

불뚱 입자 생성을 효율적으로 학습 표현하기 위한 선형 회귀

김수미^o, 김동희*, 김종현*
^o강남대학교 소프트웨어융합학부,
^{*}강남대학교 소프트웨어융합학부
e-mail: jonghyunkim@kangnam.ac.kr

Efficient Learning Representation of Fire-Flake Particle Generation with Linear Regression

Sumi Kim^o, Donghui Kim*, Jong-Hyun Kim*
^oSchool of Software Application, Kangnam University,
^{*}School of Software Application, Kangnam University

● 요약 ●

본 논문에서는 간단한 선형 회귀를 이용하여 복잡한 불 시뮬레이션(Fire simulation)에서 표현되는 불뚱 입자(Fire-flake particle)를 효율적으로 생성할 수 있는 방법을 제안한다. 일반적으로 불 시뮬레이션에서 불꽃은 격자기반으로 표현하고, 불뚱은 입자로 표현하는 입자-격자 하이브리드 시스템을 이용한다. 이러한 방식을 그대로 인공신경망에 사용을 하게 되면 상대적으로 메모리가 많이 필요하며 계산양도 증가된다. 특히, 불뚱 입자가 존재하지 않는 부분에서도 격자 공간에 대한 메모리를 할당해야 되기 때문에 고해상도 불뚱 효과를 학습하는데 있어서 많은 문제점을 가지고 있다. 이러한 문제를 완화하기 위해 본 논문에서는 격자 단위가 아닌, 입자 단위로 불뚱 입자 생성을 학습할 수 있는 방법을 제안한다.

키워드: 유체 시뮬레이션(Fluid simulation), 화염(Flame), 선형 회귀(Linear regression), 불뚱 입자(Fire-flake particle)

I. Introduction

유체 시뮬레이션은 게임 혼합현실 메타버스 등 물리기반 혼합현실을 사실성으로 표현하기 위해 활용되는 대표적인 기술이다. 최근에는 인공지능을 통해 계산량이 큰 유체 시뮬레이션을 학습하고 표현하려는 기법들이 꾸준히 제시되고 있다. 하지만, 기반유체(Underlying fluids)의 움직임에 의해 표현되는 2차 효과(Secondary effects)는 더욱더 학습하기 어렵고, 계산양도 커진다. 이런 범주에 속하는 것들은 공기방울, 스플래쉬, 거품, 불뚱 효과 등이 여기에 표현된다. 기반유체에 포함되는 물과 불의 움직임은 일반적으로 오일러리안(Eulerian) 접근법으로 근사되는 격자로 표현된다. 하지만 격자 구조로 네트워크 학습하기에는 메모리가 많이 필요하기 때문에 고해상도 테스트 결과를 만들 수 없다. 본 논문에서는 2차 효과를 입자 단위로 학습시킬 수 있는 새로운 방법을 제시하며, 그 예로 불뚱 입자 생성을 학습 표현하고자 한다.

II. The Proposed Scheme

불뚱 입자를 분류하는 네트워크에서 모델로 본 논문에서는 선형 회귀 기반의 간단한 인공신경망을 사용한다. 특징 벡터 $X = \{x_1, x_2, \dots, x_N\}$ 와 불뚱 입자에 대한 분류 값 $L = \{l_1, l_2, \dots, l_N\}$ 로 구성된 데이터 집합을 이용하여 네트워크 모델을 훈련한다. 이것들은 해당 위치($\{p_1, p_2, \dots, p_N\}$)에서 전처리 과정을 통해 생성된다. 이 모델은 특징 벡터 x_i 가 l_i 로 표시된 클래스에 있을 확률 P_s 가 최대치가 되도록 하는 게 목표이다. P_s 가 따르는 확률 분포 함수 y_s 를 고려하여 주어진 데이터에서 함수 y_s 를 근사화한다. 확률 분포 $y_s(x_i, w_s)$ 는 가중치 w_s 로 표현되는 목적 함수이다. P_s 를 y_s 로 표현하면 다음과 같다 (수식 1 참조).

