

# 스마트 거울기반 의상 가상착의와 상의 내어입기

조재현\*, 문남미\*

\*호서대학교 컴퓨터 소프트웨어학과

e-mail : jaehyeon99@naver.com

## Virtual dress up and tuck in Top on Smart Mirror

Jae-Hyeon Cho\*, Nam-Mee Moon\*\*

\*Dept of Computer Software, Hoseo University

### 요 약

‘스마트’라는 단어가 대중화가 되면서 전자기기 뿐만 아니라 거울에도 스마트가 붙게 되었다. 하지만 스마트 거울에 다양한 기능이 추가되면서 ‘꾸미는 것을 돕는다’라는 거울 본연의 기능이 부가적인 요소가 되는 경우가 있거나 매장 내에 모델링이 끝난 옷을 보여주지만 집에 있는 자신의 옷을 보는 것이 부족한 경우가 많다[1]. 본 논문에서는 OpenCV를 활용하여 옷을 여러벌 갈아입으면서 코디를 할 때 번거로움을 줄이고자 거울 앞에서 찍으면 프레임과 전경 추출 알고리즘을 사용하여 사용자의 옷을 추출하고 추출된 옷의 정확도를 위해 보정작업을 추가한다. 윤곽선의 노이즈를 줄이기 위해 Morphology 필터링을 사용하고 Clahe 히스토그램 균일화를 통해 옷의 선명도를 높였다. 추가적으로 가상으로 띄워주는 기능과 옷을 보여줄 때 HSV모델의 특성을 활용하여 채도나 명도의 변화의 상관없이 색을 추출하여 상의와 하의를 분리하여 상의를 내어입는 기능도 선택할 수 있게 구현 하였다.

### 1. 서론

거울의 본연의 기능은 나의 모습을 조금 더 정확하게 꾸밀 수 있도록 보여주는 것이고 거울에 스마트란 단어가 붙게 된다면 본연의 기능을 더 극대화하고 더 쉽게 할 수 있도록 해주는 것이다. 옷을 입은 자신의 모습을 비교하면서 스타일링을 할 수 있는 스마트 거울은 많지만 집에 있는 자신의 옷들을 비교하는 기능은 부족했다[1].

본 논문은 자신의 가상 옷 이미지 추출과 상의 내어입기를 구현하였다. 기존의 스마트 거울은 이미 모델링 된 이미지를 보여주는 방식으로 개인이 사용하기에는 부적합했지만 본 기술을 이용하여 옷가게 뿐만 아니라 가정에서도 옷의 스타일링을 돕는 스마트 미러가 더 상용화 될 수 있을 것이다[2].

### 2. 본론

#### 2-1 개요

본 논문은 OpenCV를 사용하여 옷을 추출하였다. 먼저 사용자가 추출할 옷의 종류를 선택하게 하면 그에 맞는 프레임을 보여준다. 프레임에 맞는 크기만큼 슬라이싱되고 전경추출 알고리즘을 사용하여 옷을 추출하고 보정작업을 거친다. 추출된 이미지에서 히스토그램을 이용하여 옷의 색과 밝기를 얻는다. 이렇게 추출된 옷은 사용자에

게 리스트로 보여지게 되고 이미지는 사용자가 언제든지 입어 볼 수 있다.

상의 스타일링은 히스토그램을 이용하여 사용자의 옷과 가상 이미지가 겹치는 부분의 색을 추출하여 색에 해당하는 이미지를 따로 분리 시킨 후 상의 하의 순서대로 겹치면 상의를 내어입는 효과가 나타난다. 옷 추출과 착의에 대한 구성도는 (그림 1)과 같다.



(그림 1) 구성도

2-2 옷 저장하기

```
def createFrame
{
    // findcontours = 윤곽선 추출
    f = findcontours(I)
    f.size = I.size * x
    return f
}
```

(그림 2) 프레임 생성 알고리즘

(그림 2)는 프레임을 생성하는 알고리즘이다. 옷 추출의 정확도를 높이기 위하여 프레임( $f$ ) 작업을 하였다. 먼저 옷의 종류에 맞는 사진( $I$ )을 수집한다. 이 사진의 윤곽선을 추출하고 옷 종류(반팔, 반바지, 긴팔, 긴바지)에 맞는 크기( $x$ )와 위치를 조절을 하여 배열 형태로 저장하면 프레임 작업이 끝난다.

