

# 손동작 인식을 이용한 마우스제어기법

김 정 인<sup>†</sup>

## 요 약

본 연구는 마우스의 입력을 사람의 손동작으로 대체하는 휴먼마우스 시스템에 관한 것이다. 모니터의 해상도가 커지면서 웹카메라와 모니터간의 해상도 차이로 인하여 단순히 웹카메라에서 손의 위치를 인식하여 포인터를 이동시키는 방법으로는 모니터의 모든 범위에 커서를 위치시킬 수가 없다. 따라서, 되돌아오는 동작을 반복하지 않고 마우스의 위치를 모니터의 가장자리까지 원하는 곳으로 편리하게 위치시키는 효과적인 방법을 제안한다. 또한, 다양한 마우스이벤트를 처리하기 위하여 엄지와 검지를 이용한 손가락 동작 인식을 제안하고 인식률을 측정한 결과, 97% 이상으로 그 유효성을 확인하였다.

## A Mouse Control Method Using Hand Movement Recognition

Jung-In, Kim<sup>†</sup>

## ABSTRACT

This paper proposes a human mouse system that replaces mouse input by human hand movement. As the resolution of monitors increases, it is not quite possible, due to the resolution difference between web cameras and monitors, to place the cursor in the entire range of a monitor by simply moving the pointer which recognizes the position of the hand from the web camera. In this regard, we propose an effective method of placing the position of the mouse, without repeating the returning hand movements, in the corners of the monitor in which the user wants it to be. We also propose the recognition method of finger movements in terms of using thumb and index finger. The measurement that we conducted shows the successful recognition rate of 97% that corroborates the effectiveness of our method.

**Key words:** Hand Movement(손동작), Recognition(인식), Mouse(마우스), Control(제어), Image(영상)

## 1. 서 론

현재 컴퓨터 사용자는 컴퓨터에 의사를 전달하는 수단으로 키보드, 마우스와 같은 장치를 이용한다. 또한 최근에는 화면을 직접 터치하여 조작하는 입력 방법이 제안되었고 직관성과 편리성으로 인하여 많은 기기들이 터치패드화 되어가고 있다. 그러나, 이러한 접촉식 입력은 사용자의 행동과 장비 사용을 위한 공간상의 제약을 수반한다. 앞으로는 직접적인

접촉 없이 컴퓨터에게 의사를 전달하는 방법으로 음성인식이나 동작인식에 의한 입력이 보편화될 것으로 보인다. 컴퓨터 입력의 일환으로 부가적인 하드웨어 없이 사람의 손동작을 컴퓨터의 입력 수단으로 사용한다면 컴퓨터를 좀 더 사용자 중심으로 만드는 새로운 인간친화적인 인터페이스가 가능하다. 특히 사람의 손동작 중 검지만을 펴서 표현하는 포인팅 동작은 방향을 지시하기 위하여 가장 빈번히 사용되는 동작이다. 따라서 이와 같은 손동작을 추적하여

※ 교신저자(Corresponding Author): 김정인, 주소: 부산광역시 남구 용당동 535 동명대학교 컴퓨터공학과(608-711), 전화: 051)629-1174, FAX: 051)629-1169, E-mail: jikim@tu.ac.kr  
접수일: 2012년 8월 22일, 수정일: 2012년 9월 18일

완료일: 2012년 10월 8일  
<sup>†</sup> 종신회원, 동명대학교 컴퓨터공학과  
※ 이 논문은 2011학년도 동명대학교 교내학술연구비 지원에 의하여 연구되었음(2011B018).

윈도우 시스템 사용을 위한 마우스 포인터의 움직임 을 제어 할 수 있다면 사용자는 좀 더 자연스럽게 컴퓨터에게 의사를 전달 할 수 있다[1-8].

본 연구에서는 웹카메라로 인식한 손동작과 엄지 및 검지손가락의 움직임으로 윈도우 시스템의 마우스를 제어하는 방법을 제안한다. 특히, 사용자 모니터의 모든 부분에 한번의 움직임으로 커서를 위치시킬 수 없는 문제점이 있는데, 이는 트래킹 과정에서 처리속도를 빠르게 하기 위하여 입력영상의 해상도를 스크린의 크기에 비해 작게 설정하기 때문에 발생한다. 해결방법으로 마우스가 움직이는 관성을 활용하여 해상도가 큰 화면에도 마우스 포인터를 쉽게 이동시킬 수 있는 방법을 제안한다.

