

Received August 27, 2021, accepted October 14, 2021, date of publication October 27, 2021, date of current version November 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123435

Stable and Anisotropic Freezing Framework With Interaction Between IISPH Fluids and Ice Particles

JONG-HYUN KIM¹ AND JUNG LEE²

¹School of Software Application, Kangnam University, Yongin 16979, South Korea

²School of Software, Hallym University, Chuncheon 24252, South Korea

Corresponding author: Jung Lee (airjung@hallym.ac.kr)

This work was supported in part by the Hallym University Research Fund under Grant HRF-202011-009, in part by the Research and Development Program for Forest Science Technology by the Korea Forest Service (Korea Forestry Promotion Institute) under Project 2021390A00-2123-0105, and in part by the National Research Foundation of Korea by the Ministry of Science, ICT and Future Planning under Grant 2017R1C1B5074984.

ABSTRACT In this paper, we present a novel method that can stably express the directional ice form caused by freezing of flowing water. The key to the proposed framework is to reflect the flow of fluids with viscosity in the direction of ice growth. Water is simulated by applying a new viscous technique to the implicit incompressible fluid simulation, and the proposed anisotropic freezing solution is used to express directional ice and glaze effects. The conditions under which water particles turn into ice particles are calculated according to a new energy function based on humidity and water flow. The humidity is approximated based on the virtual water film on the surface of the object, and the flow of fluid is incorporated into our anisotropic freezing solution to guide the growth direction of the ice. As a result, the proposed technique reliably produces glaze and directional freezing effects according to the flow direction of viscous water.

INDEX TERMS Freezing water, icicle shape, implicit incompressible SPH, ice glaze, anisotropic ice growth.

I. INTRODUCTION

In various VR (Virtual Reality) contents such as movies and games, the freezing effect is used in various scenes expressing freezing scenes in cold regions or extreme environments: “The Day after Tomorrow” (2004), “The Last Airbender” (2010), “The Huntsman: Winter’s War” (2016). VR content is often produced by bending ice in a certain direction due to factors such as the surface of an object, the flow of water, or wind direction, to express immersively the extreme winter, not just ice that grows downward by gravity. In the field of computer graphics, there have been many studies to show the creation and growth of ice. Im *et al.* [1] proposed a technique to express the interior of opaque ice, and Kharitonsky and Gonczarowski [12] proposed a technique to generate and grow icicles using physics-based simulation. Kim *et al.* [2] and Gagnon and Paquette [13] proposed a technique for expressing icicles and glaze effects appearing on the surface of an object using procedural methods. Since these techniques focus on expressing the icicle and glaze effects generated by slowly falling water droplets, it is difficult to express when

considering only the growth of ice due to gravity without interacting with water.

In this paper, we propose a physics-based simulation technique that reflects the flow of water in the freezing process. Our method does not use a temperature-based model, but a crystallization-based state change model like Im *et al.*'s method [32]. Like their method, we also drew inspiration from the supercooled water phenomena, which stays liquid at temperatures below zero degrees and then freezes quickly when colliding with a cold surface (see Figure 1). In addition, in order to more accurately express the characteristics of freezing occurring as water flows, the growth direction of ice is modeled anisotropically according to the fluid flow. In the previous approaches, it was difficult to express the phenomenon of freezing as water flows along the surface of a solid, and because water particles are scattered by collisions between water and solid, the freezing modeling becomes unstable, resulting in lower freezing quality. In addition, in calculating the growth direction of ice that freezes while extending like a branch, it is difficult to realistically express the growth direction with an isotropic kernel [32]. Our technique can anisotropically generate or grow ice even when the flow of water changes dynamically due to external forces or

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.



FIGURE 1. The real image of water freezing due to the supercooled water phenomenon.

collisions with objects. Therefore, it is possible to express not only the growth of ice according to the flow of water, but also the phenomenon of rapidly freezing water.

A. PROBLEM STATEMENT

In general, the flow of fluid has a significant influence on determining the shape of the ice during simulation. Various techniques have been proposed to perform freezing simulation such as icicles in the field of computer graphics, but in most cases, the form of freezing is determined only by gravity. This causes patterned freezing, making it difficult to produce realistic freezing in various virtual environments. When modeling the winter environment, ice is created in a curved form not only by gravity but also by various environmental factors such as wind and water flow, and this feature is important when creating virtual environments or games (see Figure 2).



FIGURE 2. Ice shape formed by various physical environments.

Typical examples of external forces affecting freezing surfaces are wind and liquid flow. In particular, supercooled water is more susceptible to liquid motion because it freezes immediately (see Figure 2). However, since the existing method explicitly calculates the force based on water particles, if the speed of the water particles is high or the impact is strong due to the collision with the solid, the water particles become unstable in the process of changing to ice. As a result, ice surfaces are not generated properly or noise is generated severely. In addition, when water particles collide with a solid with large curvature, the surface tension becomes unstable, resulting in scattering of ice particles, resulting in generation of low-quality ice surfaces. Also, when using the isotropic approach to generate a curved ice surface considering liquid motion, the direction cannot be accurately estimated. Therefore, in this paper, we proposed a method to improve the

quality of freezing surfaces by solving the aforementioned stability problem.

As shown in Figure 2, it can be seen that the frozen ice near a tree or house has a curved shape due to the physical environment such as wind or water flow. In this paper, we propose a new framework that can stably express this freezing shape. To implement the proposed freezing solver, the following sub-problems must be solved in the input fluid simulation:

- 1) **Integration with reliable water simulation solvers.** Most of the freezing phase changes from water to ice, and in this paper, we propose a freezing solver based on IISPH (Implicit Incompressible SPH) to stably integrate fluid motion into the growth direction of ice.
- 2) **Anisotropic calculation of freezing direction.** In the freezing process, the growth direction is calculated based on the anisotropic kernel to accurately and stably deliver the fluid motion to the freezing solver.
- 3) **Improved quality of freezing surfaces by integrating advanced surface tension and viscosity.** When the phase of water particles is changed to ice, scattering motion occurs due to impact by collision, and this instability lowers the quality of ice shape generation. In this paper, improved surface tension and viscosity terms are integrated into the IISPH-based freezing solver to alleviate these problems.

II. RELATED WORK

In this section, we explore various studies related to ice, freezing, and snow simulations similar to the proposed method.

A. ICE AND FREEZING SIMULATIONS

In the field of physics-based simulation, studies related to the formation and growth of ice have been continuously published. Kim *et al.* [3] realistically expressed frost growing on flat or curved surfaces by expressing the growth of ice through crystallization of small and thin water particles using discrete-continuous method (i.e., diffusion limited aggregation, phase field method, and stable fluid solver). Im *et al.* [1] and Nishino *et al.* [4] proposed a method to express detailed features such as opaque areas inside ice. Kim *et al.* [14] proposed a method to model cracks and scratches that appear on ice surfaces when ice is impacted. Kharitonsky and Gonczarowski [12] realistically expressed the growth of icicles, taking into account the physical properties of water droplets, based on thermodynamics.

Unlike existing approaches, Kim *et al.* [2] expressed the growth of icicles by solving the Stefan problem. This method realistically represents the shape of the icicles being merged together by simulating several icicles at the same time, but it required a lot of computation. Gagnon and Paquette [13] proposed a procedural method that allows users to easily control the ice growth simulation, making it possible to model not only the shape of the ice but also the glaze effects according to the user's intention. Compared to the previously mentioned approaches, their method produces icicles quickly, but only

expresses a single form of icicles, and it is difficult to express the merging of multiple icicles. Ishikawa *et al.* [5] simulated the freezing of water droplets as they fall under gravity based on how glaze effects and icicles are generated. Since the aforementioned methods modeled the ice formation under the precondition that the water droplets slowly flow downwards only by gravity, it is difficult to model the freezing shape generated under other conditions.

Several studies have attempted to represent the freezing of flowing water [6]–[8], [15], [16]. These methods represent an opaque freezing phenomenon based on the diffusion of air bubbles, but cannot express icicles or glaze effects. Iwasaki *et al.* [7] proposed a method to express the melting and freezing effects on a particle basis, and expressed ice spikes generated on the ice surface, but most of the results were focused on melting. Wicke *et al.* [8] simulated various materials based on implicit connectivity, and produced the result of ice formation in flowing water using a heat transfer method. Makkonen [15], [16] proposed an ice growth theory model based on observations, but it is difficult to attempt simulation or visualization because they do not propose an algorithm to discretize and implement the theory.

B. SNOW SIMULATIONS

Many approaches to simulate ice or snow using continuum mechanics have been studied and used to represent elastoplastic snow models [17]. Since Stomakhin *et al.* [18] first introduced the material point method (MPM) to the snow simulation technique in the field of computer graphics, the MPM method has been expanded in various ways to handle phase shifts [19]–[23]. Recently, Han *et al.* solved the frictional contact between snow and hair in the MPM framework [24].

In addition to MPM, there are other discretization techniques used to simulate snow. Wong and Fu [25] applied the discrete element method (DEM) to snow particles and spring mechanics to represent snow in an interactive simulation environment. Mukai *et al.* [9] simulated snow splitting down from the roof using the extended DEM method, and Dagenais *et al.* [26] simulated snow movement using position based dynamics (PBD) and level-set. Takahashi and Fujishiro [37] modeled the movement of snow using flow based on smoothed particle hydrodynamics (SPH). Abdelrazek *et al.* [27] simulated an avalanche using the Bingham viscosity model, and Goswami *et al.* expressed the snow movement in real time on a GPU [10].

