

Article

Interactive Blow and Burst of Giant Soap Bubbles

Namjung Kim ¹ and Kyoungju Park ^{2,*}

¹ Department of Image, Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University, Seoul 156-756, Korea; sogalman@naver.com

² Department of Integrated Engineering, Digital Imaging Major, Chung-Ang University, Seoul 156-756, Korea

* Correspondence: kjpark@cau.ac.kr; Tel.: +82-2-820-5823

Academic Editor: Angel Garrido

Received: 16 December 2016; Accepted: 11 April 2017; Published: 14 April 2017

Abstract: Previous studies on virtual soap bubbles mainly focused on methods for visualizing the physical and geometrical properties of soap bubbles and paid little attention to the possible ways to enhance the interaction between the simulation and the user. In this paper, a user interaction-based giant soap bubble simulation system is proposed in which the free-form shape, size, and position of giant soap bubbles are determined by the user's hand motions. Our method improves the controllability of soap bubble simulation by correcting the jerky hand trajectory and hand velocity to a smooth and gradual path. Our air flow transfer algorithm can produce detailed deformation and standing wave for soap film in real time. Our novel soap film bursting algorithm represents the process of the bursting phenomenon of soap-film and giant soap bubbles in a unified framework. The results of our experiment demonstrate that the system allows the user to experience the giant soap bubble blowing and bursting process in a virtual environment.

Keywords: soap bubble; user interaction; air flow transfer; soap film bursting; Kinect

1. Introduction

Because of their striking shapes, deformations, and various colors, giant soap bubbles are widely used in bubble shows, children's entertainment, and public presentations [1]. Figure 1 shows the photographs of a real giant soap bubble blowing and bursting. Making giant soap bubbles requires skillful techniques, a highly viscous soap solution, tools such as bubble wands, and a demonstration space. If a method can be found that simulates realistic giant soap bubbles, and which can automatically reflect the user's motion, the practicality issue could be solved in the virtual world. However, managing both the control of the user's motion and the visual realism of the giant soap bubble simulation is extremely challenging.

To address the existing issue, we propose an efficient approach that automatically demonstrates a giant soap bubble driven by the user's hand movements in a virtual environment. The proposed system controls the shape, size, and position of the bubble in real time by the user's hand movements. It also allows the user to experience various shapes of the soap bubble making process. Since our system recognizes the movement of the user's hands without an object attached to the human body, it is more comfortable and user-friendly than the motion capturing technology based device that attaches the marker to the user's body. In addition, the proposed system is more convenient and less costly than the technology that features a haptic device, because our method recognizes the movement of both the left and right hands of the user with one Kinect which is a consumer level depth camera widely used in the gaming industry.

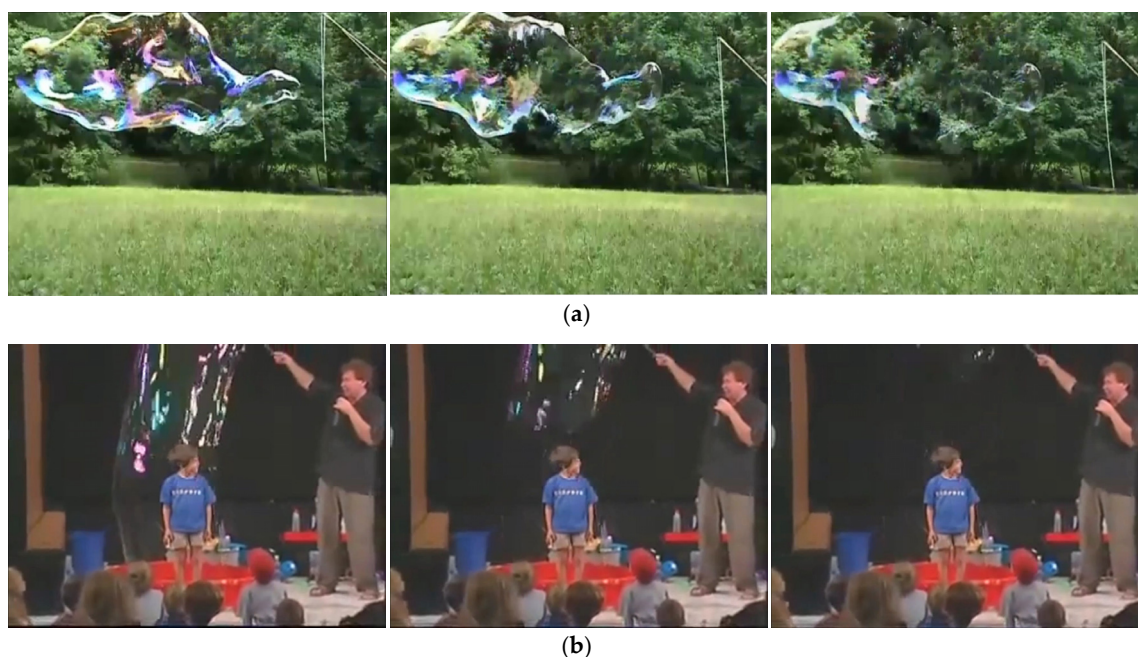


Figure 1. Photographs of a real giant soap bubble blowing and bursting. When using a tool with (a) two wands and (b) single wand.

The bursting phenomenon of small spherical soap bubbles occurs extremely quickly and is thus difficult to observe with the naked eye. Nevertheless, the bursting phenomenon of giant soap bubbles at a meter scale is an important factor for visual realism. Since the soap film is extremely large with giant soap bubbles, the progress of the soap film bursting can be visually observed. Although a previous method [2] has been introduced to generate the realistic and detailed bursting phenomenon of soap bubble, the method is still computationally expensive to use in interactive applications and did not consider drainage. This paper offers a method of simulating the details of the phenomenon of soap film bursting based on hole opening propagation and position-dependent draining in real time. In addition, the shape and the size of the soap bubble can be adjusted according to the amount of remaining soap solution.

Kim et al. [3] proposed a giant soap bubble method that makes the versatile shape of soap bubbles from a soap film by incorporating the surface tension effects into deformable models. For time-varying surface tension, they introduce Gibbs elasticity and Marangoni elasticity adaptively depending on the material properties of soap films. They also present a re-meshing method for topological change from a soap film into a soap bubble. Their method focuses on the animating the evolving process of soap films into giant soap bubbles in a way to keep minimal surfaces, show local oscillations on surfaces, and result in realistic animation of soap bubbles. However, they did not consider the effects due to external forces including bubble bursting, air inflow, and user interventions during simulation. Therefore, it is necessary to develop soap film bursting and air inflow methods for interactive simulation system of giant soap bubbles and also to modify a force generation method to apply continuous user interventions during simulation.

We extend and modify an existing method [3] so that users interactively build giant soap bubbles as they move both of their hands. Our major technical contributions are as follows: (1) we introduce an interactive blowing and bursting system of giant soap bubbles; (2) we generate smooth user exerted force over time by interpolating the tracked hand motion; (3) we present the bursting animation method of soap film that introduces local draining parameters due to deformations and gravity, makes a hole on soap film meshes, and propagates the holes gradually; and (4) we visualize the standing waves in the soap film, due to movements of wands, using an air flow transfer method.

2. Related Work

2.1. Related Work in Human Interaction Using Kinect

In recent years, a considerable number of studies have been actively conducted to recognize human motion and handle user interaction using Microsoft Kinect. Murata and Shin [4] proposed a method to recognize alphabets and numbers represented by hand movements, and Yang [5] introduced a method to recognize sign language from hand movements and shapes. A study was conducted to analyze the home gait based on Kinect and perform a fall risk assessment for the elderly [6]. Another study was proposed to estimate the control parameters of a virtual simulator for military training from trainer skeleton information [7]. Choi et al. [8] proposed a method that generates character juggling skillfully using the motions of a novice user with a Kinect. These previous works focused on methods of analyzing and recognizing human movements using Kinect, but it is difficult to find a study in which 3D bubble simulations of virtual environments are driven by human motion.

