

A simple approach for bubble modelling from multiphase fluid simulation

Bo Ren¹ ✉, Yuntao Jiang², Chenfeng Li³, and Ming C. Lin⁴

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract This article presents a novel and flexible bubble modelling technique for multi-fluid simulations using a volume-fraction representation. By combining the volume fraction data obtained from a primary multi-fluid simulation with a simple and efficient secondary bubble simulation, a range of real-world bubble phenomena are captured with a high degree of physical realism, including large bubble deformation, sub-grid bubble motion, bubble stacking over the liquid surface, bubble volume changing and dissolving etc. Without any change required to the primary multi-fluid simulator, the proposed bubble modelling approach is applicable to any multi-fluid simulator based on the volume fraction representation.

Keywords bubble, volume fraction, smoothed particle hydrodynamics, fluid simulation

1 Introduction

Real world liquids often contain bubbles, interacting and evolving together in various forms. For example, vibrant bubbles are generated when

gas is quickly trapped or injected into liquid, while flickering tiny bubbles occur when liquid slowly releases dissolved gas. Bubbles can also behave very differently in liquid: big bubbles can change shape due to liquid-gas interaction; in certain solutions such as soap-suds stacking bubbles can be observed covering the liquid surface; when gas is able to dissolve in liquid, bubble size can change dynamically and small bubbles can dissolve and disappear due to liquid motion; bubbles also quickly, if not immediately, pop when they rise and reach the liquid surface, and in case of bubble stacking the pushed-up bubbles pop firstly due to surface liquid loss. These varying dynamic phenomena contribute to the rich visual effects caused by liquid-gas interaction.

Various approaches have been proposed to model bubbles, especially bubble generation and tracking. Eulerian surface tracking framework can be used for large bubble motion and deformation [6], and this can be further improved by introducing supplemental particles [13]. Particles are also used to directly represent small (sub-grid) bubbles, simulate bubble motion and add spray effects [4][8]. While the uprising motion of bubbles are often simulated by directly adding buoyancy and drag forces, more complex methods that capture more accurately the liquid-gas interaction can introduce physical realism, such as using two-way coupling models [12][19].

From the viewpoint of physics, bubbles can be considered as a special type of interfacial multi-phase flow phenomena involving immiscible liquid and gas. In this sense, recent progresses of multi-fluid simulation using the volume fraction representation [18][20] have potential to benefit bubble modelling. While fully capturing the liquid-gas interaction, the volume fraction data accurately describe gas

1 College of Computer and Control Engineering, Nankai University, Tianjin, China. E-mail: renboeverywhere@gmail.com

2 Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: jiangyt11@mails.tsinghua.edu.cn, shimin@tsinghua.edu.cn.

3 College of Engineering, Swansea University, Swansea, UK. E-mail: c.f.li@swansea.ac.uk.

4 Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, USA. E-mail: lin@cs.unc.edu.

Manuscript received: year-month-day; accepted: year-month-day

distribution through the whole simulation space, which sequentially determines bubble distribution and volume change. Specifically, in the presence of bubbles, the local volume can be dominated by the gas phase, but due to the high density ratio between liquid and gas phases, the local average mass of gas can still be small compared to the liquid mass. Thus, bubbles can be automatically detected through such disagreement between mass and volume fractions, which fits nicely into the volume fraction representation framework. However, there have been little research to take advantage of these advanced volume-fraction-based multi-fluid simulation for modelling bubble effects.

We propose a novel method that can model various bubble effects from multi-fluid simulations using the volume fraction representation. Based on the volume fraction data, regions where the mass fraction is dominated by gas are categorized as gas regions or large bubbles, and they can be traced with isosurface-based reconstruction methods. Sub-grid bubbles are modeled in a low-cost secondary simulation, where the volume and movement of bubbles are determined from the results of the primary multi-fluid simulation. The new approach has the following advantages: 1) it is simple and efficient without any change to the primary simulator, and has the flexibility to integrate bubble effect into existing multi-fluid simulation data without the need of re-simulation; 2) it can simultaneously handle various bubble effects including deformation of large bubbles, volume change, dissolving and stacking of sub-grid bubbles etc.; 3) the result can naturally capture liquid-gas interaction taking advantage of the advanced multi-fluid simulation, with bubble distribution consistent to the physical gas fraction distribution. We demonstrate the effectiveness of the proposed approach with both SPH and grid-based multi-fluid simulations.

The rest of the paper is organized as follows. In Section 2, we review related work. In Section 3, we introduce our bubble generation algorithm based on the volume-fraction representation. We describe the specific implementation detail in Section 4 and the results in Section 5. We conclude by discussing the benefit and limitation of the new approach in Section 6.

2 Previous Works

Traditionally graphics researches focus on direct bubble simulation. Eulerian grid are used to simulate large bubbles in liquid, where the shape and deformation of bubbles can be traced through the level-set method, producing various visual effects of large bubbles [5, 6, 16]. Certain stacking effects of large bubbles over a liquid surface are also modeled, e.g. in [11, 24]. To facilitate bubble generation and shape tracking, Lagrangian particles are later introduced into some hybrid simulation methods [2, 7, 13, 15, 22], where sub-grid bubbles and foam are taken into account.

