



손 동작 인식을 통한 사용자에게 편리한 핸드마우스 인터페이스 구현

Developing User-friendly Hand Mouse Interface via Gesture Recognition

저자 (Authors)	강성원, 김철중, 손원 Sungwon Kang, Chuljoong Kim, Won Sohn
출처 (Source)	한국방송미디어공학회 학술발표대회 논문집 , 2009.11, 129-132 (4 pages)
발행처 (Publisher)	한국방송·미디어공학회 The Korean Society Of Broad Engineers
URL	http://www.dbpia.co.kr/Article/NODE01575903
APA Style	강성원, 김철중, 손원 (2009). 손 동작 인식을 통한 사용자에게 편리한 핸드마우스 인터페이스 구현. 한국방송미디어 공학회 학술발표대회 논문집, 129-132.
이용정보 (Accessed)	광운대학교 128.134.57.*** 2018/05/08 02:29 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독 계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

손 동작 인식을 통한 사용자에게 편리한 핸드마우스 인터페이스 구현

*강성원 *김철중 *손원

* 경희대학교 전자정보대학

hancoms@naver.com

Developing User-friendly Hand Mouse Interface via Gesture Recognition

*Sungwon Kang *Chuljoong Kim **Won Sohn

School of Electronics and Information, Kyung Hee University

요약

컴퓨터의 소형화로 휴대성과 공간의 제약이 없는 컴퓨터 인터페이스 방법의 필요성이 증가하고 있으며, 이와 관련하여 인간-컴퓨터 상호작용(HCI)을 위한 제스처 기반의 제어방식에 대한 연구가 활발하게 진행되고 있다. 기존의 손동작 인터페이스 구현들은 컴퓨터를 제어하기 위하여 사용방법에 대한 선행학습이 필요하였다.

이 논문은 사용자의 손 모양과 손끝 정보만을 가지고 선행학습이 요구되지 않는 간편한 인터페이스 구현방법을 제안하였다. 이를 위해 1대의 웹캠과 인텔의 오픈소스 영상처리 라이브러리 OpenCv를 사용하였다. 차영상과 화소값 기반의 영상처리과정을 통해 실시간으로 손 영역을 추적하고 이를 이진화 시켰다. 손가락의 움직임도 값이 변하지 않도록 중심모멘트를 설정하여 마우스 커서 움직임을 상대적으로 활용하였다. 상황에 따라 손 끝점을 절대적 좌표로 활용하여 손이 웹캠에서 벗어나면 움직임을 자연스럽게 연결시켰다. 마지막으로 검지의 움직임 하나만으로 마우스 클릭 이벤트를 수행함으로써 보다 사용자에게 친숙한 핸드마우스 인터페이스를 구현하였다.

1. 서론

이 논문은 사용자의 손 모양과 손끝 모양 정보만을 가지고 안정적 이면서 선행학습이 요구되지 않는 간편한 핸드마우스 인터페이스 구현방법을 목적으로 한다. 컴퓨터 산업의 발전과 더불어 컴퓨터 인터페이스는 인간에게 보다 친숙한 환경으로 변화해 가고 있다. 현재 컴퓨터에 의사전달을 하기위해서 사용자는 키보드, 마우스, 태블릿 등의 하드웨어적인 장비를 사용한다. 이러한 방식들은 장비 조작기술 습득과 공간 등의 환경적 제한을 받는다.[5]

이를 개선하기위해 부가적인 장비 없이 사람의 손만으로 컴퓨터의 입력동작을 제어한다면 보다 더 인간 중심적인 인터페이스 환경이 가능해질 것이다. 그러나 현재 손동작 제스처 기반의 인터페이스 방식들은 손모양이나 동작을 기호화하여 활용하는 것이 많다.[4] 이 같은 방식은 인식률을 높일 수 있지만 선행학습이 필요하다는 단점을 가지고 있다.

이 논문에서는 사용자에게 보다 친숙한 손동작을 가지고 이러한 단점을 보완하고자 한다. 이를 위해 손가락에서 가장 많이 사용되는 검지를 활용하였다. 검지를 이용하여 포인팅을 표시하는 동작은 방향을 지시하기 위해 쓰이는 가장 빈번한 동작이다. 따라서 이와 같은 손동작을 추적하여 마우스 포인터를 제어 할 수 있다면 사용자는 좀 더 자연스럽게 의사전달을 할 수 있다. 또한 검지만의 움직임을 추적하여 클릭을 제어한다면 선행 학습 없이 컴퓨터에 쉽게 의사 전달을 할 수 있다.

