathbf{u}^{n+1}$, we obtain $\mathbf{u}^*$ from $\mathbf{u}^n$ by computing the advection term $-(\mathbf{u}^n\cdot\nabla)\mathbf{u}^n$ and the external force term $\mathbf{f}$ in Eq. (3). This produces $\mathbf{u}^{n+1}$ from $\mathbf{u}^*$ after enforcing incompressibility, which is expressed by Eq. (4). [33] and [3] present additional details of these procedures. Our focus is the advection component of this formulation.

3.1 Semi-Lagrangian advection

The semi-Lagrangian method, introduced by [33] and illustrated in Fig. 2, has become a standard method of solving the advection equation in computer graphics. It can be written as

$$\rho_t + (\mathbf{u}\cdot\nabla)\rho = 0, \tag{5}$$

where ρ is the field variable to be advected. This equation appears as part of Eq. (1); and it must be solved to allow Eq. (3) to advect the fluid velocity $\mathbf{u}$. We must stress the importance of the interpolation technique employed within the semi-Lagrangian method. It is not a numerical update of simulation nodes but an interpolatory profiling that determines what happens between the nodes and the method of calculating the advected values, and thus determines the subgrid accuracy.

3.2 CIP advection

The CIP method was originally developed by [35, 36] and was introduced to the computer graphics community by [32]. It employs higher-order differentiable interpolation functions for the semi-Lagrangian advection step and advects additional information such as derivatives so that the advected profile function can be reconstructed more accurately. Our method is similar in spirit to this approach, which has been shown to be effective in suppressing numerical dissipation and generating more convincing turbulent visuals [17]. If we consider a one-dimensional example, it is possible to obtain the advection equation for the derivative of ρ in the x-direction, ρ_x, by differentiating Eq. (5), as follows:

$$\frac{\partial\rho_x}{\partial t} + u\frac{\partial\rho_x}{\partial x} = -u_x\frac{\partial\rho}{\partial x}. \tag{6}$$

The terms on the left of Eq. (6) can be computed using a semi-Lagrangian method. In this method, it is necessary to evaluate ρ_{xx} at sample points by considering ρ and ρ_x, which are stored on neighboring nodes in the simulation. The effect of the term on the right-hand side of Eq. (6) was visually imperceptible in our experiments, and we therefore ignore it. Throughout this paper, we followed the unsplit-CIP-interpolation approach presented by [17] for the implementation of the CIP method. The essence of our method is combining the CIP method with the MLS method, to produce a more controllable simulation with realistic appearances.

4 MLS with derivative constraints

Because MLS is essentially an over-constrained optimization method, it can be given more flexibility by simple manipulations of its constraint functions. This is how MLS that we propose is able to meet higher-order constraints such as specified values of derivatives and the incompressibility condition. These additions allow CMIP to be used to improve the semi-Lagrangian and CIP methods. More information about the moving least-squares method and its implementation can be found elsewhere [4, 25, 31].

4.1 Moving least-squares

Subgrid profiles of fluid data can be obtained by interpolating or approximating discrete simulation results. If the nodes

$\mathbf{p}_i$ (see the red dots in Fig. 2 for a 2D example) are neighbors of a sampling point $\mathbf{x} = [x, y, z]$, then a polynomial function

$$a(\mathbf{x}) = \mathbf{b}^T(\mathbf{x})\mathbf{c} \tag{7}$$

that satisfies the constraint equations S can be constructed as follows:

$$S_l(\mathbf{p}_i) = \mathbf{b}_l^T(\mathbf{p}_i)\mathbf{c} = \phi_l, \tag{8}$$

where, l is the constraint index, $\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}), \ldots, b_k(\mathbf{x})]$ is the basis function of the fitting polynomial, $\mathbf{c} = [c_1, \ldots, c_k]$ is the unknown vector of coefficients, and ϕ is the constraining value. The number of elements in $\mathbf{b}(\mathbf{x})$ and $\mathbf{c}$ is given by $k = \frac{(d+m)!}{m!d!}$, where m is the degree of $\mathbf{b}$ and d is the spatial dimension.

A moving least-squares fit is achieved by multiplying all of the constraints in Eq. (8) by the weight functions

$$\omega\big(|\mathbf{r}_i|\big) = 1/\big(|\mathbf{r}|^2 + (\epsilon dx)^2\big), \tag{9}$$

where $\mathbf{r}_i = \mathbf{p}_i - \mathbf{x}$, ϵ is a weighting parameter, and dx is the spacing of the simulation grid. These variables are then gathered together into one equation, as follows:

$$\mathbf{W}(\mathbf{x})\mathbf{B}\mathbf{c}(\mathbf{x}) = \mathbf{W}(\mathbf{x})\Phi. \tag{10}$$

In this equation, $\mathbf{W}$ is an $N \times N$ diagonal matrix of weight functions, $\mathbf{B}$ is an $N \times k$ matrix of basis functions, and Φ is a vector in $\mathbb{R}^N$, which is the space of ϕ_l in Eq. (8). After applying the method of normal equations, $\mathbf{c}(\mathbf{x})$ can be obtained by solving this equation as a linear system. Hence, the value of a can be determined from Eq. (7). In previous formulations in [31], ϵ was used to control the accuracy of interpolation; but here it is used to adjust the strength of the turbulence, as described in Sect. 5.

