마우스 포인터 제어를 위한 실시간 손 인식 알고리즘

*이동욱 *김수동 *이동석 *유지상

광운대학교

*dlehddnr86@kw.ac.kr

Hand recognition algorithm with real-time for control of mouse pointer

*Lee, Dong-Wook *Kim, Su-Dong *Lee, Dong-Seok *Yoo, Ji-Sang

KwangWoon University

요약

본 논문에서는 카메라로부터 획득된 영상을 이용하여 손을 인식하고 이를 통해 실시간으로 마우스 포인터를 제어하는 알고리즘을 제안한다. 기존의 상호 작용 멀티미디어 시스템은 펜이나 마우스등과 같은 특정 외부 입력장치들에 의존하였기 때문에 사용자에게 불편함을 주었다. 따라서 본 논문에서는 외부 입력장치가 필요 없는 손 인식 알고리즘을 이용하여 이러한 단점을 보완하였다. 제안하는 알고리즘은 카메라로부터 획득된 영상에 저주파 필터를 통과시킨 후 색 정보를 이용하여 손 영역과 배경을 분리하고, 분리된 손 영역의 중심 좌표를 이용하여 모니터 상의 마우스 포인터 좌표를 결정한다. 또한 손의 중심을 원점으로 하는 가변적인 크기의 원과 손가락과의 교차점을 이용하여 손가락의 개수를 계산하고, 이를 통해 마우스의 특정 동작을 결정한다. 제안한 알고리즘은 90% 이상의 높은 손 인식률을 나타내었으며, 스테레오 카메라를 이용한 3차원 실시간 상호작용 멀티미디어 시스템에도 적용이 가능할 것으로 기대된다.

1. 서론

손은 인간이 많이 사용하는 신체 부위중 하나로 편리한 인터페이스 도구로 활용 된다. 손의 동작을 인식하고, 인식된 정보를 사용하면 실시간 휴먼 인터페이스를 구현 할 수 있다. 이런 점에서 손 동작인식에 관해 많은 연구가 이루진다. 하지만 손의 움직임이 빠르고 복잡하기 때문에 실시간으로 정확하게 인식하기가 힘들다. 따라서 이를 실용화하는데 많은 어려움이 있다.

손동작 인식 방법은 장비를 이용한 방법과 비전을 이용한 방법이 있다. 장비를 이용한 방법은 사용자가 직접 장비를 몸에 착용하고, 장비에 부착된 센서를 이용하여 손동작을 인식하거나, 손에 마킹을 하여 이 마킹 점을 추출함으로써 손동작을 인식하는 것이다. 이 방법은 정확하게 손동작을 인식 할 수 있으나, 장비를 착용하거나 손에 마킹을 해야 하므로 사용자에게 불편함을 주는 단점이 있다. 장비 없이 비전을 이용하는 방법은 크게 모델 기반 방법과 윤곽선 기반 방법이 있다. 모델 기반 방법은 인식하려는 손 모양을 3차원적으로 모델링한 후에 입력되는 영상과 기존의 모델링된 영상들을 비교하여 손 모양을 인식한다. 이 방법은 정확하고 다양한 손동작을 인식할 수 있으나 실시간 처리 응용에 적용시키기 어렵다. 윤곽선 기반 방법은 손의 형태상의 특징을 추출하여 손 모양을 인식하는 방법이다. 이 방법은 카메라로 획득된 2차원 영상을 사용한다. 연산이 복잡하지 않기 때문에 주로 실시간 시스템에서 많이 사용된다. 하지만 손가락 움직임, 폐쇄영역, 손의 회전등으로 인한 형태상의 변화에 민감하다.[1]

본 논문에서는 윤곽선 기반의 손 인식 및 추적 방법을 사용하여 실시간 휴먼 인터페이스 시스템에 대해 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 방법에 대한 시스템 구성도와 각 부분에 대한 세부동작에 대해 설명한다. 3장에서는 실험 조건과 결과에 대해 설명하고, 마지막장에서는 결론을 내린다.

2. 본론

2.1 시스템 구성

제안하는 알고리즘은 그림 1 과 같다. 카메라로부터 획득된 영상에서 피부색의 YCbCr 모델의 색 분포를 이용하여 손 영역을 추출하고, 추출된 영역의 중심 좌표를 구한다.