2.2. Related Work in Soap Bubble Simulation

Numerous studies have been carried out to physically simulate and model soap bubbles. Durikovic [9] proposed a method to create small spherical soap bubbles from a fixed round tube and simulate elastic deformations based on a mass-spring system. Wei et al. [10] simulated deformation and oscillation of soap bubble using spherical harmonics, and Huszar and Szecsi [11] proposed a fast method that renders dense soap foam and models foam interaction. Zheng et al. [12] presented a regional level set technique to represent complex bubble clusters, and Pan et al. [13] proposed a robust method to model soap films and soap bubbles based on an energy minimization approach. Recently, Da et al. [14] proposed a technique that simulates realistic bubble clusters and deformation of soap films and bubbles using a Lagrangian vortex sheet, and Kim et al. [3] introduced a method to simulate the formation of giant soap bubbles by using the varying surface tension with Gibbs-Marangoni elasticity. These previous studies focused on developing algorithms that handle the physical and geometrical characteristics of soap films and soap bubbles, whereby the motion of the blowing tool was simulated with a manual input or fixed state. To the best of our knowledge, no efficient method is available that allows users to experience the blowing and bursting of 3D soap bubbles by their motions.

2.3. Related Work in Air Effects

Many physics-based modeling techniques have been presented to simulate the air effects of soap bubble and cloth. A study that simulates the interaction of a wind field with a lightweight deformable object including soap bubble has been proposed [10], but it did not consider the air effect inside the soap bubble. The inflatable model [15], in which air is inflated inside the deformable surface, has been proposed, but it focused on the air pressure inside the deformable surface and did not consider air flow. The giant soap bubble creation model [3] has been proposed, but it treated air effect as air pressure based on Young Laplace's law. Chen et al. [16] proposed an efficient technique that realistically simulates air effects between cloth and a deformable object based on Darcy's law and the air transfer method. However, these previous studies that deal with air effects of the soap bubble and cloth did not take into account the air flow inside the giant soap bubble.

3. System Overview

The system proposed in this study focuses on enhancing the controllability and reality in the giant soap bubble simulation. The system pipeline and system configuration for this study are shown in Figures 2 and 3. The system can be separated into three major components as follows: (1) User Interaction: refines the motion of the user's hand through the trajectory stabilization and velocity stabilization and converts the stabilized hand movement into a controllable wand motion; (2) Soap Bubble Simulation: animates giant soap bubbles through a physics-based simulation framework

including air flow and surface tension; and (3) Geometry Processing: realistically represents the soap film bursting and soap solution control by efficient mesh processing.

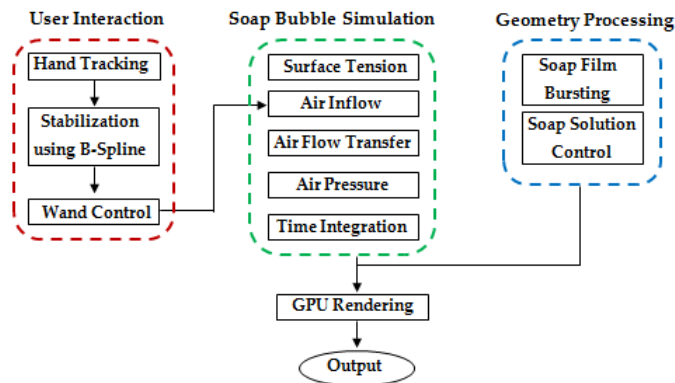


Figure 2. The system pipeline of the proposed method.

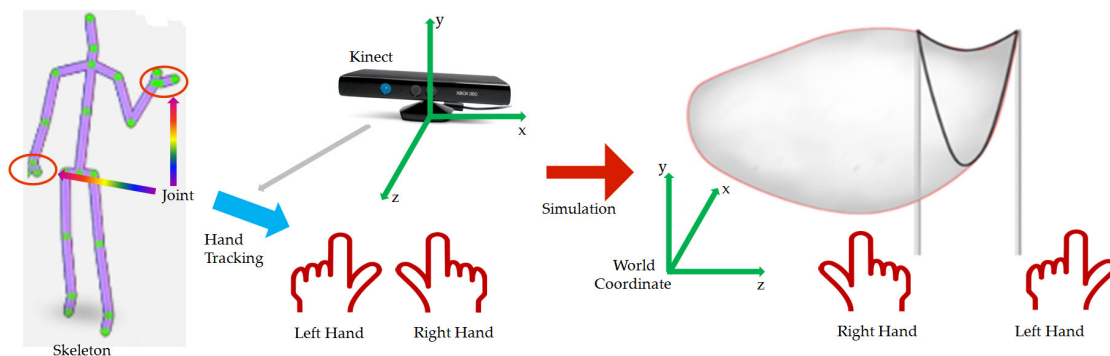


Figure 3. Proposed system configuration diagram.

4. User Interaction

We apply a stabilization method to minimize the input noise of the Kinect depth camera and to facilitate the ease of making virtual soap bubbles. Through the proposed user interaction approach, the shape, size, and position of the soap bubble can be controlled according to the user's hand movements.

4.1. Hand Position Acquisition

The shape of the soap bubbles is largely determined by the movement of the bubble wand, which is in turn influenced by the movement of the user's hands. Since the user uses the wand in the hand when manipulating the bubble wand, the individual movement of the fingers and the snap of the wrist have little effect. Therefore, we focus on the motion of the user's hands and convert the motion of the user's left hand and right hand into the movement of the left wand and right wand in the virtual world. We then use the tracked wrist positions obtained by the skeleton tracking of the Kinect SDK as the positions of the user's hands.

4.2. Hand Trajectory

Since Kinect is a camera device, it is difficult to acquire fundamentally perfect three-dimensional coordinates, and noise and drift can occur. A stabilization method is needed to minimize problems such as noise, drift, and the recognition rate of the Kinect. To overcome these problems, we employ a B-spline curve to stabilize the Kinect input data. This approach was successfully used as a rapid smoothing method for motion planning trajectories in a previous robotics study [17]. The B-spline

curve passes only the start and end control points, and excludes the control points that are significantly outside of the path. This feature of the B-spline curve makes it possible to convert jerky paths into smooth paths.

Our approach deals with stabilization using a cubic B-spline. A B-spline curve is defined as a linear combination of control points and B-spline basis functions, and the method of constructing the B-spline basis functions is based on that found in the literature [18]. The B-spline curves are updated by adding the tracked hand positions as a control points of the B-spline curve at each frame. For each frame, the points on the B-spline curve are searched sequentially to obtain smoothed hand trajectory data.

