

Chaotic Advection and Varying Buoyant Flows for Dispersed Fire-Flake Simulations

Jong-Hyun Kim*

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

[Abstract]

In this paper, we propose a chaotic advection method and various buoyant flow techniques to efficiently simulate distributed flame flows with flame behavior. Spark particles not only exhibit unique and complex behavior, but also are affected by the flame motion and the surrounding air. The motion of sparks tends to be chaotic and constantly changing, making it difficult for users to realistically model the motion of sparks. Instead of explicitly modeling the complex spark particles represented in a flame based on hydrodynamics, this paper efficiently approximates the complex motion of chaotic sparks through 1) a chaotic migration method and 2) a buoyant flow with a controlled size that varies with the temperature and age of the spark particles. For the flame migration process, a novel stochastic solver is used to compute the subgrid interactions between flame particles. The proposed method can efficiently simulate disordered spark particles in chaotic domains and provides a more detailed representation of spark particles than existing methods.

▶ **Key words:** Fire simulation, Flame, Fire-flake particle, Chaotic advection, Buoyant flow

[요 약]

본 논문에서는 화염의 움직임에 따라 분산된 불똥 흐름을 효율적으로 시뮬레이션하기 위한 혼돈 이류 방식과 다양한 부력 흐름 기법을 제안한다. 불똥 입자는 독특하고 복잡한 움직임을 보일 뿐만 아니라 화염의 움직임과 주변 공기에도 영향을 받는다. 불똥의 움직임은 무질서하고 끊임없이 변화하는 경향이 있기 때문에 사용자가 불똥의 움직임을 사실적으로 모델링하는 것은 어렵다. 화염에서 표현되는 복잡한 불똥 입자를 유체역학을 기반으로 한 명시적 모델링 대신 본 논문에서는 1) 혼돈 이류 방식과 2) 불똥 입자의 온도와 수명에 따라 다양한 크기로 제어된 부력 흐름을 통해 카오스적인 불똥의 복잡한 운동을 효율적으로 근사한다. 불똥의 이류과정에서는 새로운 확률적 솔버(Solver)를 사용하여 불똥 입자 간의 서브그리드(Subgrid) 상호작용을 계산한다. 제안하는 기법을 사용하면 카오스 영역에서 표현되는 무질서적인 불똥 입자를 효율적으로 시뮬레이션할 수 있으며 기존 기법보다 디테일한(Detail) 불똥 입자를 표현할 수 있다.

▶ **주제어:** 불 시뮬레이션, 화염, 불똥 입자, 혼돈 이류, 부력 흐름

-
- 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: 2023. 05. 08, Revised: 2023. 05. 31, Accepted: 2023. 06. 12.

I. Introduction

다양한 재질이 타거나 화염에 휩싸인 장면, 그리고 모닥 불까지 대부분의 화염 장면에서는 불뚱이 존재한다. 특히 영화나 광고, 애니메이션에서 화염 위로 솟아오르는 수많은 불뚱 입자들의 움직임은 무질서적인 형태를 지니고 있다. 화염과 불뚱은 모두 연소에 의해 생성되는 복잡한 화학 반응이며, 거시적 수준에서 불뚱 운동의 전체적인 경향은 부력, 공기저항, 소용돌이 및 확산 같은 유체역학을 기반으로 표현된다 (Fig. 1 참조).



Fig. 1. Real sparks and flakes of fire.

그러나 미시적 수준에서 불뚱 입자는 복잡하고 예측할 수 없는 방식으로 분산되거나 움직이며 이런 특징은 불뚱 입자뿐만 아니라 물속에서 표현되는 공기방울에서도 유사하게 표현된다. 공기방울은 상대적으로 유속에 의해 표현된다. 공기방울들이 상호작용으로 인해 회전하는 형태가 주로 표현된다면, 불뚱은 공기보다 가벼운 특징과 공기저항에 영향을 받기 때문에 무질서적인 움직임이 표현된다. 다수의 입자들이 유동에서 분포하는 패턴을 분산 입자 유동이라고 부른다. 이 불뚱의 무질서한 움직임과 복잡성은 분산된 불뚱 흐름을 시뮬레이션하기 더 어렵게 만든다.

오일러 표면 추적 프레임워크(Euler surface tracking framework)와 결합된 다상 유체 솔버로 입자를 시뮬레이션 할 수 있으며, 입자들은 컴퓨터그래픽스 분야에서 거품과 불뚱을 표현하는데 활용된다[1-4]. 그러나 현재까지 개발된 방법은 격자(Grid) 기반 솔버의 수치적 한계로 인해 상대적으로 거시적인 움직임을 활용하여 불뚱을 표현한다. 이 문제를 해결하기 위해 SPH(Smoothed particle hydrodynamics)기반의 입자-입자 상호작용 모델이 제시되었다[5-8]. 그러나 SPH 모델을 이용하여 공기저항과 카오스 영역에서 표현되는 불뚱 입자를 모델링하는 것은 입자간의 상호작용을 계산해야하기 때문에 효율적이지 않다 [5-8]. SPH는 입자의 개수가 계산 시간뿐만 아니라 시스템 안정성에도 영향을 주기 때문에 고품질 실시간 애플리케이션에서는 활용하기 어렵다는 한계가 있다.