마우스 포인터의 움직임과 클릭을 제어하는 데에는 기본적으로 정확한 손 영역 추출, 그리고 손의 변화와 무관하게 손의 좌표를 안정적이면서 빠르게 추적 및 인지하는 기술이 필요하다. 컴퓨터를 기반으로 하여 손 영역을 추적하는 기존의 연구들을 보면 동작은 정확하나 계산량이 많아 속도가 현저하게 느려지거나 선행학습이 요구된다는

단점이 있다. 이 논문에서는 반복적 이진화, 모폴로지, 미디언 필터, 라벨링 알고리즘 등을 사용하여 실시간으로 손의 움직임을 추적한다. 또 검지의 움직임을 카운터 하는 방식 사용하여 마우스의 좌, 우, 더블 클릭이 가능한, 즉 선행학습 없이 쉽게 사용 할 수 있는 핸드마우스 기법을 제안한다.

이 논문의 구성은 다음과 같다. 2장에서는 시스템 개요를 설명한다. 3장에서는 손영역 추출기법, 4장에서는 이를 통해 얻어진 손영역을 가지고 마우스 포인터 및 이벤트 생성하는 방법을 기술하였다. 5장 손이 촬영범위를 벗어나거나 오류가 포함된 프레임에 대한 예외처리를 다뤘다. 6장에서는 실험결과를, 7장에서는 결론을 지었다.

2. 시스템 개요

시스템 흐름도는 그림 1과 같이 손 영역 추출, 커서 처리, 마우스 이벤트 처리를 포함하는 세 가지 부 시스템으로 구분된다.

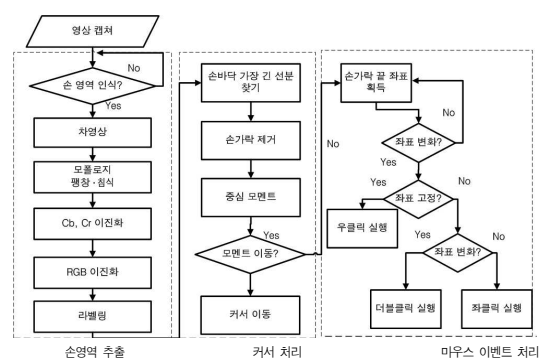


그림 1. 시스템 흐름도

첫 번째는 배경과 손을 분리 해 내는 방법이다. 두 번째는 손의 영역을 정확하게 검출하기 위해 알고리즘을 사용하여 더 정확한 손 영역만 추출하는 과정이다. 세 번째로는 추출한 손 모양을 이용해 안정된 좌표 이동 및 마우스 이벤트 생성 부분이다.

3. 손 영역 추출 기법

손과 배경을 분리하기 위해서는 차영상을 사용하고, 모폴로지 기법과 사람의 피부만 인식할 수 있는 컬러 추출, 최종적으로 주변 잡음 제거를 위한 알고리즘을 적용하였다. 이 장에서는 효율적으로 손 영역을 추출하는 알고리즘에 관하여 다룬다. 손과 배경을 분리하는 방법과 잡음을 제거하여 손 영역만 얻는 과정을 설명한다.

3.1 손 영역과 배경 분리 방법

복잡한 배경환경 속에서 손과 컬러분포가 유사한 특징을 가지는 객체가 존재할 수 있다. 따라서 우선 배경을 먼저 캡처 해 등록하는 절차가 필요하다. 손이 웹캠의 영역 안으로 들어오면 배경 영상과 손이 있는 영상을 비교하여 그림 2(a)와 같은 차영상을 생성할 수 있다.[1] 구분된 영역을 마스크로 활용한다. 이런 영상을 가지고 모폴로지 방식을 활용하여 생성한 마스크 영상 내부에 존재하는 잡음을 제거한다. 팽창과 침식 알고리즘을 순차적으로 사용하면 손 내부의 잡음이 제거된다. 내부의 잡음을 제거 하였다면 원 RGB영상에서 손의 영역만 남기고 나머지는 흰색으로 처리를 해 주어야 한다. 이를 위해 차영상과 손 내부의 잡음을 제거한 영상을 이용해 그림 2(b)와 같은 영상을 생성한다. 관심범위가 손에 근접한 영역으로 축소됐기 때문에 이진화에 따른 손 영역부분 손실이나 오차를 줄일 수 있다.[5]

