Arab J Sci Eng (2015) 40:269–278
DOI 10.1007/s13369-014-1486-8

RESEARCH ARTICLE - PHYSICS

Interface Reconstruction of Multiple Immiscible Fluids

Juraj Onderik · Michal Chládek · Roman Ďurikovič

Received: 30 December 2013 / Accepted: 30 May 2014 / Published online: 21 November 2014

Abstract We present a particle-based approach for animating multiple interacting liquids that can handle a number of immiscible fluids. We solve the usual problem of robust interface tracking by reconstructing the zero level set of our novel composite implicit function. Its recurrent formulation handles directly interfaces between any number of liquids including their free surfaces. To further enhance visual quality of the interfaces, we identify and refine particles in the vicinity of the surface. Our extraction scheme of near-surface particles robustly handles irregular distributions and rapid oscillations during the marking process. The surface is refined by upsampling new points along splines formed between neighbor particles. This strategy gives us smoother interfaces while having faster computation compared to a full simulation in a higher resolution. The proposed improvements can be easily implemented into the common smoothed-particle hydrodynamics framework.

Keywords Smoothed-particle hydrodynamics · Surface reconstruction · Upsampling · Immiscible fluids

1 Introduction

A number of great works have been published in recent years focusing simulation of single or two-phase fluids. Only a few other (eg. [1,2]) addressed the simulation and surface reconstruction of multiple fluids. In contrast to the single-phase scenario, multiple-phase simulations pose several interesting challenges ranging from immiscible to miscible problems.

Unique and consistent definition of interfaces between immiscible liquids is still an issue especially in particle-based simulations. Methods proposed for single fluid scenarios usually fail on the common interfaces. Gaps and overlaps emerge near the interface. In this paper, we propose a novel definition of a composite iso-surface which provides a unique definition of the common interface.

Concerning particle-based fluids, the final shape of the surface mainly depends on the number of particles and their distribution near the interface. In regions with low sampling, this is often a problem because surface reconstruction methods produce poor results. To cope with the low resolution, we propose an upsampling of the fluid used solely for the surface reconstruction.

Our contribution can be summarized into the following points:

Extraction of surface particles: We identify particles near the surface using a novel strategy overcoming particle clustering problems of Zhang et al. [3].
Upsampling of surface particles: We interpolate particles near the surface using a spline-based upsampling scheme generating new surface particles and thus diminishing bumpiness and restoring thin regions.
Interface reconstruction of immiscible fluids: We reconstruct the interfaces between multiple immiscible fluids using a novel composite implicit function.

Throughout this paper the following nomenclature will be used:

J. Onderik · M. Chládek · R. Ďurikovič (✉)
Faculty of Mathematics Physics and Informatics, Comenius University, Mlynská dolina, 841 05 Bratislava, Slovak Republic
e-mail: roman.durikovic@fmph.uniba.sk; durikovic@fmph.uniba.sk

J. Onderik
e-mail: onderik@sccg.sk

M. Chládek
e-mail: mychalch@gmail.com

Notation	Description
$\mathbf{x}$	Position
m	Mass
ρ	Density
p	Pressure
μ	Fluid viscosity
h	Support radius
W	Kernel function
$\mathbf{v}$	Velocity vector
$\mathbf{f}$	Force vector
$\mathbf{n}$	Surface normal
$\mathbf{t}$	Tangent vector

1.1 Surface Refinement

Point upsampling has been successfully applied [4] for techniques based on the moving least squares [5,6]. Although these methods provide high-quality results, they are computationally more expensive than proposed spline-based upsampling. A multiresolution simulations was used in [7] to get higher sampling in geometric complex regions. Similar visual improvement was achieved in [8] by combining low-resolution and high-resolution simulation within specific areas. A spline-based interpolation using local neighborhood was used in [9] to upsample surfel sets. A real-time performance for a static uniformly sampled data has been achieved in [10] by using B'ezier triangle interpolation. Unfortunately the upsampled points suffer from an uneven distribution of particles. Also the interpolation is based on the assumption that all particles are located on the surface making it difficult to adopt for an smoothed-particle hydrodynamics (SPH) fluid simulation. An efficient upsampling scheme for dynamic particles based on spherical interpolation was presented in [11]. Recently, a refined surface reconstruction was achieved in [12] by seeding particles within thin underresolved regions. In contrast to our pure geometrical solution, they modify the simulation by splitting and collapsing particles of liquid.

1.2 Immiscible Fluids

High-quality visualizations of interfaces between multiple immiscible fluids were achieved using projected multiple level sets [1]. Gradient-based projection was used to fill vacuum regions and interface overlaps. Recently, regional level sets [2] were exploited to solve the increasing number of level set variables necessary to model different materials. Although an unified particle-based model for interaction of solids and fluids has been successfully provided [13,14], visualization of interfaces between more than two particle-based liquids has not been sufficiently addressed yet. In order to cope with these problems, we propose a novel concept of composite implicit function, see Sect. 3.2. Melting and burning effects were achieved using Eulerian simulation in [15,16]. Preventing particle-based immiscible fluids from mixing was achieved by Mao and Yang [17]. In terms of SPH, a two-phase flow which could be further generalized to handle multiple phases was proposed in [18]. An interaction of water and air with temperature advection and diffusion was presented in [19] and later extended into an unified particle-based simulation of solids and fluids [14]. A fractional time-step method was introduced in [20] to achieve multiphase incompressible flow. Handling of density differences on the interfaces was improved in several works. The standard SPH solver was modified in [21] to correctly handle density discontinuities on the fluid interfaces. An approach for incompressible flow handling density ratios of 100 and more is presented in [22].

2 SPH Simulation Background

For the fluid simulation, a SPH algorithm is used that is a Lagrangian fluid simulation method. The fluid is discretized by particles taken as known samples. Any property A of the fluid is approximated at point $\mathbf{p}$ by summing over neighboring particles in weighted with kernel function W as

$$A(\boldsymbol{x}) \approx \langle \mathrm{A} \rangle (\boldsymbol{x}) = \sum_j \frac{m_j}{\rho_j} A_j W\left(\boldsymbol{x} - \boldsymbol{x}_j, h\right), \tag{1}$$

where m_j, ρ_j, A_j and $\mathbf{x}_j$ are the mass, density, property A and position of particle j, respectively. W is a smoothing kernel function with the smoothing length h. To compute pressures in the fluid, WCSPH from [23] and PCISPH from [24] were used.