def extractionCostume

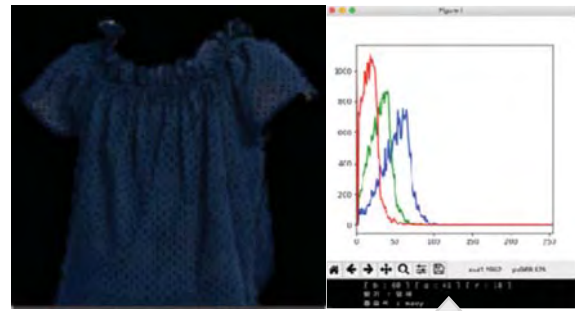
```
{
    i.size = f.size
    // grabcut = 전경 추출
    c = grabcut(i)
    // calcHist = 히스토그램 계산
    HSV = max(calcHist(c))
    if HSV.v < 40
        // mophology = 이미지 필터링
        then lmg = mophology(c, opening)
        else lmg = mophology(c, closing)
    end if
    // clahe = 히스토그램 균일화
    c = clahe(c)
    HSV = max(calcHist(c))
    return c, HSV
}
```

(그림 3) 옷 데이터 추출 알고리즘

사용자가 추출할 옷의 종류를 선택하게 하면 그에 맞는 프레임을 보여준다. (그림 3)을 보면 프레임대로 이미지를 슬라이싱하고 전경추출 알고리즘[3][4]을 사용하면 이미지가 추출되고 여기에 Mophology 필터링[3][4] 중 closing 연산을 사용하여 윤곽선의 노이즈를 제거하는데 검은색 계열의 옷일 경우 옷의 정확도를 많이 낮추게 되어 opening 연산을 사용한다. 그리고 히스토그램 균일화 연산 중에서 극단적인 값에 의한 에러를 줄이기 위해 Clahe(Contrast Limited Adaptive Histogram Equalization) [3][4] 연산을 하여 노이즈는 없애고 선명도는 올렸다[5].

이렇게 추출된 옷( $c$ )은 추천 시스템을 적용하기 위해 색상과 밝기가 같이 저장되는데 히스토그램을 이용하여 이미지에서 가장 많이 분포한 RGB 값을 얻고[6] 색조( $h$ ), 채도( $s$ ), 명도( $v$ )를 얻을 수 있는 HSV 모델로 변경한다. 본 연구에서는 색을 양자화 하여 무지개색, 흰색, 회색, 검정색으로 10가지 색으로 분류하였다[7]. 각 각 해당하는 HSV 범위 값을 지정하여 색의 분포도와 요 색의 순서도 알 수 있게 하였으며[8], 주요색의 명도를 이용하여 옷의

밝기 또한 알 수 있다(그림 4).



(그림 4) 색 추출

2-3 상의 내어입기

def tuckInTop

```
{
    f.height = 20
    n = m.copy()
    n.size = f.size
    HSV = calcHist(n)
    h = HSV.h-5 ≤ h ≤ HSV.h+5
    s = HSV.s-50 ≤ s ≤ HSV.s+50
    v = HSV.v-50 ≤ v ≤ HSV.v+50
    if m.color in [ h, s, v ]
        then t = m
    end if
    m += i
    m += t
    return m
}
```

(그림 5) 상의 내어입기 알고리즘

(그림 5)는 가상 하의를 안에 입는 스타일링을 예로들어 설명한 알고리즘이다. 상의 스타일할 때 가장 중요한 것은 상의와 하의를 구분하는 것이다. 가상 하의와 상의가 겹치는 부분( $n$ )을 슬라이싱하고 히스토그램을 이용하여 겹치는 부분에 대한 HSV 모델을 얻는다. 얻은 HSV 모델에서 색조는 최소한으로 변경하고 채도와 명도의 값에 범위를 지정하여 색의 양자화를 하여 영상( $m$ )에서 옷을 추출한다. 이렇게 추출된 옷은 색조를 중심으로 추출되기 때문에 옷이 겹쳐서 그림자가 생긴 부분도 추출할 수 있게 된다. 마지막으로 영상에 비트 연산을 이용하여 자신이 원하는 스타일에 따라서 순서대로 겹쳐서 상의( $t$ )를 내어입기를 구현하였다.

3. 실험 및 결과분석

구현은 첫 번째로 프레임을 생성하고, 옷을 저장한 후 저장된 이미지를 사용자 설정에 맞게 상의 스타일링을 하였다.

먼저 프레임 생성의 결과(그림 6)는 반팔에 해당하는 윤곽선이 잘 나타났다. 하지만 옷의 모양은 다양하므로 많은 종류의 이미지를 수집하여 다양한 프레임을 준비해서 종류 별로 정확도를 높일 수 있다.