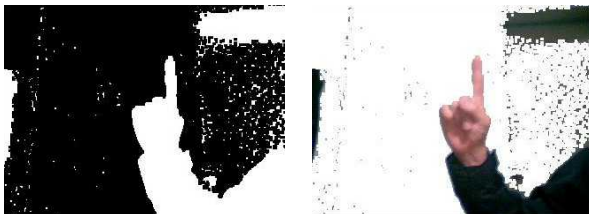


그림 2. 배경과 움직이는 사물의 분리. (a) 차영상, (b) 손과 배경 분리

그림2(a)와 같이 배경을 분리한 컬러영상을 가지고 손영역을 이진화하기 위해서 RGB컬러공간과 YCrCb공간을 활용한다. 이진화한 영상은 그림 3(a)와 같다. RGB 공간에 대한 이진화는 피부영역을 추출하는데 탁월한 알고리즘이다. RGB공간은 빛에 따른 영향이 심하기 때문에 조명환경에 적응할 수 있는 이진화 방법이 필요하였다. 이러한 단점을 보완하고자 RGB컬러공간을 YCrCb컬러공간으로 변환하여 한번 더 이진화 시켰다. RGB컬러공간을 YCrCb컬러공간으로 나누기 위해서 식(1)의 3가지 조건을 모두 만족하는 경우와 그렇지 않은 경우로 이진화 하였다.[6]

$$\begin{aligned} & \{R < 95, G > 40, B > 20\} \\ & \{MAX(R, G, B) - MIN(R, G, B) < 15\} \\ & \{|R - G| > 15, R > G, R > B\} \\ & (TURE \rightarrow 255, FALSE \rightarrow 0) \end{aligned} \quad (1)$$

Cr공간과 Cb공간에 대해서 반복적 이진화 기법을 적용시켰다. 반복적 이진화를 적용하는 이유는 웹캠의 위치가 변하거나 조명 환경이 바뀔 때마다 임계치 값을 변화시켜 환경 변화에 독립적으로 만들기 위함이다. Cr공간과 Cb공간에 대해 이진화한 결과를 OR연산한 결과는 그림 3 (b)와 같다.[6]

배경이 제거됐기 때문에 객체를 이진화시킬 수 있는 임계치값을 얻을 수 있다. 두 방법을 이용하여 이진화한 공간을 다시 OR연산 하여 한 공간으로 압축시키면 이진화된 하나의 영상을 획득할 수 있다.

결론적으로 반복적 이진화를 통해 임계치 값을 바꿔주어 환경에 독립적이면서, 동시에 RGB 공간의 이진화를 통해 피부색을 효율적으로 추출할 수 있다.



그림 3. 손영역 이진화. (a) RGB컬러공간을 통한 이진화, (b) YCrCb 컬러공간을 통한 이진화

3.2 최적화된 손 영역 추출

이진화한 영상에서 가장 큰 객체가 손이 되고 작은 객체는 잡음일 가능성이 높다. 따라서 라벨링을 통해 임의로 설정 된 면적보다 작은 객체들을 제거 해 준다. 가장 큰 객체를 추출하면 이진화된 손영역 영상을 생성할 수 있다. 그러나 손 내부의 잡음은 라벨링으로 인한 면적으로 인식되지 않았기 때문에 그림 4 (a)와 같이 내부 잡음이 제거되지 않는다. 이를 제거하기 위해 영상을 반전시켜 한 번 더 라벨링하면 손 그림 4 (b)처럼 내부의 잡음도 같이 제거할 수 있다. 이는 다음 처리과정에서 손 내부의 가로길이를 구하는데 발생하는 오류를 개선시켰다.



그림 4. 손영역 라벨링. (a) 라벨링 1회 실시, (b) 라벨링 2회 실시

4. 마우스 포인터 및 이벤트 처리

손끝 정보를 마우스 포인터로 활용하면 포인터 이동에 유리하지만 만 이벤트 처리시 손 모양 기호화의 단점이 있다. 손의 중심모멘트에서

거리정보를 활용하는 방식은 손가락의 모양의 변화에 따라 모멘트의 오차가 발생한다. 이러한 단점을 극복하고자 손 모양 변화에 무관한 중심모멘트를 구하는 방법과 검지 하나로 마우스 이벤트 처리방법을 제안한다.