계산 부담을 줄이기 위한 수단으로 영상특수효과 산업에서는 다중 레이어 프레임워크를 사용하기도 한다[9]. 이 방법에서는 스프레이(Spray), 미스트(Mist) 및 거품(Foam) 같은 효과를 표현하기 위해 여러 레이어를 추가하여 다양한 재질을 표현했으며, 게임이나 가상현실 같은 실시간 애플리케이션에서 활용되고 있다[10]. 하지만 대부분 레이어에 해당하는 텍스처를 이용하기 때문에 움직임이 제한적이며 물리적인 방식으로 제어를 할 수 없는 구조이다. 분산된 불뚱의 움직임은 입자-입자, 입자-화염, 공기저항, 카오스 영역 내 움직임 등 다양한 영향을 받기 때문에 사용자가 수작업으로 표현하기에는 너무 복잡하다.

본 논문에서는 카오스 영역에서 분산된 불뚱 입자의 흐름을 사실적이고 효율적으로 시뮬레이션하기 위한 새로운 프레임워크를 제안한다. 앞에서 언급했듯이 시뮬레이션 할 때 가장 어려운 측면은 공기저항에 의한 움직임과 다상 유체에서 발생하는 난류 등 카오스 현상에서 나타나는 무질서적인 움직임을 표현된다는 것이다. 또한 불뚱 입자 간의 개별적인 상호작용을 계산하고 서브그리드 입자에서 유체 솔버로 물리적인 운동량을 통합하는 것도 어려운 문제이다. 일반적으로 자유표면에서 표현되는 난류를 모델링한 접근법들이나, 밀도가 높은 물질이 밀도가 낮은 물질로 침투하면서 발생하는 레일리-테일러 불안정성(Rayleigh-Taylor instability)을 모델링한 접근법들이 있지만, 이러한 방법만으로는 화염 속에서 움직이는 불뚱의 디테일을 표현하기에는 충분하지 않다. 불뚱의 움직임은 지배 방정식을 시뮬레이션하여 표현하기 어렵기 때문에 본 논문에서는 불뚱 입자의 온도와 수명에 따른 다양한 부력 흐름과 확률론적 접근 방식을 이용한 혼돈 이류 방식을 제안한다. 화염과 불뚱 같은 서로 다른 매질의 경계 조건을 명시적으로 계산하는 대신, 본 논문에서는 연속체를 기반으로 분산된 불뚱 흐름의 평균 움직임을 부력을 통해 모델링한다. 격자 기반 솔버를 이용하여 전체적인 불뚱-불뚱 및 불뚱-화염 상호작용을 표현할 수 있다. 또한 불뚱 입자간의 상호작용을 시뮬레이션하기 위해 화염의 서브그리드 역학을 시뮬레이션하기 위한 새로운 확률론적 접근 방식을 소개한다. 본 논문에서는 3가지 과정으로 불뚱 입자의 움직임을 계산한다. 1) 화염의 연속체 솔버를 사용하여 불뚱 입자의 전역 운동을 효율적으로 표현한다. 2) 확률론적 혼돈 이류 기법을 통해 복합적인 움직임을 갖는 무질서한 운동을 표현한다. 또한, 3) 불뚱 입자의 수명을 기반으로 부력 흐름을 제어하여 다상 유체와 유사하게 공기 중으로 디테일하게 흘날리는 불뚱을 표현한다.

II. Related Works

Hong과 Kim은 공기방울을 시뮬레이션하기 위해 유체 부피법(Volume-of-fluid, VOF)을 도입했다[1]. 그 후 이들은 공기방울-액체 표면에서 물리적 특성으로 표현되는 불연속성을 포착하기 위해 고스트 유체 방법(Ghost fluid method)을 활용했다[2]. Song *et al.*은 기체와 액체에서 나타나는 수치손실 문제를 완화하기 위한 새로운 다상 유체 솔버를 제시했다[3]. Mihalef *et al.*은 Sussman *et al.*이 제안한 기법인 CLSVOF(Coupled level-set and volume-of-fluid) 방법을 적용하여 끓는 물을 시뮬레이션했고[11,12], Kim과 Carlson도 간단하게 끓는 물을 시뮬레이션 할 수 있는 프레임워크를 제안했다[13]. Zheng *et al.*은 매우 얇은 거품 경계면을 포착하기 위해 지역 레벨셋(Regional level-set) 기법을 개발했으며[4], Kim *et al.*은 이 연구를 확장하여 기포와 거품에 대한 체적 제어 방법을 제시했다[14]. Muller *et al.*은 SPH 기반의 다상 유체 솔버를 개발했으며[5], Cleary *et al.*은 거품과 거품을 일으키는 액체 시뮬레이션 기법을 제안했다[6]. 이 방법들은 주로 큰 기포 또는 기포-액체 인터페이스 역학에 중점을 두었다.