Scalar interpolation can be performed by polynomial basis functions. For example in three dimensions, $\mathbf{b}(\mathbf{x}) = [1, x, y, z]$ can be used when $m = 1$. This can be extended to a vector field $\mathbf{u} = [u, v, w]^T$ by gathering three scalar interpolations into one basis matrix $\mathbf{B}$ in Eq. (10). Although a vector interpolation is generally performed by interpolating each element of the vector separately, it is necessary to formulate them as a single process in which the elements constrain each other in Sect. 4.2.

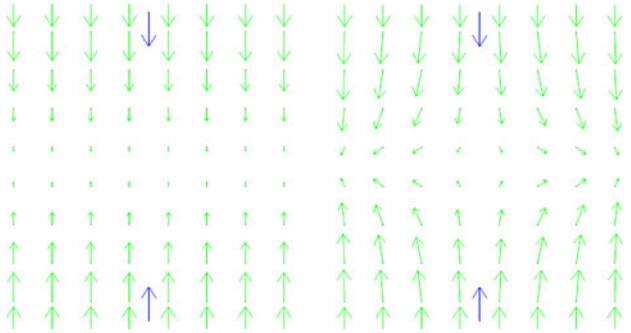
4.2 Derivative constraints

To build a bridge to fluid simulation, the MLS fit must be constrained by derivative constraints. The accurate derivative of the polynomial constraint function S, obtained by differentiating Eq. (8), is

$$\frac{\partial S}{\partial \mathbf{x}} = \frac{\partial \mathbf{b}^T}{\partial \mathbf{x}}\mathbf{c} + \mathbf{b}^T\frac{\partial \mathbf{c}}{\partial \mathbf{x}}. \tag{11}$$

Computing the second term on the right-hand side of this equation is complicated and inefficient as discussed in [14]. We therefore use the diffuse derivatives introduced by [24]:

$$\frac{\partial S}{\partial \mathbf{x}} = \frac{\partial \mathbf{b}^T}{\partial \mathbf{x}}\mathbf{c}. \tag{12}$$



(a) Linear interpolation (b) PDFMLS interpolation

Fig. 3 Comparison of linear interpolation and PDFMLS interpolation methods. *Blue arrows* are the velocities defined on simulation nodes and *green arrows* are the interpolated velocities from them

Pseudo derivative constraints can then be defined as follows:

$$S_{x\text{-derivative}} = \frac{\partial S}{\partial x} = \frac{\partial \mathbf{b}^T}{\partial x}\mathbf{c} = \frac{\partial \phi}{\partial x} \tag{13}$$

and these can be extended to pseudo divergence-free (PDF) constraints, of the form:

$$S_{\text{div}} = \nabla \cdot \mathbf{u} = \frac{\partial \mathbf{b}^T}{\partial x}\mathbf{c}_x + \frac{\partial \mathbf{b}^T}{\partial y}\mathbf{c}_y + \frac{\partial \mathbf{b}^T}{\partial z}\mathbf{c}_z = 0. \tag{14}$$

Inserting this PDF constraint into Eq. (10) formulates the DCMLS method, that gives us a physical interpolation of fluid velocity under the zero-divergence or incompressibility condition. Figure 3 shows an example of our PDFMLS interpolation profile and note that PDFMLS method generates horizontal components to satisfy divergence-free constraints, even when the given constraints have vertical components only. A more detailed introduction and experimental data are given by [13].

Since the diffusive derivatives are asymptotically accurate as proved by [14], they are not reliable when $\mathbf{x}$ is far from $\mathbf{p}_i$, due to the excessive length of the time-step $\triangle t$ (see Fig. 2). This problem is commonly encountered in many numerical simulations, and can be overcome by using a true normal method [31] to place the derivative constraint

$$S(\mathbf{x})_l = \phi_l + (\mathbf{x} - \mathbf{p}_l)^T \nabla\phi, \tag{15}$$

in some applications.

Our MLS method is easy to implement because the derivative constraints that we have formulated can be included in Eq. (10) in a similar way to the value constraints. These constraints are combined with the PDF and DCMLS constraints in an overall fluid simulation.

