불똥 입자의 이류과 삭제를 효율적으로 학습 표현하는 인공신경망

김동희[O], 김종현[*]
[O]강남대학교 소프트웨어응용학부,
[*]강남대학교 소프트웨어응용학부
e-mail: jonghyunkim@kangnam.ac.kr

An Artificial Neural Network for Efficiently Learning and Representation the Advection and Remove of Fire-Flake Particles

Donghui Kim[O], Jong-Hyun Kim[*]
[O]School of Software Application, Kangnam University,
[*]School of Software Application, Kangnam University

● 요 약 ●

본 논문에서는 유체 시뮬레이션(Fluid simulation)중 화염에서 표현되는 불똥 입자(Fire-flake particle)의 생성, 움직임과 삭제를 효율적으로 학습하고 표현할 수 있는 인공지능 기법에 대해 소개한다. 유체 시뮬레이션을 계산하기 위해서는 일반적으로 수치해석학과 같은 학문의 이해가 필요하며 불똥이나 거품과 같은 유체의 2차 효과(Secondary effect)는 기반유체(Underlying fluids)를 통해 추출되기 때문에 복잡하고 계산양이 많아진다. 이러한 문제를 완화하고자 본 논문에서는 인공신경망을 이용한 분류 모델 학습을 통해 격자 내에서 표현되어야 하는 불똥 입자의 생성을 학습하고, 다항 회귀 모델 학습을 통해 불똥 입자의 움직임을 예측한다. 또한, 불똥 입자가 삭제되어야하는 상태를 네트워크 학습을 통해 얻어내며, 수명(Lifespan) 임계값 조절하여 다양한 장면에서 불똥을 제어할 수 있다. 결과적으로 화염의 움직임을 기반으로 불똥의 움직임을 복잡한 수학식이나 디자이너에게 의존하지 않고 인공지능 학습을 통해 쉽게 제어하고 예측하는 결과를 보여준다.

키워드: 유체 시뮬레이션(Fluid simulation), 화염(Flame), 인공신경망(Artificial neural network), 불똥 입자(Fire-flake particle), 2차 효과(Secondary effects)

I. Introduction

엔지니어링 기술과 IT기술이 발전하면서 디지털 트윈(Digital twin)과 같은 컴퓨터 시뮬레이션 기술은 실제 물리현상을 정확히 묘사하고 실시간으로 분석할 수 있는 수준으로 발전하고 있다. 특히, 파도와 화염 등과 같은 움직임으로 발생하는 2차 효과(예 : 파도 거품, 불똥)등의 특징을 시각적으로 표현하는 것은 쉽지 않은 문제이며 [1-3], 다음과 같은 제약조건, 문제점이 발생한다 : 1) 수치해석학에 대한 학문의 이해가 필요하고, 2) 유체 방정식인 나비에-스토크스 방정식(Navier-Stokes equations)을 공간 및 시간에 대한 이산화와 수치 해석적인 기법을 통해 근사해를 얻어내기 때문에 계산 시간이 오래 걸린다[4] (Fig.1 참조). 따라서 실시간 표현이 어려워지기 때문에 게임이나 혼합현실과 같은 애플리케이션에서 활용이 어렵다. 3) 유체의 불연속성과 비예측성이라는 특징으로 인해 움직임을 계산하기 어려우며, 2차 효과는 더욱 복잡하기 때문에 일반적으로 많은 계산을

Fig. 1. Flame and fire-flake phenomenon.

하거나 디자이너의 수작업에 많이 의존한다. 따라서 본 논문에서는 인공신경망 학습을 기반으로 화염의 불규칙한 움직임에 대해 환경을

파악하고 불똥 입자가 생성되어야 할 위치와 움직임을 예측할 수 있는 프레임워크를 제안한다. 이는 수치해석에 대한 전문지식이 없는 사용자도 쉽게 사용가능하며 학습이 완료된 시스템에 대해 실시간으로 불똥 효과를 표현할 수 있다.

II. The Proposed Scheme

본 논문에서 이전 화염 시뮬레이션 기법을 통해 데이터를 얻었으며 [3], 인공신경망은 불똥 생성 네트워크와 움직임 네트워크가 각각 독립적으로 훈련된다. 훈련 데이터는 네트워크에 적용되기 전 학습에 적합한 형태로 전처리 과정을 거치며, 학습이 완료된 뒤 각 모델마다 테스트 결과를 시각화한다.