또한, 손동작을 이용하는 마우스제어 시스템에서는 실시간 처리가 필수적인 요건이므로 마우스 포인터의 위치를 결정하기 위한 계산이 빠르고 효율적이어야 한다. 그러나, 빠른 처리를 위하여 카메라 입력 영상에서 얻어진 손끝 좌표를 스크린 상의 한 좌표로 표현하게 되면 사용자가 원하는 위치로 마우스 포인터를 정확하게 이동하기 힘들 뿐 아니라 손의 작은 떨림에도 마우스 포인터가 심하게 진동하게 된다. 따라서, 약간의 연산이 필요하더라도 손 전체의 중심점을 계산하여 마우스의 좌표로 사용하는데 이 방법은 카메라 영상의 바깥으로 손이 움직일 때, 순간적으로 중심점을 정확히 계산할 수 없어서 마우스 제어가 어려워지는 단점이 있다. 이러한 문제점을 해결하기 위하여 본 연구에서는 손 전체를 인식하는 공간으로 카메라 영상에서 가상의 인식공간을 사각형으로 만들고 그 사각형을 벗어날 경우, 관성을 계산하는 마우스 움직임을 구현한다. 또한, 엄지와 검지에 의한 손끝의 움직임을 다양한 마우스이벤트로 활용하는 방법을 제안한다.

## 2. 관련 연구

손이나 손가락의 동작을 인식하여 컴퓨터의 입력으로 활용하는 연구는 최근 많이 진행되고 있다. 하유술 등은 사용자가 마우스를 사용하면서 느낀 불편한 점을 최소화시키기 위해 핑거마우스를 제안하였다. 이 방법은 두 대의 USB카메라를 사용하여 손을 인식하고 옵티컬 플로우(Optical Flow)방법을 기반으로 손동작의 움직임을 추적한다[1]. 이 방법은 카

메라와 모니터의 해상도가 크게 차이날 경우, 효율성이 떨어진다. 김건우 등은 동적 손동작 인식을 위하여 전처리, 손추적, 손모양 검출의 3단계로 나누었다. 그 중 전처리와 손모양 검출방법을 개선하여 성능을 향상시킨 동적 손동작인식방법을 제안하였다. 전처리 단계는 동적테이블을 이용하여 노이즈 제거의 성능을 높였고, YCbCr컬러공간을 이용한 기존의 피부색 검출방식에서 피부색의 범위를 조절할 수 있도록 하여 피부색 검출 성능을 높였다. 또한 손모양 검출에서 가이드라인을 이용한 동적 손동작 인식방법을 이용하여 인식속도를 향상시켰다[2]. 이원주는 웹캠을 이용하여 사용자의 안면이나 손동작을 인식하고 그 제스처가 나타내는 감성(회노애락)을 아이콘형태로 표현하여 상대방에게 전송하는 감성메신저를 구현하였다[3].

박상윤 등은 손 추출을 위한 방법으로 피부색과 경계에너지 정보를 이용하고 모멘트 메소드로 손바닥 중심을 구하였다. 또한 병렬신경망을 이용하여 손형상을 인식하였고 패턴을 추출하기 위하여 Fitting ellipses 방법을 이용하였다. 손동작 추적과 인식을 위해 손동작의 위치정보를 예측할 수 있는 Kalman Filter을 이용하였으며, 이 방법들이 유용함을 확인하였다[4].

강성원 등은 손 모양과 손끝 정보만을 가지고 선행학습이 요구되지 않는 간편한 인터페이스 구현방법을 제안하였다. 상황에 따라 손 끝 점을 절대적 좌표로 활용하여 손이 웹캠에서 벗어날 때 움직임을 자연스럽게 연결시켰다[5]. 손수원 등은 수신호 인식기에 쓰이기 위한 분별성 있는 손동작을 만드는 방법을 제안하였다. 기존의 수화 DB에서 손의 움직임을 분석하고 기본 동작이 되는 4가지 모션 프리미티브를 선정하고 이를 조합하여 기본 손동작 집합을 제작하였다[6].

