

# 제스처 기반 사용자 인터페이스를 위한 색상 마커 인식 알고리즘

이두희, 김 윤

강원대학교 컴퓨터정보통신공학과

e-mail: [getback9@kangwon.ac.kr](mailto:getback9@kangwon.ac.kr), [yooni@kangwon.ac.kr](mailto:yooni@kangwon.ac.kr)

## A Color Marker Detection Algorithm for Gesture-based User Interfaces

Doo-Hee Lee, Yoon Kim

Dept. of Computer Science and Engineering, Kangwon National University

### 요 약

고성능 단말기와 다양한 콘텐츠가 제공되면서 제스처 기반 사용자 인터페이스에 대한 관심이 높아지고 있다. 하지만 기존의 알고리즘을 사용하기 위해서는 센서 장치를 사용하거나 사용자가 부자연스러운 장비를 착용해야 하는 경우가 많다. 본 논문에서는 카메라를 통해 입력된 영상 정보만으로 사용자가 착용한 색상 마커를 실시간으로 검출하는 알고리즘을 제안한다. 본 논문이 제안하는 마커 인식 알고리즘은 색상 감지와 움직임 감지로 나뉜다. 단일 프레임에서 영상 성분 평균을 이용한 조건검사를 통하여 색상 영역을 검출한다. 다음으로, 인접한 프레임간의 평균 영상과 현재 영상과의 차를 가중치로 이용하여 배경 범위를 설정하고 이 범위를 벗어난 영역을 움직임 영역으로 검출한다. 마지막으로 색상 검출 영역과 움직임 검출 영역을 동시에 만족하고 이웃한 픽셀들도 위 조건을 동시에 만족하면 최종적으로 사용자의 마커로 인식한다. 본 논문이 제안하는 알고리즘은 영상 정보만 사용하기 때문에 사용자는 센서나 부자연스러운 장비를 착용할 필요가 없고 일조량에 따른 조도의 변화에 강건하기 때문에 효과적인 사용자 움직임 검출이 가능하다.

**Keyword :** 사용자 인터페이스(User Interface), 색상 검출(Color Detection), 움직임 검출(Motion Detection).

### I. 서론

최근 유비쿼터스 기술의 발전에 따라 사용자 인터페이스에 대한 연구가 활발히 이루어지고 있다. 사용자 인터페이스는 사용자와 시스템 사이의 의사소통이 가능하게 해주는 목적으로 만들어진 물리적, 가상적 매개체를 뜻한다. 특히 제스처 기반 사용자 인터페이스는 사용자와 미디어 콘텐츠간의 자연스러운 상호작용을 지원한다.

기존의 제스처 기반 사용자 인터페이스에는 접촉식과 비접촉식으로 나뉜다. 접촉식 제스처 기반 사용자 인터페이스는 사용자가 직접 센서나 장치를 착용하는 방식이다. 대표적인 예로는 마그네틱 센서와 자이로 센서 등을 이용한 데이터 글로브[1][2]와 모션 트랙커[3] 등이 대표적이다. 비접촉식 제스처 기반 사용자 인터페이스는 대부분 영상정보를 이용한다. 이 방식은 마커를 부착시키고 시각기술을 이용하여 추적하는 방법[3]과, LED나 적외선 등의 마커를 이용한 액티브 마커 등으로 연구되었다.[4][5] 하지만 기존의 제스처 기반 사용자 인터페이스는 고가의 장치를 사용하거나 사용자가 부자연스

러운 장비를 착용해야 한다는 단점이 있다. 또한 센서를 사용할 경우 시스템과 센서간 통신도 정의해주어야 한다.