5 The CMIP method

MLS is a well-known technique for scattered data interpolation and it is not difficult to implement an MLS interpo-

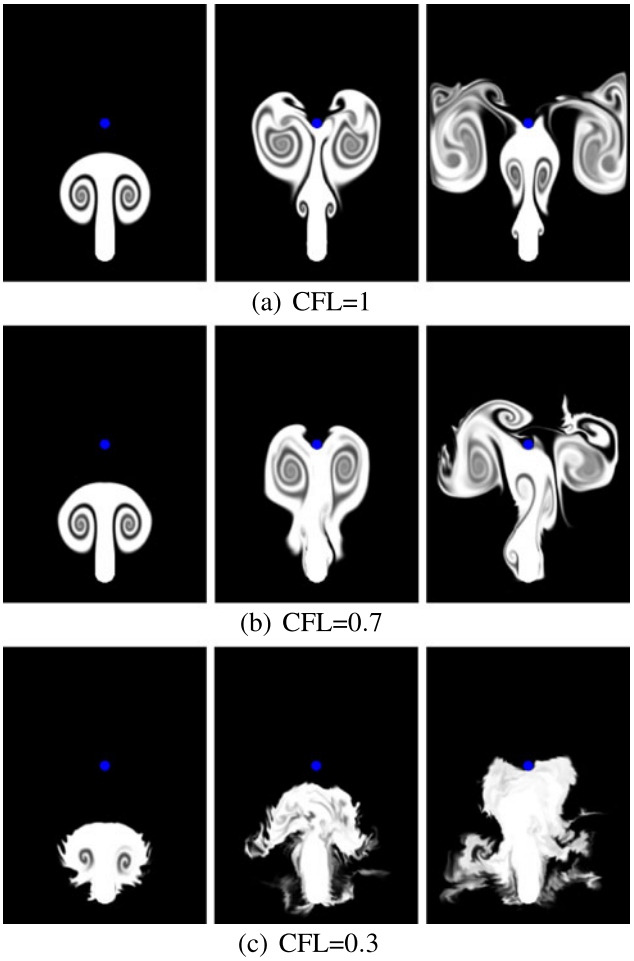


Fig. 4 We can observe that changes in temporal resolution greatly influence the turbulent details when the semi-Lagrangian method is combined with DCMLS. Smaller CFL number means smaller time steps and higher temporal resolution. Grid resolution is 200×300

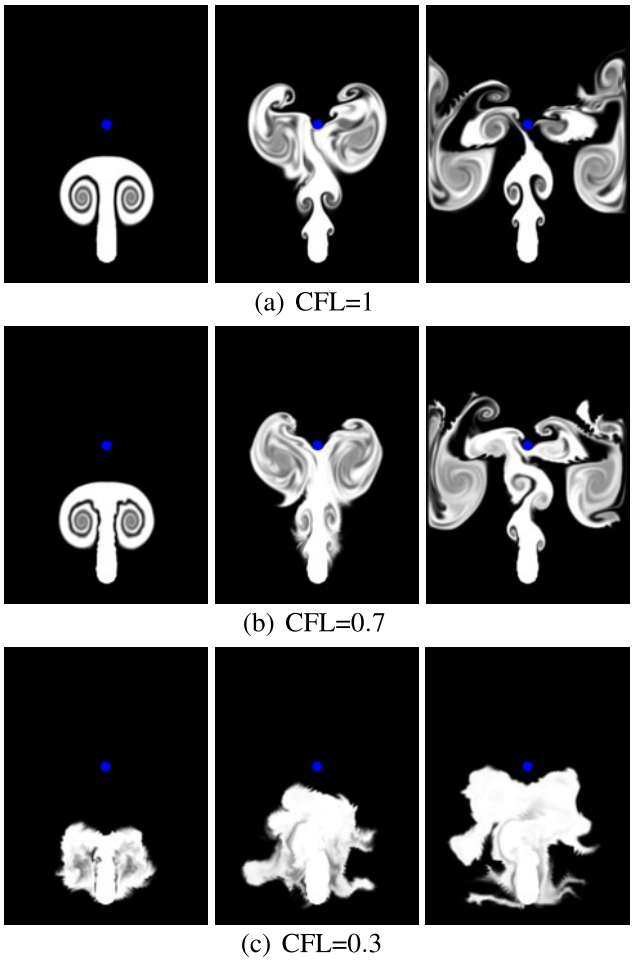


Fig. 5 The CMIP method enhances the DCMLS semi-Lagrangian method in terms of small-scale turbulence without increasing temporal resolution. See Fig. 4 for comparison. Grid resolution is 200×300 and ϵ is 3

lation module for grid-based simulations. The input is the values and positions of four nodes in two dimensions (eight nodes in 3D), and there is no singularity problem when the grid is uniform. DCMLS can easily be coupled to the semi-Lagrangian method by replacing the velocity interpolation module. For the tests in Fig. 4, we simply replaced the linear interpolation of existing semi-Lagrangian method by the DCMLS method [13] and turbulent behavior could be achieved unlike the examples in Fig. 1.

However, this DCMLS semi-Lagrangian method seems to be incapable of producing meaningful simulations, even though it produced highly turbulent effects in some of our experiments. The difficulty is that the effect of subgrid turbulence must be controlled by the magnitude of the time-step or the Courant–Friedrichs–Lewy (CFL) number, due to the asymptotic property of diffusive derivatives. This requires additional substeps to model stronger turbulence and compromises the numerical stability of the semi-Lagrangian method.

