

Efficient Opaque Ice Sphere Formation Using a Lightweight Geometric Approach

Jong-Hyun Kim*

*Associate Professor, College of Software and Convergence (Dept. of Design Technology), Inha University, Incheon, Korea

[Abstract]

In this paper, we present a particle-grid blending framework based on a geometric approach to efficiently represent opaque ice spheres with air bubbles. The water temperature is diffused through the grid and the air bubbles represented inside the ice through the particles. To solve the problem of previous methods that generate noisy dissolved air fields, we use levelsets to lighten the algorithm, i.e., the number of active particles and the initial amount of dissolved oxygen can be used to efficiently control the termination conditions of heat diffusion. We also extend the previous dissolved air field method, which only computes near air bubbles, to transparent regions to represent realistic ice spheres, and introduce a levelset-based approach to accurately compute the orientation of particles. As a result, the method presented in this paper is about three times faster than the existing methods and shows visually improved visualization of opaque ice spheres, which can be used in the field of representing physical virtual ice forms.

▶ **Key words:** Opaque ice, Geometric approach, Freezing, Dissolved air field, Air bubble

[요 약]

본 논문에서는 공기방울이 있는 불투명한 얼음 구를 효율적으로 표현할 수 있는 기하학적 접근법 기반의 입자-격자 혼합 프레임워크를 소개한다. 격자를 통해 수온이 확산되고 입자를 통해 얼음 내부에 표현되는 공기방울을 표현한다. 노이즈한 용존공기장(Dissolved air field)이 생성되는 이전 기법의 문제를 해결하기 위해 본 논문에서는 레벨셋을 이용하여 알고리즘을 경량화한다. 즉, 활성화 입자(Active particle)의 개수와 초기 용존산소량(Dissolved oxygen)을 이용하여 열확산의 종료 조건을 효율적으로 제어한다. 또한 공기방울 근처에만 계산했던 이전 용존공기장 기법을 투명한 영역에도 확산하여 사실적인 얼음 구를 표현했고, 입자의 방향성을 정확하게 계산하기 위해 레벨셋 기반의 접근법을 소개한다. 결과적으로 본 논문에서 제시한 방법은 기존 기법에 비해 약 3배정도 빨랐으며 시각적으로 개선된 불투명 얼음 구의 가시화 결과를 보여주기 때문에 물리적인 가상 얼음 형태를 표현하는 분야에 활용될 수 있다.

▶ **주제어:** 불투명한 얼음, 기하학 접근법, 빙결, 용존공기장, 공기방울

-
- First Author: Jong-Hyun Kim, Corresponding Author: Jong-Hyun Kim
 - *Jong-Hyun Kim (jonghyunkim@inha.ac.kr), College of Software and Convergence (Dept. of Design Technology), Inha University
 - Received: 2024. 01. 22, Revised: 2024. 02. 26, Accepted: 2024. 02. 27.

I. Introduction

우리 주변에서 흔히 볼 수 있는 얼음의 형태는 투명한 얼음과 불투명한 얼음으로 구분할 수 있다 (Fig. 1 참조). 투명한 형태는 뜨거운 물로 빙결 했을 때이고, 불투명하고 뿌연 얼음은 찬 물로 빙결을 했을 때 나타나는 얼음의 특징이다[1]. 이러한 재질을 표현하기 위한 개별적인 렌더링 방식은 존재하지만, 계산량이 크기 때문에 실시간 접근으로 사용하기에는 한계가 있다.



Fig. 1. Real images of clear and cloudy ice.

본 논문에서는 불투명한 얼음을 시각화하기 위한 기하학적 기반의 입자-격자 혼합기법을 제안한다. 렌더링 과정을 최적화하기 위해 물리 기반 접근법을 이용하여 빙결 과정을 기하학적으로 경량화 시키고, 얼음 내부에 존재하는 입자와 격자를 모두 사용하여 빙결 시에 표현되는 기포와 물 온도에 따른 빙결 형태를 설계한다. 얼음에 존재하는 용존산소의 생성과 확산 과정을 얼음 형태의 레벨셋(Level-set)[3]을 이용하여 효율적으로 근사된 용존산소량을 계산한다. 또한 이전 연구와 유사하게 타원체 형태의 기포를 생성할 수 있도록 한다[2]. 결과적으로 이전 기법보다 시뮬레이션 과정이 경량화되기 때문에 적은 계산으로 불투명한 고품질 얼음 렌더링을 표현할 수 있고, 이전 기법에서 나타나는 단조로운 패턴 문제를 레벨셋 기반으로 해결하여 자연스럽게 얼음 내 기포를 생성할 수 있도록 한다. 본 논문은 다음과 같이 구성되어있다. 2장에서는 얼음을 표현하기 위한 모델링과 시뮬레이션 관련 기법에 대해서 설명하고, 3장에서는 '기하학적 접근법을 이용한 효율적인 열확산', '레벨셋을 이용한 용존공기장 표현', '얼음 내 공기방울 생성' 기법에 대해서 설명한다.