The pressure and viscous forces are derived from the Navier-Stokes equations and can have various forms because of the symmetrization. We used the SPH force equations proposed in [25]:

$$\boldsymbol{f}_i^{\text{pressure}} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W\left(\boldsymbol{x} - \boldsymbol{x}_j, h\right), \tag{2}$$

$$\boldsymbol{f}_i^{\text{viscosity}} = -\mu \sum_j m_j \frac{v_j + v_i}{\rho_j} \nabla^2 W\left(\boldsymbol{x} - \boldsymbol{x}_j, h\right), \tag{3}$$

where μ is the viscosity of the fluid and p_i and v_i are the pressure and velocity of ith particle, respectively. The force acting on a particle is then computed as

$$\boldsymbol{f}_i = \boldsymbol{f}_i^{\text{pressure}} + \boldsymbol{f}_i^{\text{viscosity}} + \boldsymbol{f}_i^{\text{external}}, \tag{4}$$

where $\mathrm{f}_{\mathrm{i}}^{\text{external}}$ are external forces such as gravity or surface tension.

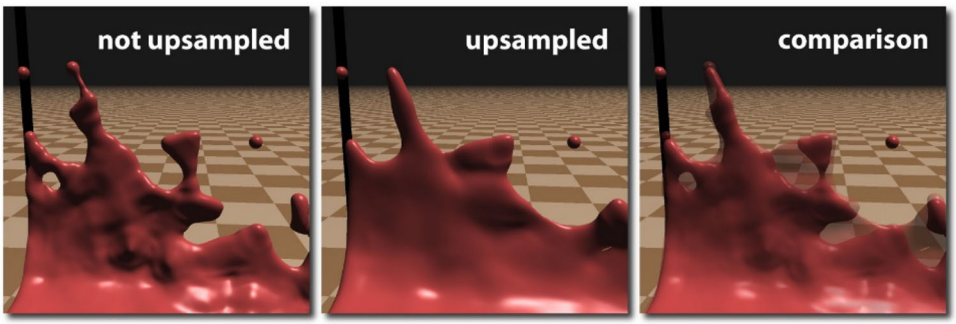

Fig. 1 Comparison of the surface reconstruction in a splashing scenario without (*left*) and with (*middle*) upsampling. Notice the filled regions (*right*) using upsampling. Iso-radius was set to $r = 1.7h$. Reconstruction was done using the distance-based approach from [26]

We have used the kernel functions proposed in [25] proposed and defined as:

$$W_{\text{poly6}}(\boldsymbol{r}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - \|\boldsymbol{r}\|^2)^3 & \|\boldsymbol{r}\| \le h \\ 0 & \|\boldsymbol{r}\| > h \end{cases}, \tag{5}$$

$$W_{\text{spiky}}(\boldsymbol{r}, h) = \frac{15}{\pi h^6} \begin{cases} (h - \|\boldsymbol{r}\|)^3 & \|\boldsymbol{r}\| \le h \\ 0 & \|\boldsymbol{r}\| > h \end{cases}, \tag{6}$$

$$W_{\text{vi}}(\boldsymbol{r}, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{\|\boldsymbol{r}\|^3}{2h^3} + \frac{\|\boldsymbol{r}\|^2}{h^2} + \frac{h}{2\|\boldsymbol{r}\|} - 1 & \|\boldsymbol{r}\| \le h \\ 0 & \|\boldsymbol{r}\| > h \end{cases}. \tag{7}$$

Kernel function from Eq. (5) is used to compute density of the fluid, while kernel functions from Eqs. (6) and (7) are used in computation of pressure and viscosity forces, respectively.

3 Methods

3.1 Surface Reconstruction with Upsampling

Compact and smooth surface is essential to obtain visually plausible results of the simulated liquids. Our approach enhances the visual quality of a fluid surface by surface particle refinement. We propose a robust strategy to extract particles near surface, overcoming the particle clustering problems of Zhang et al. [3]. Simple and efficient spline-based interpolation scheme is proposed to generate new surface particles without the knowledge of the surface geometry, see Fig. 1 (Fig. 2).

3.1.1 Extracting Particles Near the Surface

The key idea of our method of extracting near-surface particles is to measure the difference between some (interior or exterior) point and its central point used for surface reconstruction in [26]. The central point $\mathbf{C}(\mathbf{x})$ is a weighted average of the neighbor particles normalized with iso-density W_j:

$$C(\boldsymbol{x}) = \frac{\sum_j \frac{1}{w_j} W(\boldsymbol{x} - \boldsymbol{x}_j, h) \boldsymbol{x}_j}{\sum_j \frac{1}{w_j} W(\boldsymbol{x} - \boldsymbol{x}_j, h)}. \tag{8}$$

The *iso-density* w_j of jth particle is calculated using the SPH approximation of unit mass of the neighboring particles

$$w_j = W(\boldsymbol{x}_j - \boldsymbol{x}_i, h), \quad W(\boldsymbol{r}, h) = \left(1 - \frac{\|\boldsymbol{r}\|^2}{h^2}\right)^3, \tag{9}$$

where $W(\mathbf{r}, h)$ is a polynomial smoothing kernel function.