Fortunately, we have been able to resolve this problem by coupling the DCMLS semi-Lagrangian method to the CIP method as shown in Fig. 5. The CIP method uses derivatives to construct subgrid profiles, from which it interpolates the velocity $\mathbf{u}(\mathbf{x})$ and the velocity gradients $\nabla\mathbf{u}(\mathbf{x})$ at a sampling point $\mathbf{x}$, and advects them in order to reconstruct the advected subgrid profile during the next time-step.

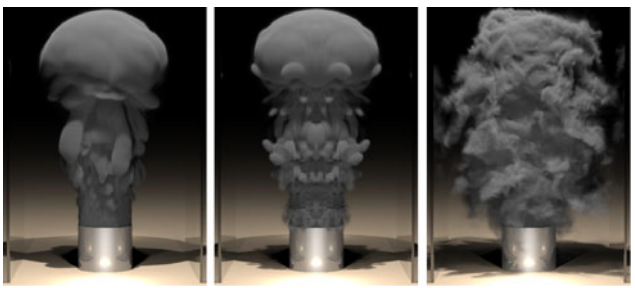
By adding the value constraint in Eq. (8) and the true normal constraint in Eq. (15) to DCMLS, we can achieve a DCMLS tri-cubic interpolation with divergence-free condition for CIP method. The diffuse derivatives of velocities $\nabla\mathbf{u}$ are calculated by the polygonal differentiation expressed by Eq. (12) in order to be advected. The pseudocode for two-dimensional CMIP method is described in Table 1. An extension to three-dimensional method is straightforward. Figure 6 demonstrates that our CMIP method can achieve convincing turbulent effects. Note that employing the CIP approach and the true normal constraints strengthens the effect of derivative constraints in advection step.

Table 1 Pseudo-code of the CMIP method for 2D semi-Lagrangian velocity advection

```
AdvectionOneTimeStep(△t)
{
  for (all nodes p_{i,j})
  {
    // Backward tracing
    find sampling point x = p_{i,j} − u^n_{i,j} △t
    find four (eight in 3D) neighboring nodes p_{l,m}     // Fig. 2

    // Adding constraints to MLS function
    Add value constraints u^n_{l,m}                       // Equation (8)
    Add derivative constraints ∇u^n_{l,m}                 // Equation (15)
    Add divergence free constraint ∇ · u(x) = 0           // Equation (14)

    // Updating velocity profile
    Solve MLS to obtain u(x) and ∇u(x)                    // Equation (10)
    u^{n+1} ← u(x)                                        // Equation (7)
    ∇u^{n+1} ← ∇u(x)                                      // Equation (12)
  }
}
```



(a) USCIP (b) BFECC (c) CMIP with $\varepsilon = 4$

Fig. 6 The proposed CMIP method simulates the improved turbulence when it is compared to the anti-dissipative advection methods such as BFECC and USCIP. Note that the small-scale dynamic motion was improved as well as the visual detail. The CFL number used in these examples is 1

Controllability is an important factor in simulations of turbulence. Figure 7 shows that the CMIP method provides robust and effective control over turbulence in an intuitive way. The idea of adjusting ϵ in the MLS weight function ω in order to control turbulence was motivated by the fact that ϵ controls the continuous transition between the interpolation and approximation of MLS (see [31] for geometric examples). Equation (9) shows that large ϵ values diminish the effects of $|\mathbf{r}|$ on determining ω and thus amplifies the effect of the divergence constraints relatively to the other constraints. We believe this is an appropriate control in a graphics context, as it allows our MLS profiler for sufficient freedom enough to serve as a subgrid solver.

