



## 웹 카메라와 손을 이용한 마우스 기능의 구현

Implementation of Mouse Function Using Web Camera and Hand

---

저자 (Authors)	김성훈, 우영운, 이광의 Seong-Hoon Kim, Young-Woon Woo, Kwang-Eui Lee
출처 (Source)	<a href="#">한국컴퓨터정보학회논문지 15(5)</a> , 2010.5, 33-38 (6 pages) <a href="#">Journal of the Korea Society of Computer and Information 15(5)</a> , 2010.5, 33-38 (6 pages)
발행처 (Publisher)	<a href="#">한국컴퓨터정보학회</a> The Korean Society Of Computer And Information
URL	<a href="http://www.dbpia.co.kr/Article/NODE06528152">http://www.dbpia.co.kr/Article/NODE06528152</a>
APA Style	김성훈, 우영운, 이광의 (2010). 웹 카메라와 손을 이용한 마우스 기능의 구현. 한국컴퓨터정보학회논문지, 15(5), 33-38.
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 구독계약을 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.

## 웹 카메라와 손을 이용한 마우스 기능의 구현

김 성 훈\*, 우 영 운\*\*, 이 광 의\*\*

# Implementation of Mouse Function Using Web Camera and Hand

Seong Hoon Kim\*, Young Woon Woo\*\*, Kwang Eui Lee\*\*

### 요 약

본 논문에서는 USB 인터페이스 방식의 웹 카메라를 통해 입력받은 영상을 영상처리 기법을 통해 손의 움직임과 손가락 개수를 파악하여 실시간으로 마우스의 기능을 구현하는 알고리즘을 제안하였다. 웹 카메라로부터 입력받은 RGB 컬러모델 영상을 조명 변화에 강한 YCbCr 컬러 모델 영상으로 변환하여 휘도 성분을 제외한 색차 성분만으로 피부색을 추출해 이진화된 영상으로 만든다. YCbCr 컬러 모델을 이용하여 피부색을 추출할 경우, 주변 환경에 의해 정확한 손 영역을 추출할 수 없어 라벨링(labeling)과 열림(opening) 연산, 닫힘(closing) 연산을 수행하여 정확한 손 영역을 추출한다. 이렇게 추출된 손 영역의 중심을 이용하여 마우스 포인터를 이동시키며 손가락 개수를 이용하여 마우스의 클릭을 수행하였다. 구현된 제안 기법을 실험한 결과, 마우스 포인터 이동을 위한 기능 성공률은 평균 94.0%, 손가락 개수 인식률은 평균 96.0%로 실용화 가능성을 보였다.

### Abstract

In this paper, we proposed an algorithm implementing mouse functions using hand motion and number of fingers which are extracted from an image sequence. The sequence is acquired through a web camera and processed with image processing algorithms. The sequence is first converted from RGB model to YCbCr model to efficiently extract skin area and the extracted area is further processed using labeling, opening, and closing operations to decide the center of a hand. Based on the center position, the number of fingers is decided, which serves as the information to decide and perform a mouse function. Experimental results show that 94.0% of pointer moves and 96.0% of finger extractions are successful, which opens the possibility of further development for a commercial product.

▶ Keyword : 웹 카메라(Web Camera), 거리변환행렬(Distance Transform Matrix), 라벨링(Labeling), 벡터 내적(Inner Product)

• 제1저자 : 김성훈 교신저자 : 이광의

• 투고일 : 2010. 02. 10, 심사일 : 2010. 03. 04, 게재확정일 : 2010. 04. 21.