4.3. Hand Velocity

If the velocity of the wands is adjusted constantly, the soap film does not burst satisfactorily, rendering the giant soap bubbles easier to form. We apply hand velocity stabilization to prevent any sudden change of wand velocity. The hand velocity is obtained using the finite difference method [19], and we add hand velocity to the control point of the B-spline curve. This stabilized hand velocity (v_{hand}^t) is applied to the Euler integration to obtain the position of the wand as shown in Equation (1). Then, the position of the wand calculated in the Kinect coordinates is converted into the 3D world coordinates where the simulation is performed. Through this, wand trajectory with the property of a smooth and gentle shortcut is created for smooth wand movement. Figure 4 shows the input hand trajectories from the Kinect, the interpolated hand trajectories using the B-spline, and the wand trajectories.

$$\mathbf{x}_{wand}^{t+h} = \mathbf{x}_{wand}^t + h\mathbf{v}_{hand}^t \quad (1)$$

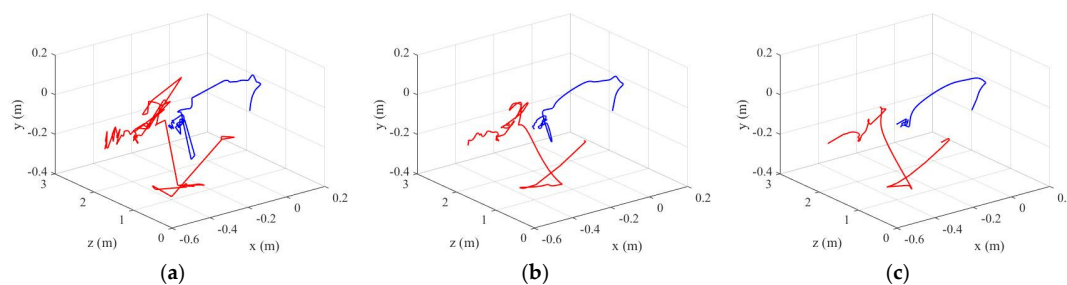


Figure 4. (a) Input hand trajectories; (b) interpolated hand trajectories; and (c) wand trajectories. Left hand (red curve), right hand (blue curve).

5. Soap Bubble Simulation

We use a unified framework to simulate soap film and bubbles, which consists of a thin shell model based on a mass-spring system. Each soap molecule is represented by a particle, each particle is connected to the neighboring particles by springs, where the rest length l_{ij} is determined by the initial distance between connected particles. Our thin shell model is governed by the Lagrangian dynamics equation of motion by the following equation:

$$m_i \ddot{x}_i = \overset{\rightarrow{s}}{\mathbf{f}}_i + \overset{\rightarrow{d}}{\mathbf{f}}_i + \overset{\rightarrow{n}}{\mathbf{f}}_i + \overset{\rightarrow{p}}{\mathbf{f}}_i + \overset{\rightarrow{x}}{\mathbf{f}}_i \quad (2)$$

where m_i is the mass of particle p_i , x_i is the position of particle p_i , and \ddot{x}_i is the second derivative of x_i with respect to time. Applied forces are given by surface tension $\overset{\rightarrow{s}}{\mathbf{f}}_i$, damping force $\overset{\rightarrow{d}}{\mathbf{f}}_i$, air inflow force $\overset{\rightarrow{n}}{\mathbf{f}}_i$, air pressure force $\overset{\rightarrow{p}}{\mathbf{f}}_i$, and external force $\overset{\rightarrow{x}}{\mathbf{f}}_i$. External force consists of gravity and other force fields, and damping force is proportional to the velocity difference projected onto the spring.

5.1. Surface Tension

A soap bubble gradually evolves into a minimal surface with symmetric properties. This occurs because of the effect of surface tension. Surface tension that minimizes the surface area of the soap bubble is given by the following equation:

$$\vec{f}_i^s = E_i \sum_j (|x_j - x_i| - l_{ij}) \frac{x_j - x_i}{|x_j - x_i|} \quad (3)$$

where l_{ij} is the rest length of a spring that is connected from p_i to p_j , x_i and x_j are the positions of particles p_i and p_j respectively. E_i is the varying elasticity of particle p_i , and is obtained separately. We apply Marangoni elasticity [3] to the initial soap film, which diffuses the surfactant evenly on the soap film. We then apply Gibbs elasticity [3] to the soap film, which allows the soap bubble to reach a minimal surface. Marangoni elasticity is proportional to the surfactant concentration and Gibbs elasticity is in inverse proportion to the thickness of soap film. E_i is given by the following:

$$E_i = \begin{cases} 2w_E \frac{e_i k}{e_i + 2k} & \text{Marangoni Elasticity : life time } < 10 \\ 4w_E \frac{e_i k^2}{(e_i + 2k)^2} & \text{Gibbs Elasticity : life time } > 10 \end{cases} \quad (4)$$

where k is typically 4 μm , w_E is typically 12.3. The thickness of soap film e_i is given by the following: $e_i = \frac{w_e}{A_{Vor}(p_i)}$, where w_e is the area weight and $A_{Vor}(p_i)$ is the Voronoi area [20] of particle p_i . The pseudo code for surface tension force and damping force is shown in Algorithm 1.

Algorithm 1 Pseudo-code for surface tension and damping force

```

1: for all springs do
2:    $x_i, x_j$ : position of connected particle by spring
3:    $\vec{f}_i^s = E_i \left( \|x_i - x_j\| - l \right) \frac{x_i - x_j}{\|x_i - x_j\|}$ 
4:    $\vec{f}_j^s = E_j \left( \|x_j - x_i\| - l \right) \frac{x_j - x_i}{\|x_j - x_i\|}$ 
5:    $\vec{f}_i^d = k_d \left( \mathbf{v}_j - \mathbf{v}_i \right) \cdot \frac{x_j - x_i}{\|x_j - x_i\|}$ 
6:    $\vec{f}_j^d = -\vec{f}_i^d$ 
7: end for

```

5.2. Air Effects

Bubble artists generate a variety of deformations and standing waves in soap film using air effects created by the movement of the wand to make giant soap bubbles with complex and detailed surfaces. This study proposes an efficient method that focuses on the air flow in order to realistically represent this air effect. First, we model the air inflow created by the movement of the wand. We then propose an air flow transfer algorithm that handles the phenomenon in which the air flowing into the soap film is transferred. After the soap film is converted to a soap bubble, we treat the air pressure in consideration of the air flow inside the soap bubble.

5.2.1. Air Inflow

We estimate the local air inflow using the relation that the air inflow is inversely proportional to the distance between each wand and each soap film particle. The air inflow for particle p_i created by the motion of the wand is calculated using the following equation:

$$\vec{air}_i = -V_{wm} (\text{MAX}(\cos \theta, 0)) \left(\frac{\vec{v}_L}{\text{dist}_{Lwand} + 1} + \frac{\vec{v}_R}{\text{dist}_{Rwand} + 1} \right) \quad (5)$$

$$V_{wm} = H_{wire} \text{Area}_{wm} \quad (6)$$

where $dist_{Lwand}$ and $dist_{Rwand}$ are the distances from the particle p_i to the left and right wands, respectively. \vec{v}_L is the velocity of the left wand, \vec{v}_R is the velocity of the right wand, and θ is the angle between the velocities of the left and right wands. As the direction and velocity of the left and right hands become more similar, the soap film becomes more inflated. For example, when the left and right wands are gathered together, the soap film does not inflate. V_{wm} is an approximation of the air volume generated by the movement of the left and right wands, and H_{wire} is the height of the wire, and $Area_{wm}$ is the area where the wands move. Figure 5 shows a schematic diagram of the air inflow.

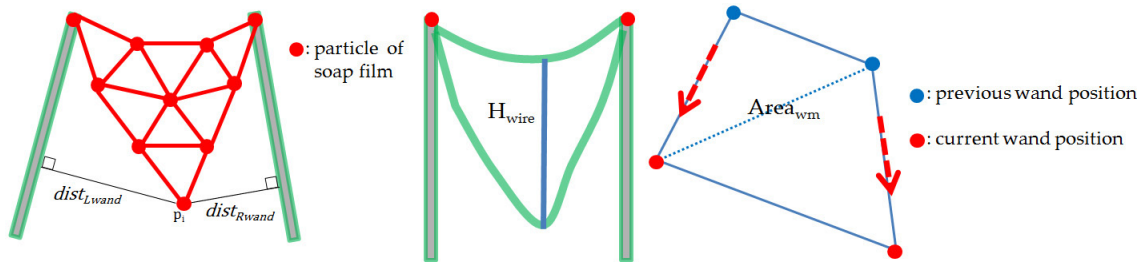


Figure 5. Schematic diagram of the air inflow.