II. Related Work

물리 기반 시뮬레이션 분야에서 고체가 녹거나 물이 빙결되는 과정을 시뮬레이션하기 위한 기술들은 상태변화

모델을 기반으로 다양하게 발전되어 왔다[4-7]. 대부분 녹거나 굳을 때의 외부형태에만 초점이 맞춰져 있기 때문에 빙결 시 표현되는 뿌연 상태변화나 다양한 형태의 기포를 모델링하는 접근법은 활발히 연구되지 못했다. Kim *et al.*은[8] DLA(Diffusion limited aggregation)[9], 상태장 접근법[10], 그리고 유체 시뮬레이션[11]을 결합하여 표면 위에 표현되는 서리를 시뮬레이션 할 수 있는 혼합 기법을 제안했다. 그러나 이 기법은 얇은 서리 층 생성에만 초점을 맞추었기 때문에 불림형태의 불투명한 얼음으로 알고리즘을 확장하기에는 충분하지 않다. 다른 접근법으로 고드름의 생성과 빠르게 얼어붙는 물리 현상을 표현한 기법들이 있지만[12,13], 이 기법들은 고드름이나 종유석과 유사한 형태만을 표현하려고 했으며, 사실적인 얼음의 시각화 부분을 다루지 않았다. Hong *et al.*은 유체 내부에서 나타나는 기포의 움직임과 변화를 표현하기 위한 입자-격자 혼합 기법을 제안했다[14]. 이 기법은 격자를 사용하여 유체와 입자를 혼합하여 표현했으며, 레벨셋을 이용하여 기포의 병합과 분열을 모델링했다. 이 기법은 격자와 입자를 사용하여 유체와 기포를 표현했고, 레벨셋을 이용하여 기포의 병합과 분열을 모델링했다. 그러나 이 기법에서는 물 내부에서의 상태변화는 다루지 않았다.

Madrazo *et al.*은 격자만을 사용하여 얼음 내부에서 표현되는 기포를 시뮬레이션했다[15]. 하지만 이 방법은 격자 해상도에 렌더링 품질이 좌우되며, 특히 격자의 중앙에만 기포가 생성되기 때문에 렌더링 품질이 단조롭게 보일 수 있다는 한계가 있다. 또한 이 방법은 천천히 얼어붙은 얼음만을 표현했으며, 얼음 내부에서 나타나는 기포를 방향성이 없는 입자로만 모델링했다. 이와는 반대로 Im *et al.*은 격자와 입자를 혼합하여 저해상도 격자에서도 고품질 얼음 렌더링을 표현할 수 있는 방법을 제안했고[2], 방향성 있는 기포를 비등방성하게 생성하기 때문에 얼음의 사실적인 특징을 잘 표현했다.

Stefan Seipel과 Anders Nivfors은 불투명한 얼음을 표현하기 위해 텍스처를 사용했으며[16], 반사와 굴절을 포함한 실시간 얼음 렌더링에 중점을 두었다. 텍스처를 사용하기 때문에 온도나 주변 환경에 따라 생성되는 시뮬레이션 기반 렌더링이 아니며, 결과적으로 정적인 장면에서만 활용된다. 또한 불룩한 형태를 가진 모델에서만 동작되는 한계가 있다.

Im *et al.*은 얼음의 디테일한 움직임을 표현하기 위해 결정화(Crystallization)기반의 빙결 시뮬레이션 기법을 제안했다[17]. 이들은 물의 흐름이나 방향성을 가진 고드름을 표현하기 위해 입자 기반 유체 시뮬레이션에[18] 통합

된 빙결 프레임워크를 제안했다. 이 방법에서는 물 입자와 고체의 충돌은 일반적인 유체-고체 해법을 사용하였고, 물의 거시적인 움직임만을 고려했기 때문에 표면을 따라 흐르는 얼음이나 동적인 얼음의 형태를 모델링하기에는 안정성이 떨어진다는 문제가 있다. 최근에 Kim *et al.*은 IISPH(Implicit incompressible SPH) 기반의 빙결 시뮬레이션 기법을 제안하여 이 문제를 효율적으로 해결했다 [19]. 이 기법에서는 유체의 점성과 표면 장력, 그리고 얼음을 비등방성하게 결정화하고 확산시키면서 얼음의 형태를 만들기 때문에 Im *et al.*[2]보다 안정적으로 수행된다.

III. The Proposed Scheme

1. Rendering Processing for Opaque Ice Formation

불투명한 얼음을 모델링한 기법 중 가장 좋은 방법은 Im *et al.*이 제안한 입자-격자 혼합방식이다[2]. 하지만, 이 방법은 격자와 입자 공간에서 빙결을 위한 에너지를 확산하고 계산하기 때문에 계산량이 크며, 종료조건이 없어서 수동으로 조건을 만족시켜야 원하는 결과를 만들어 낼 수 있다. Im *et al.*의 방식은 아래와 같은 방식으로 수행된다 (Fig. 2 참조).

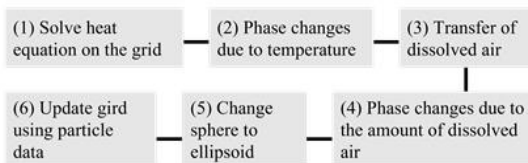


Fig. 2. Algorithm overview of previous method[2].