Alternatively, Lagrangian particles are extensively used to represent small sub-grid bubbles in particle-based bubble simulation methods under the SPH (Smoothed Particles Hydrodynamics) framework, where larger bubbles are usually reconstructed using a collection of neighboring gas particles [17, 21]. By adding cohesive forces between bubble particles and the liquid, dynamic bubble movements and stacking foams can be simulated, producing impressive results [3, 4, 9]. In these works, the motions of sub-grid bubbles are mostly modelled with one-way influenced [3, 4] or two-way coupled [7, 9] particles using various types of drag and cohesive forces, and the sub-grid bubbles have limited influence over the liquid motion. For simulation of dispersed bubbles beneath a liquid surface, [12] proposed a variable-density Poisson solver, where the local average density and pressure are influenced by bubble concentration. They also provided a stochastic solver to approximate microscale motions. Later, [19] derived a monolithical approach, where changing of bubble volumes is allowed and liquid and gas motions tightly affect each other, however bubble stacking is not modeled in their approach. Instead of directly simulating all bubble motions, [8] introduced a secondary simulation that generates spray, foam and small bubble effects from the primary single-phase Lagrangian simulation result. Their approach is partially similar to ours in that we do not directly simulate all of the bubble motions either. Based on a primary multi-fluid simulation, we reconstruct bubbles of different sizes from the fraction field, and control volume changing, dissolving and moving of bubbles in a secondary simulation which does not influence the primary simulator. Compared to the

direct bubble simulation method, our approach is able to reproduce various bubble effects in a simple and efficient way. It not only reproduces bubble phenomena where the bubble properties, distribution and movement faithfully reflect the physical gas distribution and liquid-gas two-way coupling, but also physically captures the liquid motion influenced by the gas phase thanks to the primary multi-fluid simulation. Our approach is also more flexible in that it can be applied to existing multi-fluid simulation data to integrate additional bubble effects without the need of re-simulation.

Graphics fluid simulators using a volume fraction representation dates back to [17]. Later the volume fraction representation is combined with the diffusion model to simulate multi-fluid behaviors, both in grid-based solvers [1, 10] and in SPH-based solvers [14]. More recently, multi-fluid simulators that consider the velocity difference between phases are proposed to bring in more physical realism [18, 20]. In Computational Fluid Dynamics (CFD), all commercial multi-fluid simulators are based on the volume fraction representation, such as ANSYS CFX and FLUENT. Our approach does not rely on a specific multi-fluid simulator, as long as it uses the volume fraction representation and can handle physical simulation of liquid-gas mixtures.

3 Bubble Modelling from Volume Fraction Data

The volume fraction representation in multi-fluid simulation describes the spatial distribution of the simulated phases with their fraction fields. At any point in the simulation domain, the fraction of volume locally occupied by a phase k is its volume fraction α_k ; similarly, the local fraction of mass of the phase k is its mass fraction c_k . Given the rest density of each phase, these two fraction values can be calculated from each other. Multi-fluid simulators using a volume fraction representation usually do not directly trace the interface between different phases, but the spatial fraction field can still provide direct indication of the distribution of individual phases. On the other hand, it is often difficult to clearly define and track sub-grid bubbles with the interface geometry finer than the simulation resolution. The motions of these bubbles also physically couple with the fluid motion on a sub-grid level. This indicates

that the volume-fraction based multi-fluid simulators may well fit for visual simulation of bubbles.

3.1 Categorizing Bubbles

In the physical world the densities of liquid and gas often have high relative ratios (vary from 100:1 to 1000:1). In the volume fraction representation, this leads to an obvious difference between the gas volume fraction value and the gas mass fraction value. That is, when there is locally little gas present, the volume fraction and the mass fraction of gas are both small; however, as the local gas phase increases, its volume fraction rises, while its mass fraction remains low due to the high density ratio between liquid and gas, and this can be the case even when the gas volume fraction is more than 0.95; finally, when the liquid phase is mostly propelled from the local space, the mass fraction of gas then rises.

On the other hand, real-world bubbles, especially the stacking bubbles over a liquid surface, has similar properties that the volume is largely contributed by entrapped gas, but the mass is largely contributed by the liquid forming the bubble surface. Based on these observations, we propose a simple rule to categorize bubbles into three groups: small, medium and large bubbles. Specifically, if the gas volume fraction α_g is lower than a given threshold θ_1 , we consider the local space is mainly occupied by the liquid phase and only dispersed and relatively small bubbles may exist, which are classified as “small bubbles”. If the gas volume fraction α_g is higher than θ_1 , but the gas mass fraction c_g is lower than a given threshold θ_2 , it is detected as middle-sized bubbles in the liquid or bubbles stacking over the liquid surface, and hence they are classified as “medium bubbles”. Finally, if the gas mass fraction c_g is higher than θ_2 , the location is either outside the liquid or belong to a “large bubble” region in the liquid.

An overview of our approach is demonstrated in Fig.1. Large bubbles can be conveniently generated from the simulation data through iso-surface-based surface reconstruction methods. Since the multi-fluid simulator has taken care of the physical change of the fraction field, this strategy can automatically produce deformation and volume changes (merging and splitting) for large bubbles. For medium bubbles and small bubbles, as their sizes are often of sub-grid level, it is better to directly represent them. For simplicity, we follow traditional approaches using