4.1 마우스 포인터 움직임 원리

손가락의 움직임에 무관한 중심 모멘트를 구하기 위해서는 손 전체에서 손가락에 해당하는 부분을 제거하는 방법을 이용하였다. 웹캠에서 거리가 멀어지거나 가까워지더라도 손가락과 손바닥의 비례는 변하지 않는다는 점을 이용한다. 우선 손 영역에서 가로로 가장 긴 선분을 찾는다. 그리고 가장 긴 길이를 기준으로 상단을 제거하면 손가락 부분을 제거 할 수 있다. 그리고 제거된 이진화 영상을 가지고 중심모멘트 값을 구하면 그림 5의 (a)와 (b)처럼 고정된 중심모멘트 영상을 획득할 수 있다.

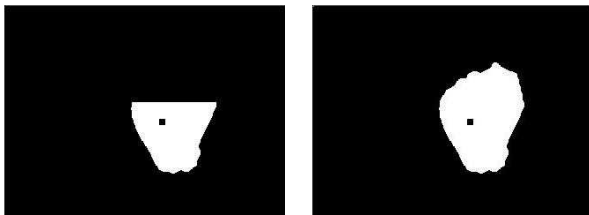


그림 5. 고정된 중심 모멘트 영상. (a) 최대 가로 적용, (b) 최대 가로 비적용

모멘트는 다음과 같은 식 (2)을 이용해 구할 수 있다.

$$\mu_{qp} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (2)$$

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (p, q = 0, 1, 2, \dots)$$

최종 계산되는 중심 모멘트 η_{pq} 는 아래의 식 (3)와 같다.

$$\eta_{pq} = 1000 \times \frac{\mu_{pq}}{(\mu_{00})^y} \quad (3)$$

$$y = (p + q + 2) / 2 \quad (p + q = 2, 3, \dots)$$

중심모멘트를 이용하여 마우스 좌표의 이동 방향과 거리를 설정할 수 있다. 마우스 포인터를 이동하는 방법에는 상대적 차이를 이용하는 방법과 손끝 점을 이용한 절대적인 좌표 이동 방법을 구현하였다.

첫 번째 방법은 이전 프레임과 현재 프레임간의 상대적인 변화를 관찰하는 방법이다. 두 프레임간 중심모멘트 값의 차이를 비교한다. 비교한 값을 통해 손이 얼마만큼 이동하였는지 상대적인 결과를 얻을 수 있다. 이 결과를 가지고 현재 마우스가 스크린 상에 위치한 커서 값을 더한다. 그리고 그 결과를 좌표위치로 리턴 시켜 최종적으로 그림 6(a)와 같은 마우스 포인터를 움직일 수 있다.

상대적인 변화를 이용하는 방식은 손가락 모양과 무관하게 포인터를 이동할 수 있다. 하지만 웹캠의 촬영범위를 벗어나거나 손영역의 크기가 작아지면 올바른 중심모멘트를 구할 수 없어 오작동을 초래한

다. 이러한 경우 그림 6(b)과 같이 손 끝점 좌표값을 이용하는 절대적인 좌표 이동 방법을 사용하였다.(4) 손 끝점의 위치를 마우스 포인터로 활용하여 모니터의 해상도에 비례하여 웹캠에 입력된 손끝 위치정보를 반환한다. 이 정보를 이용하면 손 전체가 웹캠에 존재하지 않더라도 손끝 정보만 파악하면 마우스 포인터를 이동하는데 문제가 발생하지 않는다. 하지만 손끝을 움직이면 마우스 좌표가 이동하기 때문에 손 끝 좌표를 가지고 마우스 이벤트를 처리하는데 한계가 있다.

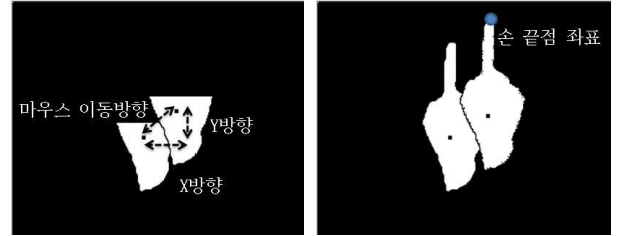


그림 6. 마우스 커서의 이동. (a) 상대적 좌표 이동 방법, (b) 절대적 좌표 이동 방법

두 방법을 상황에 따라 적절히 이용하면 사용자의 손이 웹캠에 존재하지 않을 때 마우스 포인터를 이동하지 못하도록 설정할 수 있다. 또 손 전체가 웹캠에 위치하지 않았을 때 일으키는 문제점을 보완할 수 있다.