Fig. 2에서 (1)과 (6)은 격자에서 실행되며, 나머지 과정은 입자에서 계산된다. 이 방법에서는 물을 통한 열전달을 모델링하기 위해 격자를 이용하고 이를 작은 기포로 뿌리게 표현한다. 또한 입자를 사용하여 상태를 물, 얼어붙은 얼음, 기포로 분류한다. 얼음 내부에서 용해된 공기는 기포 입자 또는 뿌연 작은 기포들로 변환된다. (1)단계에서는 빙결 시 발생하는 열전달을 시뮬레이션하며, 격자 공간에서 계산한다. 그 다음 (2)~(5)단계에서는 입자들과 격자의 물리 속성을 업데이트한다. 물 입자에서는 보간을 통해 인접한 격자들로부터 온도를 업데이트한다. 입자에 포함된 열의 양은 입자가 얼었는지 여부를 나타내는 중요한 속성이다. 얼음은 물 만큼 많은 용존산소를 포함할 수 없기 때문에 (2)과정에서는 얼어붙은 입자에서 전달된 공기를

Table 1. List of terminology used in the paper.

Item	Meaning
Dissolved oxygen	The amount of oxygen dissolved in water.
Dissolved air field	The value of dissolved oxygen calculated from a regular grid structure.
Level-set	The field where the implicit function was calculated on an arbitrary 3D model. In this paper, we use this field to calculate the amount of dissolved oxygen and directional air bubbles.
Active particle	Particles categorized using a threshold of temperature. This value is used to design the simulation end condition.

인접 입자로 이동시킨다. 얼어붙은 입자는 근처에서 얼지 않은 입자를 찾아 용해된 공기를 방출한다. 빙결된 입자는 잔류 공기의 양에 따라 큰 기포 또는 작은 기포들로 분류된다. 다음 과정에서는 빙결 속도와 용해된 공기의 양에 따라 각 입자의 방향과 길이를 계산한다.

공기의 부피는 입자에 포함된 용존산소에 의해 결정된다. 마지막 과정에서는 작은 기포들로 분류된 입자들의 용존산소의 양을 이용하여 격자 기반에서 용존공기장을 형성한다. 결국 모든 물 입자가 얼음 입자가 되면 시뮬레이션은 종료된다.

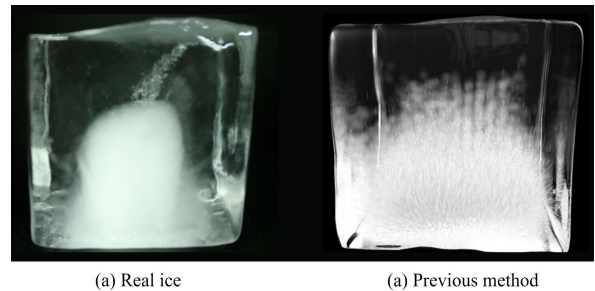


Fig. 3. Im *et al.*'s technique causes unnatural rendering artifacts when particles form a dissolved air field[2].

이 방법은 물리 기반 렌더링이지만 상대적으로 계산량이 크기 때문에 본 논문에서는 경량화된 접근법을 이용하여 유사한 결과를 빠르게 표현할 수 있는 기법을 소개한다. 뿐만 아니라 입자로부터 용존공기장을 표현할 때 나타나는 부자연스러운 문제를 완화시키기 때문에 시각적인 렌더링 품질을 개선시킬 수 있다 (Fig. 3 참조). Table 1은 본 논문에서 사용한 용어와 활용 과정을 설명한 테이블이다.

2. Heat Transfer with Geometry Approach

물이 빙결되는 과정에서 열전달을 하기 위해 본 논문에서는 Im *et al.*이 사용한 방법과 유사하게 유한차분법과

디리클레 경계 조건(Dirichlet boundary condition)을 이용하여 열 방정식을 계산한다 (수식 1 참조)[2].

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0 \quad (1)$$

여기서 α 는 열전달 가중치이며, u 는 온도, t 는 시간간격이다. 열전달을 위한 가중치 α 는 상수로 처리했으며, Im *et al.*에서 사용한 입자 가중치 기법을 사용하지 않는다[2]. 본 논문에서는 α 를 1로 설정하여 실험하였다.

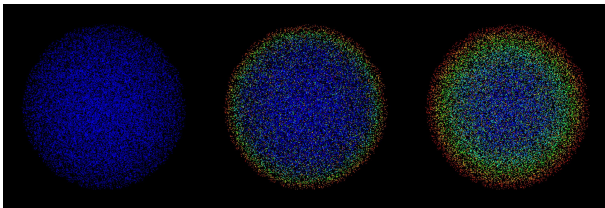


Fig. 4. Heat diffusion on the particles. The particles were colored from red(high) to blue(low) depending on the temperature.