III. PROPOSED FRAMEWORK

Previous studies explicitly calculate the force of water particles, so if the speed of water particles is high or impact is applied due to collision with solids, the process of changing water particles into ice becomes unstable. As a result, ice surfaces may not be generated properly or noise may occur severely [32]. In this paper, we use IISPH, an implicit method that can reliably calculate fluid motion, rather than an explicit solver that directly calculates force based on position.

The framework proposed in this paper consists of a fluid simulation step and a freezing simulation step. In the fluid simulation stage, fluid motion, fluid-solid interaction, surface tension and contact force are calculated according to the IISPH (Implicit incompressible SPH) [28] based solution. We calculate the interaction between the fluid and the solid by creating boundary particles [29] on the vertices, edges, and faces of the solid mesh. The contact force and surface tension were implemented by the method of He *et al.* [31], and a viscosity-based fluid was modeled to stably express the glaze of ice and the curved freezing shape. The key point of this paper is to reliably simulate the anisotropic formation and growth of ice.

Simulating the supercooled water phenomenon makes it easy to model not only the shape of ice growing downwards by gravity, but also the more complex freezing shape. Since the temperature of supercooled water is less than zero degrees, freezing proceeds by crystallization between molecules, not by temperature change. The fluid in this paper is also assumed to be supercooled water, and the crystallization step for the fluid to change to ice is newly proposed based on the nucleation energy proposed by Im *et al.* [32]. A list of symbols is available in Table 1.

TABLE 1. List of symbols used in the paper.

Name	Description
h	Kernel function's characteristic length
ρ_0	Rest density
F^p	Pressure force
F^{np}	Non-Pressure force
$\mathbf{v}_{i,l}^{np}$	Intermediate velocity
$\rho_{i,l}^{np}$	Intermediate density
α	Freezing fraction
Γ	Virtual water film
η	Ratio of current virtual water film
$\mathbf{g}\mathbf{v}$	Anisotropic growth direction of ice
W_A	Anisotropic kernel
C	Covariance matrix of water particles
f_r	Freezing factor
ce	Anisotropic nucleation energy
gz_{th}	Glaze threshold
ic_{th}	Ice threshold
F^s	Surface tension force
F^a	Air pressure force
p_{atm}	Air pressure coefficient
F^{vf}	Viscosity force of fluid
F^{vs}	Viscosity force of solid
ϕ	Level-set of ice particle

A. IMPLICIT INCOMPRESSIBLE SPH SIMULATION

The IISPH technique is one of the simulation methods to improve the incompressible problem in particle-based fluids [32]. IISPH can discretize the LHS (left hand side) of the continuity equation through the RHS (right hand side) of the basic SPH technique and the first-order accuracy of the

Euler method as follows: $\frac{D\rho}{Dt} = \rho \nabla \cdot \mathbf{v}$. As a result, the continuity equation can be rewritten as (see Equation 1).

$$\frac{\rho_i(t + \Delta t) - \rho_i(t)}{\Delta t} = \sum_{j \in S(i)} m_j \mathbf{v}_{ij}(t + \Delta t) \nabla W_{ij} \quad (1)$$

where $S(i)$ is the particles adjacent to the i -th particle, ρ_i and m_i are the density and mass of the i -th particle, and \mathbf{v}_{ij} is the difference in velocity between the i and j -th particle: $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. W is calculated as the third Spline interpolation as follows (see Equation 2).

$$W(q, h) = \frac{5}{14\pi h^2} \begin{cases} (2-q)^3 - 4(1-q)^3, & 0 \leq q < 1 \\ (2-q)^3, & 1 \leq q < 2, \\ 0, & q \geq 2 \end{cases} \quad (2)$$

where h is the length of the kernel function, q is defined as $\frac{|\mathbf{r}|}{h}$, and \mathbf{r} is: $\mathbf{r} = \mathbf{x}_{ij}$. In this paper, the radius of the simulation domain is calculated as kh to be proportional to h , and k is set to 2. W_{ij} is defined as $W|_{\mathbf{r}=\mathbf{x}_{ij}}$, ∇_i is the slope for \mathbf{x}_i (i.e. $\frac{\partial}{\partial \mathbf{x}_i}$), and is written as ∇W_{ij} in this paper. Also, \mathbf{x}_{ij} is the position difference between the i th and the j th particle: $\mathbf{x}_i - \mathbf{x}_j$. In Equation 1, $\mathbf{v}_{ij}(t + \Delta t)$ and $\rho_i(t + \Delta t)$ are not known, but the condition $\rho_i(t + \Delta t) = \rho_0$ is satisfied to maintain the incompressibility of the continuity equation, and then an equation for pressure is derived based on this equation.

Newton's second law is used to express explicitly the pressure and non-pressure due to the force acting at the position of the i th particle. In this process, the RHS for F_i is calculated using the first-order accuracy of Euler method, and $F_i = m_i \frac{D\mathbf{v}_i}{Dt}$ and F_i are calculated separately for the pressure and non-pressure parts (see Equation 3).

$$F^P(t) + F^{nP}(t) = m_i \frac{\mathbf{v}_i(t + \Delta t) - \mathbf{v}_i(t)}{\Delta t} \quad (3)$$

where F^P and F^{nP} represent the force for pressure and non-pressure, respectively. Using the above equation, the velocity is newly defined as follows (see Equation 4).

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{F_{i,l}^P(t) + F_{i,l}^{nP}}{m_i} \Delta t \quad (4)$$

Velocity is divided into two components according to the pressure contribution of force as follows (see Equation 5).

$$\mathbf{v}_{i,l+1} = \left(\mathbf{v}_{i,l} + \frac{F_{i,l}^{nP}}{m_i} \Delta t \right) + \frac{F_{i,l}^P}{m_i} \Delta t = \mathbf{v}_{i,l}^{nP} + \mathbf{v}_{i,l+1}^P \quad (5)$$

where $\mathbf{v}_{i,l}^{nP}$ and $\mathbf{v}_{i,l}^P$ are calculated using $\mathbf{v}_{i,l}^{nP} = \mathbf{v}_{i,l} + \frac{F_{i,l}^{nP}}{m_i} \Delta t$ and $\mathbf{v}_{i,l}^P = \frac{F_{i,l}^P}{m_i} \Delta t$, respectively, and the subscripts l and $l + 1$ represent discretized values at time t and $t + \Delta t$, respectively. IISPH is similar to Chorin's two-stage projection algorithm [38]: 1) Mid-velocity u^* is related to viscosity and velocity in the previous time-step, and 2) new velocity is related to pressure and mid-velocity in the new time-step. In this paper, $\mathbf{v}_{i,l}^{nP}$ is similar to mid-velocity u^* in Chorin's projection method, and $\mathbf{v}_{i,l+1}^P$ affects the pressure

gradient of Navier-Stokes equation. In Chorin's projection method, the pressure equation can be obtained by applying a divergence-free condition to the velocity at a new time-step. However, the pressure equation in IISPH can be obtained by applying the incompressibility to the density at the new time-step $t + \Delta t$. By dividing the density into the pressure part and the non-pressure part by Equation 1, we can calculate the mid-density $\rho_{i,l}^{np}$ from the mid-velocity $\mathbf{v}_{i,l}^{np}$ (see Equation 5).

$$\rho_{i,l}^{np} = \rho_{i,l} + \Delta t \sum_{j \in S(i)} m_j \mathbf{v}_{ij,l}^{np} \nabla W_{ij} \quad (6)$$

Applying Equation 5 to Equation 6 results in Equation 7.

$$\rho_{i,l+1} = \rho_{i,l} + \Delta t \sum_{j \in S(i)} m_j \mathbf{v}_{ij,l+1} \nabla W_{ij} \quad (7)$$

$$\rho_{i,l+1} - \rho_{i,l}^{np} = \Delta t \sum_{j \in S(i)} m_j \left(\mathbf{v}_{ij,l+1} - \mathbf{v}_{ij,l}^{np} \right) \nabla W_{ij} \quad (8)$$

where $\rho_{i,l}$ is $\rho_{i,l} = \sum_j m_j W_{ij}$, and $\rho_{i,l+1} = \rho_0$ must be satisfied within the limits of numerical accuracy to maintain incompressibility. Using Equation 5 and $\mathbf{v}_{ij,l+1}^P$, the above equation is made in a simpler form, and the pressure equation can be expressed implicitly (see Equation 8).

$$\begin{aligned} \rho_0 - \rho_{i,l}^{np} &= \Delta t \sum_{j \in S(i)} m_j \mathbf{v}_{ij,l+1}^P \nabla W_{ij} \\ &= \Delta t \sum_{j \in S(i)} m_j \left(\frac{F_{ij,l+1}^P}{m_i} \Delta t \right) \nabla W_{ij} \\ &= \Delta t^2 \sum_{j \in S(i)} m_j \left(F_{i,l+1}^P - F_{j,l+1}^P \right) \nabla W_{ij}. \end{aligned} \quad (9)$$

The pressure is then calculated using the above equation. Equation 9 can be rewritten by applying the $\rho_{i,l}^{np}$ of Equation 6 as shown below (see Equations 10 and 11).