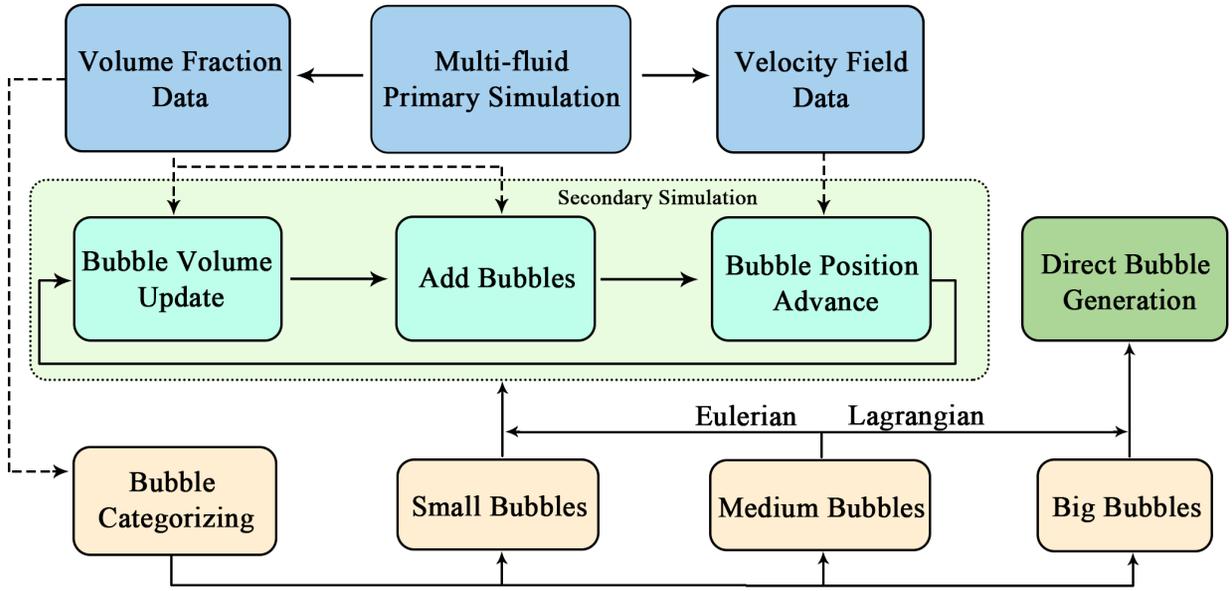


Fig. 1 Overview of bubble modelling strategy from multi-fluid simulation using a volume-fraction representation. Different types of bubbles are categorized from the volume fraction field data in the primary simulation. Bubbles are whether directly generated or created through a secondary simulator depending on their categories. The primary simulation is not influenced.

sphere-shaped sub-grid bubbles in the rendering, however the bubble shapes can be further improved by recent rendering technique [19] using a dictionary of level sets for various bubbles in the rendering process. Bubbles can come into contact with each other when they get closer. It is possible to simply merge them as a larger bubble, however in case of multiple contacts which often happen for stacking bubbles this is not the best choice. Moreover, not all bubbles in contact have an original distribution like a sphere, and the newly merged larger bubble may in turn collide with other bubbles, resulting in distribution errors. Thus, instead of merging the colliding bubbles, we adopt the tessellation method described in [3], where the bubbles in contact are represented using a weighted Voronoi diagram. Note that such tessellation method is only used to regenerate contact-surface mesh, it does not affect the general bubble motion (position) determination or volume change handling process even in the case of bubble-stacking phenomenon.

3.2 Controlling Bubble Volume

Bubbles may experience volume change in the liquid for several reasons. Since gas is relatively more compressible compared to liquid, bubble sizes usually expand when rising up. Bubble sizes also

change in cases of merging or splitting. All gases are dissolvable in the liquid to a certain extent. As a result bubbles can change its volume when moving through the liquid region, absorbing undissolved gas or losing some of its own due to dissolving, and some small bubbles may be totally dissolved in the liquid and disappear.

Fortunately the volume fraction representation provides a straight forward way to evaluate bubble volume changes. Specifically, the bubble volume must satisfy the following relation:

$$\sum_b V_b = \alpha_g V \quad (1)$$

where V_b is the volume of bubbles, α_g is the local gas volume fraction, V is the local volume. The summation performs over all bubbles b in the local volume V .

The above equation indicates three main factors that affect the bubble volume. The first factor is the local gas volume fraction at the bubble location, which affect all three types of bubbles. Intuitively, if the gas volume fraction is high, there is more gas in the local region, and the bubble volume can be larger. The second factor is the local volume V , which may change in Lagrangian simulators when the effective particle volume varies over time. The last factor is the local bubble number. Each bubble

will share less volume if there are more bubbles in the local space. This mostly affects the volume of small bubbles, since these bubbles are most likely to get dispersed in the liquid sharing the same local volume.

4 Implementation

In this section we will describe in detail how to generate multi-sized bubbles based on the principles introduced in Section 3, as well as derive volume change and motion of the bubbles from primary simulation data. In Lagrangian simulators particles in the primary simulation can serve as a reference of bubble position, however in Eulerian multi-fluid simulators using a volume fraction representation, there are no Lagrangian particles. Thus although the general framework does not differ much, the specific calculation for these two types of simulators are slightly different, which we will explain in the following sections.

4.1 Lagrangian Simulator

Among different Lagrangian methods, the SPH framework is the most popular one in computer graphics for bubble simulation and multi-fluid simulation, therefore we mainly refer to the SPH formulation in this section.

First we consider small bubbles, appearing in locations where the gas volume fraction is low and the local volume is mostly occupied by the liquid phase. From Eqn. (1), we need to determine all three factors for small bubbles in a Lagrangian primary simulation. Following the standard SPH formulation, the interpolated volume fraction value of gas for a particle i is

$$\alpha_{gi} = \sum_j V_j \bar{\alpha}_{gj} W_{ij} \quad (2)$$

where $\bar{\alpha}_{gj}$ denotes the volume fraction of gas for the particle j , V_j is the effective volume of particle j , $W_{ij} = W(\mathbf{r}_i - \mathbf{r}_j, h)$ is the smoothing kernel function with support h , and $\mathbf{r}_i, \mathbf{r}_j$ denote positions of particles i, j . The summation is performed over all fluid particles j in the neighborhood of particle i .