First, we calculate the displacement between each particle $\mathbf{x}_i$ and its central point $\mathbf{C}(\mathbf{x}_i)$. Since the center point is actually a weighted and iso-density nor-malized local center of mass, the difference vector $\mathbf{x}_i - \mathbf{C}(\mathbf{x}_i)$ points outside of the fluid (for near-surface particles). Next, we virtually shift particles along the displacement vectors into their new positions $\mathbf{r}_i$ and sample the final displacement $\mathbf{d}_i$ between new positions $\mathbf{r}_i$ and respective center points $\mathbf{C}(\mathbf{r}_i)$. Formally, this can be summarized as

$$\boldsymbol{d}_i = \boldsymbol{r}_i - \boldsymbol{C}(\boldsymbol{r}_i), \quad \text{where } \boldsymbol{r}_i = \boldsymbol{x}_i + (\boldsymbol{x}_i - \boldsymbol{C}(\boldsymbol{x}_i)). \tag{10}$$

As shown in Fig. 3 (left), surface particles detection using our improved displacement criterion still fails for isolated particles. We use a simple strategy to prevent the problem with detecting isolated particles. We define a *density decay* δ_i and combine it with the displacement length $\|\boldsymbol{d}_i\|$ to get the final detection criterion:

$$d_i = \text{clamp}\left(\frac{\gamma}{h}\|\boldsymbol{d}_i\|, 0, 1\right) + \text{clamp}(\delta_i, 0, 1),$$
$$\delta_i = \beta\left(\alpha - \frac{\rho_i}{\rho_0}\right), \tag{11}$$

where the α, β and γ are user-defined parameters and the function clamp is clamping to a (0, 1) interval. α is critical density ratio and β is density decay factor which together control how strongly the change in density affects the particle marking process. To properly balance the contribution of the density decay and the displacement, we scale the displacement by the displacement factor γ. We take the ith particle as a near-surface particle if d_i is higher than the user-defined threshold d_{surf}. We used $d_i = 0.5$.

3.1.2 Spline-Based Upsampling of Surface Particles

Once we extract near-surface particles, the upsampling process starts based on the following scheme. For each par-

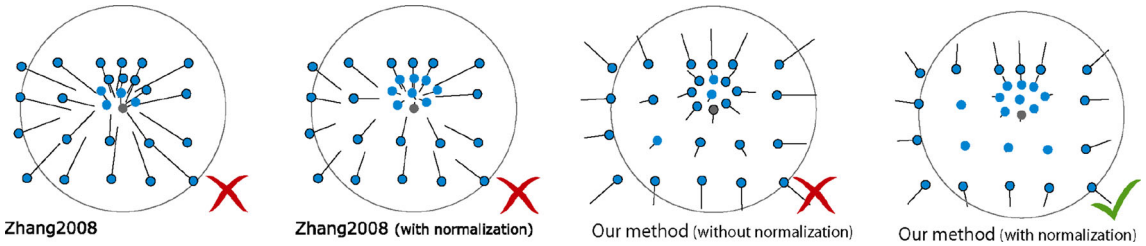

Fig. 2 Comparison of the previous method from [3] with our novel method for detecting surface particles within a cluster scenario and the contribution of the density normalization from Eq. (8). Particles with (without) *black border* are marked as surface (interior) particles. *Gray lines* represent respective displacement vectors. *Black circle* is the interaction region of the *gray particle*. *First row* both methods are without the density normalization and thus fail to correctly distinguish particles. *Second row* our method with the density normalization provides desired results

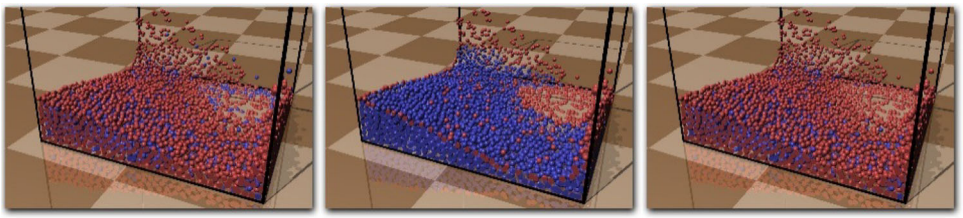

Fig. 3 *Left* the particles extracted using the displacement length. *Middle* the particles extracted using the density decay criterion. *Right* the result of the combined surface particle extraction method. The *red particles* are marked as the surface particles and the *blue* as the interior particles. These results were achieved using the following parameters: $\alpha = 0.95$, $\beta = 10$, $\gamma = 7$

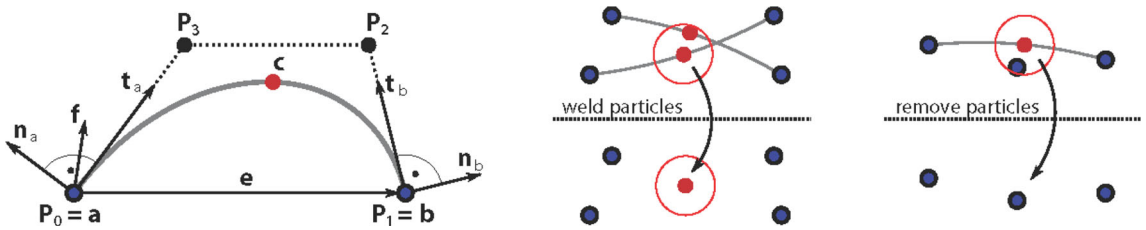

Fig. 4 *Left* our spline-based upsampling scheme for two close particles. *Middle* welding update of two middle particles that are too close to each other. *Right* middle particles created close to an original fluid particle are removed

ticle **a** and a particle **b** within the upsampling radius u, we create a new particle **c** lying at the center of a cubic Bézier curve $\mathbf{B}^3(t)$, spanned between **a** and **b**, see Fig. 4 (left). Formally, the position $\mathbf{x}_c$ of middle point **c** is defined as

$$\boldsymbol{x}_c = \frac{1}{8}\boldsymbol{P}_0 + \frac{3}{8}\boldsymbol{P}_1 + \frac{3}{8}\boldsymbol{P}_2 + \frac{1}{8}\boldsymbol{P}_3 \equiv \boldsymbol{B}^3\left(\frac{1}{2}\right), \tag{12}$$

where the cubic curve is described by its four control points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$. The starting and ending control points are equal to positions of particles **a** and **b**. Formally, the control points are defined as

$$\begin{aligned} \boldsymbol{P}_0 &= \boldsymbol{x}_a, \quad \boldsymbol{P}_1 = \boldsymbol{x}_a + \omega\|\boldsymbol{x}_a - \boldsymbol{x}_b\|\frac{\boldsymbol{t}_a}{\|\boldsymbol{t}_a\|}, \\ \boldsymbol{P}_3 &= \boldsymbol{x}_b, \quad \boldsymbol{P}_2 = \boldsymbol{x}_b + \omega\|\boldsymbol{x}_a - \boldsymbol{x}_b\|\frac{\boldsymbol{t}_b}{\|\boldsymbol{t}_b\|}. \end{aligned} \tag{13}$$