\* 경북대학교 컴퓨터정보학부 교수 \*\* 동의대학교 멀티미디어공학과 교수

※ 이 논문은 2010년 한국컴퓨터정보학회 제41차 동계학술대회에서 발표한 논문("Web Camera를 이용한 손 마우스")를 확장한 것임

※ 이 논문은 2009학년도 동의대학교 교내연구비에 의해 연구되었음(과제번호 : 2009AA200)

## I. 서론

손은 인간이 많이 사용하는 신체 부위중 하나로 편리한 인터페이스 도구로 활용된다. 손의 동작을 인식하고, 인식된 정보를 사용하면 실시간 휴먼 인터페이스를 구현할 수 있다. 이런 점에서 손동작 인식에 관해 많은 연구가 이루어지고 있다 [1][2][3]. 하지만 손의 움직임이 빠르고 복잡하기 때문에 실시간으로 정확하게 인식하기 힘들다. 따라서 이를 실용화하는데 많은 어려움이 있다[4].

손동작 인식 방법은 장비를 이용한 방법과 비전을 이용한 방법이 있다. 장비를 이용하는 방법은 사용자가 직접 장비를 몸에 착용하고, 장비에 부착된 센서를 이용하여 손동작을 인식하거나, 손에 마킹을 하여 이 마킹 점을 추출함으로써 손동작을 인식하는 것이다[5]. 이 방법은 정확하게 손동작을 인식할 수 있으나, 장비를 착용하거나 손의 마킹을 해야 하므로 사용자에게 불편함을 주는 단점이 있다. 장비 없이 비전을 이용하는 방법은 카메라로 획득한 2차원 영상을 이용하여 손 모양을 인식하는 방법이다[6]. 하지만 이 방법은 손가락의 움직임, 폐쇄영역, 손의 회전등으로 인한 형태상의 변화에 민감하다는 단점을 가지고 있다. 본 논문에서는 장비 없이 비전을 이용하는 방법으로 웹 카메라(Web Camera)를 통해 입력된 영상에서 손의 움직임과 손가락 개수를 파악하여 마우스 기능을 구현하고 실험을 통하여 실용화 가능성을 보인다.

## II. 시스템 구성

제안한 알고리즘은 그림 1과 같다. 웹 카메라는 영상을 입력 받아 컴퓨터로 영상을 전송한다. 컴퓨터는 웹 카메라를 통해 전송 받은 영상을 영상처리 기법을 통하여 마우스 기능을 수행한다. 영상처리 과정은 우선 배경과 손을 분리하여 손 영역을 추출한다. 정확한 손 영역을 추출하기 위해 라벨링(labeling)과 열림(opening) 연산, 닫힘(closing) 연산을 수행한다. 손 영역 추출 후, 손의 중심을 파악한다. 특정 영역 안에 중심이 있을 경우 음함수 원의 방정식을 이용하여 손가락 개수를 파악하며 손가락 개수 파악 과정 중 나타나는 손목부분은 벡터 내적을 이용하여 제외시킨다. 손가락 개수에 따른 마우스의 클릭기능을 수행한다. 특정 영역 안에 중심이 없을 경우 손 중심의 위치에 따라 마우스 포인터가 이동하게 된다. 그리고 손 영역이 없거나 마우스 기능을 수행하면 다시 컴퓨터가 웹 카메라를 통해 영상을 입력받는다.

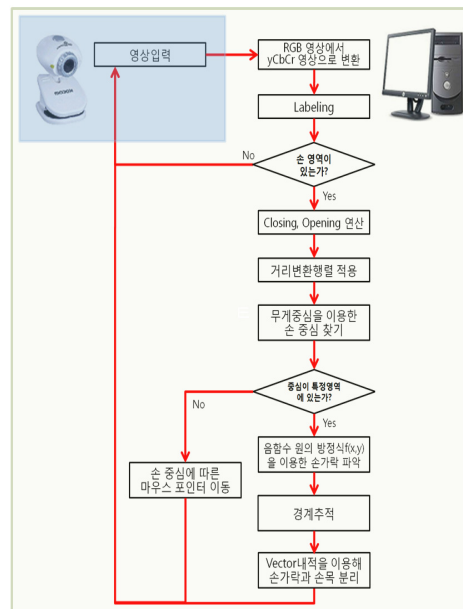


그림 1. 시스템 구성도  
Fig. 1. System architecture

### 2.1 손 영역 추출

웹 카메라로부터 입력받은 RGB 컬러모델 영상을 YCbCr 컬러모델 영상으로 변환한다. RGB 컬러모델은 영상의 전체적인 밝기의 변화에 매우 민감한 반응을 보인다. YCbCr 컬러모델의 Cb, Cr 정보는 RGB 컬러모델에 비해 밝기의 변화에 덜 민감하기 때문에 밝기의 변화에도 안정적인 결과를 보인다. 본 논문에서는 영상을 YCbCr 컬러모델 영상으로 변환하여 손 영역을 배경과 분리한다.