1. 불똥 생성 네트워크

불똥 생성 여부를 알기 위해 필요한 것은 불똥이 생성될 수 있는 조건이다. 불똥 생성 네트워크 모델에서 사용한 데이터는 격자기반 화염 시뮬레이션에서 얻은 밀도, 온도, 속도장의 데이터를 이용한다 (Fig. 1 참조)

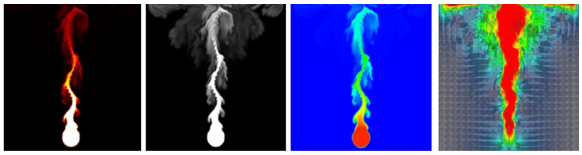


Fig. 2. Metadata : (a) flame, (b) density, (c) temperature, (d) velocity.

메타데이터를 만들기 위한 격자 해상도는 256×256이며, 각 노드마다 불똥의 생성 여부를 플래그를 통해 나타낸다. 학습 데이터는 전체 격자를 대상으로 진행했으며, 전체 노드 개수에 비해 불똥이 생성된 노드들은 극히 일부분으로서, 이런 데이터를 그대로 사용하게 되면 학습 가중치가 한쪽으로 기울 가능성이 크다. 이 문제를 완화하기 위해 본 논문에서는 고스트 셀(Ghost cell)과 1:1 샘플링 과정을 거쳐 분류 학습에 필요한 데이터 수로 정제한다. 고스트 셀은 불똥 입자가 존재하는 격자에 패딩개념을 추가하여, 주변 격자까지 유효한 값으로 설정하는 방법이다. (Fig. 3 참조). Fig. 3a에서보듯이 불똥은 (1,1)노드에 존재한다. 하지만, 실제로 이러한 접근법은 데이터 개수의 비대칭 때문에 학습을 저해하는 요소가 될 수 있다. 고스트 셀 방법은 인접 인접의 노드들을 일시적으로 유효한 상태로 만들어 가상의 데이터를 활용하는 방식이다 (Fig. 3b 참조).

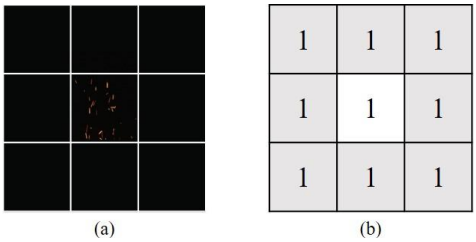


Fig. 3. Ghost cell structure (grey node : ghost cell).

고스트 셀은 주변 셀의 개수를 확보함으로써 일시적으로 유효한 가상의 데이터를 만들 수는 있지만, 전체 노드 개수에 비해서 여전히 부족하다. 그렇기 때문에 본 논문에서는 1:1 샘플링 기법을 이용한다. 샘플링은 분류 모델의 각 클래스에 대해 데이터 수를 1:1로 설정하여 불균형한 데이터 수로 인해 학습이 잘 이뤄지지 않는 것을 예방하기 위한 방법이다. 앞에서 언급한 데이터 정제 과정 이후 불똥 입자의 생성 네트워크를 학습한다. 불똥 생성 네트워크 모델은 총 3개의 층을 가지며, 활성함수는 히든 레이어 사이 함수로는 LeakyReLU, 마지막 활성함수로는 Softmax를 사용한다 (Fig. 4 참조).

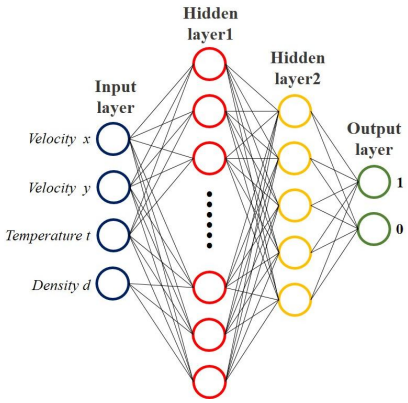


Fig. 4. Generator network of fire-flake.

LeakyReLU의 α값은 0.1 이며 이는 ReLU와 Tanh함수를 사용한 결과와 비슷한 결과를 보인다 (Fig. 5 참조). 그림에서 보듯이 3개의 활성화 함수 결과가 거의 유사하기 때문에 본 논문에서는 LeakyReLU와 Softmax를 사용하였다. Fig. 5에서 빨간색은 화염이며, 흰색은 불똥 생성 네트워크를 통해 불똥 입자가 생성되어야 할 위치를 예측한 결과이다.