$$\begin{aligned} \rho_0 - \left(\rho_{i,l} + \Delta t \sum_{j \in S(i)} m_j \mathbf{v}_{ij,l}^{np} \nabla W_{ij} \right) \\ = \Delta t^2 \sum_{j \in S(i)} m_j \left(\frac{F_{i,l+1}^P}{m_i} - \frac{F_{j,l+1}^P}{m_j} \right) \nabla W_{ij} \end{aligned} \quad (10)$$

$$\begin{aligned} \rho_0 - \rho_{i,l} - \Delta t \sum_{j \in S(i)} m_j \left(\mathbf{v}_{ij,l} + \left(\frac{F_{i,l}^{nP}}{m_i} - \frac{F_{j,l}^{nP}}{m_j} \right) \Delta t \right) \nabla W_{ij} \\ = \Delta t^2 \sum_{j \in S(i)} m_j \left(\frac{F_{i,l+1}^P}{m_i} - \frac{F_{j,l+1}^P}{m_j} \right) \nabla W_{ij} \end{aligned} \quad (11)$$

where $\mathbf{v}_{i,l}^{nP} = \mathbf{v}_{i,l} + \frac{F_{i,l}^{nP}}{m_i} \Delta t$ is used to simplify LHS, and the force on pressure with conserved momentum is calculated as follows (Equation 12).

$$F_{i,l+1}^P = -m_i \sum_{j \in S(i)} m_j \left(\frac{P_{i,l+1}}{\rho_{i,l}^2} + \frac{P_{j,l+1}}{\rho_{j,l}^2} \right) \nabla W_{ij} \quad (12)$$

Then, $F_{i,l}^{np}$ is calculated as follows (see Equation 13).

$$F_{i,l}^{np} = m_i \sum_{j \in S(i)} m_j \prod_{ij} \quad (13)$$

$$\prod_{ij} = -\frac{(\mu_i + \mu_j) \mathbf{x}_{ij} \nabla W_{ij}}{\bar{\rho}_{ij}^2 (\mathbf{x}_{ij}^2 + \epsilon \bar{h}_{ij}^2)} \mathbf{v}_{ij,l}, \quad (14)$$

where \bar{h}_{ij} is calculated as: $\bar{h}_{ij} = \frac{(h_i + h_j)}{2}$, this value is affected by the radius of the kernel, and $\bar{\rho}_{ij}$ is calculated as: $\bar{\rho}_{ij} = \frac{(\rho_i + \rho_j)}{2}$. In Equation 11, the values of the *LHS* variables are known, and the *RHS* can be further simplified through a simplified version of Equation 12 (see Equation 15).

$$\begin{aligned} \Delta t^2 \frac{F_{i,l+1}^p}{m_i} &= -\Delta t^2 \sum_{k \in S(i)} m_k \left(\frac{p_{i,l+1}}{\rho_{i,l}^2} + \frac{p_{k,l+1}}{\rho_{k,l}^2} \right) \nabla W_{ik} \\ &= \left(-\Delta t^2 \sum_k \frac{m_k}{\rho_{i,l}^2} \nabla W_{ik} \right) p_{i,l+1} \\ &\quad + \left(\sum_{k \in S(i)} -\Delta t^2 p_{k,l+1} \frac{m_k}{\rho_{k,l}^2} \nabla W_{ik} \right) \\ &= d_{ii} p_{i,l+1} + \sum_{k \in S(i)} d_{ik} p_{k,l+1}, \end{aligned} \quad (15)$$

where d_{ik} is $d_{ik} = -\Delta t^2 \frac{m_k}{\rho_{k,l}^2} \nabla W_{ik}$, and can be rewritten as $\Delta t^2 \frac{F_{j,l+1}^p}{m_j}$. Using Equation 15, the *RHS* of Equation 11 can be written as follows (see Equation 16).

$$\begin{aligned} RHS &= \sum_{j \in S(i)} m_j ((d_{ii} p_{i,l+1} + \sum_{k \in S(i)} d_{ik} p_{k,l+1}) - (d_{jj} p_{j,l+1} \\ &\quad + \sum_{j \in S(i)} d_{jk} p_{k,l+1})) \nabla W_{ij} \end{aligned} \quad (16)$$

In the above equation, the contribution to the *i*th particle is extracted from the underlined term in Equation 16, and can be written as follows (see Equation 17). The underlined term in Equation 16 expresses the contribution of water particle *i*, and expresses the influence of the *i*-th particle when the pressure is updated with an iterative numerical solution.

$$\sum_{k \in S(j)} d_{jk} p_{k,l+1} = \sum_{k \in S(j), k \neq i} d_{jk} p_{k,l+1} + d_{ji} p_{i,l+1} \quad (17)$$

Since the above equation must be solved with a numerical solution such as the Jacoby iteration method, the *RHS* of Equation 11 is rewritten as follows (see Equation 18 and 19).

$$\begin{aligned} RHS &= \sum_{j \in S(i)} (d_{ii} p_{i,l+1} + \sum_{k \in S(i)} d_{ik} p_{k,l+1} - (d_{jj} p_{j,l+1} \\ &\quad + d_{ji} p_{i,l+1} + \sum_{k \in S(j), k \neq i} d_{jk} p_{k,l+1})) \nabla W_{ij} \\ &= \sum_{j \in S(i)} p_{i,l+1} (d_{ii} - d_{ji}) m_j \nabla W_{ij} \\ &\quad + \sum_{j \in S(i)} m_j \left(\sum_{k \in S(i)} d_{ik} p_{k,l+1} - d_{jj} p_{j,l+1} \right) \end{aligned}$$

$$- \sum_{k \in S(j), k \neq i} d_{jk} p_{k,l+1} \nabla W_{ij} \quad (18)$$

$$\begin{aligned} \therefore RHS &= a_{ii} p_i + \sum_{k \in S(i)} \left(\sum_k d_{ik} p_{k,l+1} - d_{jj} p_{j,l+1} \right. \\ &\quad \left. - \sum_{k \in S(j), k \neq i} d_{jk} p_{k,l+1} \right) \nabla W_{ij} \end{aligned} \quad (19)$$

where a_{ii} , d_{ii} , and d_{ik} are defined as follows (see Equation 20).

$$\begin{aligned} a_{ii} &= \sum_{j \in S(i)} (d_{ii} - d_{ji}) m_j \nabla W_{ij} \\ d_{ii} &= -\Delta t^2 \sum_{k \in S(i)} \frac{m_k}{\rho_{i,l}^2} \nabla W_{ik} \\ d_{ik} &= \sum_{k \in S(i)} -\Delta t^2 p_{k,l+1} \frac{m_k}{\rho_{k,l}^2} \nabla W_{ik} \end{aligned} \quad (20)$$

To update the pressure with an iterative numerical solution, the *RHS* of Equation 11 is replaced by Equation 19 (see Equation 21).

$$p_i^r = \frac{\xi}{a_{ii}} \quad (21)$$

$$\begin{aligned} \xi &= \rho_0 - \rho_i^{np} - \sum_{j \in S(i)} m_j \left(\sum_{k \in S(i)} d_{ik} p_{k,l+1}^r - d_{jj} p_{j,l+1}^{r-1} \right. \\ &\quad \left. - \sum_{k \in S(j), k \neq i} d_{jk} p_{k,l+1}^{r-1} \right) \nabla W_{ij} \end{aligned} \quad (22)$$

where the superscripts $r - 1$ and r represent the previous and new iteration values, respectively, and ρ_i^{np} is defined as follows (see Equation 23).

$$\rho_i^{np} = \rho_{i,l} + \Delta t \sum_{j \in S(i)} m_j \left(\mathbf{v}_{ij,l} + \left(\frac{F_{i,l}^{np}}{m_i} - \frac{F_{j,l}^{np}}{m_j} \right) \Delta t \right) \nabla W_{ij} \quad (23)$$

The above equation must be solved repeatedly until the residual of $p_{i,l+1}^r$ becomes sufficiently small. In this paper, the SOR (Successive over relaxation) method was used, and the convergence rate of the iterative method was improved using the following method (see Equation 24).

$$p_i^r = (1 - w) p_i^{r-1} + w p_i^r \quad (24)$$

where w was determined through repeated experiments, and in this paper, we set it to 0.8. Pressure can be calculated with an iterative numerical solution based on Equation 21, and this value is used to calculate $F_{i,l+1}^p$ in Equation 15. Also, $F_{i,l}^{np}$ is calculated using Equation 13. These two values are used in Equation 3 to calculate the new velocity. The particle's position, \mathbf{x}_i , is updated using the first-order Euler method. Figure 3 shows the IISPH-based fluid simulation results, and we integrate this fluid solution with the freezing simulation to create a new ice growth result.

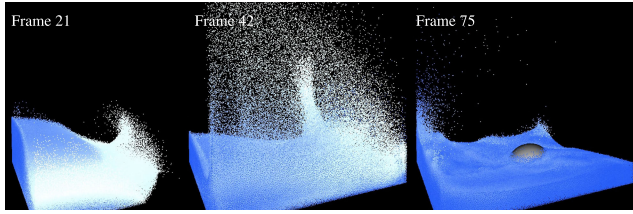


FIGURE 3. IISPH-based water simulations. Water particles are colored according to the magnitude of the velocity from white (high) to blue (low).



FIGURE 4. Characteristics of the ice surface during freezing.

B. ANISOTROPIC FREEZING SIMULATION

Our system is divided into a fluid simulation step and a freezing solver, and before explaining our framework in detail, the definition of basic elements used in the system is briefly explained in the table below (see Table 2).

TABLE 2. Notations.