이정진 등은 증강현실 응용을 위한 손동작 인식기법을 제안하였고 정승대 등은 가전제품에 활용할 수 있는 손동작에 대한 평면 추정방법을 제안하였다[7,8].

결국, 손이나 손가락을 인식하여 컴퓨터의 입력방법으로 활용하는 시스템은 사용이 쉬우면서 인식율이 높아야하고 인식시간이 짧아야 실용성이 있다고 하겠다.

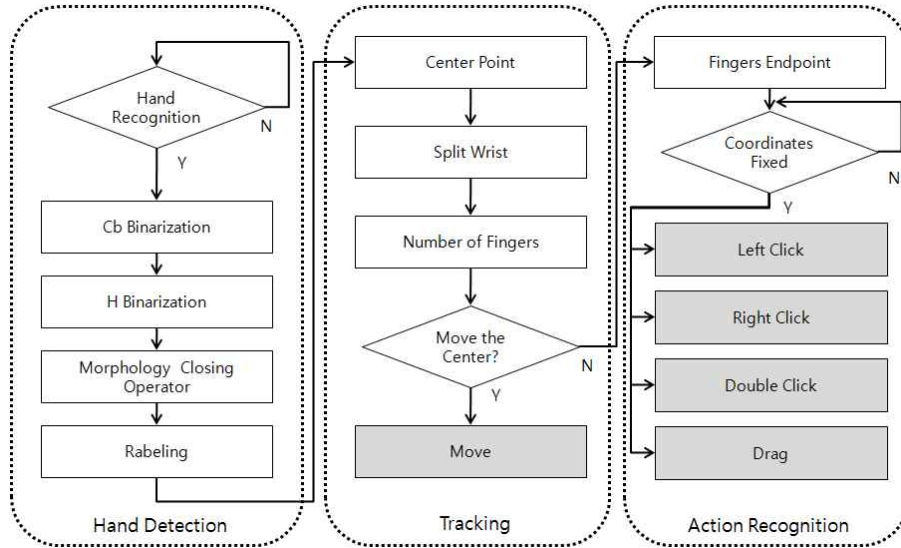


그림 1. 전체 시스템 흐름도

### 3. 손동작 인식에 의한 마우스제어 시스템

#### 3.1 전체시스템 구성도

그림 1의 전체 시스템은 크게 세 부분으로 나눌 수 있다. 첫 번째로 사용자의 피부색 및 환경에 관한 정보와 손의 동작변화로 인한 피부 영역의 국부적인 조명변화에 관한 정보를 실시간으로 학습하는 과정이다. 두 번째로 학습과정을 통해 생성된 2차원 룩업 테이블을 사용하여 피부영역의 확률분포를 얻어 손의 이동을 추적하고 손 영역의 크기와 방향정보를 이용하여 관심영역을 제한하며 손끝 좌표를 빠르게 검출하는 트래킹과정이다. 세 번째로는 두 과정을 거쳐 얻어진 각 손끝 좌표와 중심점의 거리 변화를 계산하여 이벤트를 제어하는 과정이다.

#### 3.2 손 객체의 추출

윤곽선 추출은 영상으로부터 발견되어진 물체의 가장자리의 픽셀 값의 위치를 복구하는 과정이다. 아래 그림은 원래 영상 안에서 그려진 손의 윤곽선을 보여준다. 마지막 단계에서는 똑같은 덩어리로 픽셀들을 그룹화하고 관련 있는 구성 요소 알고리즘을 적용하여 그림 2와 같은 손의 덩어리 표현을 얻는다.

그림 3의 영상(a)는 색상 단편화에 의하여 얻어진 손의 덩어리이다. 영상(b)는 타원형의 구조요소를 가진 모폴로지 개방 연산을 적용함에 의하여 복구되어

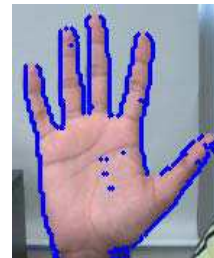
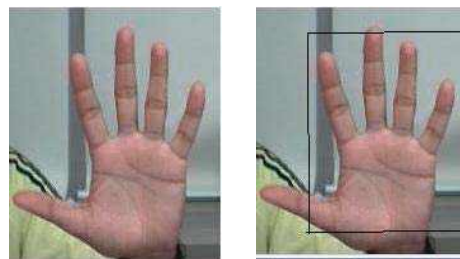


그림 2. 찾아진 손으로부터 추출된 윤곽선



(a) 손 객체 (b) 모폴로지 연산 후



(c) (a)-(b) 연산 후

그림 3. 손가락 덩어리 탐지를 위한 모폴로지 연산

지고 손으로부터 단편화된 손바닥이다. 영상(a)로부터 영상(b)를 빼기 연산함에 의하여 손가락들이 영상 안에서 발견 되어 질 수 있다.