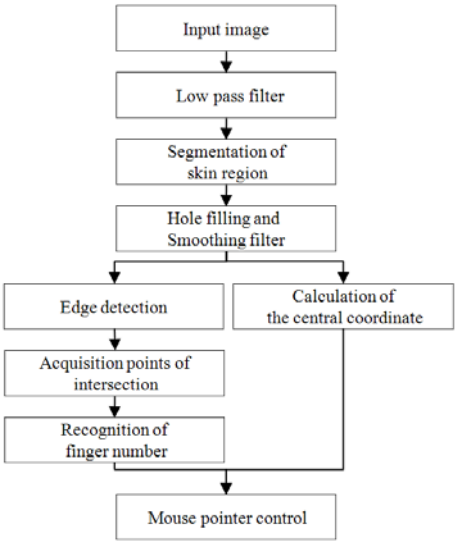


그림 1. 시스템 구성도

이를 이용하여 현재 화면에 마우스 포인터의 위치를 결정한다. 손가락의 개수를 세기위해서 추출된 영역의 경계 영상과 원과의 교점의 개수를 이용한다. 이렇게 구한 손가락 개수는 마우스 포인터의 특정 동작을 결정한다.

2.2 피부색을 이용한 손 영역 추출

카메라로부터 입력된 영상에서 관심 영역을 추출하는 과정은 Y, Cb,Cr 정보의 값을 사용한다. 사람의 피부색 분포는 자연계의 다른 물체와 구분할 수 있는 특징을 가지고 있다. 사람마다 다소 차이는 존재하지만 YCbCr 평면에서의 피부색 분포는 매우 유사하면서도 협소한 영역을 차지하고 있고, 단지 밝기의 차이만이 존재하므로 Cb, Cr성분을 이용하면 쉽게 피부 영역을 추출할 수 있다. 또한 Cb, Cr정보는 RGB 정보에 비해 조명변화에 덜 민감하기 때문에 조명 조건의 변화에도 안정적인 결과를 보인다. 본 논문에서는 YCbCr 칼라 포맷을 이용하여 배경과 손 영역을 분리해 낸다.

카메라로부터 입력받은 영상은 저주파 필터를 통과한 뒤 식 (1)을 사용하여 배경영역과 손 영역을 분리 한다.

$$\begin{aligned} 30 \le Y \le 100 \\ 100 \le Cb \le 129 \\ 128 \le Cr \le 137 \end{aligned} \qquad (1)$$

추출된 손 영역은 위 그림 2. (b)처럼 정확하지 않고 노이즈가 포함된다. 그래서 filtering을 하여 이 점을 보완한다.

(a) (b)

그림 2. color filter 과정

(a) 카메라로부터 획득된 영상 (b) color filter 통과후 영상

2.3 필터링

필터링을 하여 산술적인 식으로 걸러내지 못한 손 영역을 찾아내고, 손 영역 이외에 영역에서 생성된 노이즈를 제거한다.

hole-filling은 그림 3에서와 같이 영상을 검색하면서 값이 255이고, 같은 수평 또는 수직 방향에 놓인 임의의 두 개의 픽셀 사이 간격이 임계간격 보다 작으면 그 픽셀들 사이에 값을 모두 255로 채운다.

다음과정으로 식 (2)를 사용하여 영상을 평탄화 시켜 hole-filling 으로 제거하지 못한 노이즈를 제거한다.

$$If \sum_{k=-N/2}^{N/2} \sum_{l=-N/2}^{N/2} I(x+k, y+l) \ge 255 \times N \times N/2, I(x, y) = 255$$
$$else \qquad I(x, y) = 0 \qquad (2)$$

$I(x, y)$는 영상의 x , y 에서의 픽셀 값을 의미한다. 일정 크기의 블록 (N×N)내의 픽셀 값을 모두 더한 값이 임계치 이상이면 그 블록의 중심 값을 255, 이하이면 0 으로 변환한다.

위와 같이 설계된 필터들을 사용하여 그림 4.(c)와 같이 손 영역만 정확하게 추출된다.