Name	Description
Glaze (see Figure 4a)	Transparent ice formed by raindrops falling on the surface.
Curved ice (see Figure 4b)	Ice produced by intermolecular crystallization, such as supercooled water.
Icicle (see Figure 4c)	Ice formed when water flows and falls under the influence of gravity. Because of the kinetic energy of flowing water, it does not freeze immediately but freezes as it flows down, so it usually has a long, pointed cone shape.

1) VIRTUAL WATER FILM ON SOLID SURFACE

Ice grows in two properties, wet and dry, both of which are influenced by the freezing fraction, which affects the rate at which water turns into ice. The freezing fraction decreases as the humidity increases, and the water freezes into ice as the wet diffuse [16]. The freezing process is affected by various physical environments. In general, freezing occurs at a low temperature, but, like supercooled water, when the temperature is already below zero, freezing occurs by crystallization between molecules. In this case, curved freezing surfaces are expressed as they are greatly affected by wind or fluid movement. To change the fluid into ice, a nucleation process is required, and this nucleation energy is generated when the intermolecular forces, which are affected by temperature, humidity and other factors, become stronger. It is assumed that the fluid treated in this paper is supercooled water, and

the crystallization of the fluid into ice was implemented by modeling the intermolecular force based on the nucleation energy.

In the dry growth process, the fluid particles are converted into ice immediately after colliding with the object, and in the wet growth process, the fluid particles flow on the surface of the object, forming a water film as ice accumulates. The level-set technique [33] can be used to form the water film, but this method is generally expensive to calculate and is not suitable for particle-based simulation, making it difficult to represent the water film.

In this paper, a virtual water film is added to the solid particles to efficiently express the water film. When creating a virtual water film, Γ , the virtual mass of fluid particles adjacent to the solid particles is considered. As a result, the amount of mass transferred from the fluid particles is calculated taking into account the freezing fraction, $\alpha^{ice|solid}$ (see Equation 25).

$$\begin{aligned} \Gamma_t^{ice|solid} &= \Gamma_{t-\Delta t}^{ice|solid} + \Delta\Gamma^{ice|solid} \\ \Delta\Gamma^{ice|solid} &= \alpha^{ice|solid} \sum_j m_{fluid} \eta W(\mathbf{x}^{ice|solid} - \mathbf{x}^{fluid}) \\ \eta &= 1 - \frac{\Gamma_{t-\Delta t}^{ice|solid}}{\Gamma_{max}} \end{aligned} \tag{25}$$

where $\Delta\Gamma^{ice|solid}$ is the amount of virtual mass obtained from fluid particles. $\Gamma_t^{ice|solid}$ and $\Gamma_{t-\Delta t}^{ice|solid}$ are the amount of virtual water film in the current and previous time-steps, respectively, and $\alpha^{ice|solid}$ and m^{fluid} are the freezing fraction of solid particles and the mass of fluid particles. η is the ratio of ice and fluid particles to the virtual water film. In this paper, Γ_{max} is set as the mass of the fluid particles so that the mass of the water film on the fluid surface cannot exceed the mass of the fluid particles.

Figure 5 shows Γ calculated as water and solid particles collide. As shown in Figure 5a, Γ increases on the surface in contact with water particles, and this value diffuses around the solid particles as the contact surface widens (see Figure 5b).

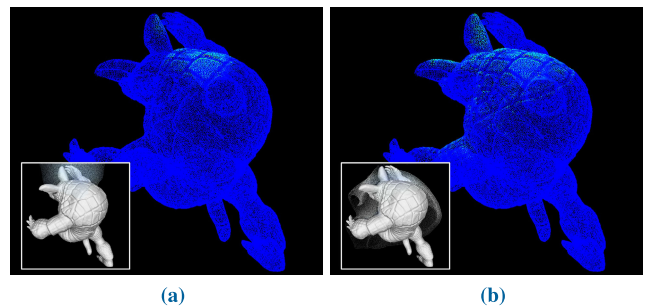


FIGURE 5. Virtual water film expressed by contact between water and solid particles. The solid particles were colored from cyan (high) to blue (low) depending on the magnitude of Γ .

2) ANISOTROPIC GROWTH DIRECTION OF ICE

Due to gravity, ice grows faster, mainly vertically, rather than horizontally. For example, icicle-like shapes are created by

slow flowing water droplets. As the water droplets flow down the surface of the ice under the influence of gravity, they form an elongated shape. Because water droplets are attached to the surface for a long time due to contact force and surface tension, icicles mainly lengthen in the vertical direction, and grow little by little in the horizontal direction due to the thin water film remaining on the surface. Depending on the environment, if an external force, such as a strong wind, acts on a water drop, it is more affected than gravity, so icicles can grow in different forms. In this paper, the growth direction of ice is modeled anisotropically by reflecting the flow of water during rapid water freezing. When water collides with an object, its velocity changes, and the growth direction of ice is calculated based on the velocity of the fluid particles at this time (see Equation 26).

$$\mathbf{g}^{\text{ice|solid}} = \frac{\sum_{j \in S(i)^{\text{fluid}}} \mathbf{v}_j^{\text{fluid}} W_A(\mathbf{x}^{\text{ice|solid}} - \mathbf{x}_j, \mathbf{G}_j)}{\|S(i)^{\text{fluid}}\|} \quad (26)$$

where $\mathbf{g}^{\text{ice|solid}}$ is the growth direction of the i th solid and ice particles, and \mathbf{v}_j and $S(i)$ are the velocity of the fluid particle and its adjacent particles, respectively. The linear transformation \mathbf{G} rotates and increases the radial vector \mathbf{r} , resulting in W_A being anisotropic weighted kernel (see Equation 27).

$$W_A(\mathbf{r}, \mathbf{G}) = \sigma \|\mathbf{G}\| \|\mathbf{G}\mathbf{r}\| \quad (27)$$

In order to calculate the anisotropic kernel from fluid particles, the covariance matrix is first calculated (see Equation 28), and principal component analysis (PCA) is performed to calculate the orientation and stretching of the fluid particles. In this paper, PCA was calculated through SVD (Singular value decomposition), and the eigenvectors and eigenvalues calculated using this are as follows (see Equation 29).

$$\mathbf{C}^{\text{fluid}} = \frac{\sum_j W_{ij} (\mathbf{x}_j - \mathbf{x}_i) (\mathbf{x}_j - \mathbf{x}_i)^T}{\sum_j W_{ij}} \quad (28)$$

$$\mathbf{G} = \mathbf{R}\Phi\mathbf{R}^T \quad (29)$$

$$\Phi = \text{diag}(\sigma^1, \sigma^2) \quad (30)$$

where \mathbf{R} is the rotation matrix calculated using principal axes, and Φ is the diagonal matrix containing eigenvalues. From this value, the anisotropic matrix \mathbf{G} is calculated. As a result, the growth direction of ice is calculated using W_A , an anisotropic kernel. Because ice forms on the surface of the object, each time a fluid particle collides with the object, the growth direction vector, $\mathbf{g}^{\text{ice|solid}}$, is calculated.

Unlike previous techniques that generate ice only in a specific direction, the proposed method uses $\mathbf{g}^{\text{ice|solid}}$ to grow ice anisotropically according to the flow of water. Figure 6 represents the growth direction of ice. As fluid particles collide with the object, freezing proceeds rapidly. In this figure, $\mathbf{g}^{\text{ice|solid}}$, which represents the growth direction, is represented by a blue arrow.

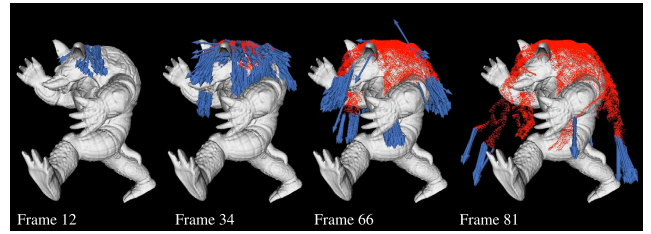


FIGURE 6. Growth direction vector calculated based on anisotropic kernel (blue arrow: $\mathbf{g}^{\text{ice|solid}}$, red particle: ice).

3) PHASE SHIFT

In this paper, we propose a method to calculate the freezing fraction that affects the phase shift of fluid particles into ice particles based on $\mathbf{g}^{\text{ice|solid}}$. If the fluid particles around the solid particles have the same freezing fraction, the ice will grow equally in the same direction and only glaze effects will be expressed. Since we want to control the freezing effect according to the flow of water, the freezing fraction is calculated as follows (see Equation 31).

$$fr = \hat{\mathbf{g}}^{\text{ice|solid}} \cdot \hat{\mathbf{x}}^{\text{isf}} \quad (31)$$

where fr is the freezing fraction of the fluid particle, and \mathbf{x}^{isf} is: $\mathbf{x}^{\text{isf}} = \mathbf{x}^{\text{ice|solid}} - \mathbf{x}^{\text{fluid}}$. It should be noted that each vector in the above equation is a normal vector ($\hat{\mathbf{x}}^{\text{isf}}$). Equation 31 shows that the freezing fraction of each fluid particle is determined by the angle between $\mathbf{g}^{\text{ice|solid}}$ and \mathbf{x}^{isf} . When this angle exceeds 90 degrees, fr becomes negative, and ice does not grow in the region where the fr value is negative. In this paper, fr is scaled as follows to adjust its range to $[0, 1]$ so that ice grows in regions with high humidity and positive fr values: $fr = \frac{fr+1}{2}$.