격자 해상도보다 작은 기포와 불뚱 같은 움직임을 포착하기 위해 격자 솔버에 입자를 활용한 하이브리드 접근법이 있다. Greenwood와 House는 입자 레벨셋(Particle level-set)을 기반으로[15], 공기 중으로 흩어진 입자인 탈출 입자(Escaped particle)들을 작은 기포로 변환하여 처리했다[16]. Hong *et al.*은 탈출 입자들을 버리지 않고 SPH로 계산하여 사실적으로 기포를 표현할 수 있는 하이브리드 프레임워크로 확장했다[7]. 이와 유사하게 Thurey *et al.*은 SPH 기포를 얇은 물 프레임워크(Shallow water framework)와 결합했고[17], Mihalef *et al.*은 마커 레벨셋(Marker level-set)을 사용하여 격자 기반 시뮬레이션에서 입자 모델을 통합할 수 있는 방법을 소개했다[8]. 그러나 대부분의 방법들이 물속에 존재하는 기포만을 표현하려고 했으며 화염에 의해 표현되는 불뚱 모델링은 연구가 많이 되지 않았다.

물리 기반 시뮬레이션에서는 화염을 표현하기 위해 다양한 방법들이 제시되었다. 격자 기반 유체 시뮬레이션이 도입된 후, 유사하게 온도장을 계산하여 화염을 표현하는 연구가 소개되었고[18], Nguyen *et al.*은 연료와 연소된 재질 사이의 경계면에서 표현되는 화염의 속도와 압력을 새롭게 모델링했다[19]. Hong *et al.*은 폭발-충격파를 곡률 기반으로 풀어내서 유체 방정식과 커플링할 수 있는 방식을 제안함으로써 사실적인 화염 패턴을 표현했다[20]. Horvath와 Geiger는 GPU를 기반으로 스크린 공간에서

화염 시뮬레이션을 빠르게 표현할 수 있는 프레임워크를 제안했다[21]. Kim *et al.*은 온도를 제어하여 목표 형상으로 화염의 움직임을 제어하는 방법을 제안했다[22]. 뿐만 아니라 폭발로 인한 시각적 이차 효과를 표현한 연구들도 있다[23,24].

격자 기반 시뮬레이션에서는 서브그리드 디테일을 표현하기 위해 입자를 활용하기도 한다. Feldman *et al.*은 입자를 활용하여 폭발로 인해 생성되는 폭발파를 표현하였다[25]. Hong *et al.*은 물속에 있는 공기방울의 움직임을 사실적으로 표현할 수 있는 입자-격자 하이브리드 기법을 제안했으며[7], 특히 SPH기반 와류기법을 제안하여 사실적인 공기방울의 움직임을 표현할 수 있는 수치기법을 소개했다. Zhao *et al.*은 GPU를 사용하여 목적 형상에 따라 움직이는 불꽃놀이 애니메이션 기법을 소개했고[26], Kim *et al.*은 움직이는 공기방울의 경로를 스케치 기반으로 제어하는 기법을 제안했다[27]. Kim *et al.*은 화염의 움직임으로 표현되는 불뚱을 공기방울과 유사하게 모델링했다[28]. 이들은 Hong *et al.*이 제안한 기포 역학과 유사하게 불뚱의 움직임을 풀어냈지만[7], 불뚱과 기포는 움직임이 다름에도 불구하고 결과상 큰 차이를 보이지 않았다. Kim과 Lee는 2차원 이미지로부터 화염의 흐름을 분석하여 실시간으로 불뚱 효과를 제작할 수 있는 프레임워크를 제안했다[29].

Table 1. Characteristics of previous methods.

Method	Feature
Hong et al.[7]	Utilizing bubble dynamics
Kim et al.[27]	Controlling shape of the bubbles
Kim et al.[28]	Modeling sparks based on bubble dynamics
Kim and Lee[29]	Modeling flame movement using image analysis
Choi et al.[30]	Learning to model the behavior of sparks

최근에 Choi *et al.*은 Kim *et al.*의 방법을 인공지능을 통해 학습시켰지만 불뚱의 디테일보다는 간단한 입자 시스템과 유사한 형태를 보였으며, 공기 중으로 흩날리는 형태의 불뚱을 표현하기에는 충분하지 않다[28,30]. Table 1은 위에서 언급한 기존 기법의 특징을 요약한 테이블이다.

III. The Proposed Scheme

1. Underlying Fluid Solver

본 논문에서는 화염을 모델링할 때 비압축성 다상 유체 흐름을 가진다고 가정한다. 비압축성 나비에-스톡스 방

정식(Navier-Stokes equation)은 운동량 보존과 질량 보존 방정식으로 주어진다 (수식 1과 2 참조).

$$u_t + (u \cdot \nabla p/\rho) = \nabla \cdot (\mu \nabla u)/\rho + f/\rho \quad (1)$$

$$\nabla \cdot u = 0 \quad (2)$$

여기서 u, p, ρ, μ, f 는 각각 속도, 압력, 밀도, 점성, 외력을 나타내며, t 는 현재 시간에 대한 물리량을 표현하려고 추가된 하위첨자이다. 또한 ∇ 은 발산 연산자를 나타낸다. 본 논문에서는 Stable fluids 프레임워크를 사용하여 위 방정식을 계산한다[31].