6 Results and discussion

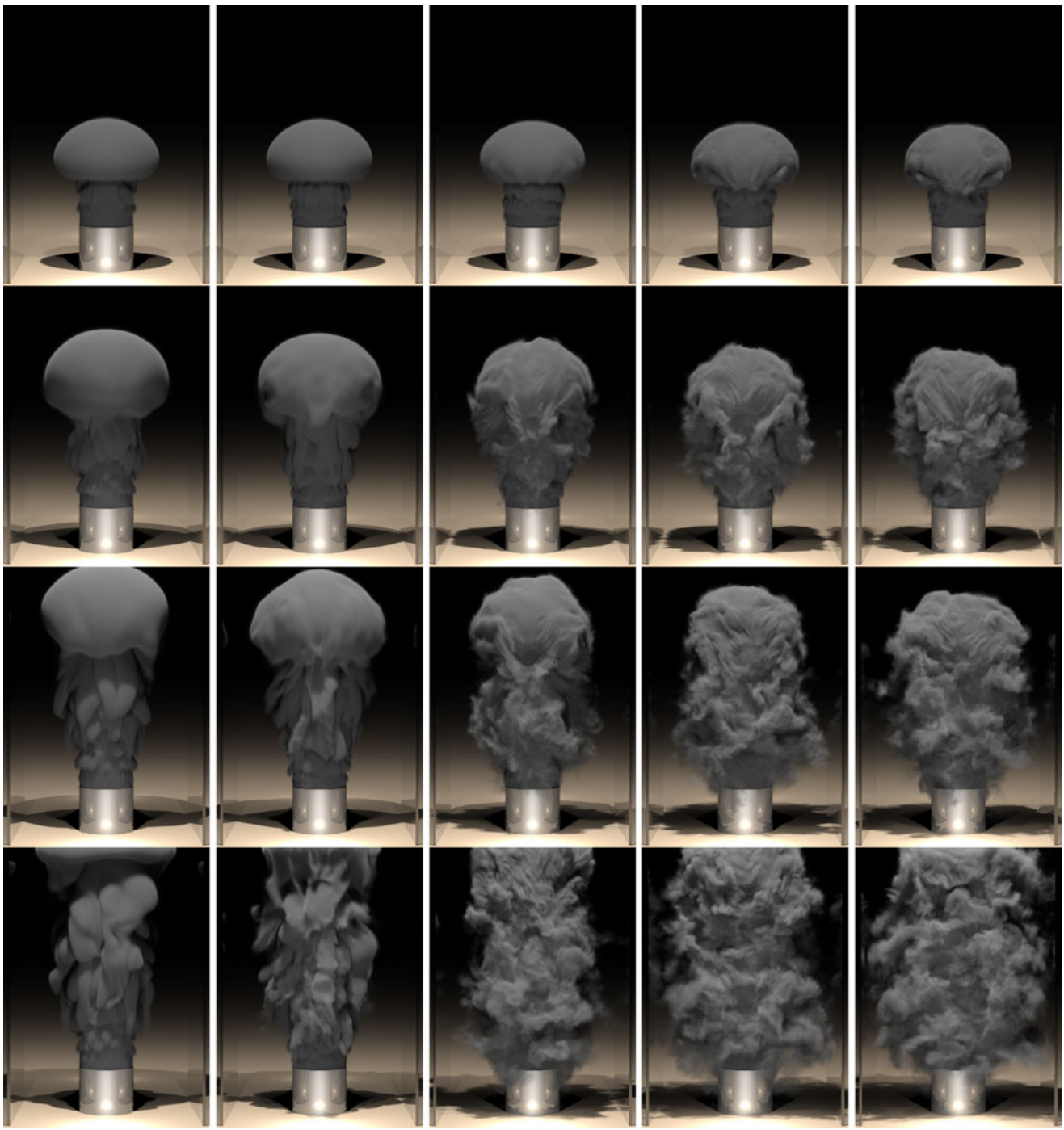
Figure 8 shows various examples generated with the CMIP method. The simulation conditions are given in Table 2. We describe that laminar-turbulent flow transitions, which are hard to capture with the steady-state turbulence model used in procedural approaches, are achieved in a natural way. Since the CMIP method can be integrated with existing fluid solvers, it could easily cope with further scenarios, controlled by established techniques involving buoyant forces, sourcing, boundary conditions, and temperature cooling, and so on. We found that the effect of a vortex-based antidissipation method such as vorticity confinement is not necessary in highly turbulent velocity fields with large values of ϵ. Results could be rendered with high temperature thanks to the strong turbulence enough to mimic premixed flames without the help of combustion dynamics. For an example, Fig. 7 shows more turbulent flows than Fig. 5 in [11] at the same resolution. A large value of ϵ, such as 4, emphasizes turbulent behavior but it can be adjusted for more subtle effects, as shown in Fig. 7. The USCIP method [17] was used for density and temperature advection in all the examples.

One limitation of CMIP simulations is a relatively high computational cost as shown in Table 2. This can be counterbalanced by the fast advances in computing machineries and the suitability of the CMIP method for parallelization or multi-core technology. The MLS calculations can be performed locally without increasing the computational complexity of the algorithm as a whole. Considering the enormous computer power routinely deployed by the visual special-effects industry to produce high-quality visuals, we believe that it would be practical to introduce our method into a current production pipeline.