Nucleation is required to change a fluid into ice, and the nucleation energy required for this process occurs when the force between molecules affected by temperature and humidity increases. In this paper, we model the force between molecules through nucleation energy. Since we are assuming supercooled water, the simulation is performed under the assumption that the temperature of the object is already below zero degrees. When the object gets wet, a water film appears on the surface of the object, so the humidity is approximated by the virtual water film described in Section III-B1, and then nucleation energy is calculated. As a result, the anisotropic nucleation energy proposed in this paper is calculated as follows (see Equation 32).

$$ce(\mathbf{x}_i^{\text{ice|solid}}, \mathbf{x}_j^{\text{fluid}}) = \Gamma_i fr W_A(\mathbf{x}_i - \mathbf{x}_j, \mathbf{G}_j) \quad (32)$$

where $ce(\mathbf{x}_i^{\text{ice}}, \mathbf{x}_j^{\text{fluid}})$ and $ce(\mathbf{x}_i^{\text{solid}}, \mathbf{x}_j^{\text{fluid}})$ represent nucleation energy for fluid and ice particles, respectively, and Γ_i represents a virtual water film. Since the nucleation energy for ice growth actually appears differently depending on the surrounding environment, in this paper, the nucleation energy for the fluid and ice particles is independently calculated. In addition, glaze and icicle can be expressed individually by adjusting nucleation energy using glaze threshold and icicle threshold.

When the ce of the fluid particles in contact with the solid particles exceeds the glaze threshold gz_{th} , they are converted into ice particles to express the glaze generated on the object surface. Similarly, when the ce of water particles in contact with ice particles exceeds the icicle threshold ic_{th} , they are converted into ice particles. Figure 7 shows the result of changing fluid particles into ice particles. As shown in the figure, as fluid particles collide with solids, rapid freezing proceeds, and changes into ice particles according to the previously described algorithm.

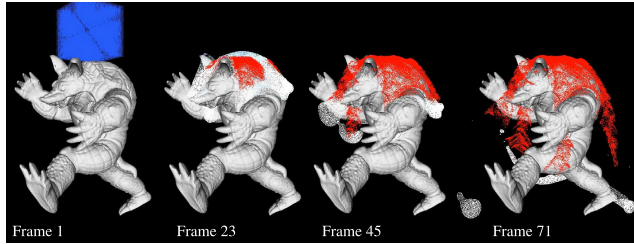


FIGURE 7. Phase shift from fluid particles to ice particles (red particle: ice, blue particle: water).

4) IMPROVING THE VISUAL QUALITY OF FREEZING SURFACE BY REDUCING INSTABILITY OF WATER PARTICLES

Fluid motion is important in modeling the shape of ice. The unnatural or unstable fluid motion degrades the quality of the freezing simulations applied based on it. Like Im *et al.*'s method [32], we express the effect of fluid particles clumping together during freezing simulation by adding air pressure and viscosity as well as surface tension to the fluid motion.

a: SURFACE TENSION AND AIR PRESSURE

Im *et al.* [32] used the surface tension method proposed by Akinci *et al.* [35], but there was a problem of numerically unstable convergence. In this paper, we improve the fluid solver by utilizing He *et al.*'s method [31] to better represent the shape of ice particles clumping together and to ensure that simulations converge reliably (see Equation 34).

$$c(\mathbf{x}) = \sum_j m_j \frac{1}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h) \quad (33)$$

$$\nabla_i c = \frac{\sum_j V_j c_j \nabla_i W(\mathbf{x}_{ij}, h)}{\sum_j V_j W(\mathbf{x}_{ij}, h)} \quad (34)$$

where V_j is the volume of the j th fluid particle, and \mathbf{x}_{ij} is the distance between the particles i and j . In the process of calculating the surface tension F_i^s , the average of the two surface tension energy densities is calculated and used in the surface tension equation to preserve the momentum (see Equation 35).

$$F_i^s = \frac{\kappa}{4} \sum_j V_j V_j \left(|\nabla c_i|^2 + |\nabla c_j|^2 \right) \nabla_i W(\mathbf{x}_{ij}, h) \quad (35)$$

where κ is the squared gradient energy coefficient [31], [36], which is set to 0.02 in this paper.

In this paper, we add air pressure force F_i^a to fluid particles to improve the effect of freezing the fluid as it flows along the surface [31] (see Equation 36).

$$F_i^a = V_i p_{atm} \sum_j V_j \nabla_i W(\mathbf{x}_{ij}, h) \quad (36)$$

where p_{atm} is the air pressure of fluid particles, which is set to 5000 in this paper. Our surface tension method includes both forces defined earlier: $F_i^{s+a} = F_i^s + F_i^a$.

Figure 8 is a comparison between the surface tension used in Im *et al.*'s method [32] and the surface tension we used. This paper did not propose a new surface tension technique, but improved the instability of the surface tension used by Im *et al.* [32]. In this paper, the surface tension is calculated in the iterative solver to calculate the pressure in IISPH, and this force is used in the integration process to update the position of the particle.

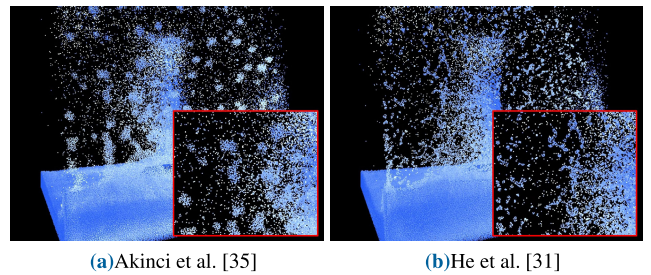


FIGURE 8. Comparison of the surface tension technique used in Im *et al.*'s freezing solver [32] (a) and our method (b) (inset image: zoom-in).

b: FLUID VISCOSITY

Im *et al.* [32] used only the surface tension of Akinci *et al.* [35] to express the curved ice that appears when a fluid flows and freezes. However, in this way, if the velocity of fluid particles colliding with the object is very slow, it sticks to the surface of the object and flows down, so you can model ice glaze or curved ice, but the faster the fluid particle speed, it becomes difficult to model the freezing shape.

As shown in Figure 9 showing the results of Im *et al.* [32], when the speed of the particle increases, it is expressed as a splash, but the contact force or surface tension of the fluid are not properly exerted. In this paper, $F_{i \in fluid}^{vf}$ and $F_{i \in solid}^{vs}$, which are forces related to viscosity, as well as contact and cohesion forces due to surface tension, are added to reduce simulation

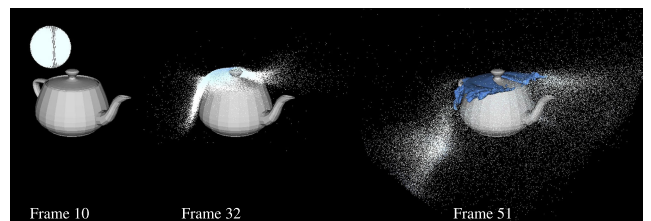


FIGURE 9. Results of freezing water simulation using the method of Im *et al.* [32] (particle: fluid, mesh: ice surface).

instability and improve the quality of freezing shape (see Equations 37 and 40).

$$F_{i \in fluid}^{vf} = \sum_j m_j \frac{\beta_0 \mathbf{u}_{ij} \beta_1}{\rho_{ij}^{avg}} \nabla_i W(\mathbf{x}_{ij}, h) \quad (37)$$

$$\mathbf{u}_{ij} = \frac{h(\mathbf{v}_{ij} \cdot \mathbf{x}_{ij})}{\|\mathbf{x}_{ij}\|^2 h^2} \quad (38)$$

$$\rho_{ij}^{avg} = \frac{\rho_i + \rho_j}{2} \quad (39)$$

where β_0 and β_1 mean bulk viscosity and sound wave respectively, and in this paper, we set them to 0.1 and 10. Since the above equation is a viscous effect only for fluid particles, it can express a sticky flow of water. Since this viscous effect helps to better express the surface tension or contact force of the fluid (see Figure 10), the freezing effect according to the fluid flow is also well expressed (see Figures 10c and 10d).

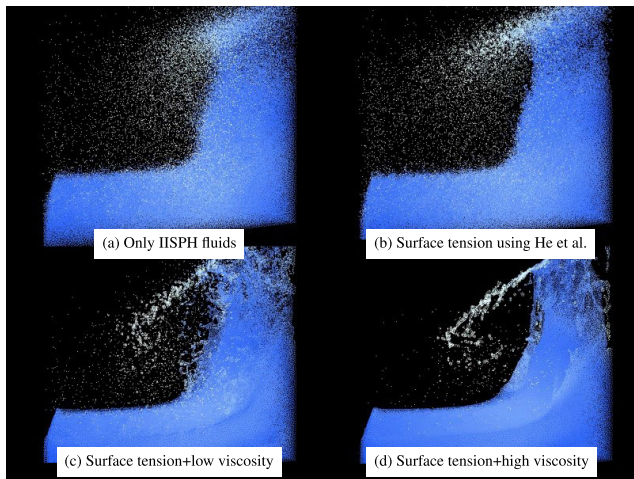


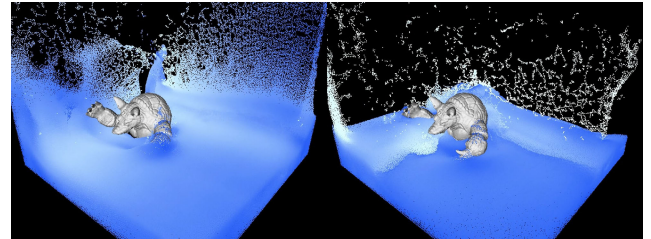
FIGURE 10. Comparison between different viscosity forces $F_{i \in solid}^{vs}$.