2. Basic Force of Fire-Flake Particles

본 논문에서는 네 가지 유형의 힘으로 불똥 입자의 기본적인 힘을 모델링한다[28]. 공기의 흐름을 나타내는 Airflow force f^{air} , 항력인 Drag force f^{drag} , 양력인 Lift force f^{lift} , 마지막으로 부력인 Buoyancy force f^{buoy} . 각 힘은 불똥 입자마다 적용되어 속도를 업데이트한다 (수식 3 참조).

$$f = f^{air} + f^{drag} + f^{lift} + f^{buoy} \quad (3)$$

불똥 입자의 이동에 가장 큰 영향을 미치는 요인은 난류로 인한 기류이다. 수식 3에서 정의했듯이 시뮬레이션 공간에서 생성된 속도장은 와류 제어를 통해 불똥 입자의 움직임을 사실적으로 표현할 수 있다. 본 논문에서는 첫 번째로 공기의 흐름인 f^{air} 를 수식 4를 이용하여 계산한다.

$$f_i^{air} = k_{air} v_i \quad (4)$$

여기서 k_{air} 는 공기 속도 계수이고, v_i 는 격자속도인 u 에서 보간된 속도이며, 보간된 위치는 입자의 위치이다. i 는 불똥 입자의 인덱스이다.

불똥 입자의 질량은 작기 때문에 공기 중에서 무질서하게 움직이며, 이런 움직임은 불똥을 사실적으로 표현하는데 있어서 중요한 역할을 한다. 본 논문에서는 항력과 양력을 이용하여 이와 유사한 움직임을 표현한다. 불똥이 공기 중에서 움직일 때 항력은 저항하는 힘이며, 양력은 불똥 입자를 둘러싼 공기 흐름에 수직으로 생성된다. 이 과정은 속도장과 불똥의 온도를 이용하여 계산한다. Hong *et al.*은 항력과 양력을 활용하여 공기방울의 움직임을 모델링했으며[7], 본 논문에서는 그들의 방법을 불똥 움직임에 맞게 수정한다 (수식 5 참조).

$$f_i^{drag} = -k_{drag} r^2 |v_i - u_i| (v_i - u_i) \quad (5)$$

여기서 k_{drag} 는 항력 계수이며, r 은 불똥 입자의 반지름이다. 마찬가지로 양력은 수식 6을 통해 계산한다.

$$f_i^{lift} = -k_{lift} V_i T_i (v_i - u_i) \times w_i \quad (6)$$

여기서 k_{lift} 는 양력 계수, V_i 는 불똥 입자의 부피, T_i 는 불똥 입자의 온도, w_i 는 기류의 회전 속도이다 : $w_i = \nabla \times u_i$. 불똥 입자의 온도는 격자온도에서 보간된 값을 사용한다. 불똥 입자의 부력은 온도를 고려하여 수식 7을 통해 계산한다.

$$f_i^{buoy} = k_{buoy} T_i f_{buoy} \quad (7)$$

여기서 k_{buoy} 는 부력 계수, f_{buoy} 는 상향 정규 벡터이다.

3. Generation of Fire-Flake Particles

물리적으로 불똥은 탄소 기반 연료에 영향을 미치는 외부 힘이나 공기 주입에 의해 생성된다. 탄소 기반 물질에서 떨어져 나온 조각은 스스로 연소할 만큼 뜨겁고 공기 중에 떠다닐 만큼 가벼워서 육안으로 불똥이라는 개체로 보인다. 불똥 입자가 생성되는 경계 영역은 온도와 속도 변화에 따라 상대적으로 휘발성이 높다. 따라서 온도가 충분하고 속도와 온도가 급변할 가능성이 있는 위치를 예측할 수 있으며, 이를 계산하기 위해 다음과 같은 에너지 함수를 한다 (수식 8 참조).

$$E_i = E_i^t + E_i^k \quad (8a)$$

$$E_i^t = k_t T_i \quad (8b)$$

$$E_i^k = \frac{1}{2} \rho v_i^2 \quad (8c)$$

여기서 E^t 는 열에너지, k_t 는 열용량 계수, E^k 는 운동 에너지, ρ 는 밀도이다. 따라서 온도가 k_t 를 초과하고 시간에 따른 에너지 변화가 k_e 를 초과하는 지역이 불똥 입자가 발생할 가능성이 높은 영역이며, 이를 찾기 위해 수식 9를 이용하여 계산한다.

$$\frac{\partial E}{\partial t} > k_e \quad E^t > k_t \quad (9)$$

여기서 k_t 와 k_e 는 임계값이며, 본 논문에서는 각각 2.0과 0.6으로 설정했고, 이 값은 실험적으로 찾은 값이다. 수식 9를 통해 선별된 모든 영역에서 불똥 입자를 생성하지만, 생성 범위를 벗어난 불똥 입자들은 별도로 처리해야 한다. 이 과정을 적용하지 않으면 너무 많은 불똥이 생성되며, 화염의 움직임과는 다른 형태로 제어되는 문제가 발

생한다. 이 문제를 완화시키기 위해 Kim *et al.*이 제안한 불똥 생성 조건을 활용한다[28].