From the definition of volume fraction, the following relation holds

$$V_j \bar{\alpha}_{gj} = V_{gj} \quad (3)$$

where V_{gj} is the gas volume of particle j . As a result,

Eqn. (2) can be rewritten as:

$$\alpha_{gi} = \sum_j V_{gj} W_{ij} \quad (4)$$

Therefore, if we treat the small bubbles in a secondary simulation, for a fluid particle i in the primary simulation, we can apply the following equation to small-bubble particles b in the secondary simulation:

$$\alpha_{gi} = \sum_b V_b W_{ib} \quad (5)$$

where V_b is the volume of small bubble b in the secondary simulation. The summation is performed over all small bubble particles b located in the spatial neighborhood of particle i . Only those fluid particles in the primary simulation with gas volume fraction lower than θ_1 require this interpolation.

Eqn. (5) provides an estimation to the interpolated volume fraction at the position of particle i due to existing small bubbles. The difference between such estimated value and the true value of α_g of particle i can be used to determine new generation of small bubbles and volume adjustment of existing small bubbles. Intuitively, if the estimated value is larger than the true value in the primary simulation, volumes of nearby small bubbles may be too large, or the small bubbles may be overpopulated, and vice versa.

Up to this point, we still need to determine the local volume and the bubble number related to the local volume shown in Eqn. (1). For each small bubble in the secondary simulation, we find the nearest fluid particle in the primary simulation in space. At the same time, for each fluid particle in the primary simulation which has volume fraction of the gas phase lower than θ_1 , we count the number n of small bubbles attached to it as ‘‘nearest’’. In this way, each fluid particle in the primary simulation with gas volume fraction lower than θ_1 defines a local volume using its own volume $V_i = m_i/\rho_{i0}$, where m_i is mass of particle i , and ρ_{i0} is its current rest density; the bubble number in this local volume is just n .

Then for a small bubble particle b in the secondary simulation, its volume change should be:

$$\Delta V_b = \frac{1}{n} (\bar{\alpha}_{gi} - \alpha_{gi}) V_i \quad (6)$$

Whenever $V_b < 0$, the small bubble particle in the secondary simulation is removed. A small bubble particle will also be directly removed if the nearest particle it finds does not satisfy the condition that

the gas volume fraction is lower than θ_1 , which indicates it has travelled too far into other regions.

After the above adjustment for existing small bubble particles, we recalculate Eqn. (5) for each fluid particle in the primary simulation with gas volume fraction lower than θ_1 . If

$$(\bar{\alpha}_{gi} - \alpha_{gi}) > \epsilon \bar{\alpha}_{gi} \quad (7)$$

where $\epsilon \in [0, 1)$ is a control factor, then a new small bubble particle is added into the secondary simulator at the position of particle i . The volume of the new small bubble is

$$V_{b,new} = (\bar{\alpha}_{gi} - \alpha_{gi})V_i \quad (8)$$

The small bubble particles in the secondary simulation are advanced by the velocity of its nearest primary particles, or they can be advanced with the gas velocity if phase velocity is provided by the primary simulator. When relative position of secondary bubble particle and its nearest primary fluid particle change, which results in falling value of the smoothing kernel, a non-zero control factor ϵ serves to prevent frequent particle adding due to such effect. We set $\epsilon = 0.5$, however it can be freely adjusted depending on different needs.

The medium bubbles does not need to be handled in the secondary simulator. For Lagrangian primary simulators it is convenient to attach a medium bubble to those primary fluid particles that have gas volume fraction larger than θ_1 but gas mass fraction smaller than θ_2 . The volume of the medium bubbles is set to $V_b = \bar{\alpha}_{gi}V_i$. This has advantage in that for large gas volume fraction values, fluctuation in the SPH interpolation more easily results in frequent adding and deleting of secondary particles, and directly attaching medium bubble to the corresponding primary fluid particle avoids this issue. The fact that secondary simulation is not needed also provides computational efficiency due to saving of interpolation calculation. Note that the size of a medium bubble will change with the volume fraction of the primary fluid particle, and a medium bubble will move with the primary fluid particle it attaches to.

Finally, those primary fluid particles that has gas mass fraction larger than θ_2 automatically form large bubbles through iso-surface based surface reconstruction methods, such as using the anisotropic kernel method [23].

4.2 Eulerian Simulator

Similar to the secondary simulation for the Lagrangian simulators, we can add Lagrangian particles representing small and medium bubbles into a secondary Eulerian grid, and calculate the volume and motion of bubbles using primary simulation data. In Eulerian grids the secondary simulation also start from updating the volume of existing bubbles. Here the local volume in Eqn. (1) can be conveniently chosen as each grid volume V , and the number of secondary particles n located in each grid serves as the bubble number in the local volume. A proper updating method should generate volume expanding, shrinking, bubble adding and deleting effects during the computation.

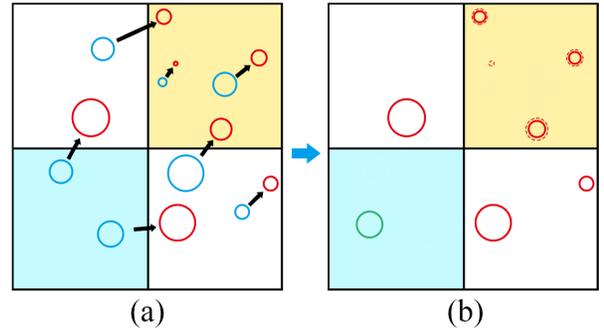


Fig. 2 (a) Volume of existing bubbles are adjusted when they are advanced. Some grids may be over-populated (light-yellow) or containing insufficient bubble volume (light-blue). Blue circles indicate previous position and volume of bubbles, red circles indicate their current position and volume; (b) In over-populated places bubble volumes are corrected downward, and in insufficient places a new bubble is generated (green circle).