Im *et al.*의 방법은 Fig. 3에서 보듯이 부자연스러운 결과를 초래하기 때문에 본 논문에서는 상수로 처리하여 열전달을 계산한다[2]. 입자의 열은 격자로부터 보간하여 계산한다. Fig. 4는 격자로 보간된 입자의 온도를 표현한 결과이며, 그림에서 보듯이 외부에서부터 열이 자연스럽게 확산되는 결과를 안정적으로 보여주고 있다.

이전 기법에서는 모든 입자에 대한 열전달이 수행되어야 열확산이 중단되지만[2], 이 방법은 효율적이지 않다. 얼음 내 용존산소량과 공기방울은 외부에서 열이 확산되기 때문에 일반적으로 얼음 표면 내부에 그 특징이 나타난다. 본 논문에서는 이를 효율적으로 풀어내기 위해 모든 입자에 대한 열전달 조건이 아닌, 활성화 입자(Active particle)인 P^* 의 개수와 초기 용존산소량 조건을 이용하여 열확산의 종료 조건을 효율적으로 제어한다. 활성화 입자의 개수는 입자의 온도를 이용했으며, 이를 계산하는 방법은 수식 2와 같다.

$$n(P^*) \leftarrow \{p | p.u > \Gamma\} \quad (2)$$

여기서 p 는 입자이고, $p.u$ 는 입자의 온도이다. 또한 Γ 는 활성화 입자로 분류하기 위한 온도의 임계값이며, 기본값으로 0.5로 설정했고 사용자가 이 값을 조절하여 원하는 장면을 제어할 수 있도록 하였다. 최종적인 열전달의 종료 조건은 다음과 같다 (수식 3 참조).

$$n(P^*) \geq \|P\| \gamma \quad (3)$$

여기서 P 는 전체 입자들의 집합이다. 또한 γ 는 초기 용존산소량을 나타내며, 본 논문에서는 0.5로 설정했다.

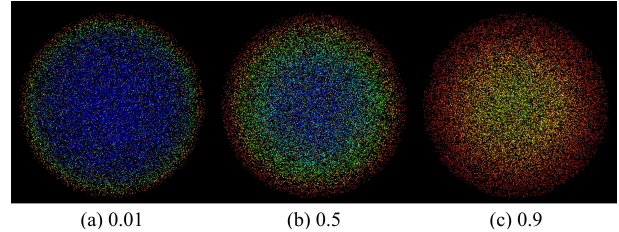


Fig. 5. Controlling heat transfer in particles with γ .

Fig. 5는 γ 를 통해 입자의 온도를 표현한 결과이다. 이 값에 따라 얼음 표면 내부에 전달되는 열이 다르게 때문에 이를 통해 다양한 얼음 렌더링을 쉽게 표현할 수 있다는 장점이 있다. 기존 방법에서는 전체 입자에 대한 열확산이 시뮬레이션의 종료조건이지만, 본 논문에서 제안하는 방법은 얼음을 표현하는 전체 입자의 개수와 활성화된 입자, 그리고 초기 용존산소량에 따라 열확산을 제어하고, 결과적으로 얼음 내 용존산소량과 공기방울을 표현할 수 있기 때문에 직관적으로 빙결 장면 제작이 가능하다.

3. Phase Change to Ice State

빙결 시 입자가 얼음으로 변하는 상태변화를 표현하기 위해 본 논문에서는 온도를 기반으로 얼음의 상태를 변화시킨다. 상태변화의 기준이 되는 온도 임계값인 κ 는 0.1로 설정했으며, 이 값의 변화에 따라 표면 내 얼음 입자의 양을 결정할 수 있다.

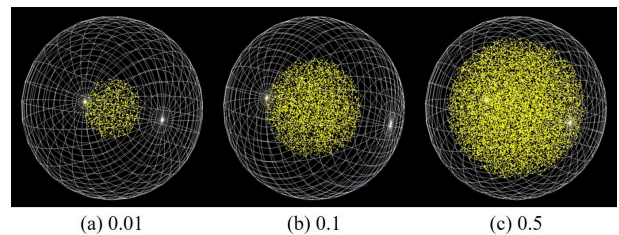


Fig. 6. Controlling phase change to ice particle with κ .

Fig. 6은 κ 를 통해 빙결 시 상태변화 된 얼음 입자의 양을 제어한 결과이다. 이 값에 따라 얼음 표면 내부에 표현되는 얼음 입자의 양이 달라지며, 결과적으로 다양한 얼음 재질을 표현할 수 있다는 장점이 있다.