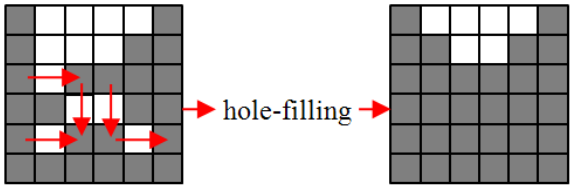


그림 3. hole-filling

(a) (b) (c)

그림 4. filtering 과정

(a)원영상 (b)hole-filling (c)smoothing filter

2.4 손 추적 및 손가락 개수 인식

필터링된 손 영역 중 손등의 중심 좌표 이용하여 마우스 포인터의 위치를 결정한다. 이 중심 좌표 계산하기 위해 두 가지 방법을 사용한다. 하나는 추출된 영역의 위치좌표들의 정보를 이용하여 계산된 중심 좌표를 사용한다. 다른 하나는 거리변환행렬을 사용하여 계산된 중심 좌표를 사용한다.

영역의 위치정보를 이용하여 중심 좌표 $C_{region}(x, y)$는 식 (3)을 사용하여 계산된다.

$$C_{distance}(x,y) = \frac{\sum_{m=0}^{N} P_m(x,y)}{N} \qquad (3)$$

$P_m(x,y)$는 추출된 영역의 x, y 좌표의 순서쌍이고, N은 추출된 영역의 전체 픽셀 개수를 의미한다. 추출된 영역의 모든 픽셀들의 x, y좌표의 순서쌍을 모두 더하여 픽셀의 개수 N 만큼을 그 더한 값에 각각 나누어주면 중심 좌표가 계산된다.

식 (3)을 사용하는 경우 손목이 많이 포함된 영상에서 계산되는 중심 좌표 $C_{region}(x, y)$는 손등이 아닌 손목 위에 존재하게 된다. 거리변환행렬을 사용하여 중심 좌표 $C_{distance}(x, y)$를 계산하기 위해서는 먼저 영상을 다운 샘플링을 한다. 다운 샘플링된 영상에서 거리변환행렬을 사용하여 $C_{distance}(x, y)$를 계산한다. 식 (4)의 행렬식에 의해 변환되는 픽셀 값들 중 가장 큰 값을 가진 픽셀이 영상의 중심 좌표가 된다. 손목이 포함된 영상일지라도 계산되는 $C_{distance}(x, y)$는 손등 위에 존재하게 된다.

$$D = \begin{bmatrix} 4.5 & 4.0 & 3.5 & 3.0 & 3.5 & 4.0 & 4.5 \\ 4.0 & 3.0 & 2.5 & 2.0 & 2.5 & 3.0 & 4.0 \\ 3.5 & 2.5 & 1.5 & 1.0 & 1.5 & 2.5 & 3.5 \\ 3.0 & 2.0 & 1.0 & 0.0 & 1.0 & 2.0 & 3.0 \\ 3.5 & 2.5 & 1.5 & 1.0 & 1.5 & 2.5 & 3.5 \\ 4.0 & 3.0 & 2.5 & 2.0 & 2.5 & 3.0 & 4.0 \\ 4.5 & 4.0 & 3.5 & 3.0 & 3.5 & 4.0 & 4.5 \end{bmatrix} \qquad (4)$$

이 방법은 다운 샘플링을 해서 사용하기 때문에 정확도가 낮아져 계산되는 $C_{distance}(x, y)$는 그림 5.(b)에서처럼 손등의 중심을 가리키지 않는다.

이러한 문제점들을 보정하기위해 두 좌표의 거리에 따라 일정한 가중치 a 를 곱하여 새로운 좌표 $C_{new}(x, y)$를 계산하고, 그 결과를 마우스 포인터의 위치에 대응시킨다.

$$C_{new}(x,y) = \alpha C_{distance(x,y)} + (1-\alpha) C_{region(x,y)} \qquad (5)$$

식 (5)를 사용하여 계산된 $C_{new}(x, y)$는 그림 5.(c)와 같이 손목이 많이 포함된 영상에서도 중심 좌표가 손등 가운데를 가리킨다.