웹 카메라로부터 입력받은 RGB 컬러모델 영상을 YCbCr 컬러모델 영상으로 변환하여 수식 (1)과 같이 손 영역을 추출하여 이진 영상을 만든다[7].

$$30 \leq Y \leq 100$$

$$100 \leq Cb \leq 129 \dots\dots\dots (1)$$

$$128 \leq Cr \leq 137$$

YCbCr 컬러모델을 이용하더라도 그림 2와 같이 주변 환경에 의해 잡음이 발생하면 정확한 손 영역을 추출하기 어렵다. 따라서 잡음을 제거하기 위해 라벨링을 수행한다. 이진 영상을 라벨링하는 가장 간단한 방법은 Grassfire 알고리즘이다[8].

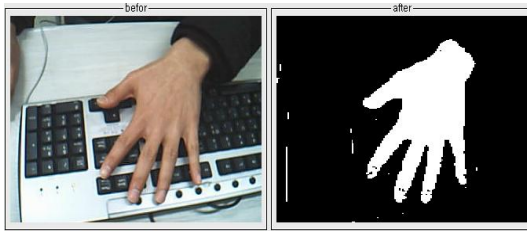
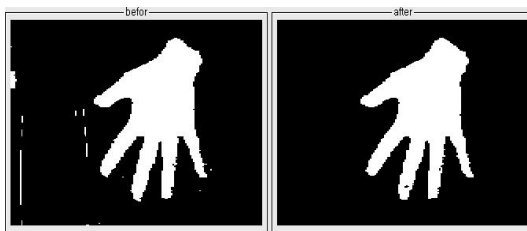


그림 2. 배경과 손 분리  
Fig. 2. Separation of background and hand area

Grassfire 알고리즘은 마른 잔디(grass)에서 불(fire)이 번져가는 모양과 비슷하게 화소를 라벨링하기 때문에 붙여진 이름이다. 이 방법은 자기호출(recursive call)을 이용하여 모든 인접요소가 라벨링될 때 까지 현재 관심화소의 주변 인접화소를 차례로 검사하면서 라벨링하는 방법이다. Grassfire 방법은 자기호출을 사용하기 때문에 함수가 자기 자신을 반복해서 호출한다. 과도한 자기호출은 시스템 스택(system stack)을 넘치게(overflow) 하기 때문에 너무 큰 크기의 물체영역을 라벨링하기에는 적당하지 않다. 또한 자기호출은 프로그램이 내부적으로 처리하는 많은 명령들로 인해 속도가 저하되는 단점이 있다[9].

본 논문에서는 자기호출을 사용하는 대신 고속으로 동작하면서 아주 큰 영역을 라벨링하는 것도 가능한 스택을 이용한 반복문 형태를 사용한다. 그림 3과 같이 라벨링을 통해 잡음을 제거한다. 라벨링을 통해 잡음을 제거 하더라도 그림4와 같이 정확한 손 영역을 추출하기 어렵다.