7 Conclusion

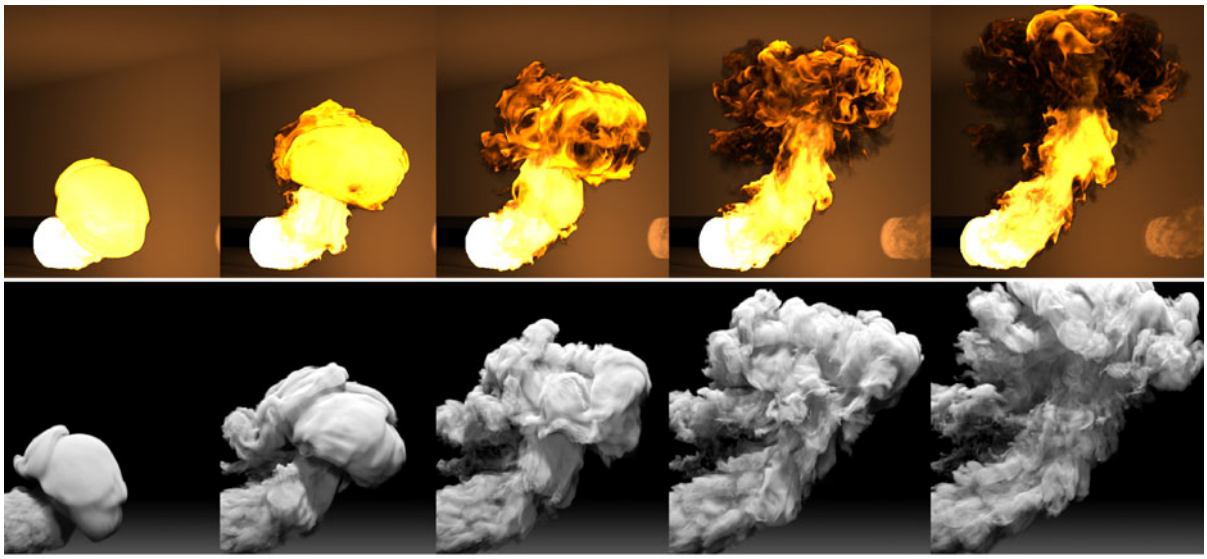
We have presented a new method of velocity interpolation which can be used to improve the ability of existing fluid simulators to show the small-scale details of highly turbulent flows. This method also permits more intuitive control, as a result of the introduction of an improved interpolation method during the advection step. We believe that our technique will be used to generate more convincing visuals of highly turbulent liquids, multiphase fluids, and explosions.

Acknowledgements This Research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2012.

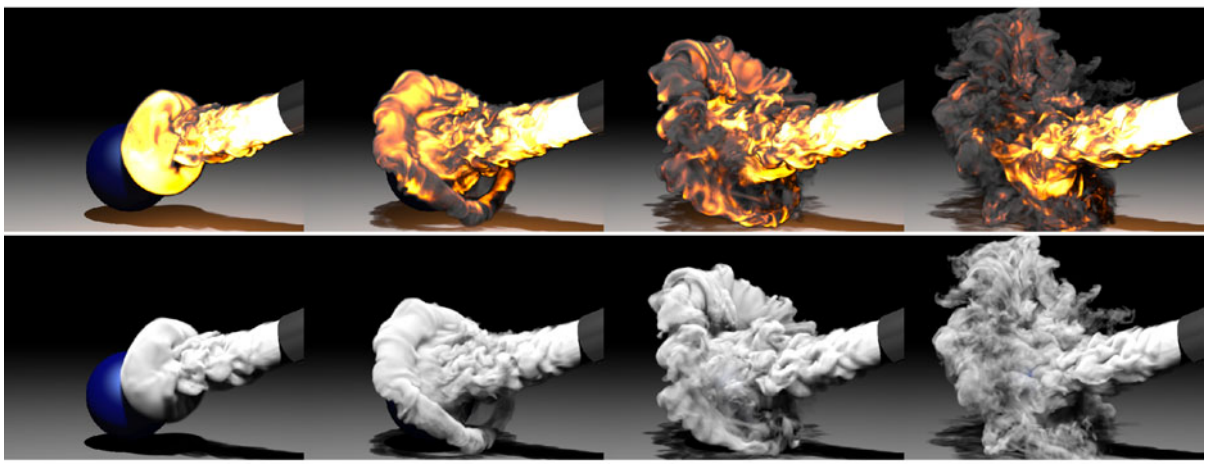


(a) $\varepsilon = 0.1$ (b) $\varepsilon = 1$ (c) $\varepsilon = 2$ (d) $\varepsilon = 3$ (e) $\varepsilon = 4$

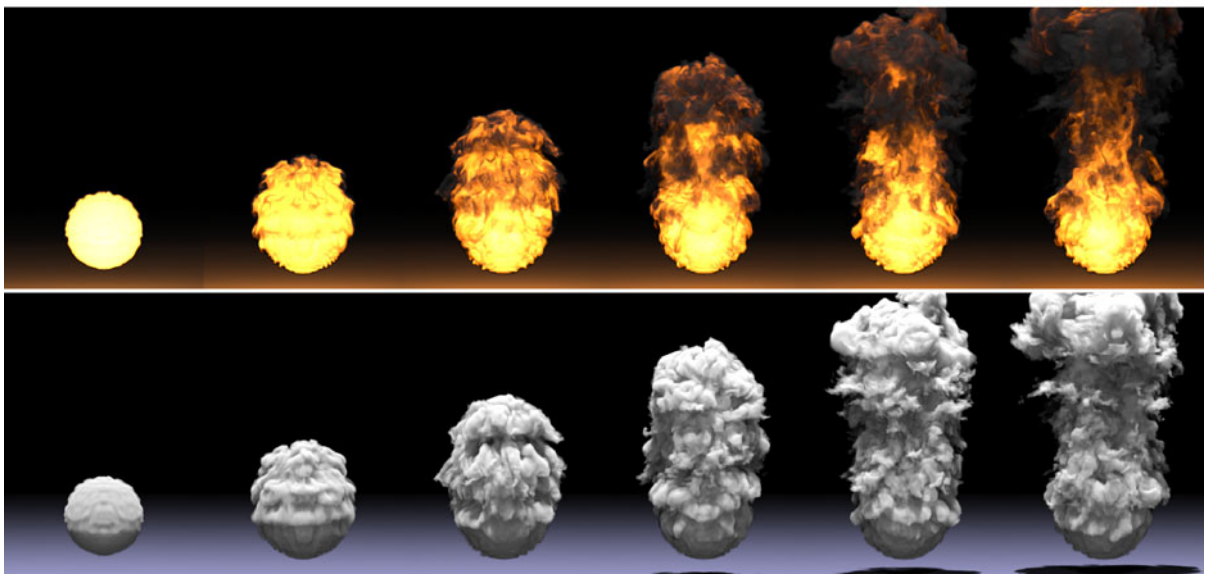
Fig. 7 *Right columns* show more turbulent behavior with larger ϵ values. Note that this turbulence control can be achieved simply by changing one parameter. Other simulation conditions were identical to the example in Fig. 6(c)