4. Varying Buoyancy for Preserving Details

이번 장에서는 정적인 부력이 아닌 다상 유체와 같은 공기 흐름으로 인한 불똥 입자의 난류를 표현하고자 새로운 부력을 모델링한다 (수식 10과 11 참조).

$$f_i^{buoy'} = k_{buoy} T_i f_{buoy} \eta \quad (10)$$

$$\eta_{t+\Delta t} = \eta_t + \|v_i\| - \epsilon \quad (11)$$

여기서 η 는 불똥 입자의 수명이고 ϵ 는 감쇠계수이다. 수식 11은 입자의 수명을 갱신하는 과정이다. 이 값은 부력과 통합하여 다양한 형태의 부력장이 되도록 모델링한다. 추가적으로 본 논문에서는 불똥 입자가 공기 중에서 흩어지면서 표현되는 난류를 모델링하기 위해 아래와 같은 힘을 추가한다 (수식 12 참조).

$$f_i^{vorticity} = \epsilon \left(N \times \frac{w}{|w|} \right) \rho_i \eta_i T_i \quad (12)$$

여기서 와류는 인접한 두 불똥 입자들 중심에서 계산한다 : $w = \nabla \times v$. 와류의 위치 벡터는 아래 수식과 계산하고

$$\eta = \frac{m_i p_i + m_j p_j}{m_i + m_j} - p_i, \text{ 이 값을 정규화하여 사용한다.}$$

추가적으로 불똥 입자의 수명과 온도를 고려하여 온도가 강하고 수명이 길수록 와류가 강하게 표현되도록 했다.

5. Chaotic Advection of Fire-Flake Particles

이번 장에서는 불똥 입자의 혼돈 이류를 구현하기 위한 확률론적 솔버를 제안한다. 일반적으로 파도의 출렁임, 물에 떨어뜨린 잉크방울의 퍼짐, 불똥 등과 같이 대부분의 자연현상이 무질서한 특성을 보인다. 본 논문에서는 무질서한 움직임을 계산하기 위해 가우시안 랜덤 난수를 기반으로 불똥 입자의 이류를 모델링한다. 상승하는 각 불똥 입자는 자신을 포함한 다른 불똥과 공기에 영향을 미치는 파동을 생성하여 카오스 영역과 같은 복잡한 유체 역학을 일으킨다. 이론적으로 불똥에서 생성되는 소용돌이와 유동을 정확하게 포착하면 되지만, 이 접근 방식은 카오스 영역에서 발생하는 환경적 요소와 고해상도 미세 불똥을 시뮬레이션 해야 되기 때문에 비효율적이다. 대신 이 현상을 본 논문에서는 이산 랜덤워크(Random walk)로 모델링한다. 이 과정을 모델링하기 위해 불똥 분산이 발생할 수 있는 공간에 대한 확률 함수 $s(\cdot)$ 를 사용하며, 이 함수는 수식 13을 이용하여 계산한다.

$$s_i = \phi(1 - T_i)v_i + \delta g_i \quad (13)$$

여기서 ϕ 는 사용자가 지정한 분산 계수이고, δ 는 랜덤 워크에 대한 가중치 계수이며, $1 - T_i$ 는 기체의 분율이고, g_i 는 분산되는 불똥의 움직임을 표현하는 가중치이다. 본 논문에서는 분산 계수인 ϕ 를 0.25로 설정했다. 각 불똥 입자에서 속도를 업데이트 한 후 s 를 측정한다. 불똥 입자의 온도는 격자의 온도보다 상대적으로 작기 때문에 공기 중으로 분산되는 불똥 입자의 움직임을 추적하기 위해 g_i 를 추가한다. 공기 중으로 분산되는 경우 속도와 온도는 매우 약하기 때문에 g_i 를 통해 새로운 방향을 계산한다 (수식 14 참조). 이 수식은 가우시안 랜덤기반의 확률론적 난수이다.

$$g \cdot x = \frac{g_{rand}}{w-1}, g \cdot y = \frac{g_{rand}}{w-1} \quad (14)$$

본 논문에서는 확률론적 솔버를 계산하기 위해 다음과 같은 과정을 거친다. 1) 시드 설정, 2) 가우시안 난수 생성, 3) 확률론적 움직임 계산. 시드 과정에서는 가우시안 랜덤 난수에 필요한 변수를 초기화하며 수식 15를 이용한다.

$$g_{addition} = \sqrt{3\zeta}, g_{fractal} = \frac{2g_{addition}}{\zeta\nu} \quad (15)$$

여기서 ν 는 RAND_MAX를 나타내며, ζ 는 랜덤 개수에 대한 가중치이고 본 논문에서는 4로 설정했다. 이 값은 실험적으로 설정한 값이다. 가우시안 난수 생성은 다음과 같은 수식을 통해 계산한다 (수식 16 참조).

$$g_{rand} = g_{fractal} \left(\sum_{i=0}^{\zeta} rand() \right) - g_{addition} \quad (16)$$

본 논문에서는 수식 14를 이용하여 불똥을 이루시킬 때 구간별로 다른 조건을 설정한다. 1) 불똥 입자의 속도 크기가 임계값보다 크다면 δ 를 50으로 설정했고, 여기서 임계값은 0.1이다. 2) 불똥 입자의 가속도 크기가 임계값보다 크다면 δ 를 5로 설정했고, 여기서의 임계값은 0.2로 설정했으며, 위 값들은 실험적으로 설정했다. 최종적으로 s_i 는 v_i 에 더해지고 오일러 적분을 통해 위치를 갱신한다.