좌표를 이동하는데 제한해야할 사항은 손 전체가 촬영되는 영상 안에 있어야 한다는 점이다. 만약 손의 일정 부분이 촬영되지 않는다면 중심모멘트 값을 구하는데 문제가 발생하기 때문이다. 따라서 촬영되는 영상의 테두리 부분에 영상이 접근하면 마우스가 이동하지 않도록 조정하였다. 마우스의 이동을 살펴보면 동일한 거리를 이동하는데 있어 짧은 시간에 이동하면 마우스 포인터가 상대적으로 더 이동하는 것을 볼 수 있다. 이와 동일한 기능을 적용하기 위해 포인터 이동에 가속도를 부여하였다.

4.2 마우스 클릭 이벤트 표현 기법

손 영역을 분할하여 손가락 모양 변화에 따라 변하지 않는 중심모멘트 값을 획득했다. 따라서 이제 손가락의 움직임은 마우스 포인터 변화에 영향을 주지 않는다. 마우스 이벤트는 5개의 손가락을 모두 사용할 수 있다. 하지만 더욱 사용자에게 친숙하고 간편한 핸드마우스 인터페이스 방법을 구현하기 위해서는 사용자에게 익숙한 손가락과 행동양식을 사용할 필요가 있었다. 이를 위해 그림 7과 같이 버튼을 누르거나 무언가를 지시할 때 가장 흔히 사용하는 검지를 이용해 마우스 클릭을 표현하였다.

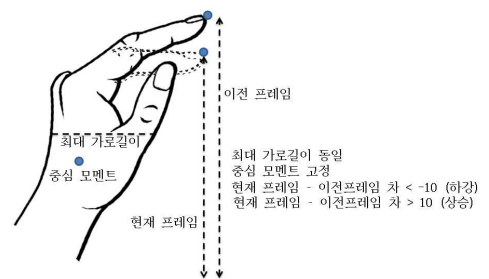


그림 7. 손 끝점을 이용한 이벤트 생성

마우스 클릭 이벤트가 발생하는 시점은 마우스 포인터가 정지했을 시점이다. 따라서 마우스 이동 중에 클릭 이벤트가 발생하지 못하도록 함으로써 이동에 따른 클릭 이벤트 발생을 최소화 하였다.

마우스의 왼쪽 버튼 클릭은 사람이 전자제품의 버튼을 누르는 것과 같이 검지를 아래로 한번 구부리는 경우이다. 실제 마우스 버튼을 클릭할 때도 동일한 행동양식을 이용하므로 별도의 학습이 필요 없는 방식이다.

그림 7과 같은 방식을 사용하기 위해 손 전체 높이 대비 끝점간의 거리 값의 변화를 인지하도록 하였다. 이진화한 손 영상에서 손가락의 끝점을 기억하면서 거리가 가까워지면 이를 컴퓨터에 저장한다. 그리고 다시 본래의 손길로 들어오면 검지를 한번 움직인 것이기 때문에 마우스 왼쪽 클릭으로 간주한다. 같은 방식으로 마우스 더블클릭은 실제 마우스 더블클릭과 같이 검지를 두 번 움직이면 된다.

마우스 오른쪽 클릭은 검지를 한번 아래로 내린 후 대기하면 실행 되도록 구현하였다. 본래의 위치로 검지가 돌아오지 않았음을 인지하게 된다. 마우스 사용 시 왼쪽클릭이 오른쪽 클릭보다 사용량이 더 많기 때문에 이 같은 방식을 적용하였고 검지 하나를 이용하여 마우스 클릭 이벤트를 표현할 수 있기 때문에 손쉽게 사용자가 적용할 수 있는 인터페이스 방법이다.

마우스 클릭이벤트가 손 끝점 정보를 활용하였기 때문에 만약 손 끝점이 웹캠의 촬영 범위를 벗어나면 인지할 수 없다. 따라서 이러한 경우 예외 처리에 해당하도록 하였다.

5. 예외 처리

사용자가 마우스를 이동하고 손이 사라지게 되면 좌표의 이동을 멈춰야 한다. 이러한 기능을 추가하기 위해 객체의 크기가 포인터를 획득할 수 없을 만큼 작을 경우 마우스 이동을 정지하도록 하였다. 또 다시 손전체가 웹캠의 촬영영역에 들어오기 전까지는 손가락의 끝점의 마우스의 절대좌표로 인지하도록 하여 자연스럽게 마우스 포인터 동작이 이어질 수 있도록 구현하였다.