where $\mathbf{x}_a$ and $\mathbf{x}_b$ are the positions of particles **a** and **b**. The user defined smoothing factor ω controls the flatness of the resulting surface near the particles. Tangent vectors $\mathbf{t}_a$ and $\mathbf{t}_b$ describe local direction (derivative) of the curve at particles **a** and **b**. Therefore, they must be perpendicular to the local normals $\mathbf{n}_a$ and $\mathbf{n}_b$ at particles **a** and **b**, respectively. Formally, $\mathbf{t}_a$ and $\mathbf{t}_b$ are defined as

$$\begin{aligned} \mathbf{t}_a &= \mathbf{n}_a \times (\mathbf{e} \times \mathbf{f}), \quad \mathbf{e} = \mathbf{x}_b - \mathbf{x}_a, \\ \mathbf{t}_b &= (\mathbf{e} \times \mathbf{f}) \times \mathbf{n}_a, \quad \mathbf{f} = 0.5(\mathbf{n}_a + \mathbf{n}_b), \end{aligned} \tag{14}$$

where **e** is the distance vector between particles **a** and **b**, and **f** is an average of local normals. As shown in Eq. (14), the tangents lie in the plane defined by **e** and **f** (having normal $\mathbf{e} \times \mathbf{f}$) and are perpendicular to the local normals as desired.

Next, we describe how to evaluate the local normals $\mathbf{n}_a$ and $\mathbf{n}_b$ at the particles.

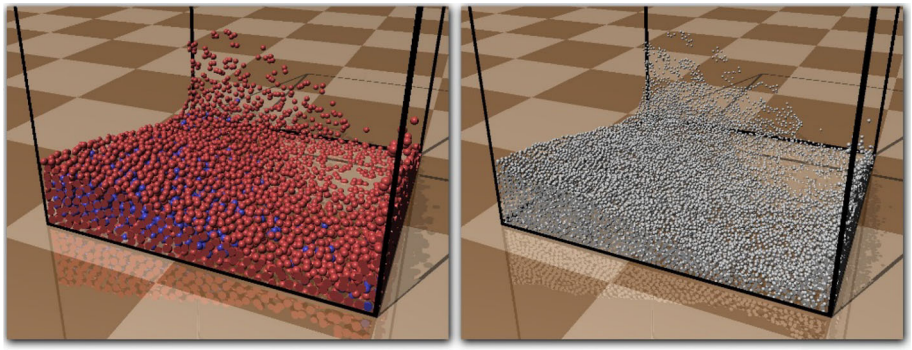

Fig. 5 *Left* the surface fluid particles are depicted with *yellow* and the interior fluid particles are depicted with *blue*. *Middle* both fluid and refinement particles. *Right* refinement particles only. Smoothing factor $\omega = 0.4$ and the welding distance was set to $0.2h$, where h is the support radius of the kernel functions used in the simulation

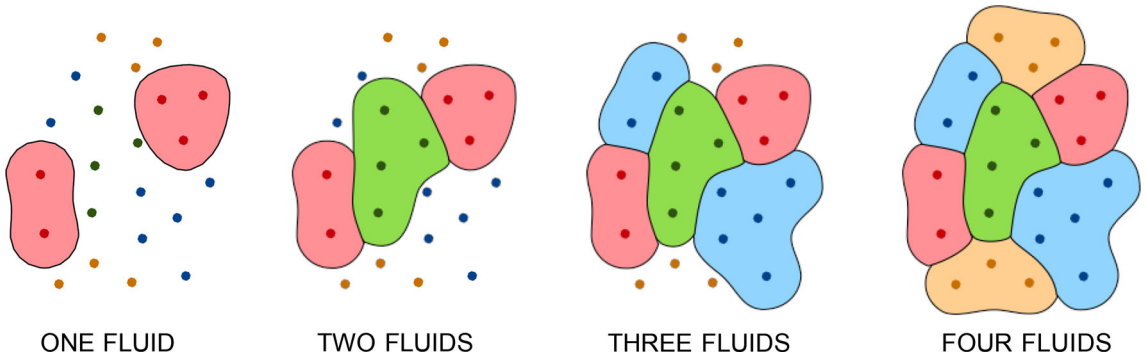

Fig. 6 2D results of our recurrent algorithm constructing a zero level set of the composite iso-potential for 4 immiscible fluids. From *left to right*, interfaces of 1,2,3 and 4 fluids are generated. Notice several sharp "corners" caused by the absolute value used in the implicit function definition

Similarly to our surface particle extraction method, for a given particle $\mathbf{a}$ we define its estimated surface normal $\mathbf{n}_a$ as

$$\begin{aligned} \boldsymbol{n}_a &= \frac{d_a}{\|d_a\|}, \\ \boldsymbol{d}_a &= \boldsymbol{r}_a - \boldsymbol{C}(\boldsymbol{r}_a), \\ \boldsymbol{r}_a &= \boldsymbol{x}_a + (\boldsymbol{x}_a - \boldsymbol{C}(\boldsymbol{x}_a)). \end{aligned} \tag{15}$$

Before a new middle particle is placed, we check whether there is any particle closer than a given *welding distance*. If we find a fluid particle being in its vicinity, we remove the generated middle particle. Otherwise, if there are two middle particles $\mathbf{c}_1$ and $\mathbf{c}_2$ too close to each other, we weld them into a new middle particle with position $\mathbf{x}_c = \frac{1}{2}(\mathbf{x}_{c_1} + \mathbf{x}_{c_2})$ by averaging their positions, see Fig. 4. The newly welded particle is then checked against the welding criterion again. The process iterates until there is no close particle to the examined particle, see Fig. 5.

Some surface visualization techniques such as splatting or screen-based techniques usually require a finer resolution thus further upsampling is needed. Our upsampling scheme can be iterated more times. In such cases, the fluid particles and the interpolation particles together become the base particles for the next iteration.

3.2 Immiscible Fluids Surface Reconstruction

We propose a general approach of surface reconstruction assuming K different immiscible fluids. We assign a fluid identifier to each particle. For the kth fluid, we have an iso-function $\varphi_k(\mathbf{x})$ and an isovalue s_k giving us an implicit function $\varphi_k(\mathbf{x}) - s_k = 0$. This implicit function defines the fluid surface with respect to the particles of kth fluid only. We have used three different definitions of the function $\varphi_k(\mathbf{x})$ from [25–27]. Such approach successfully generates an in-between interface for each fluid, and unfortunately, they penetrate each other near the common interface (Fig. 6).