본 논문에서는 고정된 시점으로 촬영되는 순수한 영상 정보만으로 사용자가 장착한 마커를 추적할 수 있는 알고리즘을 제안한다. 먼저 색상 검출 정보와 움직임 검출 정보를 이용하여 이 두 가지 검출을 모두 만족하는 픽셀의 좌표를 사용자 제스처로 판단한다. 색상 검출은 단일 프레임에서 RGB 성분의 평균을 구하고 각 픽셀단위로 평균값과 여러 조건을 검사하여 최종적으로 마커 여부를 판단한다. 또 움직임 검출은 인접한 프레임간의 영상 평균을 구하고 현재 영상과의 차이를 구하고 이를 배경 후보 범위를 결정하는 가중치로 이용한다. 현재 입력 영상의 픽셀 가운데, 위에서 설정된 범위를 벗어난 영역을 움직임 영역으로 판단한다. 최종적인 마커 검출 판단은 색상 검출 영역과 움직임 검출 영역에 모두 해당되며 일정 경계값만큼 이웃한 픽셀이 모두 위의 조건을 만족한 경우이다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 알고리즘을 기술하고 3장에서는 실험 결과를 통해 그 성능을 확인하며 최종적으로 4장에서 결론을 맺는다.

## II. 제안하는 알고리즘

### 1. 색상 검출 알고리즘

카메라로 촬영된 영상에서 사용자가 정의한 일정한 색상의 마커를 인식하기 위해선 조명과 조도에 강건해야 한다. 이를 위해 본 논문에서는 실제 사용자가 RGB 값이 각각 (255,0,0)과 유사한 성분의 마커를 사용하고 가정하였다. 실제로 일반적인 영상에서 RGB값의 분포가 극단적인 경우의 비율은 전체 영상에 비해 상대적으로 대단히 적고, 이러한 성질을 이용하면 단일 프레임 내에서 영상 평균을 구하고 조건 검사를 함으로써 마커의 색깔을 인식할 수 있다. 만약 마커의 색상을 G나 B로 하는 경우도 본 논문에서 제안하는 알고리즘을 응용하면 같은 원리로 적용할 수 있다.

#### 1.1 평균 성분 값 계산

카메라에서 촬영된 영상은 RGB컬러로 시스템에 전달된다. 마커 검출을 위해서는 단일 프레임에서의 RGB의 각각의 평균 성분 값을 구하며 그 공식은 (1)과 같다.

$$\begin{aligned} R_{mean} &= \frac{1}{K} \sum_{i=1}^K \sum_{j=1}^K R(x_i, y_i), \\ G_{mean} &= \frac{1}{K} \sum_{i=1}^K \sum_{j=1}^K G(x_i, y_i), \\ B_{mean} &= \frac{1}{K} \sum_{i=1}^K \sum_{j=1}^K B(x_i, y_i). \end{aligned} \quad (1)$$

여기서  $K$ 는 단일 프레임에서 모든 픽셀의 개수이며  $(x_i, y_i)$ 는 프레임에서의 픽셀 위치를 나타낸다.

#### 1.2 조건 검사

평균 성분 값과 각각의 픽셀의 성분 값을 이용하여 조건 검사를 시행한다. 조건 검사는 RGB 컬러 공간에서 마커의 붉은색 특성이 상대적으로 R값이 G와 B보다



(a) 입력 영상의 예.

월등하게 큼을 이용한다. 하지만 조명이나 조도에 따라 입력되는 마커의 RGB 수치는 다양하게 변화할 수 있으므로 영상 평균과의 비교를 수행한다. 이러한 조건식은 (2)와 같이 정의된다.

$$F(x,y) = \begin{cases} 1, & \text{if } R(x,y) > R_{mean}, \\ & G(x,y) < G_{mean}, \\ & B(x,y) < B_{mean} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

식 (2)의 조건에서 각각의 픽셀에 대해 평균 화소 값과의 비교를 수행함으로써 붉은 성분이 다른 성분보다 큼을 만족하지 검사한다.  $F(x,y)$ 는 이진영상으로 마커 후보로 판별되는 픽셀이며  $R(x,y)$ ,  $G(x,y)$ ,  $B(x,y)$ 는 화소의 R, G, B 값을 각각 나타낸다. 이와 같은 조건 검사를 수행하는 이유는 전체적으로 어둡거나 밝은 영상이 입력되었을 때 마커도 상대적으로 RGB 화소 값의 증감이 있는데 이때의 변화에도 적응하기 위함이다.