For a secondary particle with volume V_b , suppose it moves from a grid with gas volume fraction $\alpha_{g,t-1}$ to a grid with gas volume fraction $\alpha_{g,t}$, we first update its volume using

$$V'_{b,t} = (1 + \alpha_{g,t} - \alpha_{g,t-1})V_{b,t-1} \quad (9)$$

Then, within each grid, we calculate

$$\Delta V = \alpha_{g,t}V - \sum_b V'_{b,t} \quad (10)$$

where the summation is performed over all bubble particles in the grid. If $\Delta V \leq 0$, an amount of $\frac{1}{n}\Delta V$ is subtracted from all bubble particles inside the grid; if $\Delta V > 0$, a new secondary bubble particle is added with volume ΔV at a random position within the grid. Whenever a secondary particle has its volume less than zero or it enters a grid with gas mass fraction larger than θ_2 , the particle is removed

from the secondary simulation. After the particle volumes are updated, they are advanced by the fluid or gas velocity at their locations. This strategy is illustrated in Fig.2.

The above strategy serves for both small and medium bubbles in grid-based simulators. For large bubbles and liquid surface reconstruction, iso-surface methods similar to that in [5] can be used, with the user-defined threshold value determined by the gas volume fraction.

4.3 Bubble Reconstruction Framework

Algorithm 1 Secondary Simulation

```

1: for all secondary bubble particles  $b$  do
2:   update volume  $V_b$  (Eqn.6, Eqns.9-10)
3:   if particle outside permitted bubble region or  $V_b \leq 0$  then
4:     delete particle  $b$ 
5:   end if
6: end for
7: for all local volume  $V$  do
8:   if local bubble volume less than local gas volume then
9:     add a new bubble particle (Eqn.8, Eqn.10)
10:  end if
11: end for
12: for all secondary bubble particles  $b$  do
13:  advance using fluid or gas velocity in the primary simulation
14: end for

```

For Lagrangian and Eulerian multi-fluid simulation methods, the generation of bubbles from the primary simulation shares the same framework. Regions with high gas mass fraction are treated as large bubbles and reconstructed using iso-surface methods. Small bubbles can be handled in a secondary simulation, whose algorithm framework is shown in Algorithm. 1, and the calculation methods are described in detail for both types of simulations in Section 4.1 and Section 4.2, respectively. There are slight differences in the specific calculations between Lagrangian and Eulerian simulations, mostly due to the fact that Lagrangian simulations have built-in particle systems that can be conveniently used to indicate bubbles but Eulerian simulations do not have such facility. The local volume is defined by the grid volume in the Eulerian simulation, while it is defined by the volume of the primary fluid particles' own volume in the Lagrangian simulation.

Although we can define “medium bubble regions” for Eulerian simulation grids using θ_1 and θ_2 , it is more convenient to handle the medium bubbles in the secondary simulation.

The values of θ_1 and θ_2 are user-defined and relate to the density ratio between liquid and gas. We find $\theta_1 \in [0.1, 0.5]$, $\theta_2 \in [0.05, 0.5]$ can usually give satisfactory results in our experiments when liquid-gas density ratio is above 100:1.

The particle deleting step in Algorithm. 1 can delete bubbles in two situations. When a bubble rises above the liquid surface, it can be deleted either due to rapid drop of liquid fraction or entering into the gas phase region. This captures bubble popping that occurs at the liquid surface in the real world. Large amount of bubbles rising can result in frequent popping appearance when they reach the liquid surface, which should not be viewed as flickering artifact by immediate deletion of newly generated bubbles. The second situation is that small bubbles can naturally get deleted due to negative volume change. From a physical point of view, this corresponds to cases where gas gets dissolved in the liquid or smaller bubbles get absorbed in larger ones. On the other hand, considering rendering efficiency, it is not economical to keep bubbles smaller than a rendered pixel. Thus deleting those small bubbles beforehand has some further merits compared with keeping every existing bubble.

5 Results

The secondary simulation only involves local computations and can be readily parallelized by GPU given the primary simulation data. We implement the secondary simulator using CUDA 6.5 on an NVIDIA GeForce GTX 980 graphics card. The computational time tends to be more costly for Lagrangian simulators due to its neighborhood interpolations. However it generally adds only a small overhead compared to the primary multi-fluid simulation. In our examples the secondary simulation costs less than 4% computational time of the primary simulation. The figures in this section can be enlarged to view more details, while a supplemental video is also provided to demonstrate the examples.