(a) 연산 전(before operation) (b) 연산 후(after operation)

그림 3. 라벨링 연산을 통한 잡음 제거  
Fig. 3. Noise removal using labeling operation

정확한 손 영역을 추출하기 위해 열림 연산과 닫힘 연산을 수행한다. 열림 연산의 경우 먼저 침식(erosion) 연산을 수행 후 그 결과 영상에다 다시 팽창(dilation) 연산을 수행하면 되고, 닫힘 연산의 경우 먼저 팽창 연산을 수행 후 그 결과 영상에다 다시 침식 연산을 수행하면 된다. 열림 연산의 경우

영역 내의 갈라진 틈(gap)들을 제거하는 효과가 있으며, 닫힘 연산의 경우 영역 내의 갈라진 틈들을 확장하는 효과가 있다. 그림 5, 그림 6과 같이 열림 연산과 닫힘 연산을 통해 정확한 손 영역을 추출한다.

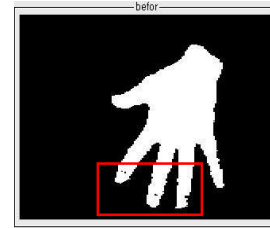
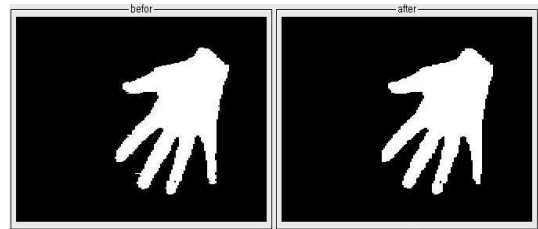
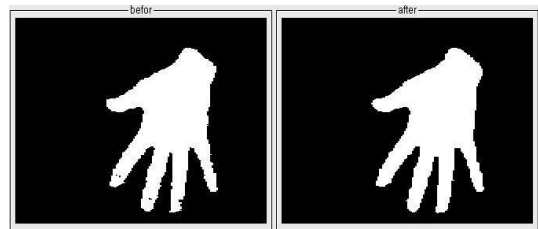


그림 4. 라벨링 수행 후 영상  
Fig. 4. Image after labeling operation



(a) 연산 전(before operation) (b) 연산 후(after operation)

그림 5. 열림 연산  
Fig. 5. Opening operation



(a) 연산 전(before operation) (b) 연산 후(after operation)

그림 6. 닫힘 연산  
Fig. 6. Closing operation

## 2.2 손 중심 찾기

정확한 손 영역을 추출한 후, 마우스 포인터의 이동을 위해 손의 중심을 찾는다. 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 위

치정보를 이용하여 중심 좌표  $C(x, y)$ 는 수식 (2)를 사용하여 계산된다.

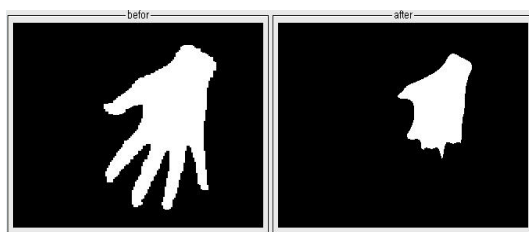
$$C(x, y) = \frac{\sum_{m=0}^N P_m(x, y)}{N} \dots\dots\dots (2)$$

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

식 (2)를 사용하는 경우, 손가락이 포함된 영상에서 계산되는 중심 좌표  $C(x, y)$ 는 손가락의 움직임에 의해 중심좌표가 움직이게 된다. 거리변환행렬을 사용하여 중심 좌표  $C(x, y)$ 를 계산한다. 수식 (3)의 행렬식에 의해 변환되는 픽셀 값들 중 가장 큰 값을 가진 픽셀이 영상의 중심 좌표가 된다. 손가락의 움직임이 있더라도 중심 좌표  $C(x, y)$ 는 크게 벗어나지 않는다.

$$D = \begin{bmatrix} 4.5 & 4.0 & 3.5 & 3.0 & 3.5 & 4.0 & 4.5 \\ 4.0 & 3.0 & 2.5 & 2.0 & 2.5 & 3.0 & 4.0 \\ 3.5 & 2.5 & 1.5 & 1.0 & 1.5 & 2.5 & 3.5 \\ 3.0 & 2.0 & 1.0 & 0.0 & 1.0 & 2.0 & 3.0 \\ 3.5 & 2.5 & 1.5 & 1.0 & 1.5 & 2.5 & 3.5 \\ 4.0 & 3.0 & 2.5 & 2.0 & 2.5 & 3.0 & 4.0 \\ 4.5 & 4.0 & 3.5 & 3.0 & 3.5 & 4.0 & 4.5 \end{bmatrix} \dots\dots\dots (3)$$

그림 7, 그림 8과 같이 위의 거리변환행렬을 사용하여 중심 좌표를 계산한다.