5.2.2. Air Flow Transfer

Dynamic volumetric air flow methods (such as that of [21]) can simulate realistic air flows at a reasonable computational cost, but they can cause large deformations and self-collisions in giant soap film mesh undergoing topological change. We exploit the triangle elements that convey thickness of soap film meshes and material properties based on Gibbs and Marangoni elasticity. These properties are varied spatially and temporally during simulation and are the key factors of our proposed method. Therefore, our approach requires the same mesh connectivity from the beginning and is based on an explicit modeling method that is difficult to handle topological changes. This study proposes a new air flow transfer method, inspired by an existing, efficient air transfer method between cloth and a deformable body [16], as an alternative to solving this problem in real time.

The effect of air flow transferred into the inflated soap film is processed in a fast and efficient manner. The schematic diagram of the algorithm is shown in Figure 6 and the pseudo code is shown in Algorithm 2. Since air inflow is created due to the movement of the wand, the air flow transfer is handled from the end particles of soap film which is first affected by the movement of the wand. First, the generated air inflow \vec{air}_i is assigned to the end particles. We then progressively propagate the \vec{air} of each particle to its neighboring particles of which the direction is similar to \vec{air} . Whether the direction is similar to \vec{air}_i is determined by $a_{dir} > \lambda_{air}$, where λ_{air} is typically selected from 0.5 to 0.7, and a_{dir} is given by the following:

$$a_{dir} = \text{MAX} \left(\frac{(x_j - x_i) \cdot \vec{air}_i}{|x_j - x_i| |\vec{air}_i|}, 0 \right) \quad (7)$$

$$\vec{air}_j^f = \frac{\sum a_{dir} \vec{air}_i}{\sum a_{dir}} \quad (8)$$

where x_i is the position of the currently selected particle and, x_j is the location of a neighboring particle.

Algorithm 2 Pseudo-code for Air Flow Transfer

```

1: if air inflow $\vec{f}$  is created
2:   assign  $air_i$  to the end particles
3: for each particle  $p_i$  do
4:   if particle  $p_i$  has air flow then
5:     for each particle  $p_j \in$  one-ring neighborhood particles of particle  $p_i$  do
6:       if  $a_{dir} > \lambda_{air}$  then
7:         add  $air_i$  to  $air_j$ 
8:       end for
9:     delete  $air_i$ 
10:  end for

```

We apply the air inflow force \vec{f}_i before the soap film is converted to the soap bubble, which is given by the following equation:

$$\vec{f}_i = \text{proj}_{\vec{n}_i}^{\vec{f}} air_i + \text{proj}_{\vec{n}_i}^{\vec{f}} air_i \quad (9)$$

where \vec{n}_i is normal vector of particle p_i . $\text{proj}_{\vec{n}_i}^{\vec{f}} air_i$ represents the force applied to the soap film by the air flow transferred to the inflated soap film, and $\text{proj}_{\vec{n}_i}^{\vec{f}} air_i$ represents the force with which the air inflow air directly acts on the soap film. The air inflow force describes the air flow transferred to the inflated soap film and the air inflow directly acting on the soap film.

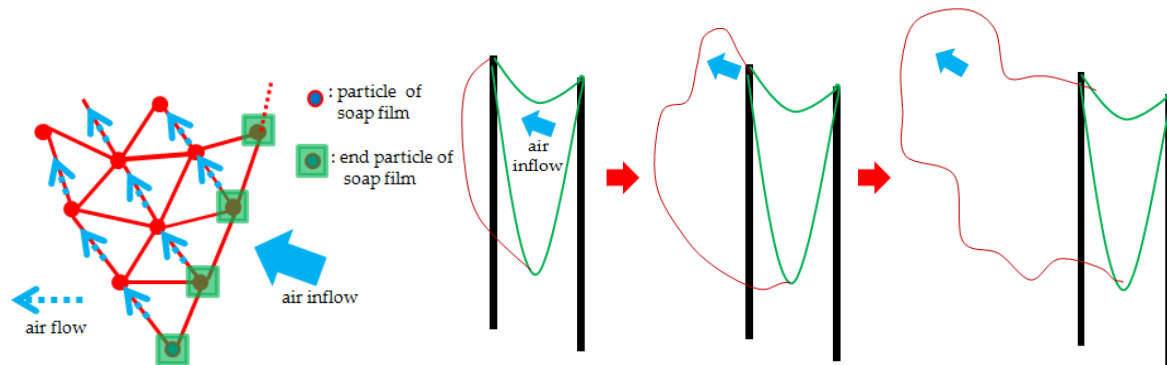


Figure 6. Schematic diagram of air flow transfer (starting with the end particles of the soap film, air is introduced into the adjoining soap film particles).

5.2.3. Air Pressure

A large amount of air is contained inside the giant soap bubble, and the air flow is high. Because of this internal air flow, the deformation and standing waves of the giant soap bubbles are enhanced. Our proposed method efficiently represents the phenomenon of air flow interaction inside a giant soap bubble based on air flow transmission, whereby the air flow transfer is performed until the bubble bursts. The air pressure force of each soap film particle is given by following equation:

$$\vec{f}_i^p = P_{cur} \text{proj}_{\vec{n}_i}^{\vec{f}} air_i \quad (10)$$

where P_{cur} is the pressure coefficient of the current bubble calculated based on Boyle's law, which explains that the pressure of the bubble is inversely proportional to the volume of the bubble. P_{cur} is given by the following equation:

$$P_{cur} = (P_0 V_0) / V_{cur} \quad (11)$$

where $P_0 = 1.01$ kPa is the atmospheric pressure coefficient, V_0 is the volume of the initial bubble, and V_{cur} is the volume of the current bubble.

5.3. Time Integration

We use semi-implicit time integration to update the position and velocity of particles at each time step. The efficiency of semi-implicit time integration was recently effectively used for a deformable surface model of a water drop [22], flower [23], and cloth [24]. Since the motion of the soap bubble is dominated by the boundary surface due to the surface tension effect, semi implicit time integration can preserve the surface detail. The position and velocity of a particle is updated using Equation (3) at each time step,

$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + h \overset{\rightarrow}{\mathbf{f}}_i^t / m_i \quad (12)$$

$$\mathbf{x}_i^{t+h} = \mathbf{x}_i^t + h \mathbf{v}_i^{t+h} \quad (13)$$

where h is the time step, \mathbf{x}_i^t and \mathbf{v}_i^t are the current position and velocity of particle p_i , respectively, and \mathbf{x}_i^{t+h} and \mathbf{v}_i^{t+h} are the position and velocity of particle p_i , respectively, at the next time step.

6. Soap Film Bursting

The main cause of the bursting of the soap bubbles is draining [25] or excessive stretching. The bursting of giant soap bubbles is an important factor for visual realism because the progress of the soap film bursting is discernible to the naked eye, unlike small spherical soap bubbles. In order to represent this phenomenon realistically, this study is based on the principle of hole opening propagation which has been widely studied in the field of physics. The principle of hole opening propagation [26] explains the phenomenon in which the cracks propagate outward from the point of puncture when the soap film cracks. This phenomenon is described by Taylor Cullick's Law [27,28]. Based on this principle, we propose a method that generates the soap film bursting in real-time for the soap film and the soap bubble in a unified framework.

6.1. Hole Opening Propagation

The conceptual diagram of the proposed method is shown in Figure 7 and the pseudo code is shown in Algorithm 3. First, we define the occurrence of the cracks in the particles of which the thickness is thinner than thickness threshold (λ_{thi}) due to draining or excessive stretching. The crack is transferred by removing the adjacent particles of the crack. When all three particles constituting the triangle are removed, the corresponding triangle is removed from the mesh. In addition, we represent the appearance of aerosol [25] composed of soap molecules and water molecules constituting the soap film when the soap film burst by generating spherical particles at the center of gravity of the removed triangles. Each spherical particle is evenly distributed over the properties of the three vertices corresponding to a triangle, such as force, velocity, and mass.