Fig. 2는 불똥 입자가 부력에 의해 위로 상승한다고 가정하고 s 를 적용한 결과이다. 카오스 영역에서 표현되는 불똥의 지그재그한 패턴이 사실적으로 잘 나타났다.

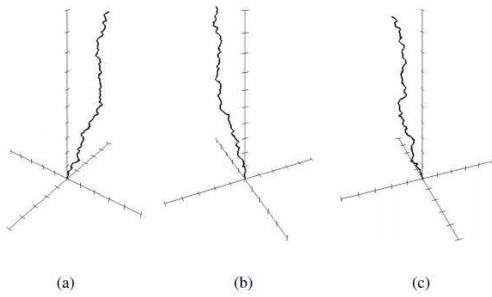
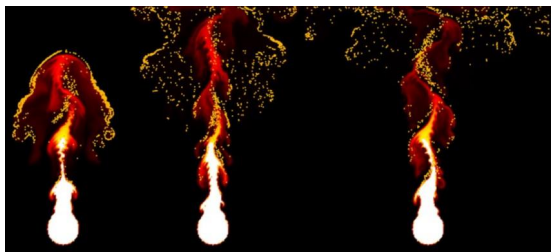


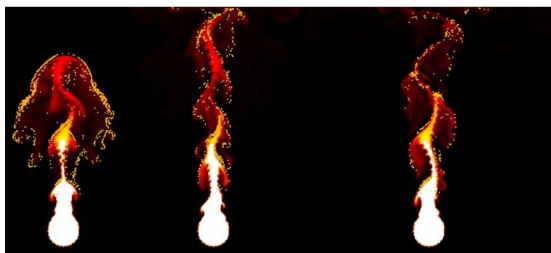
Fig. 2. This plot shows the unstable path(zigzag pattern) of a fire-flake particle.

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. 본 연구에서는 화염에 의해 분산된 불뚱의 움직임과 디테일한 불뚱 흐름을 효율적으로 표현하고자 확률론적 기반의 혼돈 이류 방식과 다양한 부력 흐름을 제안했다. 본 논문에서는 제안한 방법의 효율성과 정확성을 실험하기 위해 화염이 주입되는 장면과 속도장을 통해 검증했고, 이를 OpenGL로 구현했다.



(a) Our method



(b) Previous method [28]

Fig. 3. Rising flame (orange particle : fire-flake).

Fig. 3은 부력에 의해 상승하는 화염 시뮬레이션에서 제안하는 방법과 이전 방법의 결과를 비교한 그림이다. 모든 장면에서 화염을 표현하기 위해 격자 해상도는 256×256 를 이용했으며 타임스텝은 0.1로 설정했다. 시뮬레이션 초기인 37프레임에서는 화염의 경계부근에서 불뚱 입자가 생성됐지만, Kim *et al.*에서는 온도가 낮아지면서 분산된

움직임을 갖는 불뚱이 거의 보이지 않았으며, 오직 화염의 움직임에만 의존하는 결과만을 보여줄 뿐, 무질서한 불뚱을 표현하기에는 충분하지 않다 (Fig. 3(b) 참조). 상대적으로 제안하는 방법에서는 온도가 낮은 부분에서도 분산된 불뚱의 움직임이 명확하게 표현됐다 (Fig. 3(a) 참조).

불뚱의 움직임과 공기방울은 차이가 있다. 바로 카오스 영역 내 속도가 다양하다는 것이다. 예를 들어, 물속에 존재하는 공기방울의 경우 부력에 의해 공기방울들이 뭉쳐 지기도 하고 분리되기도 한다[7]. 하지만, 유속 내에서 이러한 현상들이 표현되기 때문에 대부분 기반유동에 의존하는 결과가 나타난다. 이전 기법에서도 기반유체의 흐름에 의존하는 결과가 나타났으며, 이 원인은 공기로 흩어지는 불뚱의 움직임을 표현하기 어렵게 한다.

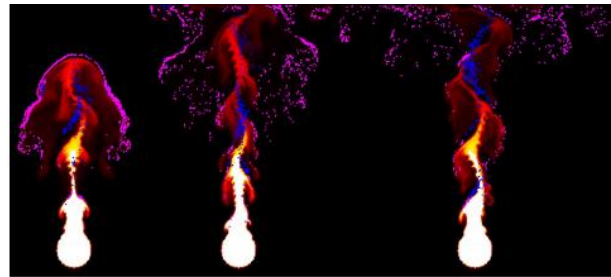


Fig. 4. Fire-flake particles with different movements and velocities in a chaotic region (input flame : Fig. 3, pink particle : dispersed fire-flake, blue particle : fire-flake generated near flame).

Fig. 4는 제안하는 방법을 통해 만들어진 결과에서 공기에 의해 분산된 입자들을 표현한 그림이다. 분홍색은 공기에 의해 분산된 불뚱이며, 여기서의 움직임은 v 보다 s 의 영향을 많이 받는 입자들이다.