We overcome this drawback by defining a composite implicit surface $\phi(\mathbf{x}) = 0$ using the following recurrent formulation

$$\begin{aligned} \phi_k(\boldsymbol{x}) &= |\varphi_{k-1}(\boldsymbol{x}) - \max(0, \phi_k(\boldsymbol{x}))|, \\ \phi_0(\boldsymbol{x}) &= 0, \\ \phi(\boldsymbol{x}) &= \phi_k(\boldsymbol{x}) - s, \end{aligned} \tag{16}$$

where $k = 1, \ldots, K$, K is the number of different fluids and s is the isolevel defining the implicit function. It is worth to note that the definition from Eq. (16) is not commutative and it depends on the order in which the fluid surfaces are combined. The fluid interface is defined by the order of the fluids. However, the simulation result using a different order of the fluids provides similar plausible results (see Fig. 7).

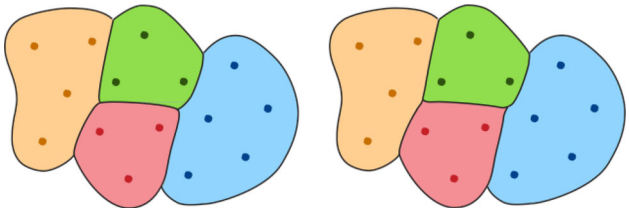

Fig. 7 Different order of fluid combining. *Left* the fluids are combined in the order *red*, *green*, *blue*, and *brown*. *Right* the fluids are combined in the order *brown*, *blue*, *red*, and *green*

As shown in Fig. 6, our composite iso-potential provides a unique zero level implicit surface definition for both the free surfaces and the fluid interfaces. Using the absolute value in our composite iso-function makes the resulting surface only piece-wise smooth and thus enabling to model the sharp features desired for complex interfaces of multiple immiscible fluids.

Further processing (e.g. ray tracing) of an interface usually requires to evaluate the normal and identify the fluid to which the examined point belongs to. Noticing that the gradient of absolute of real function $f(\mathbf{x})$ is $\nabla |f(\boldsymbol{x})| = \frac{f(\boldsymbol{x})}{|f(\boldsymbol{x})|} f(\boldsymbol{x}) = \operatorname{sgn}(f(\boldsymbol{x})) \nabla f(\boldsymbol{x})$ the surface normal $\mathbf{n}_x$ of our composite implicit function $\varphi(\mathbf{x})$ is given as

$$\begin{aligned} \boldsymbol{n}_x &= \nabla \phi(\boldsymbol{x}) \\ &= \nabla \phi_K(\boldsymbol{x}), \nabla \phi_k(\boldsymbol{x}) = \nabla |\phi_{k-1}(\boldsymbol{x}) - \max(0, \phi_k(\boldsymbol{x}))| \\ &= \operatorname{sgn}(\phi_k(\boldsymbol{x})) \nabla \phi_{k-1}(\boldsymbol{x}) \\ &\quad -\operatorname{sgn}(\max(0, \varphi_k(\boldsymbol{x}))) \nabla \phi_k(\boldsymbol{x}). \end{aligned} \tag{17}$$

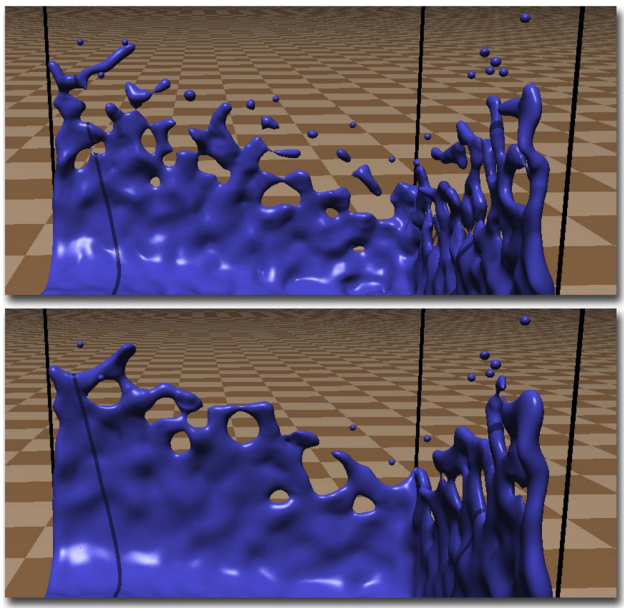

Fig. 9 Comparison of different upsampling radii r used in the upsampling of fluid surface. The upsampling radii are set according to the support radius h used in the fluid simulation. *Top* $r = h$. *Bottom* $r = 2\,\mathrm{h}$

Assuming we have K fluids identified with numbers $1, \ldots, K$, we can assign the fluid identifier at point $\mathbf{x}$ by the following classification function $F(\mathbf{x})$

$$\begin{aligned} F_k(\boldsymbol{x}) &= \begin{cases} k & 0 < \phi_k(\boldsymbol{x}) < \varphi_k(\boldsymbol{x}) \\ F_{k-1}(\boldsymbol{x}) & \text{otherwise} \end{cases}, \\ F(\boldsymbol{x}) &= F_K(\boldsymbol{x}), \end{aligned} \tag{18}$$