4. Generation of Dissolved Air Field

물리적으로 입자가 얼어붙으면 용해된 공기를 인접 입자로 밀어내는 특성을 가진다. 이전 연구에서는 용존산소

량을 주변 입자들도 전달하는 방식을 이용하여 물리적인 특징을 알고리즘에 적용하려고 노력했다[2]. 하지만, 이 방식은 인접 입자의 분포에 따라 부드럽게 표현되어야 하는 얼음 내 용존산소량이 입자 형태 그대로 표현되는 문제가 발생한다 (Fig. 3 참조). 이 문제는 렌더링 품질에 악영향을 끼치기 때문에 좋은 방식이 아니며, 계산 비용 측면에서도 효율적이지 못하다. 본 논문에서는 이 문제를 해결하기 위해 레벨셋을 기반으로 용존산소량을 표현할 수 있는 방법을 제시한다 (수식 4 참조).

$$DO(x) = -\phi(x)\epsilon \quad (4)$$

여기서 ϕ 는 삼각형 메시로 구성된 얼음 표면의 레벨셋이며, 본 논문에서는 적응형 레벨셋 기법을 이용하여 이 과정을 계산했다[20]. 3차원 표면에서의 레벨셋은 샘플링 위치인 x 로부터 표면까지의 가장 가까운 거리이기 때문에 얼음 표면 내부에서는 음수를 가지게 된다. 용존산소량은 음수를 가질 수 없기 때문에 -1 를 곱하여 이 문제를 피한다. 또한, ϵ 는 용존산소량의 강도를 제어하는 가중치이며, 본 논문에서는 3으로 설정했다. 이 값은 사용자가 원하는 얼음 재질에 따라 수정이 가능하며, 이 값에 따라 용존산소량의 강도를 쉽게 제어할 수 있다 (Fig. 7 참조).

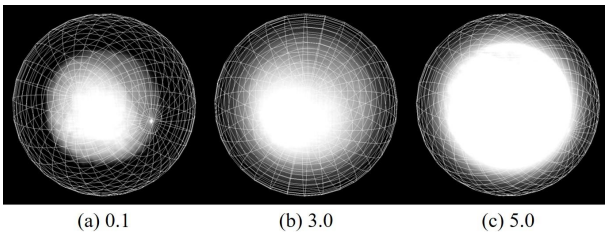


Fig. 7. Controlling dissolved air field in grid with ϵ .

5. Generation of Directional Air Bubbles

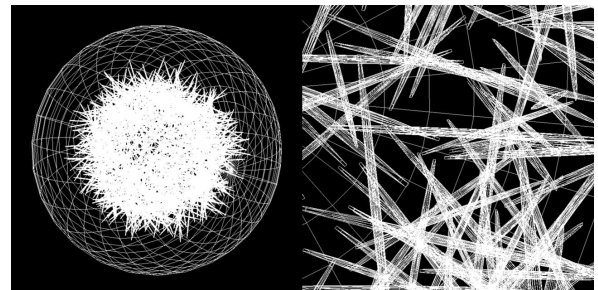
실제 현상에서는 물이 빙결하는 과정에서 용해된 공기가 인접 물 입자로 이동하게 된다. 빙결 과정의 속도가 이동하는 공기의 속도보다 빠르기 때문에 공기는 얼음이 성장하는 방향에 맞춰 공기방울이 생성된다[2]. 부피는 용해된 공기의 양에 따라 달라지며, 모양은 빙결 과정의 방향과 속도에 영향을 받는다. 빙결 속도가 빠를수록 공기방울은 더 얇고 날카로운 모양을 형성한다. 본 논문에서는 이전 기법과 유사하게 타원체 입자를 사용하여 이 과정을 설계한다[2].

얼음 부피 내부에서 공기방울이 생성되는 정확한 과정은 아직도 연구 중이기 때문에, 본 논문에서는 공기방울의 모양과 부피가 빙결 속도와 용존 공기의 양에 따라 달라진다고 가정한다[21]. Im *et al.*은 얼음 부피를 계산하여 입

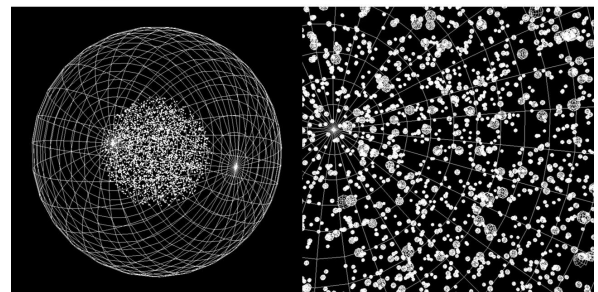
자의 용존산소량에 곱해주었지만[2], 본 논문에서는 입자가 아닌, 격자에서 직접적으로 레벨셋 기반의 용존산소량을 계산했기 때문에 이 과정이 생략된다. 대부분 용존산소량 근처에서 방향성을 가진 공기방울이 나타나는 특성이 있기 때문에 얼음 입자의 상태를 레벨셋을 기반으로 구분하여 처리한다 (수식 5 참조).

$$p^{ice}(x) = \begin{cases} |\nabla\phi(x)|\mu, & \text{if } u(x) \geq \nu \\ p^{iso}(x), & \text{else} \end{cases} \quad (5)$$

여기서 ϕ 는 레벨셋이며, $u(x)$ 는 x 에서 샘플링하여 얻어진 온도이다. μ 는 방향성을 가진 공기방울의 길이를 제어하는 가중치이고, 본 논문에서는 이 값을 0.1로 설정하였다. 이 계산 과정은 모든 공간에서 진행되는 것이 아닌, 용존산소량이 급격하게 변화하는 부분에 생성되기 때문에 임계값인 ν 보다 큰 영역에서 방향성을 가진 공기방울을 생성하고, 그렇지 않은 영역에서는 등방성하게 공기방울을 표현했으며, 본 논문에서 ν 는 0.04로 설정했다. 이 방식은 물리적이진 않지만, 에너지 확산이나 주변 입자들의 정보를 몰라도 되기 때문에 빠르게 얼음 입자의 상태를 구분할 수 있으며, 실제로 결과의 품질도 이전 결과와 유사하게 얼음 내 공기방울을 잘 표현해냈다.