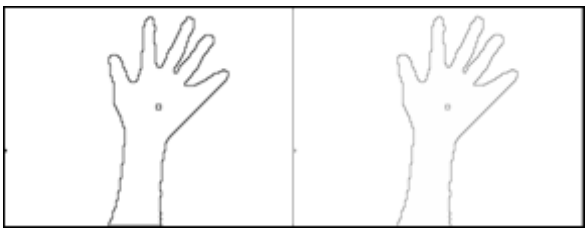
(a) (b) (c)

그림 5. 새로운 중심 좌표계산 과정

(a) region info. (b) distance transform (c) proposal

손가락의 개수는 추출된 영역의 경계 영상과 식 (5)에서 계산된 $C_{new}(x, y)$를 중심으로 하는 원과의 교점의 개수를 이용하여 계산한다.

추출된 영역에서 소벨 마스크를 사용하여 경계 영상을 구한다. 다음 과정으로 구해진 경계는 세선화 과정을 거치게 된다. 세선화하지 않았을 경우 원과의 교점이 비슷한 위치에서 여러 개가 존재할 수 있다. 이진화 된 영상에서 경계를 구하였기 때문에 경계를 세선화 하기 위하여 경계 영상과 원영상의 차 영상을 이용한다.



(a) (b)

그림 6 : (a) edge 영상 (b) 세선화 영상

위 그림 6은 경계 영상을 세선화한 영상이다. 원의 반지름은 추출된 손 영역의 크기에 따라 가변적으로 변하게 된다. 그림 7처럼 $C_{new}(x, y)$부터 22. 5°씩 16방향을 검색하여 경계와 만나는 점과 $C_{new}(x, y)$의 거리를 이용해 추출된 손 영역의 크기를 추정한다. 추정된 크기에 가중치 a를 곱하여 반지름 R을 구한다.

$$R = \alpha \sum_{k=2}^{14} D_k / 12 \qquad (6)$$

식 (6)에서 D_k 는 중심 좌표들로부터 각각의 16개의 경계와 만나는 점들과의 거리를 내림차순으로 정렬한 값이다. 16개의 점들과의 거리 중에서 그림 7에서와 같이 경계와의 교점이 아닌 영상의 가장자리와의 교점이 포함될 수 있다. 뿐만 아니라 경계와의 교점이 너무 가까운 경우 손의 크기를 추정하는데 방해요소로 작용할 수 있다. D_k 중 큰 값2개, 작은 값2개를 제외한 나머지 12개의 값의 평균값을 사용하여 R의 크기를 결정한다. 원과 만나는 교점의 개수 IP는 식 (7)을 이용하여 손가락의 개수 N으로 변환 할 수 있다.

$$N = ROUND(IP/2) - 2 \qquad (7)$$

$ROUND()$ 는 소수 첫째자리에서 반올림을 한다. 식 (7) 을 사용하여 구한 N 값으로 마우스의 특정 동작을 결정한다.

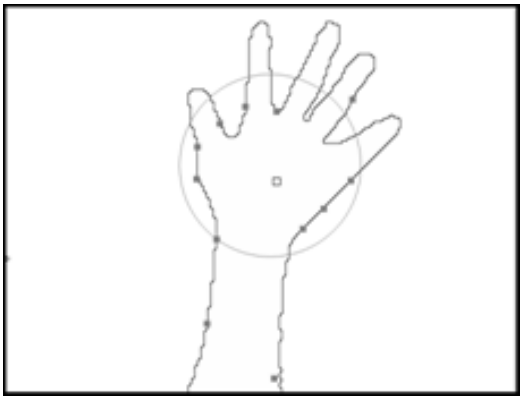


그림 7. 경계와의 교점을 이용하여 반지름을 구한 영상

3. 실험 조건 및 결과

본 논문에서는 범용 USB 카메라를 이용하였고 펜티엄4 CPU 2.80GHz 에서 구현하였다. 640×480의 입력 영상을 사용하였고, hole-filling의 임계간격은 9 pixel, 평탄화 필터의 블록의 크기는 3×3 을 사용하였다. 처리 속도는 초당 약 14 frame 의 속도를 보인다.

제안한 시스템을 효율적으로 구현하고 정확한 실험을 하기 위해 그림 8과 같이 손 인식 프로그램을 제작하여 사용하였다.