효과적인 손 검출을 위하여 cascade Adaboost기법을 이용하여 손 모양을 검출하였다[9]. Adaboost는 얼굴 검출을 위하여 개발되었지만 본 연구에서는 고정된 손 모양을 검출하기 위하여 사용하였다. Adaboost의 경우 정적이며 변화가 없는 물체에 대한 검출에는 매우 효과적이지만, 손과 같이 그 모양이 변하고 시점이 변화에 따라 다른 모양을 나타내는 객체를 검출하는 것은 쉽지 않다. 이러한 문제를 해결하기 위하여 고정된 손 모양을 종류별로 학습하도록 하였다. 또한 기존의 알고리즘은 작은 회전에는 검출률이 많이 떨어지는데 이를 보완하기 위하여 기존의 Harr 특징 뿐만 아니라 확장된 Harr 특징을 이용하였다[10]. 사용된 손 모양은 손바닥, 주먹을 사용하였고 각각에 대하여 Positive 이미지(손바닥 3299개, 주먹 2670개), Negative 이미지 1942개를 사용하여 학습하였다. 그림 4는 손모양 검출 결과를 보여 준다.

Adaboost 검출은 매우 빠른 편이지만 영상에서 손을 찾기 위해서 이미지 전체에 대해서 검색을 해야 한다. 이를 개선하기 위해서 차영상 기반 ROI(Region of interest) 설정을 통하여 성능을 향상시켰다. 이전 영상과 현재 영상의 차를 이용하여 움직임이 일정 이상인 경우 움직임이 있는 것으로 판단하고 그 주변을 ROI로 설정한다. ROI 영역에 대해서만 adaboost 검출을 함으로써 좀 더 효율적으로 손을 검출하였다.

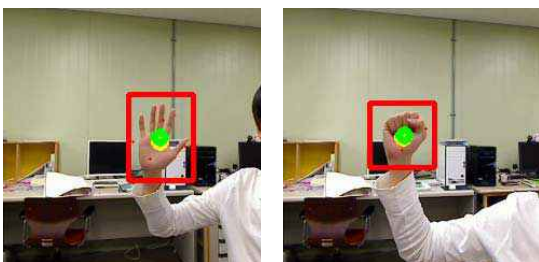


그림 4. Adaboost를 이용한 손 모양 검출

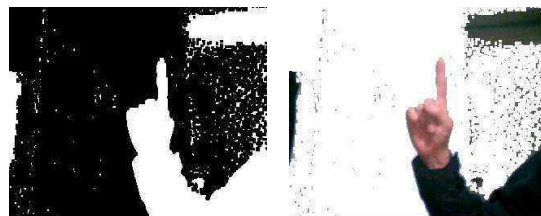
### 3.3 손 영역과 배경 분리 방법

복잡한 배경환경 속에서 손과 컬러분포가 유사한 특징을 가지는 객체가 존재할 수 있다. 따라서 우선 배경을 먼저 캡처해 등록하는 절차가 필요하다.

손이 웹캠의 영역 안으로 들어오면 배경 영상과 손이 있는 영상을 비교하여 그림 5(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 5(b)와 같은 영상을 생성한다. 관심범위가 손에 근접한 영역으로 축소됐기 때문에 이진화에 따른 손 영역부분 손실이나 오차를 줄일 수 있다.