(a) Buoyant smoke



(b) Object interaction



(c) Burning sphere

Fig. 8 Examples of highly turbulent fluid motions with improved details and realistic laminar-turbulent transitions generated by our simulation method

Table 2 Simulation conditions and performance data

	Min/frame	Timestep condition	Grid resolution	Computing machine
Fig. 6(a)	5.25	CFL 1	160 × 240 × 160	PC cluster with Intel® Core™ i7 CPU@2.66 GHz
Fig. 6(b)	3.57	CFL 1	160 × 320 × 160	HP Z400 Cluster with Intel® Xeon® CPU@2.66 GHz
Fig. 6(c)	6.4	CFL 1	160 × 240 × 160	PC cluster with Intel® Core™ i7 CPU@2.66 GHz
Fig. 8(a)	14	CFL 2	600 × 400 × 400	SUN Blade Cluster with 6048 AMD Opteron™ CPU@2.3 GHz
Fig. 8(b)	7	CFL 1.5	300 × 200 × 200	HP Z400 Cluster with Intel® Xeon® CPU@2.66 GHz
Fig. 8(c)	8	7 substeps/frame	200 × 300 × 200	HP Z400 Cluster with Intel® Xeon® CPU@2.66 GHz

References

1. Bridson, R., Houriham, J., Nordenstam, M.: Curl-noise for procedural fluid flow. ACM Trans. Graph. **26**(3), 46 (2007). SIGGRAPH Proc.
2. Chorin, A.: A numerical method for solving incompressible viscous flow problems. J. Comput. Phys. **2**, 12–26 (1967)
3. Fedkiw, R., Stam, J., Jensen, H.: Visual simulation of smoke. In: Proc. of SIGGRAPH 01, pp. 15–22 (2001)
4. Feldman, B., O'Brien, J., Klingner, B.: Animating gases with hybrid meshes. ACM Trans. Graph. **24**(3), 904–909 (2005). SIGGRAPH Proc.
5. Foster, N., Metaxas, D.: Realistic animation of liquids. Graph. Models Image Process. **58**, 471–483 (1996)
6. Foster, N., Metaxas, D.: Modeling the motion of a hot, turbulent gas. In: Proc. of SIGGRAPH '97, pp. 181–188 (1997)
7. Fresch, U.: Turbulence: The Legacy of A. N. Kolmogorov. Cambridge University Press, Cambridge (1996)
8. Gao, Y., Li, C.F., Hu, S.M., Barsky, B.A.: Simulating gaseous fluids with low and high speeds. Comput. Graph. Forum **28**(7), 1845–1852 (2009). Proc. Pacific Graphics, 2009
9. Greenwood, S.T., House, D.H.: Better with bubbles: enhancing the visual realism of simulated fluid. In: Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Animat., pp. 287–296 (2004)
10. Hong, J.M., Kim, C.H.: Discontinuous fluids. ACM Trans. Graph. **24**(3), 915–920 (2005). SIGGRAPH Proc.
11. Hong, J.M., Shinar, T., Fedkiw, R.: Wrinkled flames and cellular patterns. ACM Trans. Graph. **26**(3), 471–476 (2007). SIGGRAPH Proc.
12. Hong, J.M., Lee, H.Y., Yoon, J.C., Kim, C.H.: Bubbles alive. ACM Trans. Graph. **27**(3), 481–484 (2008). SIGGRAPH Proc.
13. Hong, J.M., Yoon, Y.C., Kim, C.H.: Divergence-constrained moving least squares for fluid simulation. Comput. Animat. Virtual Worlds **19**(3–4), 469–477 (2008). CASA Proc.
14. Huerta, A., Vidal, Y., Villon, P.: Pseudo-divergence-free element free Galerkin method for incompressible fluid flow. Comput. Methods Appl. Mech. Eng. **193**, 1119–1136 (2004)
15. Kim, J., Cha, D., Chang, B., Koo, B., Ihm, I.: Practical animation of turbulent splashing water. In: Proc. of the 2006 ACM SIGGRAPH/Eurographics Symp. on Comput. Animat., pp. 335–344 (2006)
16. Kim, B., Liu, Y., Llamas, I., Rossignac, J.: Advections with significantly reduced dissipation and diffusion. IEEE Trans. Vis. Comput. Graph. **13**, 135–144 (2007)
17. Kim, D., Song, O., Ko, H.S.: A semi-Lagrangian CIP fluid solver without dimensional splitting. Comput. Graph. Forum **27**(2), 467–475 (2008). Proc. Eurographics
18. Kim, T., Thurey, N., James, D., Gross, M.: Wavelet turbulence for fluid simulation. ACM Trans. Graph. **27**(3), 50 (2008). SIGGRAPH Proc.