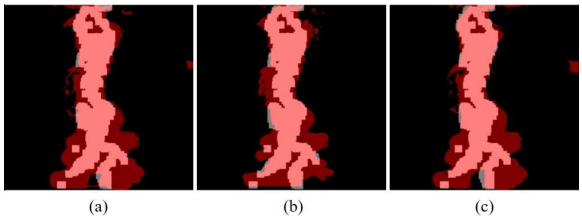


Fig. 5. The location of fire-flake particles predicted through various activation functions : (a) ReLU, (b) LeakyReLU, (c) Tanh.

2. 불똥 움직임 네트워크

이번 장에서는 불똥의 움직임을 예측하는 네트워크 학습방법에 대해 소개한다. 제안된 모델은 현재 프레임에 대한 불똥의 입자 속도를 직접적으로 추론하는 것이 아닌, 다음과 같이 인공지능을 통해 불똥 입자의 속도변화량인 Δv^{ai}를 추론한다.

입력 특징 벡터(Input feature vector)인 X는 불똥 입자가 존재하는 노드와 해당 노드에 이웃하는 노드의 화염 속도와 해당 불똥 입자의 속도를 가진다. 이 때 이웃하는 노드는 2차원에서 상, 하, 좌, 우의 노드들을 의미한다. 불똥의 움직임 네트워크의 레이어는 생성 네트워

크보다 한 층 더 깊으며 활성함수는 ReLU, 비용함수는 Least square error를 사용한다 (Fig. 6 참고)

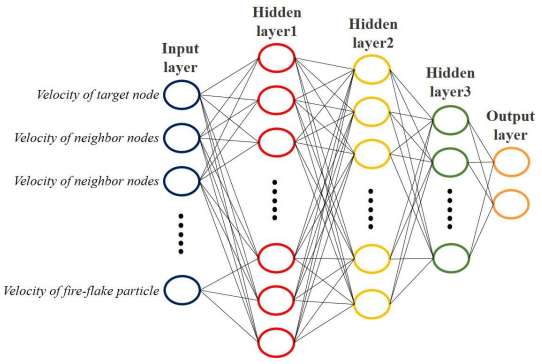


Fig. 6. Velocity modifier network of fire-flake particles.

불똥 생성 네트워크를 통해 불똥의 생성 위치를 결정하고 움직임 네트워크를 통해 불똥의 속도 변화량인 Δv^{ai}를 얻은 후 다음과 같이 불똥의 위치를 갱신한다 (수식 2 참조).

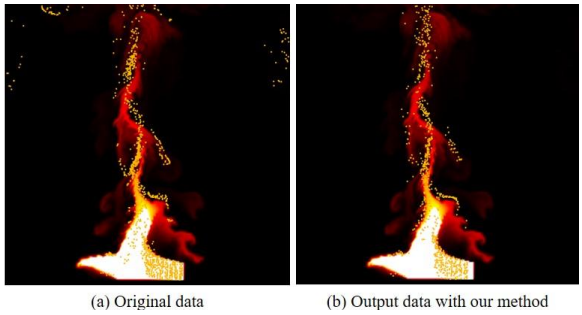
$$v_{t+\Delta t} = v_t + \Delta v^{ai}\Delta t \quad (2)$$
$$p_{t+\Delta t} = p_t + v_{t+\Delta t}\Delta t$$

3. 불똥 삭제 네트워크

이번 장에서는 불똥 입자의 삭제를 제어하는 네트워크 방법을 제안한다. 불똥 입자의 수명은 매 프레임 입자의 운동 에너지의 크기를 누적하고, 사용자가 정의한 임계값만큼을 차감하는 식으로 설계했다 : $\left(\sum_t \frac{1}{2}mv_t^2\right)$. 최종적으로, 수명이 음수가 되는 불똥 입자들에 대해서만 삭제를 했으며, 운동 에너지가 거의 미비한 입자들이 주로 삭제되는 경향이 보였다. 불똥의 수명은 불똥 입자의 속도에 의존하기 때문에 불똥 입자의 속도 값과 운동에너지를 입력벡터로 가지는 네트워크로 추가학습을 진행했으며 출력으로 수명이 추론되는 구조이다. 네트워크 형태는 움직임 네트워크와 유사하며, 본 논문에서는 수명이 0.05이하의 값을 삭제하였고, 이 값을 제어하여 불똥 입자의 개수 조절이 가능하다.