If we observe the actual phenomenon, when the water touches a cold solid, the heat is absorbed by the solid, and the water sticks to the solid surface and flows down. The fluid viscosity is well represented by $F_{i \in solid}^{vs}$, but it is not sufficient to model ice glaze effects or curved ice. This is because fluid particles should flow down by sticking to the surface and not scattering after colliding with the solid surface. As shown in Figures 10c and 10d, after the fluid collides with the wall, the effect of the fluid particles being sticky and clumping together by viscosity is well expressed, but the movement as if sticking to the wall and flowing down is not expressed. To solve this problem, in this paper, we calculate $F_{i \in solid}^{vs}$ to express additional viscous effects based on boundary particles (see Equation 40).

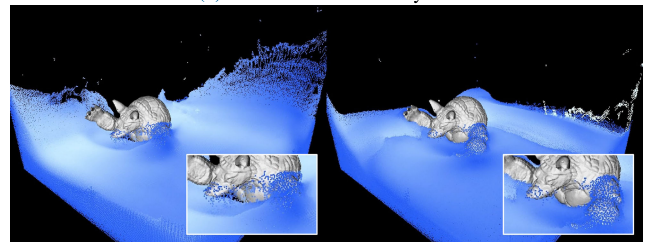
$$F_{i \in solid}^{vs} = \sum_j \Psi_j \frac{\mathbf{u}_{ij} \beta_1}{\rho_{ij}} \nabla_i W(\mathbf{x}_{ij}, h) \quad (40)$$

where Ψ_j is the volume of boundary particles proposed by Akinci *et al.* [29], and it is recommended to refer to their study for more detailed volume calculation.

Figure 11 is the comparison of viscous effects near the boundary by F^{vf} and F^{vs} . Compared to Figure 11a, which is the result of applying only the volume of the fluid, Figure 11b is more sticky and flowing down the wall or arm of the armadillo. When a fluid collides with a solid and freezes, a noisy surface is often created when fluid particles are scattered (see Frame 51 in Figure 9), but our method can stably express glaze effects of a solid surface or curved ice by applying viscosity near the boundary.



(a) Viscous fluids with only F^{vf}



(b) Viscous fluids with F^{vf} and F^{vs}

FIGURE 11. Comparison between different viscosity forces $F_{i \in solid}^{vs}$ and $F_{i \in solid}^{vs}$ (inset image: zoom-in).

5) SURFACE RECONSTRUCTION FROM ICE PARTICLES

When reconstructing the ice surface, the SVD value used to calculate the anisotropic direction of ice growth is used. In this paper, the ice surface was reconstructed in detail using the anisotropic property of SVD in the algorithm proposed by Yu *et al.* [34], and the level-set of ice particles is calculated as follows (see Equation 41).

$$\phi(\mathbf{x}) = \min_i \left(\left\| D_i^{ice} (\mathbf{x}_i - \mathbf{x}) \right\| \right) \quad (41)$$

where D_i^{ice} represents the transformation matrix for the ice particle i , whose properties are shown below (see Equation 42).

$$D_i^{ice} = \frac{1}{k_s} \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix}^T \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix}^{-1} \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \quad (42)$$

where k_s is the scaling constant, limiting σ_n to within a certain range to preserve the stretching with the greatest variation. For a more detailed explanation, it is recommended to refer to the work of Yu *et al.* [34]. In this paper, the GPU-based Marching Cubes algorithm [11] is used, and the minimum stretching value is set to be larger than half the width of the grid so that the ice surface is well represented.

Algorithm 1 Pseudo-Code of Our Algorithm

```

For each frame Do
  // Base water solver
  Advect 3D water particles using IISPH //
  Section III-A
  Compute the interaction of water & solids
  // Freezing water solver
  Viscosity force // Section III-B4
  Surface tension & air pressure forces // Section III-B4
  Virtual water film // Section III-B1
  Anisotropic direction for growthing // Section III-B2
  Phase transition from water to ice // Section III-B3
  Ice surface reconstruction // Section III-B5
End For
  
```

IV. IMPLEMENTATION

The experiments in this paper were implemented in the following environment: Intel i7-7700k 4.20GHz CPU, 32GB RAM, NVIDIA GeForce GTX 1080Ti graphics card. The IISPH-based fluid solution was implemented in the GPU environment and used as an underlying water simulation [28]. The Marching Cubes algorithm was used to reconstruct ice surfaces by adding a grid of $200 \times 200 \times 200$. The boundary particle method proposed by Akinci *et al.* [29] was used to handle the collision between fluids and solids. Pseudo-code of our foam simulation algorithm is shown in Algorithm 1.

Autodesk 3ds Max was used for high quality rendering for the final result. Since our results mainly deal with freezing surfaces, we did not render opaque regions or bubbles of ice. In this paper, rendering was performed only on ice surfaces, and rendering quality was improved using opacity, bump, reflection, refraction, and displacement mapping.

V. RESULTS

In order to analyze the technique proposed in this paper from various viewpoints, we experimented in various scenarios, and it is explained in a simulation view and a high quality rendering section.

A. SIMULATION VIEW

Figure 12 shows the result of rapid freezing when a spherical liquid is dropped onto the Stanford Bunny model. For the simulation of this scene, the time-step was set to 0.006 and approximately 50,000 fluid particles were used. As mentioned earlier, with Im *et al.*'s method [32], the spherical liquid is scattered when it collides with the object, so that the virtual water film cannot be properly formed on the solid surface, and as a result, ice cannot be expressed (see inset image in Figure 12a). Even in their method, there is a region where ice is expressed, but because the fluid is scattered, it is more like noise than freezing while flowing through the object surface. On the other hand, the proposed method well expressed the glaze effects of ice produced on solid surfaces. Not only that, but also curved ice that freezes along the solid surface was well represented (see Figure 12b).

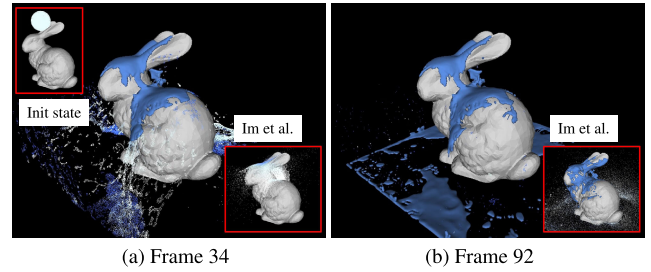


FIGURE 12. Freezing of water falling on the Stanford Bunny model using our method (inset image: Im *et al.* [32], particle: fluid, mesh: ice surface).

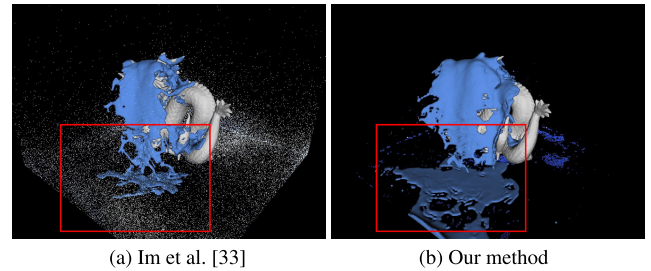


FIGURE 13. Comparison of the boundary regions between (a) Im *et al.* [32] and (b) our method (red rectangle: boundary region, particle: fluid, mesh: ice surface).

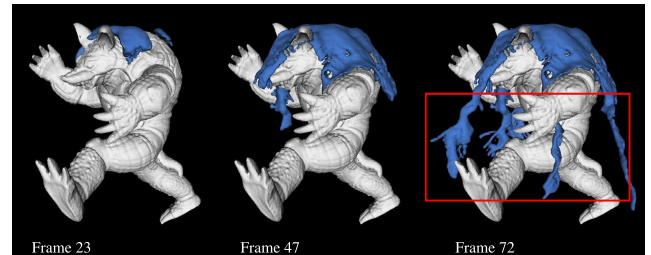


FIGURE 14. Freezing of water falling on the Armadillo model using our method (mesh: ice surface, red rectangle: anisotropic ice surfaces).

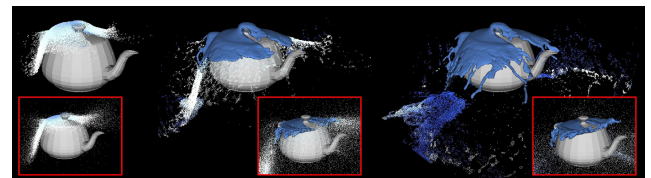


FIGURE 15. Freezing of water falling on the Utah Teapot model using our method (particle: fluid, mesh: ice surface, inset image: Im *et al.* [32]).

Figure 13 compares the quality and movement of ice generated near the boundary. Im *et al.*'s method [32] did not properly express the freezing effect due to the unstable movement of fluid particles near the boundary (see Figure 13a), but the proposed method stably expressed not only the freezing near the boundary but also the thin sheets of ice expressed during rapid freezing (see Figure 13b).

Figure 14 shows the result of rapid freezing when a spherical liquid is dropped onto the Armadillo model. As it freezes downwards by gravity, it is affected by the movement of the fluid and freezes in a zigzag shape. At the top of the



FIGURE 16. Ice surfaces and glaze effects that freeze anisotropically when water falls downward (red arrow: falling direction of water).