Algorithm 3 Pseudo-code for Soap Film Bursting

- 1: Capture particles of which the thickness are thinner than λ_{thi}
 - 2: Set up these particles as *bursting particles*.
 - 3: Set one-ring neighborhood of *bursting particles* as *next-bursting particles*
 - 4: Remove *bursting particles* and their springs.
 - 5: **if** all three particles constituting the triangle are removed **then**
 - 6: Remove the corresponding triangle from the mesh.
 - 7: Copy *next-bursting particles* to *bursting particles*.
 - 8: Remove *next-bursting particles*.
 - 9: Repeat step 3~8 until all particles have burst.
-

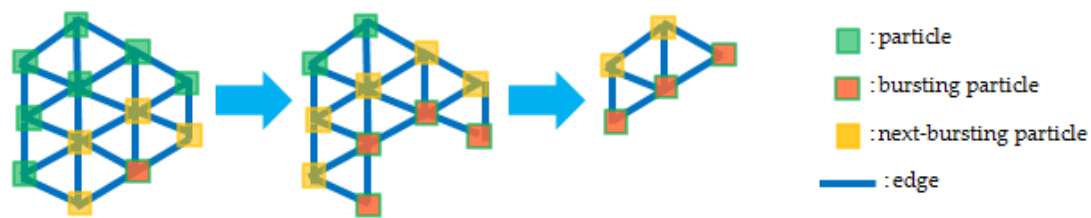


Figure 7. Conceptual diagram of soap film bursting.

6.2. Position-Dependent Draining

Drainage is a natural phenomenon in which the water in the walls of the soap bubble drains to the bottom of the soap bubble. Because the previous draining model [12] linearly decreases the thickness of the small bubble, their model does not handle position-dependent movement of a water molecule for a large bubble. We propose a novel drainage model based on water molecule propagation which considers the movement of the water molecule and topology change. Our drainage model progressively propagates the water molecule of each particle to its neighboring particles, the directions of which are most similar to the gravity direction using the following equation:

$$d_{dir} = \text{MAX} \left(\frac{(\mathbf{x}_j - \mathbf{x}_i) \cdot \vec{g}}{|\mathbf{x}_j - \mathbf{x}_i| |\vec{g}|}, 0 \right) \quad (14)$$

$$e_j + = d_{dir} / \text{Sticky} \quad (15)$$

where \vec{g} refers to gravity. *Sticky* refers to the viscosity of the soap solution, where *Sticky* is increased to delay drainage and extend the life of the soap bubble. d_{dir} determines the degree to which the direction of the neighboring particles are similar to the gravity direction. Figure 8 shows our drainage results, where the top region becomes thinner and the bottom region becomes thicker because water molecules on the top region move to the bottom region over time under the influence of gravity.

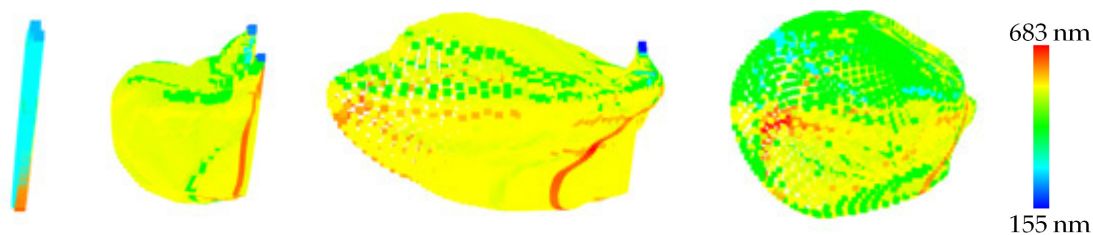


Figure 8. Draining results. Blue color represents thin regions, red color represents thick regions.

7. Soap Solution Control

Because our model is based on the assumption that a wire frame is made wet by a soap solution, the first soap film is initially created. As the amount of remaining soap solution is decreased, the amount of soap film stretching also decreases, so that the shape of the soap bubble becomes rounder and the size decreases. To mimic this phenomenon, we dynamically adjust the mesh resolution of the soap film in proportional to the amount of remaining soap solution. We adjust the number of soap film particles placed on the wire frame in proportion to the amount of residual soap solution, and place the soap film particles on the wire frame using spherical linear interpolation. We then create a triangular mesh using Delaunay Triangulation [29] and then modify the edges of the mesh into springs. Figure 9 shows that the shape and size of the soap bubble varies in proportion to the amount of remaining soap solution.

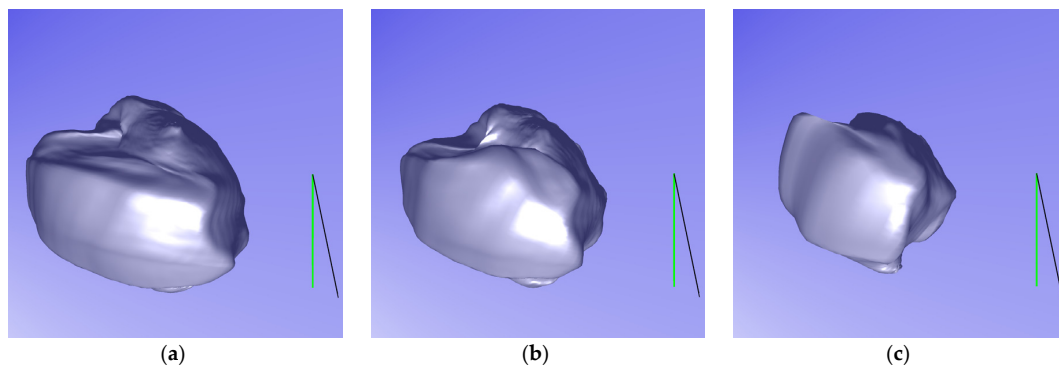


Figure 9. Results by the amount of remaining soap solution: (a) 1.0; (b) 0.75; and (c) 0.5.

8. Experimental Results and Analysis

We implemented the system using KinectSDK-v1.8 (Microsoft, Redmond, WA, USA) and multi-threading, which runs on an Intel i5 3.2 GHZ (Intel Corporation, Santa Clara, CA, USA), 16 GB RAM, Nvidia GTX 560 (NVIDIA, Santa Clara, CA, USA) PC with one Kinect. The proposed system controls the wand motions based on the user's hand motions that are tracked with from Kinect input, and the air inflow, air flow transfer, and soap film bursting are simulated according to the wand's motion. We employ a previous rendering method [3] for realistic soap bubble rendering to represent interference and reflection, as well as the refraction phenomenon in real time. The self-collisions of the soap bubble are detected using a bounding volume hierarchy [30] and the collision response is computed using collision forces [22], which result in slipping of the colliding particles. If the impact with which the soap film collides is higher than the colliding threshold (λ_{col}), the soap film bursts.

Figure 10 shows the simulation results according to the user's hand movements (left), including the left wand trajectory (middle), and the right wand trajectory (right). It also shows the input hand trajectory (dotted line), wand trajectory (full line), starting point (+), end point (\times), and current point (square). Figure 10a shows that when the trajectories of the left and right wands are straight with few redirections, the soap bubble grows in a straight direction. Figure 10b shows that when the trajectory of the left wand moves in a circle and the trajectory of the right wand is straight, ridges and valleys appear on the left side of the soap bubble. Figure 10c shows that when the trajectories of the left and right wands all move in a circle, ridges and valleys appear throughout the soap bubble. Figure 10d shows that when the trajectories of the left and right wands are increasingly redirected and twisted, and the number of movements is large, the soap film bursts. Figure 10e shows that when the trajectories of the left and right wands are less twisted, the number of movements is large, and when the left and right wands move up and down, the soap bubble is inflated further. Figure 10f shows that when the left and right wands are moved up from the bottom, the soap bubble is tilted downward and ridges and valleys appear at the top of the bubble.