$$P_s(l_i | x_i) \sim P(l_i | y_s(x_i, w_s)) \quad (1)$$

위 확률을 최대화하고자 다음과 같은 우도 함수를 산출하고, 본 논문에서는 이것을 최대화하기 위해 분류 네트워크 손실에 대해 잘 정립된 소프트맥스(Softmax)를 사용한다 (수식 2 참조).

$$L_d(L | X) = \prod_{i=1}^N P(l_i | y_s(x_i, w_s)) \quad (2)$$

위 수식은 인공신경망을 사용하여 각 레이어에 여러 노드가 포함된 다층 레이어 네트워크로 모델링된다. 이러한 네트워크는 연결된 노드가 있는 레이어로 구성되며, 레이어 L 의 출력 벡터 y_L 은 일반적으로 다음과 같이 계산된다 (수식 3 참조).

$$y_L = \Phi(w_L y_{L-1} + b_L) \quad (3)$$

여기서 Φ 는 각 구성 요소에 적용되는 활성화 함수이며, w_L 은 레이어의 가중치 행렬이고, b_L 은 레이어의 바이어스 벡터이다. 입력 벡터 x 는 물방울 입자의 위치와 가속도, 그리고 밀도, 압력, 점도 9개의 값을 갖는다. 각 레이어의 모든 출력의 활성화 함수는 탄젠트 함수를 사용한다. 손실함수에는 소프트맥스 크로스 엔트로피를 사용하였고, 이것을 사용하기 위해 Y 는 원핫(One-Hot) 인코딩을 사용하였다. 네트워크 훈련에는 Adam 옵티마이저 사용하였고 훈련을 위한 학습률은 10^{-4} 로 설정하였으며, 네트워크 모델 형태는 다음과 같다 (Fig. 1 참조).

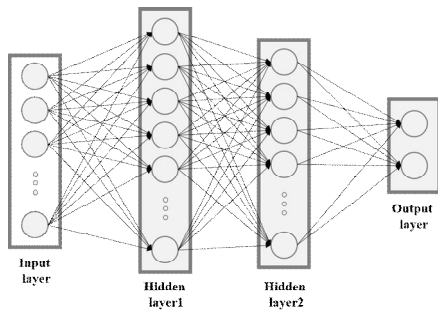


Fig. 1. Fire-flake generator network.

네트워크 훈련의 필요한 데이터는 실제 불뿔 입자를 시뮬레이션 하여 추출하였다[1]. 이 네트워크를 통해서 실제 불뿔 입자 생성 여부와 예측 불뿔 입자 생성 여부의 일치성을 확률로 추정하였고, 모델의 성능은 약 80%의 정확도를 보여주었다 (Fig. 2 참조).

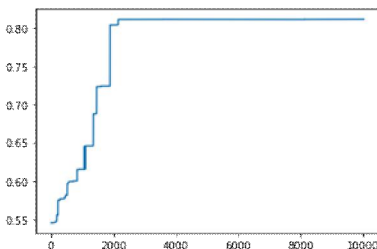


Fig. 2. Accuracy evaluation.

III. Result

본 논문에서는 기존의 불뿔 데이터를 이용해서 훈련된 모델을 테스트했다. Fig 3은 우리의 방법을 통해 생성된 불뿔 입자를 보여주는 결과이며, 입자의 이류는 이전 기법을 통해 계산하였다. 본 논문에서 제안하는 선형 회귀 방법은 시뮬레이션을 통해 만들어진 결과와 거의 유사한 형태를 만들어 냈다. 이러한 프레임워크는 복잡한 수치해석이나 알고리즘을 모르는 사용자도 쉽게 불뿔 입자를 생성해낼 수 있다는 장점을 보여주고 있다.



Fig. 3. Fire-flake effects represented by moving flames.

IV. Conclusions

본 연구에서는 불뿔 입자 생성을 효율적으로 학습할 수 있는 선형 회귀 기법을 제안했다. 향후, 카오스 영역에서 표현되는 불뿔 입자의 난류까지 학습할 수 있는 방법에 대해서 확장 연구할 계획이다.

REFERENCES

[1] Kim, TaeHyeong, Eunki Hong, Jaeho Im, Dohyeon Yang, Youngbin Kim, and Chang-Hun Kim. "Visual simulation of fire-flakes synchronized with flame." *The Visual Computer* 33, no. 6 (2017): 1029-1038.