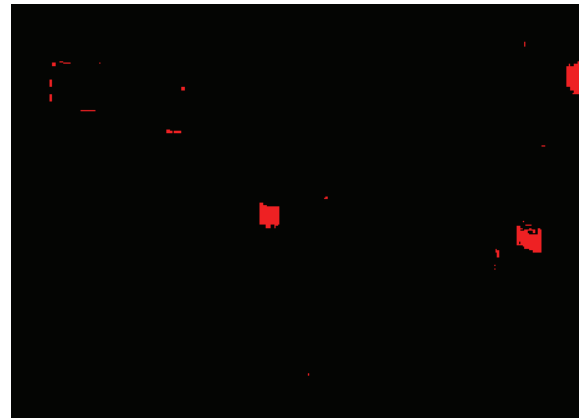
이렇게 마커 후보를 구한 후에는 마커 후보에 해당 하는 픽셀의 R성분이 G와 B보다 월등히 높은지 검사하였다. 이 또한 절대적인 경계값을 이용하지 않고 R값은 G값과 B값의 합보다도 클 것이라는 가정으로 이루어졌다. 이와 같은 조건식은 (3)과 같이 정의되고 그 예는 그림 1과 같다.

$$Marker(x,y) = \begin{cases} 1, & \text{if } R(x,y) > G(x,y) + B(x,y), \\ & F(x,y) = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

### 2. 움직임 검출 알고리즘

움직임 검출은 현재 영상까지의 누적 평균과 현재 영상과의 차를 이용한다. 배경을 갱신하기 위해서는 먼저 RGB 평균 영상을 구한다. 식 (4)은 평균 영상을 구하는 공식을 나타낸다.

$$Mean(x,y) = \frac{1}{N} \sum_{i=1}^N I_i(x,y). \quad (4)$$



(b) 결과 영상의 예

그림 1. 색상 검출 예.

Fig. 1. Example of Color Detection Image.

$N$ 은 참조할 근접 프레임의 개수를 말하며  $I_i$ 는  $i$ 번째 입력 영상을 나타낸다. 그 다음은  $Mean(x,y)$ 와 현재 영상과의 차영상을 구하고 그 차이에 제곱을 취해줌으로써 가중치를 부여한다. 또한 이렇게 구해진 가중치에  $thr1$ 만큼의 상수를 곱해줘서 그 범위를 확대하고, 최종적인 비율을 맞추기 위해 제곱근의 형태로 나타낸다.  $thr1$ 는 실험을 통해 2로 결정하여 사용하였다. 최종적으로  $K$ 장의 프레임만큼의 평균값을 구한다. 이렇게 얻어낸  $B(x,y)$ 는 평균 영상과 현재 영상을 비교할 때 얼마만큼의 차이만큼을 배경으로 인식할 것인지 범위를 결정하게 된다. 식 (5)은 이러한 과정을 나타내는 공식이다.

$$B(x,y) = \frac{1}{K} \sum_{i=1}^K \sqrt{|Mean(x,y) - I_i(x,y)|^2 \times thr1}. \quad (5)$$

식 (5)를 통해 얻어진 가중치를 이용하여 배경이 가질 수 있는 범위를 갱신한다. 평균 영상에서 가중치와 경계값의 합을 빼준 영상을  $Low(x,y)$ , 평균 영상에서 가중치와 경계값의 합을 더해진 영상을  $Upper(x,y)$ 라고 정의하며 이를 나타낸 공식은 식 (6)와 같다. 경계값인  $thr2$ 는 실험을 통해 10으로 결정하여 사용하였다.