(a) Directional air bubbles (right figure : zoom-in)



(b) Isotropic air bubbles (right figure : zoom-in)

Fig. 8. Representation of direction and isotropic air bubbles.

Fig. 8은 본 논문에서 제안한 방법을 이용하여 방향성을 가진 공기방울과 등방성한 공기방울을 표현한 결과이다. Fig. 8a는 사용자가 지정한 영역에서 방향성을 가진 공기방울이 $\nabla\phi$ 방향으로 표현된 결과를 보여주고 있으며, 얇

은 볼륨 형태를 유지하면서 얼음 내 분포된 기포를 잘 보여주고 있다. Fig. 8b는 등방성한 공기방울을 구 형태로 표현된 결과를 보여주며, 결과적으로 앞에서 계산한 얼음 입자는 둘 중에 하나의 공기방울로 변하게 되며, 서로 중복된 상태를 갖지 않는다.

IV. Experiment and Results

본 연구의 결과들을 만들기 위해 실험한 환경은 Intel Core i7-7700K CPU, 32GB RAM, Geforce GTX 1080Ti GPU가 탑재된 컴퓨터를 이용하였다. 렌더링 과정에서 공기방울은 타원체로 표현했고, 용존산소량은 OpenGL 기반의 볼륨 텍스처를 이용하여 처리하였다. 정확한 렌더링 시간은 렌더링해야 될 입자들의 개수에 의존하지만, 대부분의 얼음 구는 실시간으로 렌더링 되었다.

장면을 구성할 때는 얼음 입자들이 서로 겹치지 않도록 배치했고, 얼음 형태 외부에 위치한 입자들은 제거했다. 입자 사이의 간격이 넓어지거나 과도한 군집화를 방지하기 위해 입자들이 서로 일정한 간격을 유지하도록 했다. 일반적으로 움직이는 많은 영역 내에서 열린 물은 급격한 움직임을 보이지 않기 때문에 시뮬레이션 중에 입자는 움직이지 않는다. 물 입자가 움직이지 않는다는 가정을 바탕으로 본 논문에서는 SPH(Smoothed particle hydrodynamics)대신 격자를 사용하여 열 방정식을 계산했다. SPH는 물리량을 직접 가지고 이루어지기 때문에 활발하게 움직이는 액체와 점성 시뮬레이션을 계산할 때 보다 정확한 해를 계산할 수 있다는 장점이 있다. 그러나 고품질 시뮬레이션 결과를 얻기 위해서는 입자의 개수가 상대적으로 많이 필요하며, 적은 개수에서는 때문에 우수한 렌더링 품질을 만들어 내지 못한다. 그렇기 때문에 논문에서는 이 과정을 격자 기반 볼륨 렌더링을 통해 표현했다.

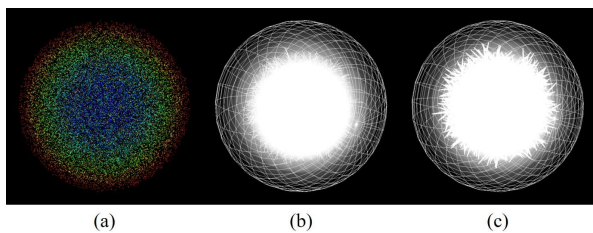


Fig. 10. Representing spherical ice using our method (number of anisotropic/isotropic air bubbles : 2,028/2,655).

Fig. 10은 본 논문에서 제안한 방법을 이용하여 구 형태의 얼음을 표현할 때 나타는 뿌연 공기방울과 방향성을 가

진 공기방울을 모두 표현한 결과이다. Fig. 10a는 얼음을 구성하는 입자들에서 열확산을 진행한 결과이며, Fig. 10b는 방향성을 가진 공기방울을 표현한 결과이고, Fig. 10c는 공기방울의 볼륨을 좀 더 크게 설정하여 두꺼운 공기방울이 표현되도록 만든 결과이다.

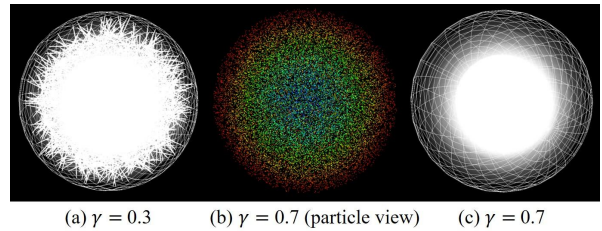


Fig. 11. Controlling dissolved air field and air bubbles with γ (number of anisotropic/isotropic air bubbles in (a) : 2,032/10,185, number of anisotropic/isotropic air bubbles in (b) : 505/0).