19. Kim, D., Song, O.Y., Ko, H.S.: Stretching and wiggling liquids. ACM Trans. Graph. **28**(5), 120 (2009)
20. Lentine, M., Zheng, W., Fedkiw, R.: A novel algorithm for incompressible flow using only a coarse grid projection. ACM Trans. Graph. **29**, 114:1–114:9 (2010)
21. Lorenz, E.N.: The Essence of Chaos. University of Washington Press, Seattle (1996)
22. Losasso, F., Talton, J., Kwatra, N., Fedkiw, R.: Two-way coupled SPH and particle level set fluid simulation. IEEE Trans. Vis. Comput. Graph. **14**(4), 797–804 (2008)
23. Narain, R., Sewall, J., Carlson, M., Lin, M.: Fast animation of turbulence using energy transport and procedural synthesis. ACM Trans. Graph. **27**(5), 166 (2008). SIGGRAPH Asia Proc.
24. Nayroles, B., Touzot, G., Villon, P.: Generalizing the finite element method: diffuse approximation and diffuse elements. Comput. Mech. **10**, 307–318 (1992)
25. Nealen, A.: An as-short-as possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. Tech. rep., TU Darmstadt (2004)
26. Nguyen, D., Fedkiw, R., Jensen, H.: Physically based modeling and animation of fire. ACM Trans. Graph. **29**, 721–728 (2002). SIGGRAPH Proc.
27. Pfaff, T., Thuerey, N., Gross, M.: Lagrangian vortex sheets for animating fluids. ACM SIGGRAPH 2012 Papers (2012)
28. Schechter, H., Bridson, R.: Evolving sub-grid turbulence for smoke animation. In: Proc. of the 2008 ACM/Eurographics Symp. on Comput. Animat. (2008)
29. Selle, A., Rasmussen, N., Fedkiw, R.: A vortex particle method for smoke, water and explosions. ACM Trans. Graph. **24**(3), 910–914 (2005). SIGGRAPH Proc.
30. Selle, A., Fedkiw, R., Kim, B.M., Liu, Y., Rossignac, J.: An unconditionally stable MacCormack method. J. Sci. Comput. **35**, 350–371 (2008)
31. Shen, C., O'Brien, J.F., Shewchuk, J.: Interpolating and approximating implicit surfaces from polygon soup. ACM Trans. Graph. **31**, 321–328 (2004). SIGGRAPH Proc.
32. Song, O., Shin, H., Ko, H.S.: Stable but non-dissipative water. ACM Trans. Graph. **24**(1), 81–97 (2005)
33. Stam, J.: Stable fluids. In: Proc. of SIGGRAPH 99, pp. 121–128 (1999)
34. Thuerey, N., Sadlo, F., Schirm, S., Muller-Fischer, M., Gross, M.: Real-time simulations of bubbles and foam within a shallow water framework. In: Proc. of the 2007 ACM SIGGRAPH/Eurographics Symp. on Comput. Animat., pp. 191–198 (2007)
35. Yabe, T., Aoki, T.: A universal solver for hyperbolic equations by cubic-polynomial interpolation I. One-dimensional solver. Comput. Phys. Commun. **66**(2–3), 219–232 (1991)
36. Yabe, T., Ishikawa, T., Wang, P.Y., Aoki, T., Kadota, Y., Ikeda, F.: A universal solver for hyperbolic equations by cubic-polynomial interpolation II. Two- and three-dimensional solvers. Comput. Phys. Commun. **66**(2–3), 233–242 (1991)

Sun-Tae Kim Graduated in Computer Science in 2010 and currently enrolled in the Master's Program at the Dongguk University in Seoul. His research interests include physically-based simulation. He is member of EnterTainment Computing Research Center of Dongguk University.

Jeong-Mo Hong is an assistant professor at Computer Science and Engineering Department. He received the B.S. degree in 2000 and the M.S. degree in 2002 in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea. After obtaining the Ph.D. degree in computer science from the Korea University in 2005, he developed several simulation techniques for visual effects as a research fellow at the Computer Science Department of Stanford University.