손가락 개수 인식은 손목의 포함여부에 따라 두 종류의 영상으로 실험하였다. 손목의 끝부터 손가락 끝까지의 거리 보다 손목이 더 포함된 경우 손목이 포함된 영상이라고 정의하고, 그렇지 않은 경우 포함되지 않은 영상이라고 정의하였다.

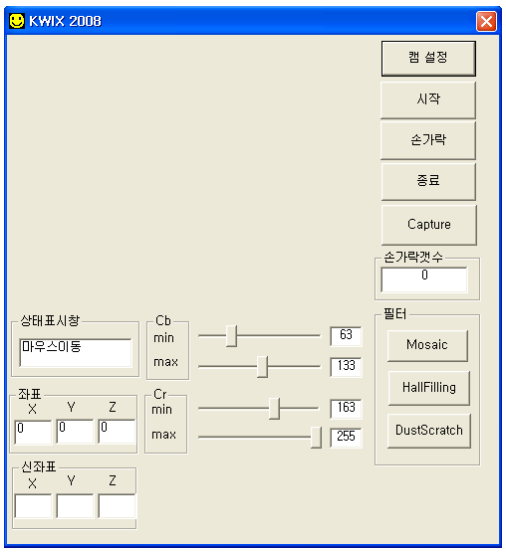


그림 8. 손 인식 프로그램

표 1. 손가락 개수 인식률

손가락 개수	손목이 포함된 영상 (%)	손목이 포함되지 않은 영상 (%)	평균 (%)
1	89	93	91
2	87	92	89.5
3	90	93	91.5
4	97	94	95.5
5	88	94	91
평균(%)	90.2	93.2	91.7

표 2. 마우스 동작 성공률

마우스 동작 구분	성공률(%)
이동	97
우클릭	73
Drag	68
더블 클릭	63

표 1에서와 같이 손목이 포함된 영상은 평균적으로 90.2% 의 인식률을 보였고 손목이 포함되지 않은 영상에서는 93.2% 의 인식률을 보였다.

마우스 동작 구분은 손가락의 개수에 따라 다르게 대응시켰다. 손가락의 개수가 5개일 때는 마우스가 이동 동작에 대응시키고 , 4개일 때는 우클릭 동작에 대응시켰다.

표 2에서 마우스 이동 동작에서는 97%의 성공률을 보였으나 동작이 복잡해질수록 성공률은 낮아 졌다.

4.결론

본 논문에서 제안한 손 인식 알고리즘은 카메라로부터 획득된 영상에서 실시간으로 손을 인식하여 마우스 대신으로 사용함으로써 마우스의 이동 및 특정동작들을 대신한다. 제안한 방법은 빠른 수행속도를 보이고 또한, 안정적으로 마우스 포인터를 제어 할 수 있다. 그러나 컬러정보를 이용하기 때문에 살색 계통의 물체가 손과 함께 나타날 경우 손을 잘 인식하지 못하고, 손가락의 개수가 빠르게 변할 시 개수의 인식률이 낮아지는 단점이 있다. 이러한 문제점은 향후 연구 및 보완되어야 할 것이다.

5. 참고 문헌

[1] Jang-Woon Kim, "Hand Posture Recognition using Data of Edge Orientation Histogram," Department of Computer Engineering, Chonnam Nation University, 2006.

[2] 김종민, 이칠우, "윈도우 플레이어 제어를 위한 에지 방향성 히스토그램 손 형상 인식," 한국정보과학회, 한국정보과학회 학술발표논문집 한국정보과학회 2003년도 가을 학술발표논문집 제30권 제2호 (Ⅱ), pp. 628 ~ 630, 2003.

[3] C. Shan, Y. Wei, T. Tan and F. Ojardias, "Real Time Hand Tracking By Combining Particle Filtering and Mean Shift", Proc. IEEE International Conference on Automatic Face and Gesture Recognition (FG'04), 669-674, Seoul, South Korea, May 2004.

[4] 김계영,장석우,"유형기반 스테레오 정합을 통한 영상변이 측정,"한국컴퓨터정보학회, 한국컴퓨터정보학회 논문지 한국컴퓨터 정보학회 논문지 제11권 제2호, pp. 83 ~ 92, 2006.

감사의 글

본 연구는 한국과학재단 특정기초연구(R01-2006-000-10199-0(2008)) 지원으로 수행되었음.