$$Low(x,y) = Mean(x,y) - B(x,y) - thr2, \quad (6)$$

$$Upper(x,y) = Mean(x,y) + B(x,y) + thr2.$$

$$Motion(x,y) = \begin{cases} 0, & \text{if } Low(x,y) \leq I_n(x,y) < Upper(x,y) \\ 1, & \text{otherwise.} \end{cases} \quad (7)$$

영상 범위가 구해졌을 때 현재 입력영상  $I_n(x,y)$ 의 픽셀 값이 범위 내에 포함되는지 검사하는 수식은 (7)과 같고 그 예는 그림 2와 같다. 범위 내에 있는 픽셀은 1, 범위 밖의 값을 갖는 픽셀은 0으로 이진화한다. 여기서 0은 배경 영역, 1은 움직임 영역이다. 실제 실험에서는 배경이 움직임영역보다 누적되는 속도를 빠르게 하여 새롭

게 발견되는 영역에 대한 민감도를 높였다.

### 3. 최종 사용자 제스처 인식 알고리즘

최종 사용자 제스처 인식은 입력 영상 중에서 움직임 영역과 마커 영역이 모두 1에 해당하고 이 좌표의 주변 픽셀이 경계값 이상의 크기일 때 최종적인 좌표로 인식하게 된다.

$$G(x,y) = \begin{cases} 1, & \text{if } Maker(x,y) = 1, \\ & Motion(x,y) = 1. \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

$$Point(x,y) = \prod_{i=-thr3}^{thr3} \prod_{j=-thr3}^{thr3} G(i,j). \quad (9)$$

식 (3)에서 구해지는  $Maker(x,y)$ 와 식 (7)에서 구해지는  $Motion(x,y)$ 의 값이 모두 1을 만족하는 최종 마커 후보  $G(x,y)$ 를 구한다. 또  $G(x,y)$ 중에서  $thr3$ 만큼 이웃한 픽셀이 모두 식 (9)을 만족하는지 검사한다. 실험에서는  $thr3$ 은 1로 설정하여 이웃한 8방향 픽셀을 검사하였다. 최종적으로 위의 조건을 모두 만족하는  $Point(x,y)$ 를 사용자의 마커가 위치한 좌표로 인식한다.

## III. 실험 결과

실험은 영상의 각 프레임의 마커 인식 및 움직임 발생 여부를 관찰하여 해당 알고리즘이 얼마만큼의 프레임을 검출 혹은 오검출 하는지 기록하였다.

표 1. 실험에 사용한 조명의 구성 목록.

Table 1. Lighting list using in an experiment

목록	어두움	보통	밝음
RGB 평균	90~110	110~130	140~160
프레임수	1000	1000	1000



(a) 입력 영상의 예.



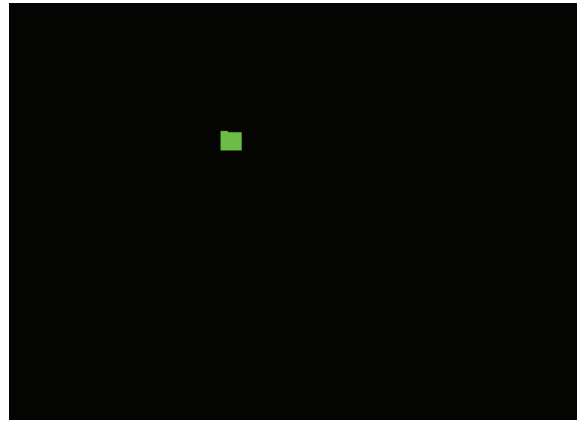
(b) 결과 영상의 예

그림 2. 움직임 검출 예.

Fig. 2. Example of Motion Detection Image.



(a) 입력 영상의 예.



(b) 결과 영상의 예

그림 3. 마커로 인식하는 영상의 예.

Fig. 3. Example of Marker Detection Image.

