

# 영상에서 화염의 이류를 이용한 움직임 예측

유수빈<sup>o</sup>, 김종현<sup>\*</sup>

<sup>o</sup>강남대학교 소프트웨어융합학부,

<sup>\*</sup>인하대학교 소프트웨어융합대학(디자인테크놀로지학과)

e-mail: jonghyunkim@inha.ac.kr<sup>\*</sup>

## Motion Prediction with Flame Advection from Video

Subin Yu<sup>o</sup>, Jong-Hyun Kim<sup>\*</sup>

<sup>o</sup>School of Software Application, Kangnam University,

<sup>\*</sup>College of Software and Convergence (Dept. of Design Technology), Inha University

### ● 요약 ●

기존의 불 영역 검출을 위한 이미지 전처리 기법은 임계값에 영향을 많이 받기 때문에 사진의 색상 분포에 의존하는 결과가 나온다. 이를 보완하고자 본 논문에서는 화염이 가지고 있는 특유의 비선형적이고 불규칙한 움직임을 추출할 수 있는 방법을 제안한다. 유체 시뮬레이션에서 활용되는 준 라그랑지안(Semi-Lagrangian) 활용하여 유동을 분석하고 이를 기반으로 영상에서 화염의 움직임을 추출한다.

**키워드:** 유체 시뮬레이션(Fluid simulation), 모션 벡터(Motion vector), 화염(Flame), 준 라그랑지안(Semi-Lagrangian)

## I. Introduction

영상에서 화염을 인식하기 위해 색 범위는 일반적으로 사용되는 방법 중 하나다. 실생활에서 화재와 불은 주로 빨간색, 주황색, 노란색과 같은 색으로 특징지어지는데, 이러한 색상을 활용하여 불의 존재를 감지하고 분류할 수 있다. 색 범위 기반의 이미지 인식은 컬러 스페이스에서 특정 범위의 픽셀을 필터링하거나 색상 분포를 분석하여 불의 색상을 식별한다. 색 범위를 활용한 화염 인식은 빠르고 효과적이지만, 다양한 환경에서의 일반화하기 어려우며, 정확도 측면에서 제한이 있을 수 있다. 따라서 정확성을 높이기 위해 다양한 방법과 기술을 혼합하여 사용하는 것이 일반적이다. 본 논문에서는 정확한 인식을 위해 불꽃의 움직임을 이용해 학습하여 화염을 추출하고자 하며, 이 과정에서 중요한 화염의 움직임을 추출할 수 있는 방법을 제시한다.

링을 수행할 수 있다[1].

불은 일반적으로 상승하는 운동을 하며, 이는 불이 가열되고 가벼운 공기와 함께 상승하는 현상으로, 뜨거운 공기가 위로 상승함에 따라 불은 위로 향한다. 또한 불은 공기의 흐름과 열의 변동으로 인해 흔들리고 굴절하는 모습을 보이기도 한다. 또한 불은 불안정하고 비선형적이며, 불규칙적인 움직임을 띤다. 이는 불의 열과 환경 조건에 따라 변할 수 있으며, 불의 불안정성을 시각적으로 나타낸다.

## 2. Flame Motion

컨투어(Contour)란 동일한 색 또는 동일한 강도를 가지고 있는 영역의 경계선 정보다. 물체의 윤곽선, 외형을 파악하는데 사용된다. OpenCV의 contours 기능은 흑백이미지 또는 이진화된 이미지만 적용할 수 있다. 따라서 전처리 기법으로 화재이미지에서 불 영역 마스크를 생성하여 contour추출을 진행하였다[2] (Fig. 1 참조).