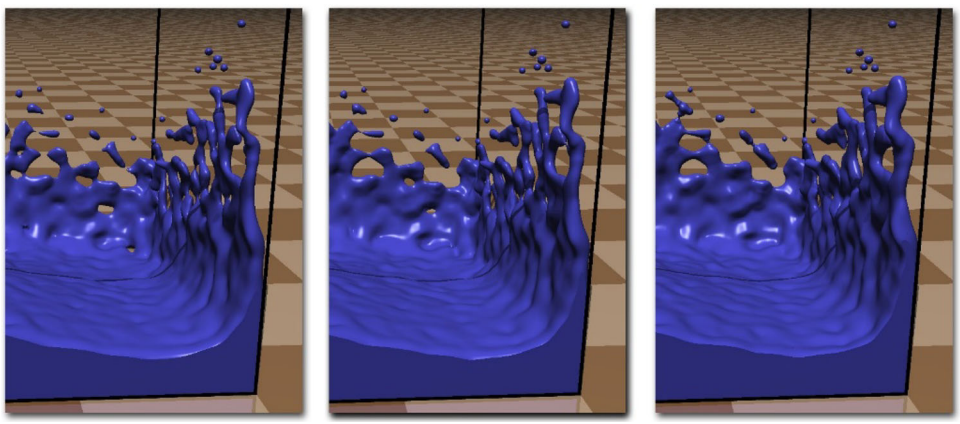

Fig. 8 Refinement of a dam break scene with 15,625 fluid particles in an aquarium with dimensions $0.25\,\mathrm{m} \times 0.25\,\mathrm{m} \times 0.25\,\mathrm{m}$. Density of the fluid is set to $1{,}000\,\mathrm{kg/m^3}$. Results using different parameter ω from Eq. (13) are compared. *Left* $\omega = 0.1$. *Middle* $\omega = 0.4$. *Right* $\omega = 0.9$

Table 1 The times of interpolation and the number of created surface particles of the spheric and the spline-based interpolation

Scene	Spheric interpolation		Spline Interpolation		Number of Interpolations
	Time	# Particles	Time	# Particles	
Dam break	0.1254 s	42,685	0.1244 s	41,712	58,909
Fountain	0.1491 s	37,051	0.1454 s	34,526	56,079

All the results in the table are measured in the dam break scene with 15,625 particles

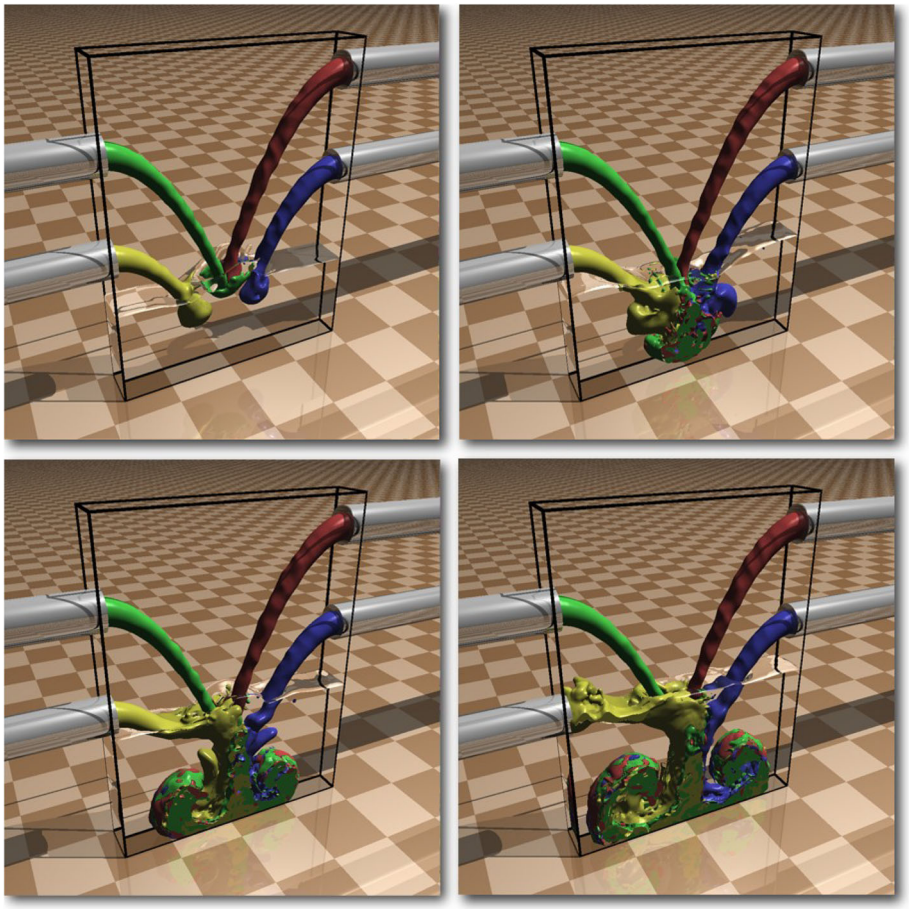

Fig. 10 An animation of four different immiscible fluid mixing in the water in an aquarium with dimension $0.65\,\text{m} \times 0.12\,\text{m} \times 0.85\,\text{m}$. All four fluids have density $1{,}000\,\text{kg/m}^3$ and flow rate $0.00242\,\text{m}^3/\text{s}$. The animation is captured in four different times

where $k = 1, \ldots, K$. Points that are outside of any fluid are identified by $F_0(\mathbf{x}) := 0$.

4 Discussion and Results

The first step of the upsampling process is the extraction of the fluid particles near the interface. The previous methods have suffered from inaccuracy in the clustered regions as can be seen in Fig. 2. The particles from inside of the fluid and near the clustered region were falsely detected as particles near the interface. The density normalization from Eq. (8) improved the results but has not solved the problem completely. The newly proposed method from Eq. (10) with the density normalization has given most accurate results. All particles from the scenario in Fig. 2 were detected correctly.

We have experimented with various setups for our extraction of near-surface particles; however, best results were achieved when setting the kernel radius h in Eqs. (8) and (9) 1.7-times larger than the kernel radius used in the fluid simulation. Further increasing of the kernel radius has no benefit in the quality of the result. The chosen parameters results in marking 2 layers of particles near the surface, see Fig. 5. Marking 2 layers is desirable because the refined surface based on a single layer can suffer from holes in the regions with lower density. Isolated particles were detected with the proposed density decay by setting critical density ratio to $\alpha = 0.95$ and density decay factor to $\beta = 10$ in Eq. (11). In the same equation, final marking was controlled by the combination of density decay with the displacement by setting displacement factor to $\gamma = 7$. Because both values in Eq. (11) are clamped to [0, 1] interval, the threshold parameter $d_{\text{surf}} = 0.5$ was used for the surface thresholding. Using these setting, the isolated particles were properly detected. According to our testing, the density decay approach and a simple criterion based on the number of neighbors produce similar results when combined with our displacement length criterion.