테스트는 320x240의 크기와 초당 15프레임을 갖는 입력 영상을 사용하였고 영상입력에는 Microsoft의 VX3000 USB Webcam을 사용하였다. 표 1은 실험 환경의 조명에 대한 정보를 나타낸다. 표 1에서 나타나듯이 전체 프레임에 대한 RGB평균값을 이용하여 조명 환경을 변화를 주었다. 실험은 각 조명마다 5회씩 실험하고 평균을 구하였다. 그림 4를 통해 실험 결과를 그래프로 확인할 수 있다. 또한 사무실과 같은 일상적인 장소를 실험 환경으로 선정하였다.

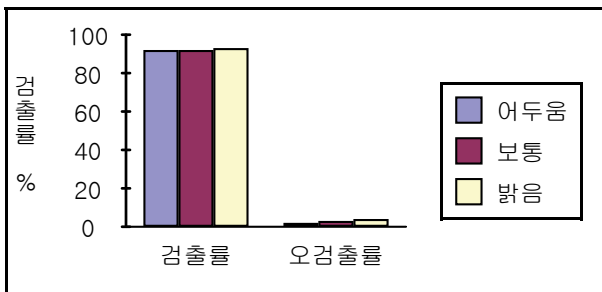


그림 4. 조명에 따른 검출 성능 비교.

Fig. 4. Performance Compare about Light.

그림 4에서 나타나듯이 조명의 밝기에 상관없이 사용자의 마커를 인식하는 검출율은 90%이상의 성능을 유지했고, 마커를 착용한 사용자가 없을 때 마커를 오검출하는 경우도 5%미만으로 검출하였다. 이는 일상적인 생활에 있어서 움직이는 사람의 옷과 마커의 색상이 비슷함으로 인한 결과이다. 같은 원리로 마커의 색상과 같아도 움직이지 않는 물체에 대해선 검출하지 않았다. 그림 3의 (a)는 입력 영상의 예이고 (b)는 최종적으로 마커로 인식되는 이진 영상의 예이다.

#### IV. 결론

본 논문에서는 RGB 컬러 모델에서의 평균 성분과

인접한 프레임의 평균 영상을 이용한 배경의 범위를 정하는 방법으로 마커와 움직임을 검출하였다. 특히 마커 감지 알고리즘은 일조량에 따른 조도의 변화에 강건하고, 움직임 검출 또한 단위 시간 동안의 변화량을 고려하기 때문에 새롭게 변화하는 구간을 효과적으로 검출한다. 위와 같은 마커와 움직임 검출 알고리즘을 통해 사용자가 장착한 마커의 움직임을 실시간으로 탐지 가능하며, 이를 통해 유비쿼터스 시대에 상응하는 다양한 사용자 인터페이스를 기대할 수 있다. 향후에는 다양한 마커를 인식하기 위한 검출 알고리즘의 개발을 추후 연구를 통해 제시하고자 한다.

#### 참고문헌

- [1] D. J. Sturman and D. Zeltzer, "A Survey of Glove-based Input," IEEE Computer Graphics and Applications, Vol. 14, No. 1, 1994, pp. 30-39.
- [2] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A Hand Gesture Interface Device," in Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface, 1987, pp. 189-192.
- [3] G. B. Guerra-Filho, "Optical Motion Capture: Theory and Implementation," Journal of Theoretical and Applied Informatics (RITA), Vol. 12, No. 2, 2005, pp. 61-89.
- [4] K. Kurihara, S. Hoshino, K. Yamane, and Y. Nakamura, "Optical Motion Capture System with Pan-Tilt Camera Tracking and Realtime Data Processing," in Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Vol. 2, 2002, pp. 1241-1248.
- [5] H. Chen, G. Qian, and J. James, "An Autonomous Dance Scoring System using Marker-based Motion Capture," in Proceedings of IEEE 7th Workshop on Multimedia Signal Processing, 2005, pp. 1-4.