## II. The Proposed Scheme

### 1. Characteristics of the Flame

불꽃은 종종 교차하거나 갈라지는 형태를 띤다. 불꽃의 입자와 열기류의 상호작용으로 인해 형성되는 이러한 현상은 불꽃의 움직임을 풍부하고 다양하게 만들어준다. 사진상에서 불꽃은 날카로운 질감을 가지고 있어 모서리가 많이 생긴다는 특징이 있다. 따라서 코너가 집중적으로 발생하는 지역에서는 화염일 가능성이 높고, 정밀한 필터

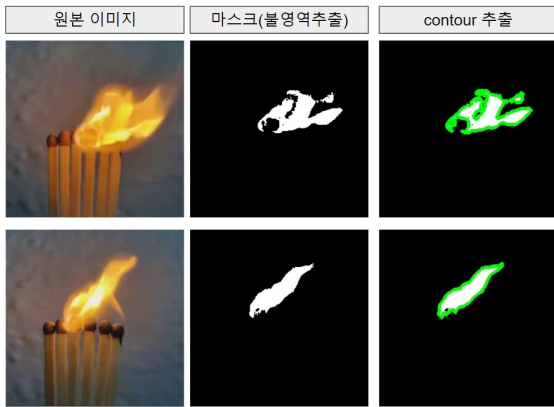


Fig. 1. Extracted contour.

이 전에 추출했던 contour를 통해 모션벡터를 계산하였다. calcOpticalFlowFarneback 함수를 사용하여 차이 이미지에서 모션 벡터를 계산하고, 계산된 차이 이미지에 대해 모션 벡터를 계산하여 화살표로 표시하는 기능을 수행한다 (Fig. 2 참조).

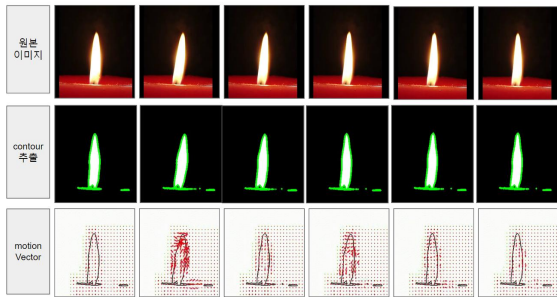


Fig. 2. Some motion vectors extracted from video frames.

이전 프레임에서의 벡터와 유사한 방향을 갖는 현재 프레임의 벡터는 불이 같은 방향으로 계속 이동할 것으로 예측할 수 있다. 예를 들어, 벡터의 각 성분(dx, dy)이 양수인 경우 불이 오른쪽 위로 이동하고, 음수인 경우 왼쪽 아래로 이동하는 것으로 해석하거나 벡터의 크기가 증가하거나 변화율이 증가하는 경우, 불의 확산 또는 이동 속도가 증가할 수 있다고 추론할 수 있다.

### 3. Interpolation

앞에서 계산한 모션벡터는 영상의 프레임에서 선형적으로 빼서 계산하기 때문에 불이라는 유체의 특성을 고려하지 않는다. 따라서 OpenCV에서 제공하는 오픈킵 플로우를 이용해 예측하는 불의 움직임은 정확하지 않을 수 있다는 문제점이 있다. 컴퓨터 그래픽스에서 불은 종종 물과 유사한 시각적 효과를 표현하기 위해 사용되는데, 이는 불이 물과 같은 흐름이나 굴절 반사, 그림자 등의 시각적 특성을 갖고 있기 때문이다. 불을 시뮬레이션하기 위한 그래픽스 기술은 유체 시뮬레이션 기법을 사용하여 구현될 수 있다. 유체 시뮬레이션은 액체나 기체와 같은 유체의 행동과 특성을 모사하는 기술로, 물, 연기, 불 등을 실제와 비슷하게 시각화하는 데 사용된다. 이러한 물과 불의 대류항 문제는 준 라그랑지안 기법으로 풀 수 있다[3].