Figure.3 shows gas rising in a soap solution simulated using a Lagrangian simulator. Vigorously

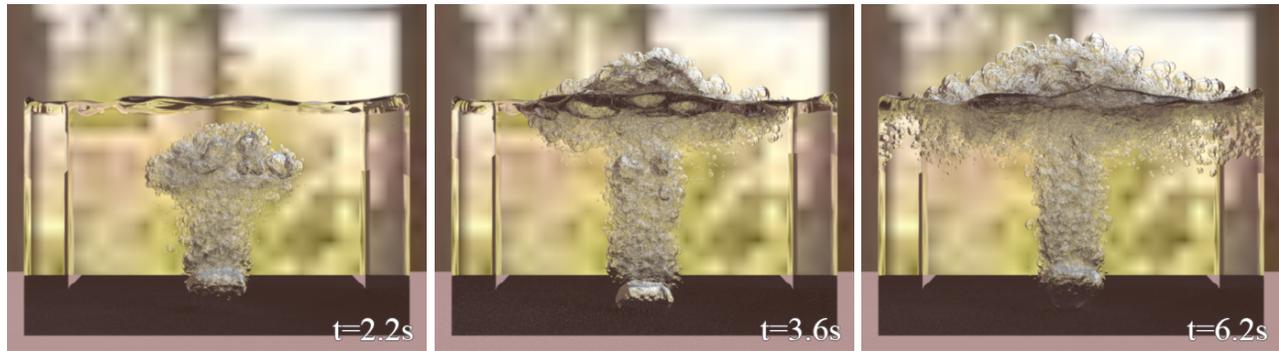


Fig. 3 Bubbles in soap solution. Gas is injected from the center of the bottom creating bubbles of various sizes, and realistic bubble volume evolving, stacking and breaking are recovered by our approach.

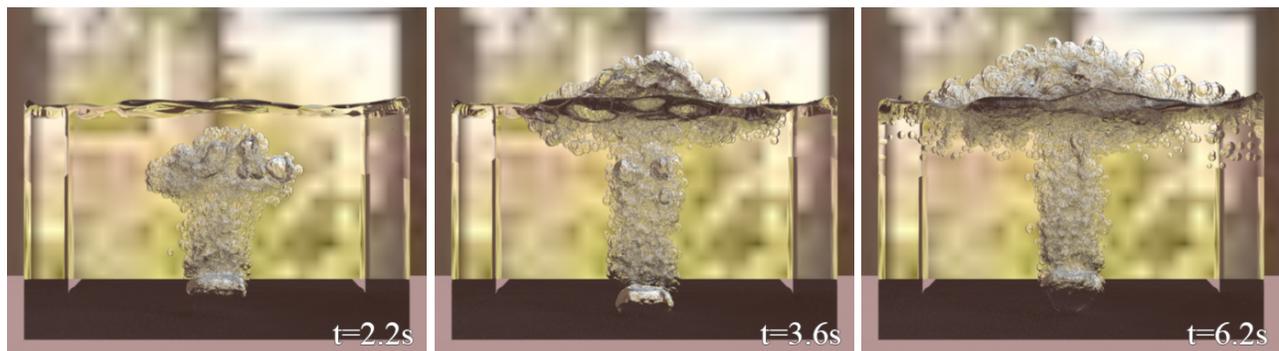


Fig. 4 Bubbles in soap solution. Only the medium and large bubbles are rendered. While less sub-grid bubbles exist in the result, the stacking effect and the deformation and motion of large bubbles are preserved.

evolving multiple-sized bubbles are formed as gas rising up, with tight liquid-gas coupling in the solution, containing about 3000 small bubbles and 6000 medium bubbles (not counting those that have broken or dissolved). Each rendered frame consists of 8 simulation steps, and the secondary simulation takes 56ms per frame (there are 25 frames in 1 second). Realistic deformation of large bubbles, sub-grid bubble motion in liquid and stacking bubbles over the liquid surface are recovered using the proposed approach. The bubbles automatically break and disappear at the top when the liquid fraction drops below the threshold.

Figure.4 shows the same case but without the small bubbles in the secondary simulation. It can be observed that there are much less sub-grid small bubbles, but the stacking effect and the deformation and motion of large bubbles are preserved. Not needing the secondary simulator, the bubble regeneration is even simpler while preserving many of the interesting visual effects.

In Figure.5, a scene with two transparent chemical solutions meeting and reacting to produce gas

(e.g. hydrochloric acid and soda solution react to produce carbon dioxide) is simulated using a Lagrangian simulator, which contains about 7000 medium bubbles. Realistic bubble effect are produced using the proposed approach. Bubbles are quickly generated in the reacting region, and form a thin layer over the liquid surface during continuous generation and breaking.

Figure.6 shows an bubbly liquid-gas mixture running through a pipe containing a small box-shaped obstacle. The primary simulation is performed using an ANSYS CFX Eulerian simulator, and 139,000 bubbles are recovered from an effective $20 \times 20 \times 100$ grid. Each rendered frame consists of 5 simulation steps, and the secondary simulation takes 33ms per frame. The volume ratio of water and gas is set to 7:3 at the left inlet of the pipe. In such a bubbly mixture containing dense bubbles, the gas motion is heavily coupled with the liquid motion while the gas velocity differs greatly from the water velocity. The fully-rendered result at the top row shows that the bubble distribution is consistent to the physical distribution of gas in the simulation;