그림 5(b)와 같이 배경을 분리한 컬러영상을 가지고 손영역을 이진화하기 위해서 RGB 컬러공간과 YCrCb 공간을 활용한다. 이진화한 영상은 그림 6(a)와 같다. RGB 공간에 대한 이진화는 피부영역을 추출하는데 탁월한 알고리즘이다. RGB 공간은 빛에 따른 영향이 심하기 때문에 조명환경에 적용할 수 있는 이진화 방법이 필요하다. 이러한 단점을 보완하고자 RGB 컬러공간을 YCrCb 컬러공간으로 변환하여 한번 더 이진화시켰다. RGB 컬러공간을 YCrCb 컬러공간으로 나누기위해서 식(1)의 3가지 조건을 모두 만족하는 경우와 그렇지 않은 경우로 이진화하였다[6].

$$\begin{aligned}
 & \text{if } ((R < 95, G > 40, B > 20) \text{ and} \\
 & \quad (\text{MAX}(R,G,B) - \text{MIN}(R,G,B) < 15) \text{ and} \\
 & \quad (|R-G| > 15, R > G, R > B)) \\
 & \text{then} \\
 & \quad (\text{TRUE} \rightarrow 255) \\
 & \text{else} \\
 & \quad (\text{FALSE} \rightarrow 0)
 \end{aligned} \tag{1}$$



(a) 차영상 (b) 손과 배경 분리

그림 5. 배경과 움직이는 사물의 분리

Cr 공간과 Cb 공간에 대해서 반복적 이진화 기법을 적용시켰다. 반복적 이진화를 적용하는 이유는 웹캠의 위치가 변하거나 조명 환경이 바뀔 때마다 임계치 값을 변화시켜 환경 변화에 독립적으로 만들기 위함이다. Cr 공간과 Cb 공간에 대해 이진화한 결과를 OR 연산한 결과를 그림 6(b)에 나타낸다.

배경이 제거됐기 때문에 객체를 이진화시킬 수 있는 임계치 값을 얻을 수 있다. 두 방법을 이용하여 이진화한 공간을 다시 OR 연산하여 한 공간으로 압축시키면 이진화된 하나의 영상을 획득할 수 있다. 결론적으로 반복적 이진화를 통해 임계치 값을 바꿔 주어 환경에 독립적이면서, 동시에 RGB 공간의 이진화를 통해 피부색을 효율적으로 추출할 수 있다.

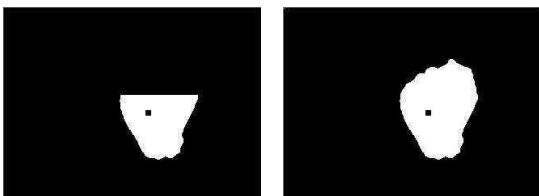


(a) RGB 컬러공간을 통한 이진화 (b) YCrCb 컬러공간을 통한 이진화

그림 6. 손영역 이진화

### 3.4 마우스 포인터의 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 제거할 수 있다. 그리고 제거된 이진화 영상을 가지고 중심모멘트 값을 구하면 그림 7의 (a)와 (b)처럼 고정된 중심모멘트 영상을 획득할 수 있다.



(a) 최대 가로 적용 (b) 최대 가로 비적용

그림 7. 고정된 중심 모멘트 영상

중심모멘트를 이용하여 마우스 좌표의 이동 방향과 거리를 설정한다. 이 방법은 이전 프레임과 현재 프레임간의 상대적인 변화를 관찰하는 방법이다. 두 프레임 간 중심모멘트 값의 차이를 비교한다. 비교한 값을 통해 손이 얼마만큼 이동하였는지 상대적인 결과를 얻을 수 있다. 이 결과를 가지고 현재 마우스가 스크린 상에 위치한 커서 값에 더한다. 그리고 그 결과를 좌표위치로 리턴시켜 최종적으로 그림 8과 같이 마우스 포인터를 움직일 수 있다. 상대적인 변화를 이용하는 방식은 손가락 모양과 무관하게 포인터를 이동할 수 있다.

좌표를 이동하는데 제한해야할 사항은 손 전체가 촬영되는 영상 안에 있어야 한다는 점이다. 만약 손의 일정 부분이 촬영되지 않는다면 중심모멘트 값을 구하는데 문제가 발생하므로, 촬영되는 영상의 가장자리 부분에 손이 접근하면 마우스 포인터의 움직임이 비정상적이 된다. 마우스의 이동을 살펴보면 동일한 거리를 손이 이동하는데 마우스 포인터가 상대적으로 덜 이동하는 것을 볼 수 있다. 이를 해결하기 위한 방법으로 카메라 영상에서 자연스러운 움직임의 손을 인식하는 영역을 지정할 필요가 있다.