이진화가 제대로 되지 않거나 최대 가로길이에 해당하는 높이 좌표 반환에 오류가 발생하면 마우스 오작동이 일어난다. 손이 정지되어 있음에도 불구하고 마우스가 이동하는 것이다. 이러한 오차를 줄이기 위해 이전프레임과 현재프레임은 손의 면적과 최대 가로길이의 변화가 작다는 점을 활용하였다. 프레임 간 면적과 가로길이의 변화가 심하면 해당 프레임을 오류프레임으로 인정하여 해당프레임을 사용하지 않는다.

6. 실험 결과

실험을 위해 조도 300~400LUX의 일정한 조명하에 마우스 포인터 이동 제적과 클릭을 실험 해 보았다. 또한 320×240해상도를 가지는 웹캠을 사용하여 영상을 입력받으며 사용자와 웹캠간의 거리를 50~60cm로 유지하였다.

마우스 이동과 좌, 우, 더블클릭의 인식률을 알아보기 위해서 왼쪽 하단에서 오른쪽 상단의 아이콘까지 이동한 후 클릭 실험을 하였다. 단 각 실험 당 일정 시간제한을 두었다. 실험 수행결과 각각의 인식률은 표 1과 같은 결과를 얻었다.

표 1. 핸드마우스 인식률

구 분	제한 시간	실험 횟수	인식 횟수	인식률
포인터 이동	4초	200	173	86.5%
좌클릭	2초	200	188	94%
우클릭	2초	200	180	90%
더블클릭	3초	200	185	92.5%

7. 결론

이 연구는 사용자의 손 모양과 손끝 모양 정보만을 가지고 안정적 이면서 선행학습이 요구되지 않는 간편한 핸드마우스 인터페이스 구현방법을 제안하였다.

마우스 좌표 이동의 경우 마우스의 좌표를 손의 중심으로 사용하고 이를 이용하여 일정 크기의 원을 그려 원 외각에 존재하는 손가락의 개수를 이벤트로 활용했기 때문에 손이 웹캠에 근접할 때 손가락의 개수를 잘못 센다거나 손가락의 움직임으로 손 전체의 중심모멘트가 달라져 오작동을 일으킨다는 문제점이 있었다. 제안한 방법을 적용하여 마우스 좌표를 이동 할 경우 손가락 부분을 제거한 중심 모멘트를 구해 손이 움직이지 않고, 손가락만 움직인다 하더라도 중심 모멘트의 값은 변하지 않아 안정적으로 포인터 이동을 제어 할 수 있었다. 또한 손이 움직일 때 모멘트의 좌표가 일정량 이상 변하면 이전 프레임과 현재 프레임의 중심모멘트를 비교하여 상대적인 결과 값을 구했다. 이를 현재 위치한 좌표 값을 받아 결과에 입력하면 마우스 포인터가 자연스럽게 이동 한다는 것을 확인 하였다.

마우스 클릭의 경우 손가락의 개수를 센다거나 손을 기호화 시키는 경우가 많아 사용방법을 익히는 문제점이 있었다. 이 연구에서는 사전 학습 없이 검지 하나만을 이용해 검지가 굽혀 질 경우 전 프레임과 현재 프레임 간 끝점을 기억한 후 끝점과 손바닥의 거리를 비교해 클릭을 구분하도록 해 이 문제를 개선하였으며 이를 실험 결과로 확인 하였다. 이 연구를 통해 제안한 기법으로 적용하면 보다 효율적으로 사용자 친화적인 인터페이스를 구축 할 수 있을 것이라 기대한다.

참고 문헌

- [1] 황선규, "(IT expert) 영상처리 프로그래밍 by Visual C++", 한빛미디어
- [2] 정성환, 이문호, "(오픈소스 OpenCV를 이용한) 컴퓨터 비전 실무 프로그래밍 : 기본편", 홍릉과학, 2007
- [3] 하영호, 남재열, 이웅주, 이철희, "디지털영상처리", 도서출판 그린, 2008
- [4] 이창주, 이준호, "깊이 에지 기반의 Curvature Scale Space Map을 이용한 손 제스처 인식", 한국정보처리학회 춘계학술발표대회 논문집 제14권 제1호, 2007
- [5] 김유호, 김종선, 박지영 "휴먼 마우스 구현을 위한 효율적인 검지 좌표 검출 및 추적", HCI 학술대회, 한국정보과학회 10권 1호, 2001
- [6] 윤종원, "적응적 피부색 검출과 에지 특징을 이용한 유해 이미지 분류 방법", 한양대학교 대학원 석사학위논문, 2009