FIGURE 17. Ice surfaces and little glaze effects that freeze anisotropically when water is thrown in the left direction (red arrow: throwing direction of water).

Armadillo model, the glaze effects of ice are expressed, and in Frame 72, ice thin like an icicle is being created. In addition,

it freezes and spreads directionally according to gv , which is the anisotropic growth direction of ice (see red rectangle in Figure 14).

Figure 15 shows that when the liquid is dropped, it freezes rapidly, similar to the results shown earlier. Unlike the unstable freezing modeling by Im *et al.*'s method [32] (see inset image in Figure 15), the proposed method realistically represents the glaze effects of ice and the ice surface frozen by the flow of fluid.

As seen in the results shown earlier, the proposed technique showed more stable and realistic freezing quality than the state-of-the-art freezing technique, Im *et al.* [32].

B. HIGH QUALITY RENDERING

In this section, we look at how water particles are frozen by throwing them in various directions.

Figure 16 is a scene where water particles fall downward, and glaze effects and curved ice surfaces are well represented on the head and back of the Armadillo model. In this scene, $g_{z_{th}}$ and $i_{c_{th}}$ were set to 0.004 and 0.0001, respectively. Since water was dropped from the upper middle of the Armadillo model, the ice surfaces spread isotropically and freeze after the impact on the back.

Figure 17 is a scene in which water particles are thrown in a diagonal direction from the top left, and curved ice surfaces appear well on the head and back of the Armadillo model as in the previous result. Compared to the scene in

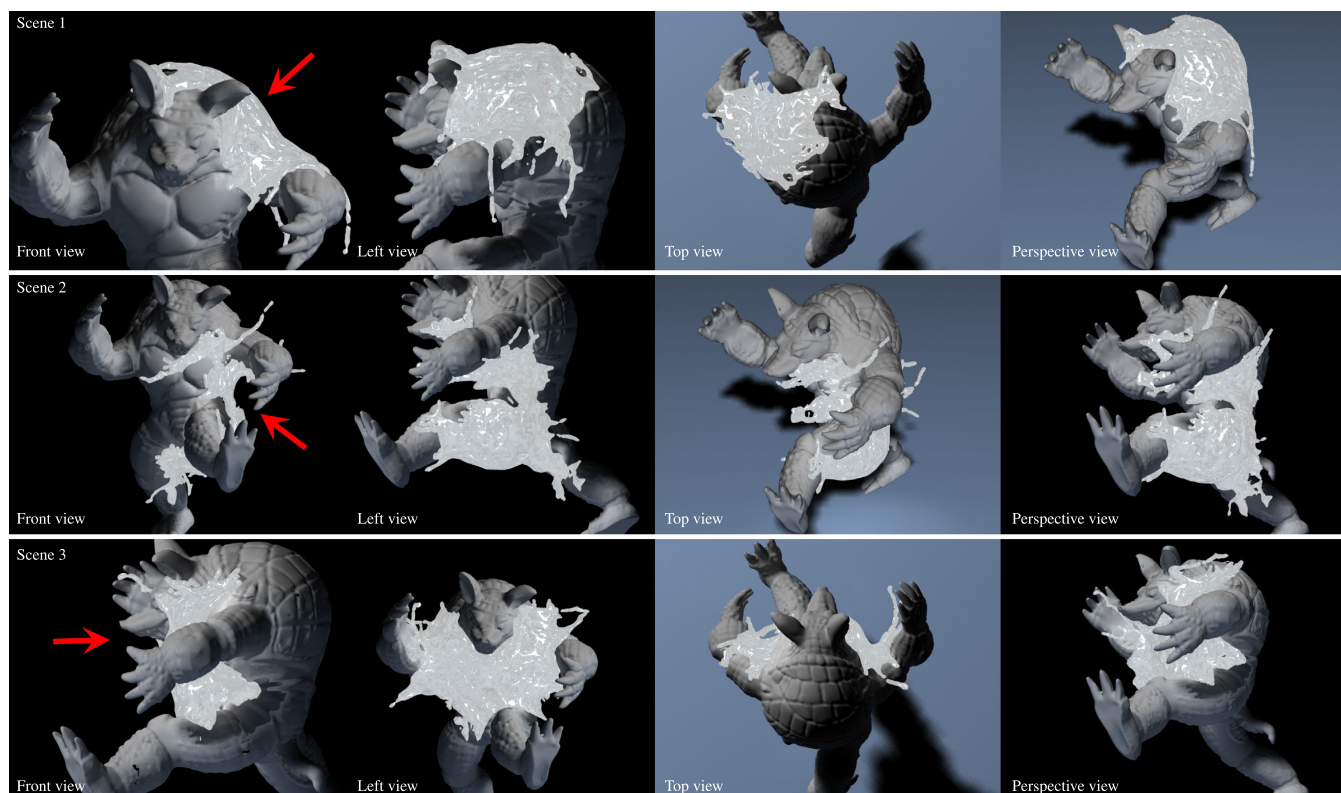


FIGURE 18. Various shapes of freezing ice (red arrow: throwing direction of water).

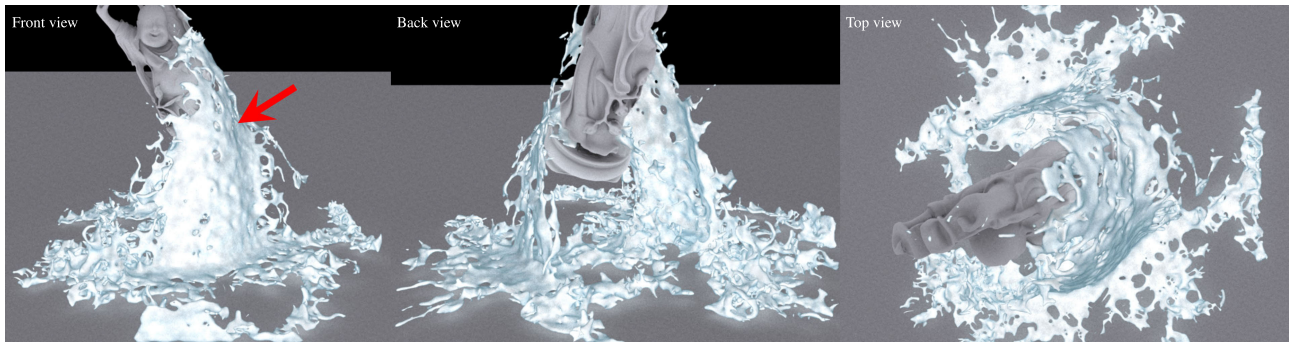


FIGURE 19. Freezing ice on the ground and on the rotating Happy Buddha model (red arrow: throwing direction of water).

TABLE 3. Simulation parameters to generate the experimental results shown in this paper.

Figure	α	gz_{th}	ic_{th}	p_{atm}	Water particles	Solid particles	Grid res. for ice surfaces
16	0.2	0.004	0.0001	5000.0	190,842	126,822	200×200×200
17	0.2	0.007	0.0001	5000.0	190,842	126,822	200×200×200
18	0.2	0.007	0.0001	5000.0	202,631	126,822	200×200×200
19	0.2	0.007	0.0001	5000.0	340,234	135,873	200×200×200

Figure 16, the gz_{th} was set to be larger so that glaze effects were less expressed, and an icicle with a relatively anisotropic shape appeared. In particular, the curved ice shape, which is well seen in the supercooled water phenomenon, was clearly shown. This shape is frozen in the red arrow direction, as shown in Figure 17, because it is affected by the throwing direction of the water.

Figure 18 shows the result of throwing water particles in various directions. We rendered from various viewpoints to observe whether the ice surfaces were well represented according to the throwing direction of the water particles. Simulation stability is essential to create such a complex scene, and Figure 18 shows the glaze effects and icicles of ice surfaces stably.

Figure 19 is a scene where water particles are dropped on the rotating Happy Buddha model, and the ice surfaces are well represented not only on the surface of the 3D model, but also on the ground. In particular, in this scene, we tried to express the fluctuating movement of water by rotating the Buddha model (see FigureFigure 19). Due to this, irregular patterns appeared on the freezing water surfaces, unlike the results shown earlier, and the ice surfaces connected from the surface of the Buddha model to the ground in the form of a bridge were well expressed. Table 3 shows the parameters and scene configuration used to produce the experimental results in this paper. The user can control the freezing surfaces using the corresponding parameters. The growth of the glaze is more constrained as the gz_{th} increases. gz_{th} is set lower than the ic_{th} to show the ice glaze more clearly. As the icicle threshold becomes larger, the icicle grows more slowly.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a framework for stable and realistic representation of glaze effects and curved ice surfaces expressed in supercooled water phenomena. The growth

direction of ice was calculated anisotropically according to the flow of the fluid, and the stability of freezing simulation was improved by adding surface tension, air pressure, and viscosity to the movement of the fluid to express the glaze effects and curved ice surfaces. Unlike the existing technique that expressed rapid freezing only in specific scenes, the proposed method showed stable and improved freezing results in various scenes.