그림 8.상대적 좌표 이동 방법

### 3.5 인식영역 범위 밖으로의 마우스 이동

스크린의 모든 범위에 커서를 위치시키기 위해서 적절하게 커서 활동 범위를 제한한다. 손의 중심점이 활동 범위 밖으로 벗어날 때의 방향과 관성을 이용하여 커서를 자동으로 이동시켜준다. 본 영상은 1초에 30 프레임씩 읽어 들이며, 활동 범위를 벗어나기 전 프레임의 중심 좌표와 벗어난 후 현재 프레임 중심 좌표의 차이를 연산하여 속도와 방향을 결정한다. 연산한 결과를 현재 프레임좌표에 연산하여 마우스 커서를 지속적으로 이동하게 한다. 이로써 해상도에 영

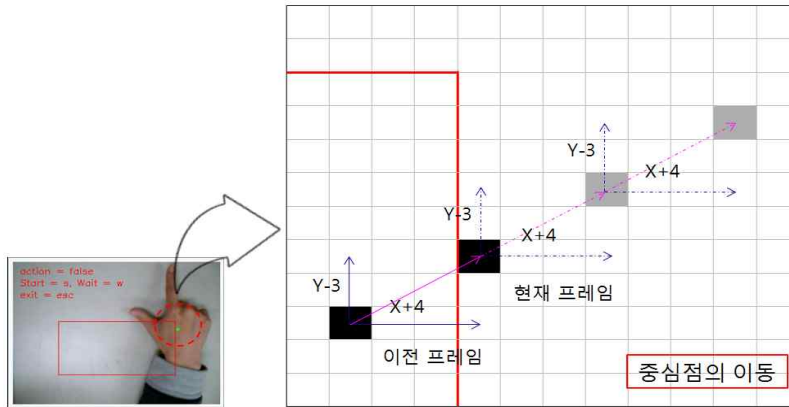


그림 9. 인식영역 사각형 밖에서 중심점 이동

향을 받지 않고 스크린 상의 모든 좌표에 접근이 가능하다. 다음은 중심점이 커서 활동 범위 안에서 밖으로 이동할 때 진행되는 그림 9와 공식이다.

$$F_N(x, y) = F_C(x, y) + \{F_C(x, y) - F_P(x, y)\}$$

$F_N$  = Next Frame  
 $F_C$  = Current Frame  
 $F_P$  = Previous Frame

(2)

식 (2)를 사용함으로써 모든 범위에 마우스 커서를 위치시킬 수 있다. 하지만 프레임간의 계산이므로 작은 움직임에도 약간의 오차가 발생한다. 이를 보완하는 방법으로 이전보다 앞 프레임을 같이 연산하면 오차를 줄일 수 있다.

### 3.6 동작에 따른 마우스 이벤트 제어

손동작과 마우스 이벤트의 연결은 다양한 방법이 있다. 손바닥을 펴서 마우스 제어를 하게 되면 5개의 손가락 위치를 모두 인식하고 각각의 움직임을 정밀히 측정해야하는 부담이 있어서, 여기서는 가장 편하게 움직일 수 있는 엄지와 검지 손가락 2개를 활용하도록 한다. 5개의 손가락을 모두 사용하는 것이 아닌

2개의 손가락만을 가지고 마우스의 이벤트를 제어하기 때문에 훨씬 적은 오차를 가지고 실험에 임할 수 있었다. 또한, 손의 중심점과 손끝과의 거리 정보를 가지고 마우스의 이벤트를 발생시켜서 쉽게 계산하며 오차를 줄일 수 있다. 그림 10은 마우스 이벤트를 나타내는 손동작들이다.

### 4. 실험 조건 및 결과 검토

본 논문에서는 범용 USB 카메라를 이용하였고 펜티엄4 CPU 2.93GHz에서 구현하였다. 320×240의 입력 영상을 사용하였고 처리 속도는 1초당 30 frame의 속도이다.

표 1에서와 같이 평균적으로 97.2%의 인식률을 보였다. 마우스 동작 구분은 기본적으로 사용하기 편한 검지와 엄지로 동작에 대응시켰다. 엄지손가락은 우클릭 동작에 대응시키고, 검지손가락은 좌클릭과 드래그 동작에 대응시켰다. 검지와 엄지손가락 2개로 더블 클릭에 대응시켰다.