Fig. 11은 초기 용존산소량인 γ 를 변경하여 실험한 얼음 구를 보여주는 결과이다. γ 가 0.3일 경우에는 상대적으로 크기가 큰 용존공기장과 방향성을 가진 공기방울의 양이 많이 생성된 결과를 보여준다. 상대적으로 γ 가 0.7일 경우에는 열확산이 많이 적용되어 용존공기장이 작아지고 그로 인해 공기방울의 개수가 적게 표현되는 얼음 구가 만들어 지기도 한다. 이러한 얼음 구 형태는 공기방울이 많이 없으며, 실제 위스키에서 사용되는 얼음 구의 형상과 유사하게 나타난다 (Fig. 12 참조). 본 논문에서 보여준 모든 얼음 구는 $50 \times 50 \times 50$ 격자에서 생성했으며, 초기 용존산소량인 γ 는 0.5, 활성화 입자를 구분하기 위한 입자의 온도 임계값인 T 는 0.5로 설정하여 사용했다. 그리고 용존산소량의 크기를 제어하는 ϵ 는 0.3으로 사용했다. 방향성 있는 입자의 길이를 제어하기 위한 값인 μ 는 0.1로 설정했고, 방향성과 등방성 입자의 상태를 구분하기 위한 ν 는 0.04로 설정했다.



Fig. 12. Whiskey ice.

본 논문에서 제안하는 방법은 얼음 구를 표현할 수 있는 이전 기법[2]과 비교했을 때 계산 측면에서 약 3배정도 빠른 결과를 보였다. 시각적인 품질 측면에서는 용존공기장

을 표현하는데 있어서 이전 기법에서는 입자의 분포에 의존하기 때문에 노이즈가 나타나기도 하지만 (Fig. 3 참조), 본 논문에서 제안하는 방법은 레벨셋을 통해 깔끔하게 노이즈 없이 용존산소량을 표현했다. 마지막으로, 모든 입자에 대한 열전달 조건이 종료되어야 시뮬레이션이 종료되는 이전 기법과 달리[2], 본 논문에서는 활성화 입자의 개수와 초기 용존산소량 조건을 이용하여 열확산의 종료 조건을 효율적으로 제어했다.

V. Conclusion

본 논문에서는 빙결 시 공기방울이 표현되는 얼음 구를 효율적으로 표현할 수 있는 기하학적 접근법 기반의 입자-격자 프레임워크를 제안했다. 기존의 방법은 계산량이 클 뿐만 아니라, 용존산소량을 표현하는 과정에서 입자의 이질감이 렌더링 결과에서 표현되기 때문에 렌더링 품질이 떨어진다. 또한, 시뮬레이션 종료 조건이 사용자에게 의해서 결정되기 때문에 원하는 장면을 제작하는데 있어서 반복적인 노력이 필요하다. 본 논문에서는 제안하는 방법은 이 문제들을 완화시켰다.

계산량이 크고 입자로 인해 노이즈한 용존공기장이 생성되는 이전 기법의 문제를 해결하기 위해 본 논문에서는 레벨셋을 이용하여 전체적인 알고리즘을 경량화했다. 즉, 입자에 대한 열 조건이 아닌, 활성화 입자의 개수와 초기 용존산소량을 이용하여 열확산의 종료 조건을 효율적으로 제어했다. 또한 공기방울 근처에만 계산했던 이전 용존공기장 기법을 투명한 영역에도 자연스럽게 확산하여 사실적인 얼음 구를 표현했고, 입자의 방향성을 정확하게 계산하기 위해 용존공기장이 아닌 레벨셋 기반으로 계산할 수 있는 방법을 제시했다. 결과적으로 본 논문에서 제시한 경량화된 기하학적 접근법은 기존 기법에 비해 약 3배정도 빨랐으며 시각적으로 한 층 개선된 불투명 얼음 구의 가시화 결과를 보여주었다.

그럼에도 불구하고 몇 가지 한계점이 있다. 첫 번째로 현재 기법은 불투명한 얼음만 가정했기 때문에 투명한 얼음으로부터 프로그레시브(Progressive)한 결과를 얻기 힘들다. 이 문제는 렌더링 과정에서 물의 온도가 고려되어 용존산소에 영향을 줄 수 있도록 새롭게 설계가 되어야 한다. 두 번째로는 용존산소의 형태가 레벨셋을 기반으로 고려되기 때문에 얼음 모델의 형상이 용존산소에도 영향을 준다. 결과적으로 구가 아닌 임의의 모델에서는 등방성이 아닌 얼음 모델 형상과 유사한 용존산소 형태가 표현될 수

있다. 이 한계점은 이전 문제에서도 나타나며[2], 향후 이 문제를 해결할 수 있도록 알고리즘을 확장할 계획이다.