(Fig. 3 참고)

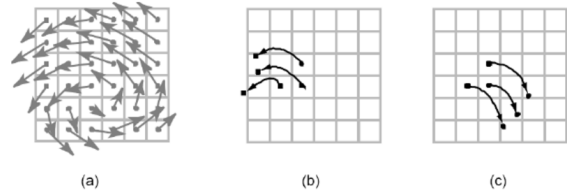


Fig. 3. Semi-Lagrangian advection.

준 라그랑지안을 이용하여 모션벡터의 보간을 수행한 결과이다 (Fig. 4참조).

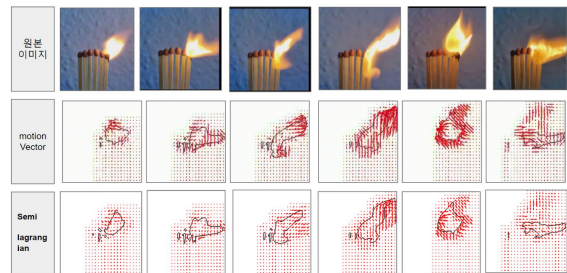


Fig. 4. Semi-Lagrangian interpolation vs. no interpolation.

## III. Result

Fig 5.에서 파란색 벡터는 준 라그랑지안을 이용하여 보간한 모션벡터를 표시한다. 또한 no interpolation의 빨간색 벡터는 보간을 따로 수행하지않고 계산한 모션벡터이며, VectorDifference는 interpolation과 no interpolation의 벡터간의 차이를 노란색으로 표시하였다.

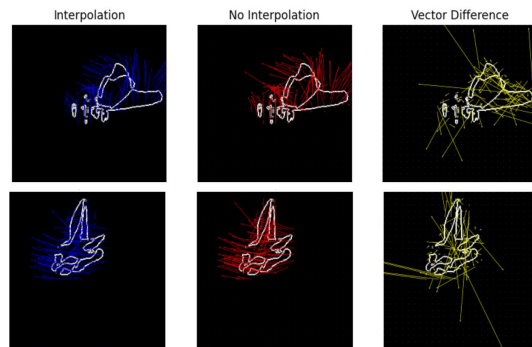


Fig. 5. Comparison results.

## IV. Conclusions

불꽃의 움직임으로 불 영역을 추출할 경우 색상 이외의 다른 특징들을 포착하고 인식에 활용할 수 있으므로, 보다 정확한 불 인식이 가능하다. 색 범위로 인식하는 경우에는 조명 조건이나 배경 등의

변동이 결과에 영향을 줄 수 있지만, 움직임 학습하는 방법은 이러한 영향을 상대적으로 줄일 수 있으므로, 불꽃의 움직임을 학습하는 방법은 색상 정보에 의존하지 않기 때문에 다양한 환경에서의 불 인식에 더 강건한 결과를 제공할 수 있을 것이라 예상된다.

불꽃의 움직임을 학습하여 불을 인식하는 방법은 더 정확하고 강건한 결과를 제공하며, 다양한 환경에서의 일반화에 유리한 장점을 가진다. 또한, 다양한 특징과 시간적인 정보를 활용함으로써 더 풍부한 불 인식이 가능하게 될 것이라 기대한다. 향후, 본 논문의 방법대로 딥러닝 모델을 이용하여 불꽃의 움직임을 통해 불 영역 추출을 진행할 계획이다.

## REFERENCES

- [1] J. Ryu and D. Kwak, "A study on a complex flame and smoke detection method using computer vision detection and convolutional neural network," *Fire*, Vol. 5, No. 4, pp. 108, 2022.
- [2] EungNam Kim, MyungJin Choi, Sun-Jeong Kim, Chang-Hun Kim, "Image-based fire area segmentation method by removing the smoke area from the fire scene videos", *Journal of the Korea Computer Graphics Society*, Vol. 28, No. 4, pp. 23-30, 2022.
- [3] Sejong Jeon and Oh-young Song, "A conservative USCIP simulation method for shallow water", *Korea Computer Graphics Society*, Vol. 25, No. 5, pp. 21-30, 2019.