동 작					
Mouse Event	Move	Left Click	Right Click	Double Click	Drag

그림 10. 손동작에 따른 마우스이벤트

표 1. 핸드 마우스 인식률

동작 모드	실험 횟수	인식 횟수	인식률(%)
커서 이동	1000	984	98.4
커서계속이동	1000	949	94.9
좌클릭	1000	978	97.8
우클릭	1000	976	97.6
더블 클릭	1000	988	98.8
드래그	1000	961	96.1

5. 결 론

본 논문에서는 손동작을 이용한 마우스의 포인터 이동과 이벤트 처리방법을 제안하였다. 특히, 카메라 영상에 손의 위치가 가장자리에 있을 경우의 부자연스런 마우스의 움직임은 해결하기 위하여 인식가능 영역을 따로 설정하였다. 또한, 제안한 커서 움직임 알고리즘은 카메라로부터 획득된 영상에서 실시간으로 손동작을 인식하여 그 움직임을 관성이 적용되도록 구현함으로써 해상도와 상관없이 모니터의 모든 위치에 커서를 위치시킨다. 제안한 방법은 빠른 수행속도를 보이고, 사용자가 원하는 모든 위치에 커서를 위치시킬 수 있다. 그러나, 컬러 정보만을 이용하기 때문에 살색 계통의 물체가 손과 함께 갑자기 나타날 경우 손을 잘 인식하지 못하고, 노란색 불빛인 경우, 마우스 움직임과 동작 인식률이 낮아지는 단점이 있다. 이러한 문제점은 향후 연구로 보완될 예정이다.

참 고 문 헌

[1] 하유솔, 고은지, 김명준, “손동작 인식 기반의 마우스 입력장치,” 정보과학회논문지, 제38권, 제10호, pp. 524-535, 2011.

[2] 김건우, 이원주, 전창호, “가이드라인을 이용한 동적 손동작 인식,” 대한전자공학회 논문지-CI, 제47권, 제3호, pp. 1-9, 2010.

[3] 이원주, “웹캠을 이용한 동적 제스처 인식 기반의 감성 메신저 구현 및 성능 분석,” 한국컴퓨터정보학회논문지, 제15권, 제7호, pp. 75-81, 2010.

[4] 박상윤, 이용주, “복잡한 영상에 강인한 손동작 인식방법,” 멀티미디어학회논문지, 제13권, 제7호, pp. 1000-1015, 2010.

[5] 강성원, 김철중, 손원, “손동작인식을 통한 사용자에게 편리한 핸드마우스 인터페이스 구현,” 한국방송공학회 학술발표대회 논문집, 제2009권, 제11호, pp. 129-132, 2009.

[6] 손수원, 배정훈, 양철중, 왕한,고 한석, “영상기반의 안정적 수신호 인식기를 위한 손동작 패턴 설계방법,” 대한전자공학회논문지-SP, 제48권, 제4호, pp. 30-37, 2011.

[7] 이정진, 김종호, 김태영, “증강현실 응용을 위한 손 끝점 추출과 손동작 인식기법,” 멀티미디어학회논문지, 제13권, 제 2호, pp. 316-323, 2010.

[8] 정승대, 장경호, 정순기, “실시간 손동작 인식을 위한 동작 평면 추정,” 정보처리학회논문지B, 제16권, 제5호, pp. 347-358, 2009.

[9] Paul Viloa and Michael Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” *CVPR*, Vol. 1., pp.I-511-I-518, 2001.

[10] R. Lienhart and J. Maydt, “An Extended Set of Harr-like Features for Rapid Object Detection,” *IEEE ICIP 2002*, Vol. 1, pp. 900-903, 2002.



김 정 인

1991년 4월~1993년 3월, 게이오 대학 계산기과학전공 공학석사  
 1993년 4월~1996년 3월, 게이오 대학 계산기과학전공 공학박사

1996년 5월~1998년 2월, 포항공과대학교 정보통신연구소 연구원, 기계번역시스템 설계  
 1998년 3월~현재, 동명대학교 컴퓨터공학과 부교수  
 관심분야: 기계번역, 기계학습, 시멘틱웹, 웹2.0