We demonstrate the surface upsampling in several examples. In Fig. 1, a detail of a dam break scene, where the non-refined surface is compared to the refined surface, can be seen. Different results are achieved using various parameters ω in Eq. (13). The results can be seen in Fig. 8. Setting $\omega = 0.4$ gives us the best results. Setting ω to a lower value produces holes in the surface and setting ω higher bloats the surface too much.

We have done a comparison with the spherical interpolation approach from [11]. Similar result can be achieved with

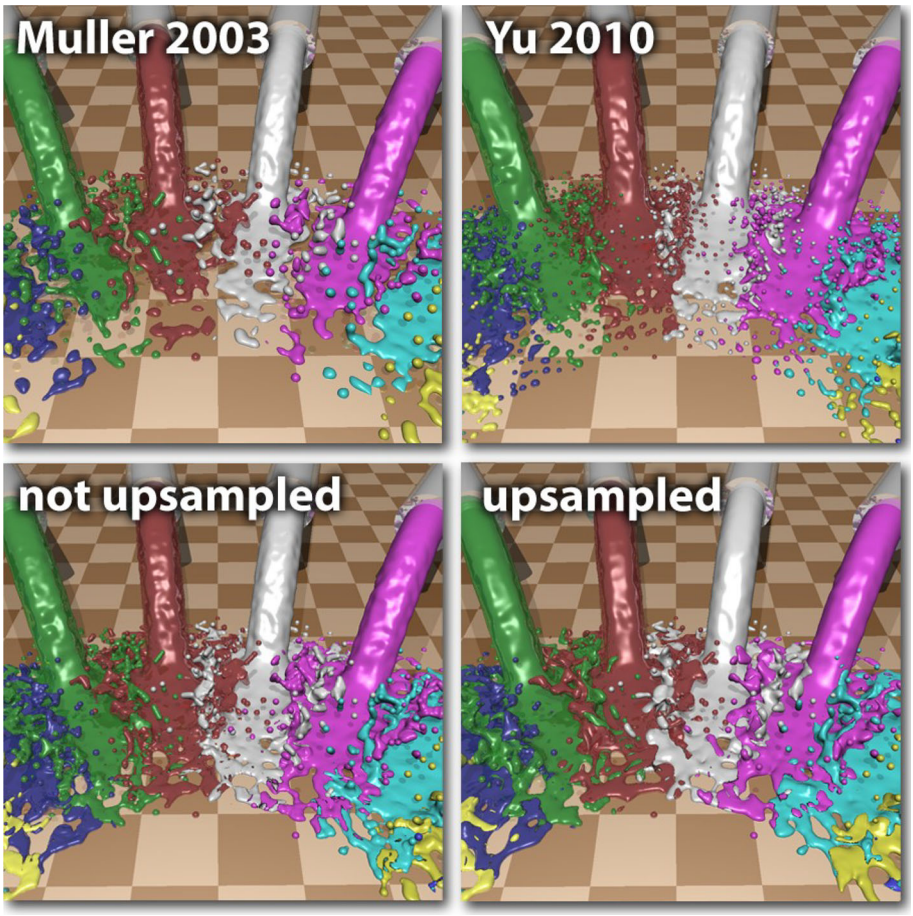

Fig. 11 Detail of mixing of 14 immiscible fluids having density 1,000 kg/m^3 and flow rate 0.0181 m^3/s. Different surface reconstruction methods are compared. *Top left* approach from [25]. *Top right* approach from [27]. *Bottom left* approach from [26] without refinement. *Bottom right* approach from [26] with our proposed refinement method

the spherical interpolation but the control over the bumpiness of the surface is lacking. The times of the interpolation are summarized in the Table 1. The same number of interpolation has been done but a different number of weldings occurred thus producing a different number of created particles. According to our measurements, the spline-based interpolation performed the same number of interpolation in a slightly shorter time.

Various results can be achieved choosing different upsampling radii. A upsampling radius equal to the support radius used in the fluid simulation was chosen in [11]. A comparison of two different radii is shown in Fig. 9. If filling the gaps in thin fluid regions and a smoother surface are desired, a higher radius has to be chosen.

Figure 10 shows a reconstruction of multiple fluids. Several fluids are poured into the water or mixed together. A clear interface produced by our composite implicit function (Eq. 16) can be seen. To convince the reader, we have set up different scenes with up to 14 different liquids having various materials, see Fig. 11 for more detail. We have tested our composite isosurface using three single-phase isofunctions [25–27] reconstructing level set at various thresholds. Higher values provide flatter and less blobby results. However, as shown in Fig. 11 (top left), reconstructing level set of $\phi(\mathbf{x})$ from Eq. (16) with the isofunction proposed in [25] the thin regions of fluids near the interfaces are discarded, resulting in a unnatural loss of the fluid volume. Using the approach from [26], these regions are nicely preserved, see Fig. 11 (bottom). In regions where several fluid mix, the sampling of a single fluid is low, and thus, the approach using PCA (principal component analysis) form [27] gives inaccurate results in such scenarios (Fig. 11 top right). Performing surface reconstruction [27] with our upsampling technique did not give us good results (Fig. 11). One reason is that performing PCA solely on the particles in the vicinity of the surface produces inadequate results for the surface reconstruction. The second reason is the lack of regularity of the created upsampled particles that is present in the fluid particles. This irregularity produces a bumpy surface. Therefore, the surface reconstruction technique from [26] was used.

Table 2 summarizes the time requirements of the refinement process. All the simulations were done on a machine with Core2 Duo E6750 processor. Simulation times are small compared to the refinement times, but if we have to take into account that the surface refinement is not needed in every time step. The simulation in a higher resolution would take

Table 2 Summarization of surface particle extraction and upsampling of the test scenes

Scene	Simulation (radius = h)		Extraction (radius = 1.7 h)		Upsampling (radius = h)		Upsampling (radius = 1.5 h)		Upsampling (radius = 2 h)	
	# Particles	Time (s)	# Particles	Time (s)	# Particles	Time (s)	# Particles	Time (s)	# Particles	Time (s)
Dam break	15,625	0.058	9,598	0.1244	50,709	0.1692	59,975	0.6674	65,884	1.5712
Fountain	9,943	0.033	8,257	0.0607	30,472	0.1097	43,384	0.4475	55,031	1.1426
3 Pipes	11,163	0.032	10,031	0.0731	42,216	0.1867	61,921	0.8278	77,284	2.3451
14 Fluids	128,226	0.490	87,408	1.0840	428,335	1.6313	553,082	6.0902	648,412	14.5273

Results of three different radii used in the upsampling were measured. All values in the table are measured for a single time step

a longer time as well. The extraction upsampling times also strongly depend on the chosen radius of the kernel function used in the process.