In this paper, the freezing shape was reliably modeled by integrating particle-based water simulations and a freezing solver. Our method is physically inspired because it uses a physics-based fluid solver, but it cannot be called a complete physics-based simulation technique because pressure and heat transfer were not considered in the process of changing the phase from water to ice. Nevertheless, as shown in the results, it shows stable integration with particle-based fluids. As mentioned in the problem statement, in the previous technique, it is difficult to model the ice shape of a curved surface because only gravity is affected. We think that our technique can be improved by conserving physical momentum as in the Stefan problem, or by using the existing physics-based approach that can accurately solve the boundary problem between phases during phase transition.

In this paper, humidity is calculated when water particles and solid particles come into contact, and a phase shift is carried out due to crystallization-based energy". However, in the process of particle contact, heat transfer to each material was not considered. Since the algorithm was initially designed on the premise that the main target is supercooled water, the accuracy or results may vary because heat transfer is not considered. We will come up with a supplement to this problem in the future.

In the future, we plan to study how to efficiently express freezing simulation using the flow of surrounding air or wind, not fluid particles. The flow of fluid is an important factor

in expressing the detailed curved ice surface, but because of the large amount of calculation, we will study how to efficiently process it. In addition, we will study how to physically express glaze effects caused by water or ice sticking to cold objects based on heat transfer.

REFERENCES

- [1] J. Im, H. Park, J.-H. Kim, and C.-H. Kim, "A particle-grid method for opaque ice formation," *Comput. Graph. Forum*, vol. 32, no. 2, pp. 371–377, May 2013.
- [2] T. Kim, D. Adalsteinsson, and M. C. Lin, "Modeling ice dynamics as a thin-film Stefan problem," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2006, pp. 167–176.
- [3] T. Kim, M. Henson, and M. C. Lin, "A hybrid algorithm for modeling ice formation," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation (SCA)*, 2004, pp. 305–314.
- [4] T. Nishino, K. Iwasaki, Y. Dobashi, and T. Nishita, "Visual simulation of freezing ice with air bubbles," in *Proc. SIGGRAPH Asia Tech. Briefs (SA)*, 2012, pp. 1–4.
- [5] T. Ishikawa, Y. Dobashi, Y. Yue, M. Kakimoto, T. Watanabe, K. Kondo, K. Iwasaki, and T. Nishita, "Visual simulation of glazed frost," in *Proc. ACM SIGGRAPH Posters*, 2013, p. 1.
- [6] Y. Miao and S. Xiao, "Particle-based ice freezing simulation," in *Proc. 14th ACM SIGGRAPH Int. Conf. Virtual Reality Continuum Appl. Ind.*, Oct. 2015, pp. 17–22.
- [7] K. Iwasaki, H. Uchida, Y. Dobashi, and T. Nishita, "Fast particle-based visual simulation of ice melting," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2215–2223, Sep. 2010.
- [8] M. Wicke, P. Hatt, M. Pauly, M. Müller, and M. H. Gross, "Versatile virtual materials using implicit connectivity," in *Proc. Eurograph. Symp. Point-Based Graph.*, 2006, pp. 137–144.
- [9] N. Mukai, Y. Eto, and Y. Chang, "Representation method of snow splitting and sliding on a roof," in *Proc. 5th Int. Conf. Adv. Eng. Technol.*, 2017, pp. 100–103.
- [10] P. Goswami, C. Markowicz, and A. Hassan, "Real-time particle-based snow simulation on the GPU," in *Proc. Eurograph. Symp. Parallel Graph. Vis.*, 2019, pp. 1–9.
- [11] C. Dyken and G. Ziegler, "GPU-accelerated data expansion for the Marching Cubes algorithm," in *Proc. PGU Technol. Conf.*, 2010, pp. 115–123.
- [12] D. Kharitonsky and J. Gonczarowski, "A physically based model for icicle growth," *Vis. Comput.*, vol. 10, no. 2, pp. 88–100, Feb. 1993.
- [13] J. Gagnon and E. Paquette, "Procedural and interactive icicle modeling," *Vis. Comput.*, vol. 27, no. 6, p. 451, 2011.
- [14] J.-H. Kim, J. Im, C.-H. Kim, and J. Lee, "Subtle features of ice with cloudy effects and scratches from collision damage," *Comput. Animation Virtual Worlds*, vol. 27, nos. 3–4, pp. 271–279, May 2016.
- [15] L. Makkonen, "A model of icicle growth," *J. Glaciol.*, vol. 34, no. 116, pp. 64–70, 1988.
- [16] L. Makkonen, "Models for the growth of rime, glaze, icicles and wet snow on structures," *Philos. Trans. Roy. Soc. London A, Math. Phys. Sci.*, vol. 358, no. 1776, pp. 2913–2939, 2000.
- [17] G. Meschke, C. Liu, and H. A. Mang, "Large strain finite-element analysis of snow," *J. Eng. Mech.*, vol. 122, no. 7, pp. 591–602, Jul. 1996.
- [18] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, "A material point method for snow simulation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–10, Jul. 2013.
- [19] A. P. Tampubolon, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, and K. Museth, "Multi-species simulation of porous sand and water mixtures," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Jul. 2017.
- [20] J. Wretborn, R. Armiento, and K. Museth, "Animation of crack propagation by means of an extended multi-body solver for the material point method," *Comput. Graph.*, vol. 69, pp. 131–139, Dec. 2017.
- [21] M. Gao, X. Wang, K. Wu, A. Pradhana, E. Sifakis, C. Yuksel, and C. Jiang, "GPU optimization of material point methods," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, Jan. 2019.
- [22] Y. Fang, M. Li, M. Gao, and C. Jiang, "Silly rubber: An implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–13, Jul. 2019.
- [23] S. Wang, M. Ding, T. F. Gast, L. Zhu, S. Gagniere, C. Jiang, and J. M. Teran, "Simulation and visualization of ductile fracture with the material point method," *Proc. ACM Comput. Graph. Interact. Techn.*, vol. 2, no. 2, pp. 1–20, Jul. 2019.
- [24] X. Han, T. F. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran, "A hybrid material point method for frictional contact with diverse materials," *Proc. ACM Comput. Graph. Interact. Techn.*, vol. 2, no. 2, pp. 1–24, Jul. 2019.
- [25] S.-K. Wong and I.-T. Fu, "Hybrid-based snow simulation and snow rendering with shell textures," *Comput. Animation Virtual Worlds*, vol. 26, nos. 3–4, pp. 413–421, May 2015.
- [26] F. Dagenais, J. Gagnon, and E. Paquette, "An efficient layered simulation workflow for snow imprints," *Vis. Comput.*, vol. 32, nos. 6–8, pp. 881–890, Jun. 2016.
- [27] A. M. Abdelrazek, I. Kimura, and Y. Shimizu, "Numerical simulation of a small-scale snow avalanche tests using non-Newtonian SPH model," *Trans. Jpn. Soc. Civil Eng.*, vol. 70, no. 2, pp. 681–690, 2014.
- [28] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, "Implicit incompressible SPH," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 3, pp. 426–435, Mar. 2014.
- [29] N. Akinci, J. Cornelis, G. Akinci, and M. Teschner, "Coupling elastic solids with smoothed particle hydrodynamics fluids," *Comput. Animation Virtual Worlds*, vol. 24, nos. 3–4, pp. 195–203, May 2013.
- [30] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, "Versatile rigid-fluid coupling for incompressible SPH," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–8, Aug. 2012.
- [31] X. He, H. Wang, F. Zhang, H. Wang, G. Wang, and K. Zhou, "Robust simulation of sparsely sampled thin features in SPH-based free surface flows," *ACM Trans. Graph.*, vol. 34, no. 1, pp. 1–9, Dec. 2014.
- [32] J. Im, J.-H. Kim, W. Kim, N. Park, T. Kim, Y. B. Kim, J. Lee, and C.-H. Kim, "Visual simulation of rapidly freezing water based on crystallization," *Comput. Animation Virtual Worlds*, vol. 28, nos. 3–4, p. e1767, May 2017.
- [33] S.-Y. Lii and S.-K. Wong, "Ice melting simulation with water flow handling," *Vis. Comput.*, vol. 30, no. 5, pp. 531–538, May 2014.
- [34] J. Yu and G. Turk, "Reconstructing surfaces of particle-based fluids using anisotropic kernels," *ACM Trans. Graph.*, vol. 32, no. 1, pp. 1–12, Jan. 2013.
- [35] N. Akinci, G. Akinci, and M. Teschner, "Versatile surface tension and adhesion for SPH fluids," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–8, Nov. 2013.
- [36] J. W. Cahn and J. E. Hilliard, "Free energy of a nonuniform system. I. interfacial free energy," *J. Chem. Phys.*, vol. 28, no. 2, pp. 258–267, Feb. 1958.
- [37] T. Takahashi and I. Fujishiro, "Particle-based simulation of snow trampling taking sintering effect into account," in *Proc. ACM SIGGRAPH Posters*, 2012, p. 1.
- [38] A. J. Chorin, "A numerical method for solving incompressible viscous flow problems," *J. Comput. Phys.*, vol. 135, no. 2, pp. 118–125, Aug. 1997.



JONG-HYUN KIM received the B.A. degree from the Department of Digital Contents, Sejong University, in 2008, and the M.S. and Ph.D. degrees from the Department of Computer Science and Engineering, Korea University, in 2010 and 2016, respectively. He is currently an Associate Professor with the School of Software Application, Kangnam University. His research interests include physics-based simulation and natural phenomenon modeling and virtual production.



JUNG LEE is currently an Assistant Professor with the Department of Convergence Software, Hallym University. His current research interests include augmented/virtual reality, fluid animation, and computer graphics.

...