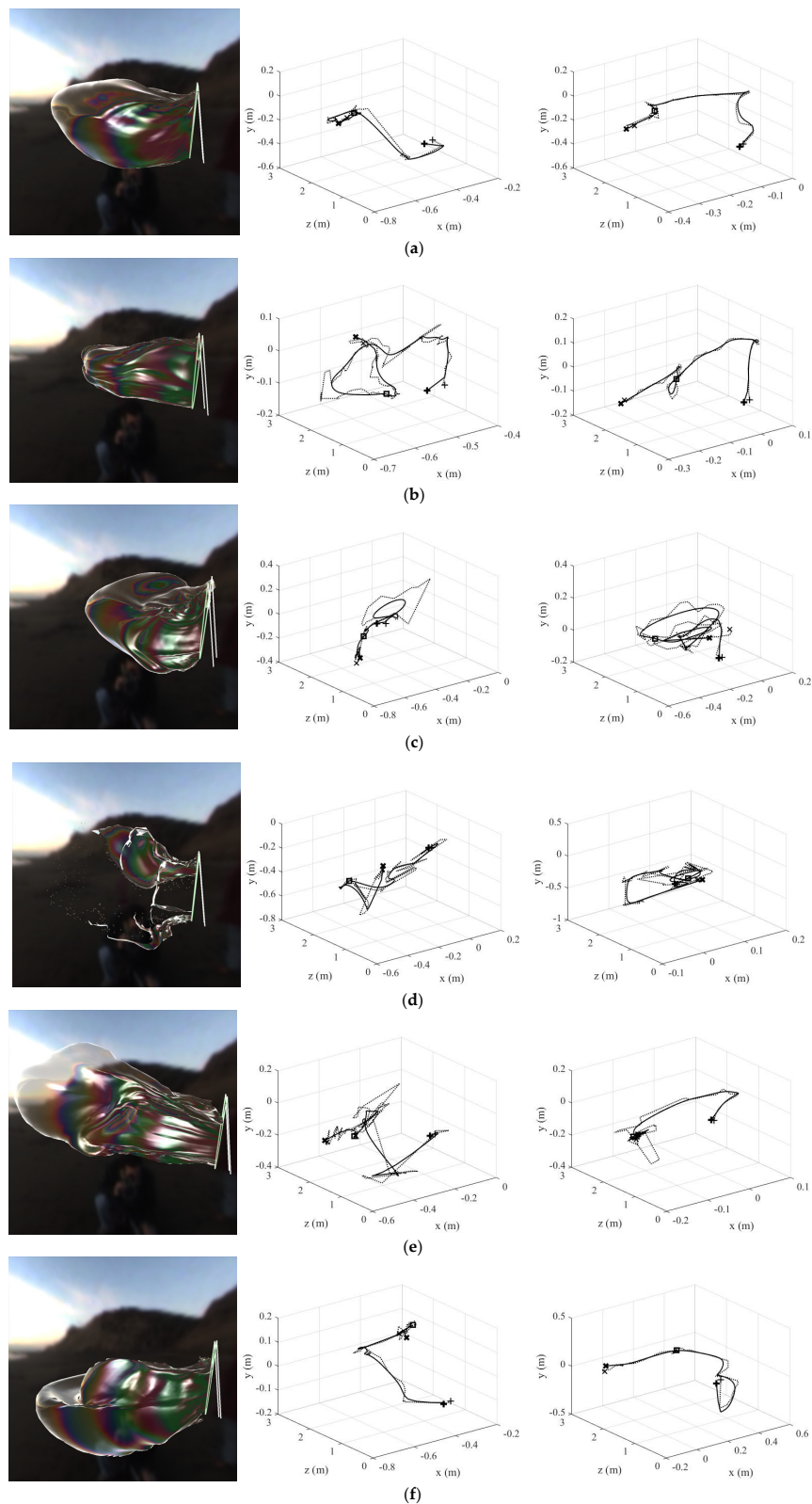


Figure 10. Simulation results (left), left wand trajectory (middle), right wand trajectory (right). Input hand trajectory (dotted line), wand trajectory (full line), starting point (+), end point (x), current point (square). When the user's hands movements are gentle (a–c). When user's hands movements are sudden (d–f).

Figure 11 shows the simulation results of the soap film bursting method. As the crack propagates in the outward direction, the bursting soap films transform into spherical particles similar to those of an aerosol.

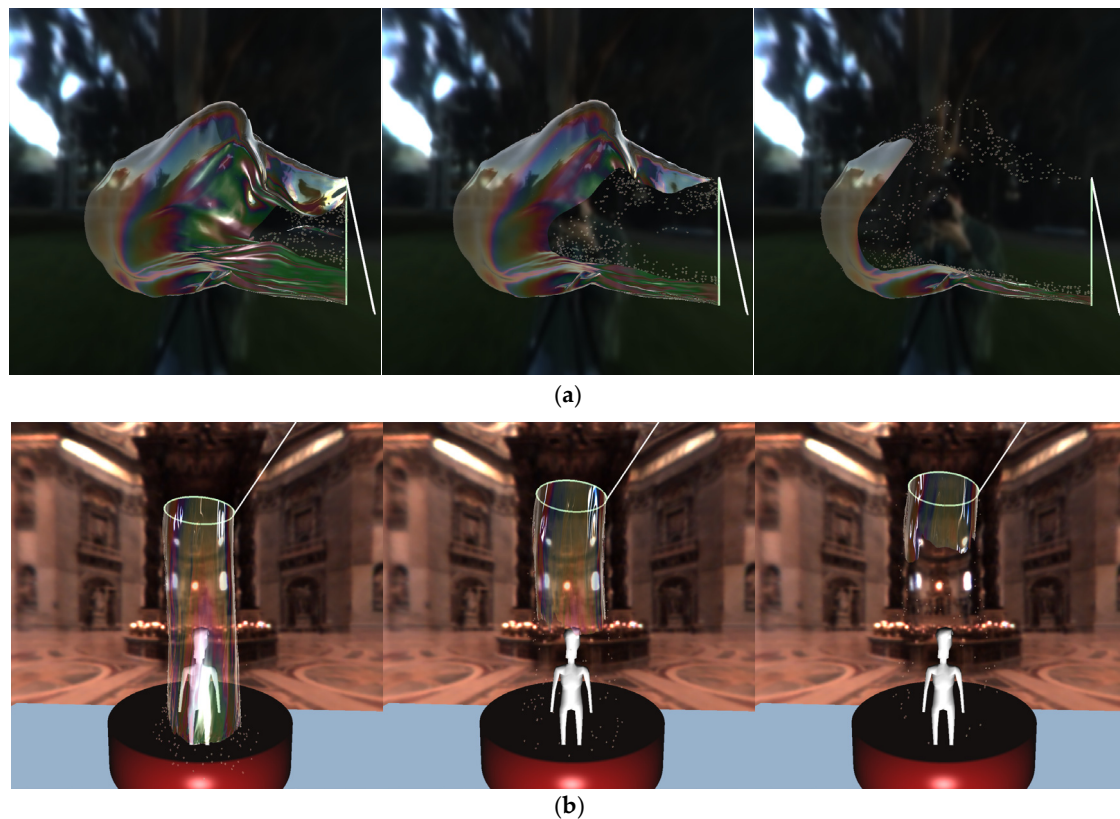


Figure 11. Simulation results by our soap film bursting method. (a) When using a tool with two wands; and (b) when using a tool with single wand.