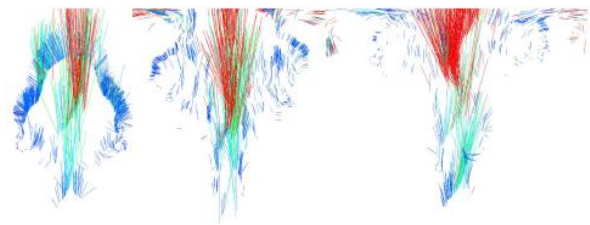


Fig. 5. Visualizing the velocity of fire-flake particles (input : Fig. 3, red : fast, blue : slow).

Fig. 5는 불뚱 입자의 속도를 표현한 결과이다. 다양한 속도의 크기를 가진 움직임들이 혼재된 것을 볼 수 있으며, 이러한 무질서한 움직임이 불뚱의 디테일을 표현하게 된다. 기존의 방법은 속도장 주변이 유사한 크기를 가지지만, 제안하는 방법은 같은 공간에서도 다양한 속도를 표현

할 수 있기 때문에 기존에 표현하기 어려웠던 무질서적인 움직임을 효율적으로 표현할 수 있다는 장점을 가진다.

V. Conclusions

본 논문에서는 화염의 움직임에 따라 분산된 불뚱과 무질서한 불뚱 움직임을 효율적으로 시뮬레이션할 수 있는 기법을 제안했다. 화염의 움직임에만 의존하는 기존의 불뚱 입자를 다양한 형태의 부력과 새로운 이류 방식을 통해 디테일하게 개선시켰다. 기존의 랜덤워크 방식처럼 무작위로 잡음이 혼합된 움직임을 추가하는 것이 아닌 화염의 크기와 방향을 고려하기 때문에 대부분의 장면에서 매개변수 수정 없이 안정적으로 수행됐다. 그럼에도 불구하고 몇 가지 한계점이 있다. 불뚱 입자들끼리의 상호작용이나 그로 인해 표현되는 난류를 고려하지 않았기 때문에 마이크로 단위의 디테일이나 많은 개수의 불뚱을 표현하기에는 아직 물리적인 속성이 부족하다 (예 : 불뚱의 충돌, 병합, 모양 변화, 중첩 문제 등). 이러한 문제를 방지하려면 대규모 불뚱 입자들 간의 근접 테스트 또는 충돌 감지를 실행해야 한다. 본 논문에서는 불뚱 간의 직접적인 상호작용을 고려하지 않았기 때문에 부자연스러운 부분이 있을 수 있지만, 공기방울 같이 병합하는 과정에서 모양이 크게 변형되지 않기 때문에 큰 문제로는 이어지지 않는다. 불뚱 입자별로 상호작용을 수행하면 알고리즘 성능이 저하되기 때문에 본 논문에서는 입자의 속성별로 부력과 이류를 적용하여 근사치를 계산했다.

본 논문에서 제안하는 방법은 동일한 환경에서 디테일한 불뚱 입자를 표현할 수 있다는 장점을 가지기 때문에 효율적이다. 여기서 말하는 디테일은 화염에만 의존하는 움직임이 아닌, 다상 유체와 같은 계산양이 많고 복잡한 카오스 영역을 적은 계산으로 표현할 수 있다는 것이다. 향후에는 인공지능을 통해 불뚱의 움직임을 학습적으로 표현하도록 확장할 계획이다. 또한 시뮬레이션 없이 화염의 온도와 기하학적인 형태만을 이용하여 불뚱의 움직임을 효율적으로 변형할 수 있는 방법을 새롭게 연구할 계획이다.

ACKNOWLEDGEMENT

This work was supported by INHA UNIVERSITY Research Grant.

REFERENCES

- [1] J.-M. Hong and C.-H. Kim, "Animation of bubbles in liquid," in *Computer Graphics Forum*, vol. 22, no. 3. Wiley Online Library, pp. 253-262, 2003. DOI: 10.1111/1467-8659.00672
- [2] J.-M. Hong and C.-H. Kim, "Discontinuous fluids," *ACM Transactions on Graphics(TOG)*, Vol. 24, No. 3, pp. 915-920, 2005. DOI: 10.1145/1186822.1073283
- [3] O.-Y. Song, H. Shin, and H.-S. Ko, "Stable but nondissipative water," *ACM Transactions on Graphics (TOG)*, Vol. 24, No. 1, pp. 81-97, 2005. DOI: 10.1145/1037957.1037962
- [4] W. Zheng, J.-H. Yong, and J.-C. Paul, "Simulation of bubbles," *Graphical Models*, Vol. 71, No. 6, pp. 229-239, 2009. DOI: 10.1016/j.gmod.2009.08.001
- [5] M. Muller, B. Solenthaler, R. Keiser, and M. Gross, "Particlebased fluid-fluid interaction," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 237-244, 2005. DOI: 10.1145/1073368.1073402
- [6] P. W. Cleary, S. H. Pyo, M. Prakash, and B. K. Koo, "Bubbling and frothing liquids," in *ACM SIGGRAPH 2007 papers*, pp. 97, 2007. DOI: 10.1145/1276377.1276499
- [7] J.-M. Hong, H.-Y. Lee, J.-C. Yoon, and C.-H. Kim, "Bubbles alive," *ACM Transactions on Graphics (TOG)*, Vol. 27, No. 3, pp. 1-4, 2008. DOI: 10.1145/1360612.1360647
- [8] V. Mihalef, D. Metaxas, and M. Sussman, "Simulation of two-phase flow with sub-scale droplet and bubble effects," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, pp. 229-238, 2009. DOI: 10.1111/j.1467-8659.2009.01362.x
- [9] W. Geiger, M. Leo, N. Rasmussen, F. Losasso, and R. Fedkiw, "So real it'll make you wet," in *ACM SIGGRAPH 2006 Sketches*, pp. 20, 2006. DOI: 10.1145/1179849.1179874
- [10] Y. Qiu, L. Yang, S. Li, Q. Xia, H. Qin, and A. Hao, "Novel fluid detail enhancement based on multi-layer depth regression analysis and flip fluid simulation," *Computer Animation and Virtual Worlds*, Vol. 28, No. 5, pp. e1741, 2017. DOI: 10.1002/cav.1741
- [11] M. Sussman, "A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles," *Journal of Computational Physics*, Vol. 187, No. 1, pp. 110-136, 2003. DOI: 10.1016/S0021-9991(03)00087-1
- [12] V. Mihalef, B. Unlusu, D. Metaxas, M. Sussman, and M. Y. Hussaini, "Physics based boiling simulation," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 317-324, 2006. DOI: 10.1145/1218064.1218108
- [13] T. Kim and M. Carlson, "A simple boiling module," in *Symposium on Computer Animation: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Vol. 2, No. 4, pp. 27-34, 2007.