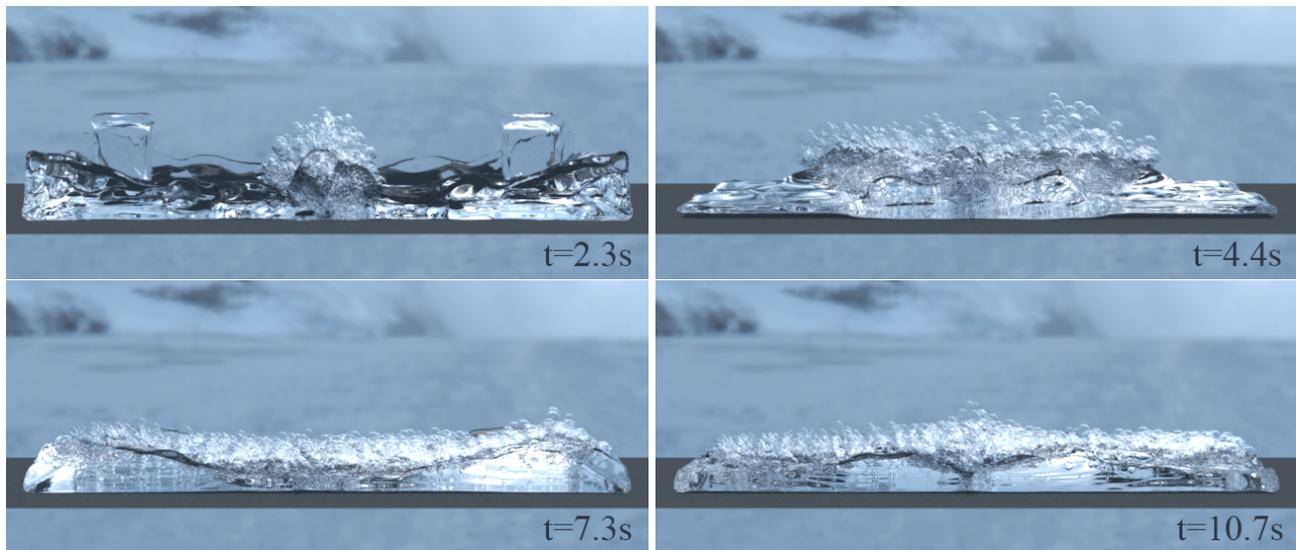


Fig. 5 Reacting bubbles. Two transparent solutions meet and react at the center producing gas. Bubbles are quickly generated in the reacting region, and form a thin layer over the liquid surface during continuous generation and breaking.

the partly-rendered result at the bottom row shows more clearly the varying size and motion of different bubbles reproduced. The water surface mesh is unchanged in the latter case, only deleting the un-rendered bubbles in the rendering process.

6 Conclusion

We have presented a novel and efficient bubble modelling strategy for multiple-fluid simulation using volume fraction representation. Through simple and efficient computations, various bubble effects can be recovered from the primary simulation data without any changes required to the primary simulator, including deformation of large bubbles, volume change, dissolving and stacking of sub-grid bubbles. In the results, the bubble motion and liquid motion naturally reflect the two-way coupled liquid-gas interaction, and the bubble distribution is also consistent to the physical gas distribution in the primary simulation. The proposed bubble modelling approach can be easily and independently applied to any multiple-fluid simulator based on a volume fraction representation, and is able to integrate bubble effects into existing multi-fluid simulation data without the need of re-simulation. The idea of utilizing fraction fields for region recognizing could also be useful in recovering other natural phenomena such as mud sliding, efflorescence, dissolution and crystallization, which may offer several future directions.

The current strategy only produces at most one single bubble within each local volume if possible, however multiple tiny bubbles generated at the same time using certain patterns may be desirable in some cases. For a Lagrangian simulator, when a primary fluid particle is considered to be attached by a medium bubble, there may actually be multiple smaller bubbles existing in the local volume, which is not considered by our approach. Further investigation on these aspects can potentially enhance the flexibility of the proposed approach. Also for Lagrangian simulators, though the bubble distribution largely corresponds to the physical gas distribution, they are not exactly the same. In principle, iterations over the volume correction and particle adding steps will reduce the errors and feedback to the primary simulator may also be adopted; however the marginal improvement could be offset by the more complex computation.

Since bubble positions and volume are calculated from the primary simulation data, the total bubble volume may change slightly after the Voronoi tessellation, losing some of the overlapped volumes. This mostly affects medium bubbles in Eulerian simulations where no pressure force propels particles when they become too close to each other. Though such volume loss can to some extent reflect the weakly compressible feature of gas in the Lagrangian simulation, neighborhood contact detection and volume correction strategy may be desired to

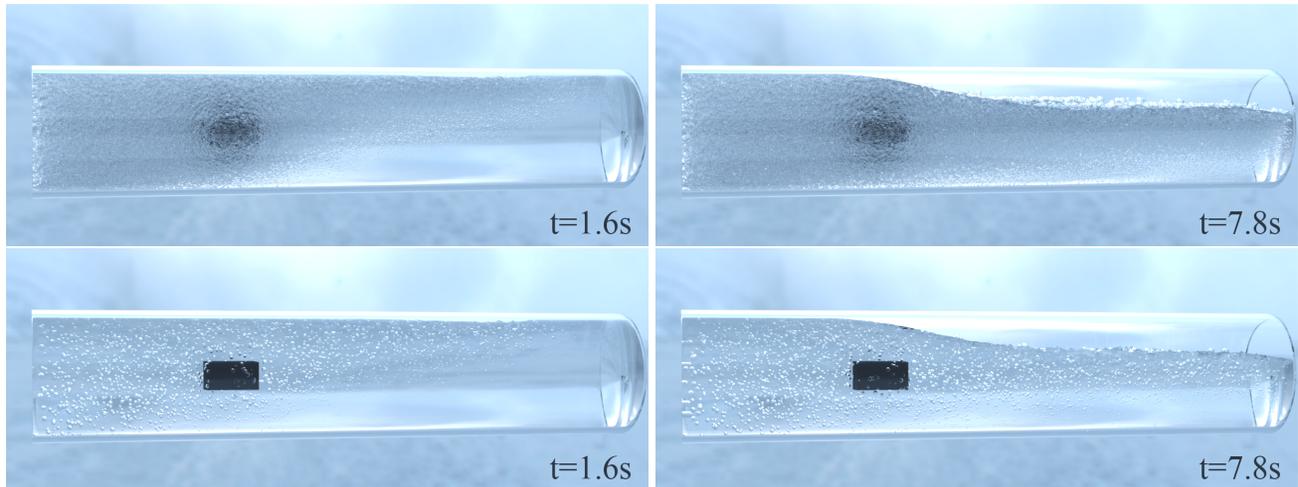


Fig. 6 Bubbly liquid-gas mixture running through a pipe containing a small box-shaped obstacle. Top: All bubbles are rendered. Distribution of the dense bubbles is consistent to the gas volume fraction distribution in the primary simulation. Bottom: Only one out of every 30 bubbles is rendered. Each bubble moves differently according to the physical velocity and has varying volume during its motion.

alleviate this problem for the Eulerian simulation, but at a higher computational cost.