5 Conclusion and Future Work

To conclude, we have proposed several improvements toward particle-based fluid simulation which enabled us to simulate arbitrary number of immiscible fluids. In order to track interfaces between immiscible fluids, we devised a novel composite implicit function. We further enhanced the interfaces and free surfaces by upsamling the particles in the vicinity of the surface. To detect the particles in the vicinity of the surface a novel method is used, which gives better results in clustered regions than previous methods.

In the future, we will parallelize the surface refinement algorithm and exploit the power of GPU.

References

1. Losasso, F.; Shinar, T.; Selle, A.; Fedkiw, R.: Multiple interacting liquids. In: SIGGRAPH'06: ACM SIGGRAPH 2006 Papers, pp. 812–819. ACM Press, New York (2006)
2. Kim, B.: Multi-phase fluid simulations using regional level sets. ACM Trans. Graph. **29**, 175:1–175:8 (2010)
3. Zhang, Y.; Solenthaler, B.; Pajarola, R.: Adaptive sampling and rendering of fluids on the GPU. In: Eurographics/IEEE VGTC Symposium on Point-Based Graphics, pp. 137–146. Los Angeles (2008)
4. Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; Silva, C.T.: Computing and rendering point set surfaces. IEEE Trans. Vis. Comput. Graph. **9**, 3–15 (2003)
5. Levin, D.: Mesh-independent surface interpolation. In: Geometric Modeling for Scientific Visualization, pp. 37–49 (2003)
6. Pauly, M.; Kobbelt, L.P.; Gross, M.: Point-based multiscale surface representation. ACM Trans. Graph. **25**, 177–193 (2006)
7. Adams, B.; Pauly, M.; Keiser, R.; Guibas, L.J.: Adaptively sampled particle fluids. ACM Trans. Graph. **26**(3), 48:1–48:7 (2007)
8. Solenthaler, B.; Gross, M.: Two-scale particle simulation. ACM Trans. Graph. **30**, 81:1–81:8 (2011)
9. Guennebaud, G.; Barthe, L.; Paulin, M.: Dynamic surfel set refinement for high-quality rendering. Comput. Graph. **28**, 827–838 (2004)
10. Guennebaud, G.; Barthe, L.; Paulin, M.: Interpolatory refinement for realtime processing of point-based geometry. Comput. Graph. Forum **24**(3), 657–667 (2005)
11. Solenthaler, B.; Zhang, Y.; Pajarola, R.: Efficient refinement of dynamic point data. In: Symposium on Point-Based Graphics, pp. 65–72 (2007)
12. Ando, R.; Tsuruno, R.: A particle-based method for preserving fluid sheets. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '11, pp. 7–16. ACM, New York, NY, USA (2011)
13. Solenthaler, B.; Schläfli, J.; Pajarola, R.: A unified particle model for fluid solid interactions. Comput. Anim. Virtual Worlds **18**, 69–82 (2007)
14. Keiser, R.; Adams, B.; Gasser, D.; Bazzi, P.; Dutre, P.; Gross, M.: A unified lagrangian approach to solid-fluid animation. In: Gross, M.; Pfister, H.; Alexa, M.; Rusinkiewicz S. (eds.) Symposium on Point-Based Graphics, pp. 125–133. Eurographics Association, Zurich, (2005)
15. Losasso, F.; Irving, G.; Guendelman, E.: Melting and burning solids into liquids and gases. IEEE Trans. Visual. Comput. Graph. **12**(3), 343–352 (2006). Member-Ron Fedkiw
16. Carlson, M.; Mucha, P.J.; Van Horn III, R.B.; Turk, G.: Melting and flowing. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer animation, SCA'02, pp. 167–174. ACM, New York (2002)
17. Mao, H.; Yang, Y.H.: Particle-based immiscible fluid–fluid collision. In: Proceedings of Graphics Interface 2006, GI'06, pp. 49–55 Canadian Information Processing Society, Toronto (2006)
18. Monaghan, J.; Kocharyan, A.: Sph simulation of multi-phase flow. Comput. Phys. Commun. **87**(12), 225–235 (1995)
19. Müller, M.; Solenthaler, B.; Keiser, R.; Gross, M.: Particle-based fluidfluid interaction. In: SCA'05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 237–244. ACM Press, New York (2005)
20. Hu, X.Y.; Adams, N.A.: An incompressible multi-phase SPH method. J. Comput. Phys. **227**, 264–278 (2007)
21. Solenthaler, B.; Pajarola, R.: Density contrast SPH interfaces. In: SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 211–218. Eurographics Association, Aire-la-Ville, Switzerland (2008)
22. Hu, X.; Adams, N.: A constant-density approach for incompressible multiphase SPH. J. Comput. Phys. **228**(6), 2082–2091 (2009)
23. Becker, M.; Teschner, M.: Weakly compressible SPH for free surface flows. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA'07, pp. 209–217. Eurographics Association, Aire-laVille, Switzerland (2007)
24. Solenthaler, B.; Pajarola, R.: Predictive-corrective incompressible SPH. ACM Trans. Graph. **28**, 40–1406 (2009)
25. Müller, M.; Charypar, D.; Gross, M.: Particle-based fluid simulation for interactive applications. In: SCA'03: Proceedings of the

2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 154–159. Eurographics Association, Aire-la-Ville, Switzerland (2003)
26. Onderik, J.; Chládek, M.; Ďurikovič, R.: SPH with small scale details and improved surface reconstruction. In: SCCG'11: Proceedings of the Spring Conference on Computer graphics. Comenius University, Bratislava, Slovakia (2011)
27. Yu, J.; Turk, G.: Reconstructing surfaces of particle-based fluids using anisotropic kernels. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA'10, pp. 217–225. Eurographics Association, Aire-la-Ville, Switzerland (2010)