(a) 연산 전(before operation) (b) 연산 후(after operation)

그림 7. 거리 변환 행렬 연산  
Fig. 7. Distance transform matrix operation

## 2.3 손가락 파악

마우스 클릭 기능을 구현하기 위해 손가락 개수를 파악한다. 손가락 개수 파악은 수식 (4)와 같이 음함수 원의 방정식을 이용하여 손등 또는 손바닥과 손가락은 분리한다.



그림 8. 중심 좌표 찾기  
Fig. Finding a center of a hand area

$$f(x, y) = x^2 + y^2 - r^2$$

$$f(x, y) < 0 \text{ 원의 내부}$$

$$f(x, y) = 0 \text{ 원의 경계선} \dots\dots\dots (4)$$

$$f(x, y) > 0 \text{ 원의 외부}$$

음함수 원의 방정식을 이용하여 영상의 모든 좌표를 방정식에 대입해 그림 9와 같이 두 원 사이의 손가락 영역을 찾게 된다. 원의 방정식을 이용해 손가락 영역을 추출한 후, 손가락 개수 파악을 위해 경계 추적을 한다. 경계추적을 통하여 영상에서 연결된 영역을 하나의 손가락으로 판단한다. 연결된 영역마다 중심 좌표를 이용하여 각 손가락의 중심 좌표를 찾는다. 그림 10과 같이 경계추적과 중심 좌표를 이용해 손가락 개수를 파악한다.

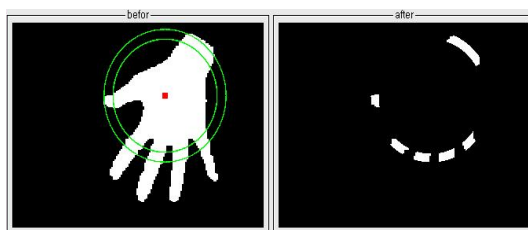


그림 9. 원의 방정식을 이용한 손가락 영역 추출  
Fig. 9. Extraction of finger areas using a circle equation

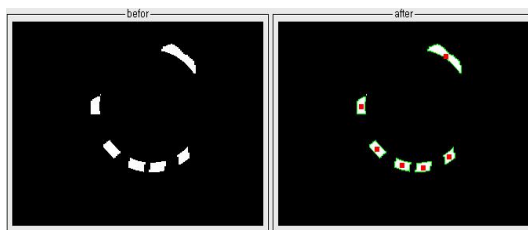


그림 10. 손가락 개수 파악  
Fig. 10. Calculation of the number of fingers

경계추적을 통해 손가락을 추출하면 손가락과 손목부분이 함께 나타난다. 손목부분이 영상에 계속 없을 경우에는 문제가 되지 않지만 영상에 나타날 경우 손가락 개수 파악에 문제가 발생한다. 이러한 문제를 해결하기 위해 벡터 내적을 이용하여 손목부분을 손가락 개수에서 제외시킨다. 손의 중심에서 손가락(손목부분 포함)으로 향하는 각 벡터를 구한다. 각 벡터를 구한 후, 벡터 내적을 이용하여 각 벡터 마다 자기 자신을 제외한 사이 각도가 가장 작은 벡터를 구한다. 각 벡터마다 사이 각도가 가장 작은 벡터를 구한 후, 벡터 내적 공식에서  $\cos\theta$ 값의 범위(-1 ~ 1)를 이용하여 손목을 제외시킨다. 손가락 사이의 각도는 최대 90°이하 이며 손목과 손가락 사이의 최소 각도는 90°이상 이라는 사실을 이용하여  $\cos\theta < 0$  인 영역을 손가락 개수 파악에서 제외시킨다. 그림 11, 수식 (5)와 같이 벡터 내적을 이용하여 손목부분을 손가락 개수에서 제외시킨다.