REFERENCES

- [1] Pounder, Elton Roy. *The physics of ice*. Elsevier, 2013.
- [2] Im, Jaeho, Hanwook Park, Jong-Hyun Kim, and Chang-Hun Kim. "A Particle-Grid Method for Opaque Ice Formation." In *Computer Graphics Forum*, Vol. 32, No. 2pt3, pp. 371-377. Oxford, UK: Blackwell Publishing Ltd, 2013. DOI: 10.1111/cgf.12057
- [3] Museth, Ken, David E. Breen, Ross T. Whitaker, and Alan H. Barr. "Level set surface editing operators." In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 330-338. 2002. DOI: 10.1145/566654.566585
- [4] Carlson, Mark, Peter J. Mucha, R. Brooks Van Horn III, and Greg Turk. "Melting and flowing." In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 167-174. 2002. DOI: 10.1145/545261.545289
- [5] Fujisawa, Makoto, and Kenjiro T. Miura. "Animation of ice melting phenomenon based on thermodynamics with thermal radiation." In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pp. 249-256. 2007. DOI: 10.1145/1321261.1321307
- [6] Iwasaki, Kei, Hideyuki Uchida, Yoshinori Dobashi, and Tomoyuki Nishita. "Fast particle-based visual simulation of ice melting." In *Computer graphics forum*, Vol. 29, No. 7, pp. 2215-2223. Oxford, UK: Blackwell Publishing Ltd, 2010. DOI: 10.1111/j.1467-8659.2010.01810.x
- [7] Gan, Hui, James J. Feng, and Howard H. Hu. "Simulation of the sedimentation of melting solid particles." *International journal of multiphase flow*, Vol. 29, No. 5, pp. 751-769, 2003. DOI: 10.1016/S0301-9322(03)00035-1
- [8] Kim, Theodore, Michael Henson, and Ming C. Lin. "A hybrid algorithm for modeling ice formation." In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 305-314. 2004. DOI: 10.1145/1028523.1028564
- [9] Witten, Thomas A., and Leonard M. Sander. "Diffusion-limited aggregation." *Physical review B*, Vol. 27, No. 9, 1983.
- [10] Grinstein, Geoffrey. *Directions in condensed matter physics: memorial volume in honor of Shang-keng Ma*. Vol. 1. World Scientific, 1986.
- [11] Stam, Jos. "Stable fluids." In *Seminal Graphics Papers: Pushing the Boundaries*, Volume 2, pp. 779-786. 2023. DOI: 10.1145/311535.311548
- [12] Kharitonsky, Dany, and Jakob Gonczarowski. "A physically based model for icicle growth." *The Visual Computer*, Vol. 10, pp. 88-100, 1993. DOI: 10.1007/BF01901945

- [13] Kim, Theodore, David Adalsteinsson, and Ming C. Lin. "Modeling ice dynamics as a thin-film stefan problem." In Symposium on Computer Animation: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation: Vienna, Austria, Vol. 2, No. 04, pp. 167-176. 2006.
- [14] Hong, Jeong-Mo, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. "Bubbles alive." ACM Transactions on Graphics (TOG), Vol. 27, No. 3, pp. 1-4, 2008. DOI: 10.1145/1399504.1360647
- [15] Madrazo, Carlos, Takeshi Tsuchiya, Hiroaki Sawano, and Keiichi Koyanagi. "Air bubbles in ice by simulating freezing phenomenon." J. Soc. Art Sci, Vol. 8, No. 1, pp. 35-42, 2009.
- [16] Seipel, Stefan, and Anders Nivfors. "Real-time rendering of ice." In Proceedings of the ninth IASTED international conference on computer graphics and imaging, CGIM, Vol. 7, pp. 60-66. 2007.
- [17] Im, Jaeho, Jong-Hyun Kim, Wook Kim, Nuri Park, Taehyeong Kim, Young Bin Kim, Jung Lee, and Chang-Hun Kim. "Visual simulation of rapidly freezing water based on crystallization." Computer Animation and Virtual Worlds, Vol. 28, No. 3-4, pp. e1767, 2017. DOI: 10.1002/cav.1767
- [18] Macklin, Miles, and Matthias Müller. "Position based fluids." ACM Transactions on Graphics (TOG), Vol. 32, No. 4, pp. 1-12, 2013. DOI: 10.1145/2461912.2461984
- [19] Kim, Jong-Hyun, and Jung Lee. "Stable and anisotropic freezing framework with interaction between IISPH fluids and ice particles." IEEE Access, Vol. 9, pp. 146097-146109, 2021. DOI: 10.1109/ACCESS.2021.3123435
- [20] Frisken, Sarah F., Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. "Adaptively sampled distance fields: A general representation of shape for computer graphics." In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 249-254, 2000. DOI: 10.1145/344779.344899
- [21] Carte, A. E. "Air bubbles in ice." Proceedings of the Physical Society, Vol. 77, no. 3, pp. 757, 1961. DOI: 10.1088/0370-1328/77/3/327

Authors



Jong-Hyun Kim received the B.A. degree in the Department of Digital Contents at Sejong University in 2008. He received M.S. and Ph.D. degrees in the Department of Computer Science and Engineering at Korea University,

in 2010 and 2016. Prof. Kim is an Associate Professor in the College of Software and Convergence (Dept. of Design Technology) in Inha University. His current research interests include fluid animation and virtual reality.