Currently bubble motion is purely driven by the primary simulation, and adding feedbacks toward the primary simulation may facilitate bubble buoyancy. The phenomenon with very large bubbles stacking above the liquid surface (as in previous work [11]) are not captured by the proposed approach, since these very-large bubble regions will tend to be detected as gas region.

Acknowledgements

Acknowledgements of people, grants, funds, etc., should be placed in a separate section before the References section. The name of funding organizations should be written in full. Do not include acknowledgements on the title page, as a footnote to the title or otherwise.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- [1] K. Bao, X. Wu, H. Zhang, and E. Wu. Volume fraction based miscible and immiscible fluid

- animation. *Computer Animation and Virtual Worlds*, 21(3-4):401–410, 2010.
- [2] L. Boyd and R. Bridson. Multiflip for energetic two-phase fluid simulation. *ACM Trans. Graph.*, 31(2):16:1–16:12, Apr. 2012.
- [3] O. Busaryev, T. K. Dey, H. Wang, and Z. Ren. Animating bubble interactions in a liquid foam. *ACM Trans. Graph.*, 31(4):63:1–63:8, July 2012.
- [4] P. W. Cleary, S. H. Pyo, M. Prakash, and B. K. Koo. Bubbling and frothing liquids. *ACM Trans. Graph.*, 26(3), July 2007.
- [5] J.-M. Hong and C.-H. Kim. Animation of bubbles in liquid. *Computer Graphics Forum*, 22(3):253–262, 2003.
- [6] J.-M. Hong and C.-H. Kim. Discontinuous fluids. *ACM Trans. Graph.*, 24(3):915–920, July 2005.
- [7] J.-M. Hong, H.-Y. Lee, J.-C. Yoon, and C.-H. Kim. Bubbles alive. *ACM Trans. Graph.*, 27(3):48:1–48:4, Aug. 2008.
- [8] M. Ihmsen, N. Akinci, G. Akinci, and M. Teschner. Unified spray, foam and air bubbles for particle-based fluids. *Vis. Comput.*, 28(6-8):669–677, June 2012.
- [9] M. Ihmsen, J. Bader, G. Akinci, and M. Teschner. Grapp (2011) algarve (portugal) animation of air bubbles with sph.
- [10] N. Kang, J. Park, J. Noh, and S. Y. Shin. A hybrid approach to multiple fluid simulation using volume fractions. *Computer Graphics Forum*, 29(2):685–694, 2010.

- [11] B. Kim, Y. Liu, I. Llamas, X. Jiao, and J. Rossignac. Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph.*, 26(3), July 2007.
- [12] D. Kim, O.-y. Song, and H.-S. Ko. A practical simulation of dispersed bubble flow. *ACM Trans. Graph.*, 29(4):70:1–70:5, July 2010.
- [13] P.-R. Kim, H.-Y. Lee, J.-H. Kim, and C.-H. Kim. Controlling shapes of air bubbles in a multi-phase fluid simulation. *Vis. Comput.*, 28(6-8):597–602, June 2012.
- [14] S. Liu, Q. Liu, and Q. Peng. Realistic simulation of mixing fluids. *Vis. Comput.*, 27(3):241–248, Mar. 2011.
- [15] V. Mihalef, D. Metaxas, and M. Sussman. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Computer Graphics Forum*, 28(2):229–238, 2009.
- [16] V. Mihalef, B. Unlusu, D. Metaxas, M. Sussman, and M. Y. Hussaini. Physics based boiling simulation. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 317–324, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [17] M. Müller, B. Solenthaler, R. Keiser, and M. Gross. Particle-based fluid-fluid interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 237–244, New York, NY, USA, 2005. ACM.
- [18] M. B. Nielsen and O. Osterby. A two-continua approach to eulerian simulation of water spray. *ACM Trans. Graph.*, 32(4):67:1–67:10, July 2013.
- [19] S. Patkar, M. Aanjaneya, D. Karpman, and R. Fedkiw. A hybrid lagrangian-eulerian formulation for bubble generation and dynamics. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 105–114, New York, NY, USA, 2013. ACM.
- [20] B. Ren, C. Li, X. Yan, M. C. Lin, J. Bonet, and S.-M. Hu. Multiple-fluid sph simulation using a mixture model. *ACM Trans. Graph.*, 33(5):171:1–171:11, Sept. 2014.
- [21] X. Shao, Z. Zhou, and W. Wu. Particle-based simulation of bubbles in water-solid interaction. *Computer Animation and Virtual Worlds*, 23(5):477–487, 2012.
- [22] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross. Real-time simulations of bubbles and foam within a shallow water framework. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pages 191–198, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [23] J. Yu and G. Turk. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.*, 32(1):5:1–5:12, Feb. 2013.
- [24] W. Zheng, J.-H. Yong, and J.-C. Paul. Simulation of bubbles. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 325–333, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.