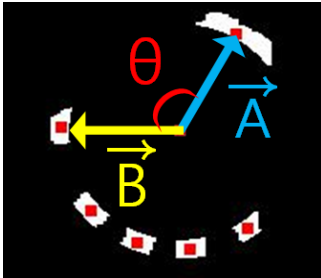


그림 11. 벡터 내적을 이용한 손가락 개수 파악  
Fig. 11. Calculation of the number of fingers using inner product

$$\vec{A} \cdot \vec{B} = |\vec{A}| |\vec{B}| \cos \theta \quad (5)$$

$$\text{여기서, } \cos \theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| |\vec{B}|} = \frac{a_1 b_1 + a_2 b_2}{\sqrt{a_1^2 + a_2^2} \sqrt{b_1^2 + b_2^2}}$$

### III. 실험 결과 및 고찰

본 논문에서는 KOCOM회사의 KMC-90 VGA USB 웹 카메라 모델을 사용하였고 Pentium(R) 4 CPU 2.80GHz, 1.00GB RAM에서 구현하였다. 제안한 알고리즘을 효율적으로 구현하고 정확한 실험을 하기 위해 그림 12와 같이 손 마우스 프로그램을 제작하여 사용하였다.

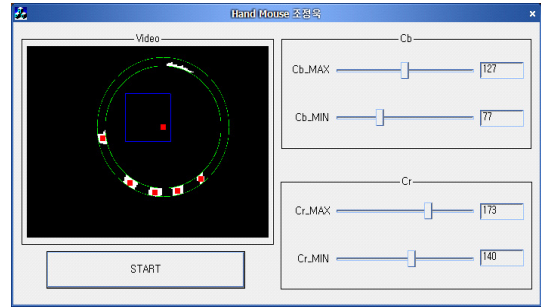


그림 12. 제작된 손 마우스 프로그램  
Fig. 12. Implemented hand mouse program

거리변환 행렬은 본 논문에서 제시한 7×7행렬이 아닌 21×21행렬을 사용하였다. 7×7행렬을 사용할 경우 손가락영역이 나타났으며 최대한 손가락 영역을 줄이기 위해 21×21행렬을 사용하였다.

마우스 포인터의 자연스러운 이동을 위해 설정한 파란색 사각형 영역 안에 손 중심이 있을 경우, 마우스 클릭 기능이 수행되며 마우스 포인터 이동은 수행되지 않는다. 설정한 파란색 사각형 영역 안에 손 중심이 없을 경우, 마우스 클릭 기능이 수행되지 않으며 마우스 포인터 이동이 수행된다.

표 1. 손가락 개수 인식률

Table 1. Recognition rate of the number of fingers

손가락 개수	손목이 포함되지 않은 영상	손목이 포함된 영상	평균(%)
1	100	100	100
2	97	97	97
3	95	95	95
4	94	94	94
5	94	94	94
평균	96.0	96.0	96.0

표 2. 마우스 기능 성공률

Table 2. Success rate of mouse function

기능	성공률(%)
마우스 포인터 이동	94.0
클릭	67.0
드래그	62.0

표 1에서와 같이 손목이 포함되지 않은 영상과 손목이 포함된 영상 모두 같은 인식률을 보였으며 손가락 개수가 많아질수록 인식률은 낮아진다는 것을 볼 수 있다. 마우스 클릭 기능은 손가락 개수에 따라 좌 클릭과 우 클릭을 수행하게 설

정하였다. 손가락의 개수가 4개일 경우 좌 클릭을 수행하며 손가락의 개수가 2개일 경우 우 클릭을 수행한다. 표 2에서와 같이 마우스 포인터 이동은 94%의 성공률을 보였으나 동작이 복잡해질수록 성공률은 낮아진다.