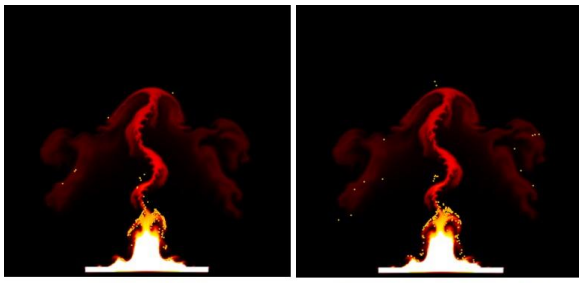
III. Results

본 논문에서는 화염에 따른 불똥의 움직임을 보여주기 위해서 다양한 장면에서 실험을 진행했다. Fig. 7은 화염의 움직이는 소시 위치가 좌우로 이동하는 장면이며, 이 과정에서도 불똥의 생성과 움직임, 그리고 삭제가 자연스럽게 표현되었다.



(a) Original data (b) Output data with our method

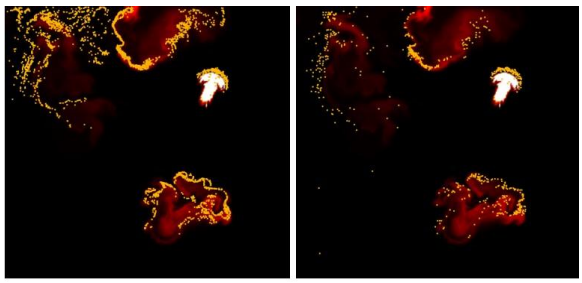
Fig. 7. Fire-flake effects represented by moving flames.



(a) Original data (b) Output data with our method

Fig. 8. Fire-flake effects expressed due to buoyancy flame.

Fig. 8은 고정된 위치에서 주입되는 화염이 부력에 의해 이류하고, 그로 인해 표현되는 불똥 효과이다. 화염이 강하게 표현되는 가운데 영역뿐만 아니라, 화염이 약간 바깥쪽 부근에서도 공기에 의해 표류하는 불똥의 디테일한 움직임을 잘 표현해냈다.



(a) Original data (b) Output data with our method

Fig. 9. Fire-flake effects expressed by random flame injection.

Fig. 9는 고정된 위치가 아닌 랜덤한 위치에서 주입되는 화염에 의해 표현된 불똥 효과이다. 인공지능기반으로 불똥의 생성과 움직임을 추론했음에도 불구하고 유체 방정식 기반의 시뮬레이션과 유사한 결과를 만들어냈다.

IV. Conclusions

본 논문에서는 인공신경망 기반으로 불똥의 생성과 움직임, 그리고 삭제를 제어하는 새로운 인공지능 기반 프레임워크를 제안했다. 우리의 기법은 비선형적이며 불규칙한 움직임을 가지는 불똥의 움직임을 안정적이고 사실적으로 표현하였다. 우리의 방법은 불똥 입자 외에도

격자-입자 방식을 하이브리드 구조에서 다양한 재질을 표현할 수 있을 것으로 기대한다. 향후, 우리는 본 프레임워크를 거품입자에 적용하는 연구를 진행할 것이며, 더욱더 사실적인 표현을 위해 브라운 운동 기반의 네트워크 학습 기법을 디자인할 계획이다.

REFERENCES

[1] Takahashi, Tsunemi, et al. "Realistic animation of fluid with splash and foam." Computer Graphics Forum. vol. 22. no. 3. Oxford, UK: Blackwell Publishing, 2003.

[2] Kim, Jong-Hyun, and Jung Lee. "Fire sprite animation using fire-flake texture and artificial motion blur." IEEE Access, vol. 7, pp. 110002-110011, 2019.

[3] Kim, TaeHyeong, et al. "Visual simulation of fire-flakes synchronized with flame." The Visual Computer, pp. 1029-1038, 2017.

[4] Stam, Jos. "Stable fluids." Proceedings of the 26th annual conference on Computer graphics and interactive techniques. 1999.