Figure 12 shows a comparison between the existing method [3] and our method. Because the existing method [3] does not consider air flow inside the bubble and applies internal pressure to the soap film, it is difficult to generate the detailed deformation and an overall flat surface is often produced. Our model controls the air flow inside the soap bubble based on an air flow transfer algorithm, in order to generate detailed deformation and standing waves. In addition, our model processes air flow transfer with negligible computational cost using the efficient operations between one-ring neighboring particles, which are directly accessible in the mass-spring system. Figure 13 shows the color-coded results of Figure 12b. The blue color indicates weak air flow while the red color indicates strong air flow. When the wand moves towards the right direction, air flow moves towards the left. Air flow accumulates as the wand is moved, which shows that the intensity of the air flow is stronger than in the previous state. Table 1 shows the computation times for each result and Table 2 shows the parameters used in our system.

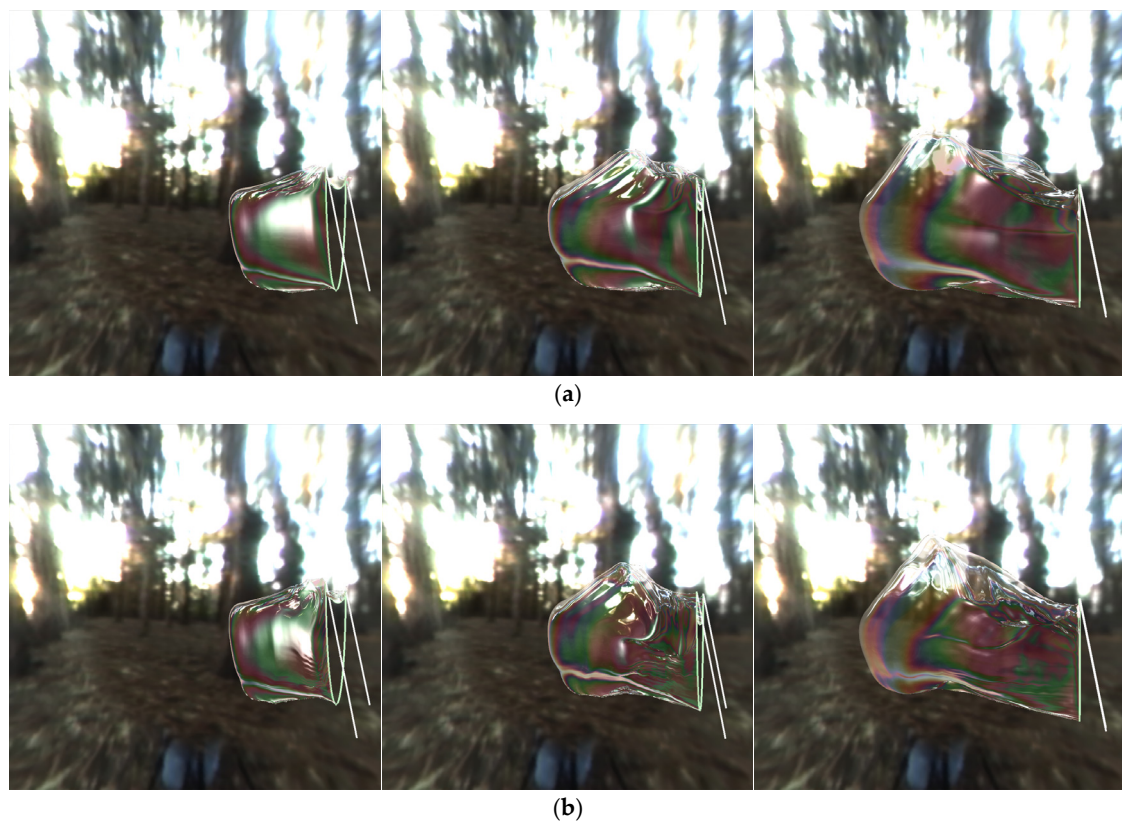


Figure 12. Comparison of air effects. (a) Existing method [3]; and (b) our method.

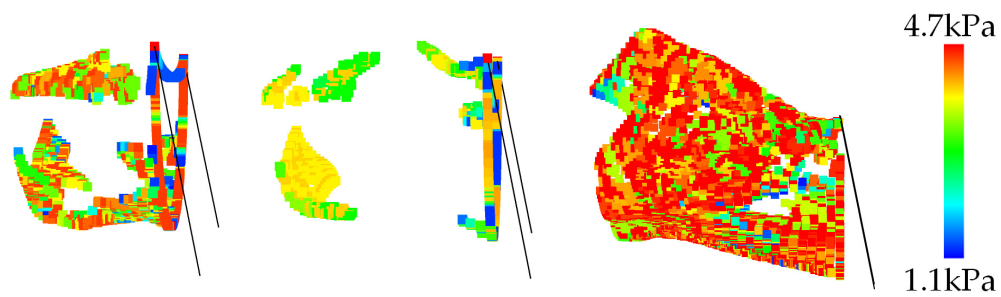


Figure 13. Results of the color coding of the strength of the air flow in Figure 12b.

Table 1. The computation times (frames per second) for each result.

Example	Avg. Particle Size	Avg. Triangle Size	Avg. Time (fps) (Including Rendering)
Figure 10a	9452	15,324	31.2
Figure 10b	9088	12,542	35.0
Figure 10c	9868	21,036	28.5
Figure 10d	9252	14,282	31.3
Figure 10e	16,020	32,568	20.1
Figure 10f	13,882	28,324	22.5
Figure 11a	13,056	28,112	23.1
Figure 11b	25,650	41,208	18.1
Figure 12a	10,026	20,328	26.5
Figure 12b	11,328	23,580	27.3

Table 2. Parameters used in our system.

Notation	Parameter Name	Range
w_E	Varying elasticity coefficient (Section 5.1)	[12.0~13.0]
w_e	Area weight (Section 5.1)	[0.02~0.05]
λ_{air}	Air direction threshold (Section 5.2.2)	[0.5~0.7]
h	Time step (Section 5.3)	0.02
λ_{thi}	Thickness threshold (Section 6.1)	[20~50]
<i>Sticky</i>	Viscosity of the soap solution (Section 6.2)	[20.0~30.0]
λ_{col}	Colliding threshold (Section 8)	[1.5~2.0]

Figure 14 shows the comparisons of our simulation results with real footage. Figure 14a,b show the actual footage and simulation results, respectively, when using a tool with two wands. The simulation results show the inflation of the soap film in the left direction, opposite the direction in which the wand moves, as shown in the footage. The simulation results also show that the soap film surface is very smooth and curved as seen in the footage. Figure 14c,d show the real footage and simulation results, respectively, when creating a giant soap bubble from a soap solution using a single wand tool. The simulation results show that the soap bubble is created according to the direction in which the wand moves from bottom to top, as shown in the footage. In addition, although the new surface continues to be generated according to the motion of the wand, the simulated results show that the soap film surface is smooth and the standing wave is expressed as seen in the footage.

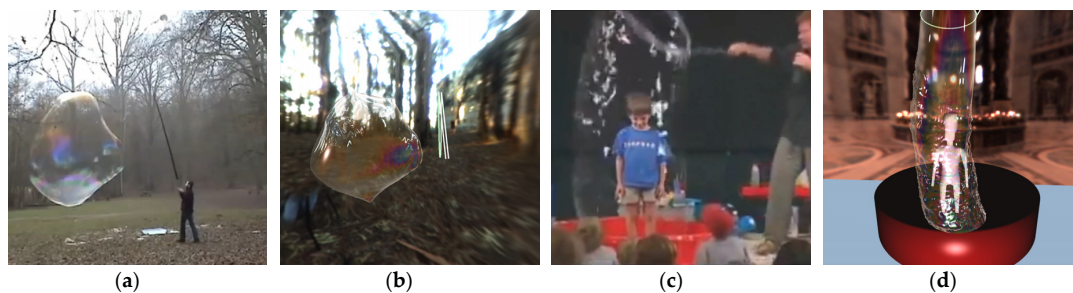


Figure 14. Comparison of simulation results and real footage. (a) The real footage and (b) simulation result, respectively, when using a tool with two wands. (c) The real footage and (d) simulation result, respectively, when using a tool with single wand.