- [14] B. Kim, Y. Liu, I. Llamas, X. Jiao, and J. Rossignac, "Simulation of bubbles in foam with the volume control method," *ACM Transactions on Graphics (TOG)*, Vol. 26, No. 3, pp. 98, 2007. DOI: 10.1145/1275808.1276500
- [15] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surfaces," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 736-744, 2002.
- [16] S. T. Greenwood and D. H. House, "Better with bubbles: enhancing the visual realism of simulated fluid," in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 287-296, 2004. DOI: 10.1145/1028523.1028562
- [17] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross, "Real-time simulations of bubbles and foam within a shallow water framework," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 191-198, 2007. DOI: 10.1145/1272690.1272716
- [18] Z. Melek and J. Keyser, "Interactive simulation of fire," in *10th Pacific Conference on Computer Graphics and Applications*, 2002. *Proceedings. IEEE*, pp. 431-432, 2002.
- [19] D. Q. Nguyen, R. Fedkiw, and H. W. Jensen, "Physically based modeling and animation of fire," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 721-728, 2002. DOI: 10.1145/566570.566643
- [20] J.-M. Hong, T. Shinar, and R. Fedkiw, "Wrinkled flames and cellular patterns," *ACM Transactions on Graphics (TOG)*, Vol. 26, No. 3, pp. 47, 2007.
- [21] C. Horvath and W. Geiger, "Directable, high-resolution simulation of fire on the gpu," *ACM Transactions on Graphics (TOG)*, Vol. 28, No. 3, pp. 1-8, 2009.
- [22] T. Kim, J. Lee, and C.-H. Kim, "Physics-inspired controllable flame animation," *The Visual Computer*, Vol. 32, pp. 871-880, 2016.
- [23] G. Kawada and T. Kanai, "Procedural fluid modeling of explosion phenomena based on physical properties," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 167-176, 2011. DOI: 10.1145/2019406.2019429
- [24] N. Kwatra, J. Gretarsson, and R. Fedkiw, "Practical animation of compressible flow for shockwaves and related phenomena," in *Symposium on Computer Animation*, pp. 207-215, 2010.
- [25] B. E. Feldman, J. F. O'Brien, and O. Arikan, "Animating suspended particle explosions," in *ACM SIGGRAPH 2003 Papers*, pp. 708-715, 2003. DOI: 10.1145/1201775.882336
- [26] H. Zhao, R. Fan, C. C. Wang, X. Jin, and Y. Meng, "Fireworks controller," *Computer Animation and Virtual Worlds*, Vol. 20, No. 2-3, pp. 185-194, 2009. DOI: 10.1002/cav.287
- [27] P.-R. Kim, H.-Y. Lee, J.-H. Kim, and C.-H. Kim, "Controlling shapes of air bubbles in a multi-phase fluid simulation," *The Visual Computer*, Vol. 28, pp. 597-602, 2012. DOI: 10.1007/s00371-012-0696-x
- [28] T. Kim, E. Hong, J. Im, D. Yang, Y. Kim, and C.-H. Kim, "Visual simulation of fire-flakes synchronized with flame," *The Visual Computer*, Vol. 33, pp. 1029-1038, 2017.
- [29] J.-H. Kim and J. Lee, "Fire sprite animation using fire-flake texture and artificial motion blur," *IEEE Access*, Vol. 7, pp. 110002-110011, 2019. DOI: 10.1109/ACCESS.2019.2934163
- [30] M. Choi, J. A. Wi, T. Kim, Y. Kim, and C.-H. Kim, "Learning representation of secondary effects for fire-flake animation," *IEEE Access*, Vol. 9, pp. 17620-17630, 2021. DOI: 10.1109/ACCESS.2021.3054061
- [31] J. Stam, "Stable fluids," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 121-128. DOI: 10.1145/311535.311548

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.