#### IV. 결론

본 논문에서 제안한 손 마우스 알고리즘은 웹 카메라로부터 입력된 영상에서 실시간으로 손을 인식하여 마우스 대신으로 사용함으로써 마우스 포인터 이동 및 마우스 클릭 기능을 대신한다. 실험 결과 제안한 방법은 안정적으로 마우스 포인터를 제어 할 수 있었다. 그러나 기존의 마우스보다 정교한 제어를 할 수 없으며 컬러 정보를 이용하기 때문에 피부색과 유사한 색의 물체가 손과 함께 나타날 경우 손을 잘 인식하지 못하고, 손가락의 개수가 빠르게 변하거나 손가락이 겹쳐질 경우 개수의 인식률이 낮아지는 단점이 있다. 이러한 문제점은 향후 연구 및 보완되어야 할 것이다.

#### 참고문헌

- [1] 이동욱, 김수동, 이동석, 유지상, "마우스 포인터 제어를 위한 실시간 손 인식 알고리즘," 한국방송공학회 학술발표대회논문집, 211-214쪽, 2008년 11월.
- [2] 김상호, "손 제스처를 사용한 홈 엔터테인먼트 시스템을 위한 비전 기반의 인터페이스," 경북대학교대학원 석사학위논문, 2004년.
- [3] 김주황, 박정훈, 전중서, 강민구, 이승현, "OpenCV를 이용한 마우스의 손 동작 인식," 한국인터넷정보학회 학술발표대회 논문집, 제20권, 303-306쪽, 2009년 10월.
- [4] 김종민, "예지 방향성 히스토그램을 이용한 실시간 손 형상 인식," 전남대학교대학원 석사학위논문, 2004년.
- [5] 김인철, 이남호, 이용범, 진성일, "Polhemus 센서의 궤적 정보 해석을 이용한 스트로크 기반의 손 제스처 인식," 전자공학회논문지C, 제36권, 제8호, 46-53쪽, 1999년 8월.
- [6] 김건우, 이원주, 전창호, "웹캠을 이용한 손동작 인식 방법," 대학전자공학회 학술대회논문집, 619-620쪽, 2008년 6월.
- [7] N. Otsu, "A threshold selection method from gray-level histogram," IEEE Trans. on System Man Cybernetic, Vol.9, No.1, pp.62-66, Jan., 1979.

- [8] 강 동중, 하 중은, "Visual C++을 이용한 디지털 영상 처리," 사이텍미디어, 서울, 2003년 3월.
- [9] 이정현, 이준형, "영상처리 알고리즘 구현에서 소스코드 재사용을 위한 제너릭프로그래밍 방법에 관한 연구," 한국컴퓨터정보학회논문지, 제 10권, 제 3호, 19-34쪽, 2005년 3월.

#### 저 자 소 개



##### 김 성 훈

1996년 2월 : 연세대학교 본대학원  
전자공학과(공학박사)  
1996년 3월~2006년 2월:  
영동대학교 컴퓨터공학과 부교수  
2006년 3월~현재 :  
경북대학교 컴퓨터정보학부 조교수  
관심분야 : 인공지능, 패턴인식, 지능  
형콘텐츠



##### 우 영 운

1991년 8월 : 연세대학교 본대학원  
전자공학과(공학석사)  
1997년 8월 : 연세대학교 본대학원  
전자공학과(공학박사)  
1997년 9월~현재:  
동의대학교 멀티미디어공학과 교수  
관심분야 : 인공지능, 패턴인식, 퍼지  
이론, 의료정보



##### 이 광 의

1997년 8월 : 서강대학교 대학원 전  
자계산학과(공학박사)  
1997년 12월~2001년 2월 :  
한국전자통신연구원(선임연구원)  
2001년 3월~현재:  
동의대학교 멀티미디어공학과(부교수)  
관심분야 : 계산이론, 병렬처리, 인공  
생명, 기계학습