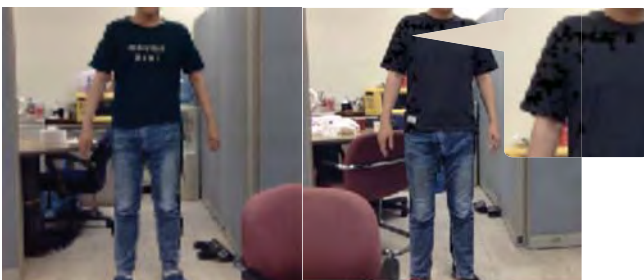
(그림 6) 프레임 생성

그 다음으로 (그림 7)은 옷을 저장하는 과정에선 검흰 스프라이트 티, 검은색 바지, 회색 배경으로 실험한 모습이다. 이 과정에서 정확도에 많은 영향을 주는 것은 배경, 상의와 하의의 색 비교가 얼마나 잘 되는가, 사용자가 프레임에 얼마나 정확하게 맞춰서 위치하는가이다. 프레임보다 넘치게 위치하면 배경과 비슷한 테두리 부분이 함께 잘리고, 프레임보다 너무 작다면 배경이 덜 잘리는 현상을 보였다. (B)를 보면 어깨쪽 테두리 부분의 인식률이 부족하다는 것을 알 수 있다. 이 부분을 보완하기 위해서 밝은 색 계통은 Closing 연산을, 어두운 색 계통은 Opening 연산에 대한 커널 매트릭스를 세분화하여 보다 더 정확한 옷 이미지를 추출할 수 있다.



(A) 사진 (B) 추출된 이미지

(그림 7) 오른쪽이 최종적으로 저장되는 이미지



(a) 하의가 위에 있는 모습 (b) 상의가 위에 있는 모습

(그림 8) 오른쪽이 상의를 내어입는 모습

상의를 내어입는 부분(그림 8)은 사용자가 매칭을 쉽게 할 수 있도록 이미지 작업이 아닌 영상 작업으로 하였다. (b)를 보면 상의와 하의의 경계 부분에 대한 색을 추출하기 때문에 상의와 하의는 명확한 구분이 가능하지만 어깨 부분의 인식률이 부족하다는 것을 알 수 있다. 이 부분을 보완하기 위해 히스토그램 균일화와 색조 범위 조정을 한다면 어깨 부분의 정확도를 높일 수 있다.

#### 4. 결론

본 논문은 스마트 거울의 활용 범위를 가정까지 넓히기 위해 기존의 이미지 처리방식에 변화를 주었다. 미리 모델링을 하는 기존의 방법보다 정확도 면에서 조금 부족하지만 사용자가 옷을 골라서 입어보는 것에 어색함이 없었다.

하지만 2D라는 한계점이 존재한다. 향후에는 한계점을 극복하고자 게임 모델링에도 쓰이는 유니티 3D를 이용하여 사람이 스마트 거울 앞에서 한바퀴 돌면 3D모델링을 하여 옆 모습이나 뒷 모습까지 보여줄 수 있도록 할 것이다. 또한 비슷한 색의 옷을 입고 있을 때 정확도가 현저히 낮아지는 문제가 있는데 이미지 학습을 통해 점차 정확도를 높여 갈 것이다.

이 발표논문은 2017년 대한민국 교육부와 한국연구재단의 중견연구자지원사업의 지원을 받아 수행된 연구(한국연구재단-2017년-2017008886)임.

#### 참고문헌

- [1] 최병호 (2012). 스마트 미러의 방향. 정보과학회지, 30(4), 25-28.
- [2] 지능형 패션 코디네이션 시스템에서 유사의류 추천방법 Journal of Korea Multimedia Society Vol. 12. No. 5. May 2009
- [3] OpenCV-Python Study documentation, <http://opencv-python.readthedocs.io/en/latest/>
- [4] Python Numpy Tutorial, <http://aikorea.org/cs231n/python-numpy-tutorial/>
- [5] Image Processing Design Flow for Virtual Fitting Room Applications used in Mobile Devices, 978-1-4673-0818-2/12/\$31.00 ©2012 IEEE
- [6] 김병우, 이가혜, 이승형, 이승규 (2017). OpenCV를 이용한 CBIR기반 의류 검색 시스템. 한국정보과학회 학술 발표논문집, 1845-1847.
- [7] 황인성, 조법근, 전승우, 최윤식 (2014). 의류 검색용 회전 및 스케일 불변 이미지 분류 및 검색 기술. 방송공학 회논문지, 19(3), 396-404.
- [8] 강석원, 이순이, 박지웅 (2009). OpenCV를 사용한 화재 영상 처리. 한국콘텐츠학회 종합학술대회 논문집, 7(1), 79-82.