9. Conclusions and Future Work

We present a novel method that automatically generates realistic animated giant soap bubbles according to the user's hand motions. Our system allows real-time control of the shape, size, and position of soap bubbles in accordance with the user's hand trajectory, direction of motion and distance travelled. We obtain smooth and gentle wand trajectories using a smoothing method for hand motions. Our air flow transfer algorithm represents the detailed deformation and standing wave of soap film. Our soap film bursting algorithm produces the bursting effect of soap films and bubbles based on hole opening propagation and position-dependent draining in a unified framework. The proposed system can be used for physics education and media art, and enables users to experience virtual giant soap bubble blowing and bursting using the consumer level depth camera Kinect.

In future work, we intend to investigate full body capture and minimize the occlusion problem that arises with the frequent turning and twisting motion. We will also attempt to accelerate our model by parallelizing our simulation method entirely on the GPU.

Supplementary Materials: The following are available online at <http://www.mdpi.com/2073-8994/9/4/56/s1>, Supplemental Video S1.

Acknowledgments: This work (NRF-2016R1A2B4016239) was supported by Mid-career Researcher Program through NRF grant funded by the MEST.

Author Contributions: Namjung Kim implemented the proposed technique and performed the experiments. Namjung Kim and Kyoungju Park wrote the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ballet, P.; Graner, F. Giant soap curtains for public presentations. *Eur. J. Phys.* **2006**, *27*, 951–967. [[CrossRef](#)]
2. Zhu, B.; Quigley, E.; Cong, M.; Solomon, J.; Fedkiw, R. Codimensional surface tension flow on simplicial complexes. *ACM Trans. Graph.* **2014**, *33*. [[CrossRef](#)]
3. Kim, N.; Oh, S.; Park, K. Giant soap bubble creation model. *Comput. Anim. Virtual Worlds* **2015**, *26*, 445–455. [[CrossRef](#)]
4. Murata, T.; Shin, L. Hand gesture and character recognition based on kinect sensor. *Int. J. Distrib. Sens. Netw.* **2014**. [[CrossRef](#)]
5. Yang, H.-D. Sign language recognition with the Kinect sensor based on conditional random fields. *Sensors* **2014**, *15*, 135–147. [[CrossRef](#)] [[PubMed](#)]
6. Stone, E.E.; Skubic, M. Unobtrusive, continuous, in-home gait measurement using the microsoft kinect. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 2925–2932. [[CrossRef](#)] [[PubMed](#)]
7. Lee, S.; Park, S.; Chung, K.; Cho, C. Kinematic skeleton based control of a virtual simulator for military training. *Symmetry* **2015**, *7*, 1043–1060. [[CrossRef](#)]
8. Choi, J.; Kim, S.; Kim, C.; Lee, J. Let's be a virtual juggler. *Comput. Anim. Virtual Worlds* **2016**, *27*, 443–450. [[CrossRef](#)]
9. Durikovic, R. Animation of soap bubble dynamics, cluster formation and collision. *Comput. Graph. Forum* **2001**, *20*, 67–75. [[CrossRef](#)]
10. Wei, X.; Zhao, Y.; Fan, Z.; Li, W.; Yoakum-Stover, S.; Kaufman, A. Blowing in the wind. In Proceedings of the Symposium Computer Animation, San Diego, CA, USA, 26–27 July 2003; pp. 75–85.
11. Huszar, T.; Szecsi, L. GPU-supported bubble and foam rendering. In Proceedings of the CESC 2010, Budmerice, Slovakia, 10–12 May 2010; pp. 149–155.
12. Zheng, W.; Yong, J.-H.; Paul, J.-C. Simulation of bubbles. *Graph. Models* **2009**, *71*, 229–239. [[CrossRef](#)]
13. Pan, H.; Choi, Y.-K.; Liu, Y.; Hu, W.; Du, Q.; Polthier, K.; Zhang, C.; Wang, W. Robust modeling of constant mean curvature surfaces. *ACM Trans. Graph.* **2012**, *31*. [[CrossRef](#)]
14. Da, F.; Batty, C.; Wojtan, C.; Grinspun, E. Double bubbles sans toil and trouble: Discrete circulation-preserving vortex sheets for soap films and foams. *ACM Trans. Graph.* **2015**, *34*. [[CrossRef](#)]
15. Li, L.; Volkov, V. Inflatable models. *J. Comput. Sci. Technol.* **2006**, *21*, 154–158. [[CrossRef](#)]
16. Chen, Z.; Feng, R.; Wang, H. Modeling friction and air effects between cloth and deformable bodies. *ACM Trans. Graph.* **2013**, *32*. [[CrossRef](#)]
17. Pan, J.; Zhang, L.; Manocha, D. Fast smoothing of motion planning trajectories using b-splines. In Proceedings of the Robotics: Science and Systems, Los Angeles, CA, USA, 27–30 June 2011.
18. Bartels, R.H.; Beatty, J.C.; Barsky, B.A. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*; Morgan Kaufmann: Los Altos, CA, USA, 1986.
19. Janabi-Sharifi, F.; Hayward, V.; Chen, C.-S.J. Discrete-time adaptive windowing for velocity estimation. *IEEE Trans. Control Syst. Technol.* **2000**, *8*, 1003–1009. [[CrossRef](#)]
20. Meyer, M.; Desbrun, M.; Schroder, P.; Barr, A. Discrete differential-geometry operators for triangulated 2-manifolds. *Vis. Math.* **2002**, *3*, 34–57.
21. Pfaff, T.; Thuerey, N.; Gross, M. Lagrangian vortex sheets for animating fluids. *ACM Trans. Graph.* **2012**, *31*. [[CrossRef](#)]
22. Zhang, Y.; Wang, H.; Wang, S.; Tong, Y.; Zhou, K. A deformable surface model for real-time water drop animation. *IEEE Trans. Visual. Comput. Graph.* **2012**, *18*, 1281–1289. [[CrossRef](#)] [[PubMed](#)]
23. Li, J.; Liu, M.; Xu, W.; Liang, H.; Liu, L. Boundary-dominant flower blooming simulation. *Comput. Anim. Virtual Worlds* **2015**, *26*, 433–443. [[CrossRef](#)]
24. Tang, W.; Wan, T.R.; Huang, D. Interactive thin elastic materials. *Comput. Anim. Virtual Worlds* **2016**, *27*, 141–150. [[CrossRef](#)]

25. Lhuissier, H.; Villermaux, E. Bursting bubble aerosols. *J. Fluid Mech.* **2012**, *696*, 5–44. [[CrossRef](#)] [[PubMed](#)]
26. Culick, F.E.C. Comments on a ruptured soap film. *J. Appl. Phys.* **1960**, *31*, 1128–1129. [[CrossRef](#)]
27. Petit, P.C.; le Merrer, M.; Biance, A.-L. Holes and cracks in rigid foam films. *J. Fluid Mech.* **2015**, *774*, R3. [[CrossRef](#)]
28. Bico, J. Cracks in bursting soap films. *J. Fluid Mech.* **2015**, *778*, 1–4. [[CrossRef](#)]
29. O'Rourke, J. *Computational Geometry in C*; Cambridge University Press: Cambridge, UK, 1998.
30. Tang, M.; Manocha, D.; Tong, R. Fast continuous collision detection using deforming non-penetration filters. In Proceedings of the Symposium Interactive 3D Graphics and Games, Bethesda, MD, USA, 19–21 February 2010; pp. 7–13.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).