

# 휴먼마우스 구현을 위한 효율적인 손끝좌표 추적 및 마우스 포인터 제어기법

## Efficient Fingertip Tracking and Mouse Pointer Control for Implementation of a Human Mouse

박 지 영<sup>†</sup> 이 준 호<sup>\*\*</sup>  
(Ji-Young Park) (June-Ho Yi)

**요 약** 본 연구는 마우스의 입력을 사람의 손동작으로 대체하는 휴먼마우스 시스템을 위한 새로운 손 끝 좌표 추적기법과 마우스 포인터 움직임 결정법을 제안한다. 손끝좌표 추적에는 얼굴영역 추적을 위해 제안되었던 CAMSHIFT 알고리즘을 개선하여 적용하였다. 정확한 손 영역 검출을 위하여 각 사용자의 환경에 최적화된 피부영역 컬러 정보를 얻을 수 있는 실시간 피부영역 학습과 손의 자유로운 움직임으로 인하여 발생하는 손 영역의 크기 및 방향변화를 고려한 영역제한 기법을 적용하였다. 또한 손의 주축을 이용한 손끝좌표 계산법을 통해 빠르고 정확하게 손끝의 위치를 찾을 수 있다. 실시간 손끝좌표 검출에는 처리속도의 한계가 있어 얻어지는 좌표의 연속성이 결여되어 마우스 포인터의 움직임이 연결되지 않는 문제점이 있다. 연속적인 마우스 포인터의 움직임 표현을 위해 손끝좌표의 이동거리를 통해 마우스 포인터의 속도와 가속도를 계산하여 마우스 포인터의 운동방정식을 정의하고 이를 이용하여 마우스 포인터의 위치를 결정한다. 제안한 알고리즘을 적용하여 실험한 결과 빠르고 정확한 손끝좌표 추적과 마우스 포인터의 자연스러운 움직임이 가능함을 보였다.

**키워드** : 휴먼 컴퓨터 인터페이스(HCI), 휴먼마우스, 트래킹

**Abstract** This paper discusses the design of a working system that visually recognizes hand gestures for the control of a window based user interface. We present a method for tracking the fingertip of the index finger using a single camera. Our method is based on CAMSHIFT algorithm and performs better than the CAMSHIFT algorithm in that it tracks well particular hand poses used in the system in complex backgrounds. We describe how the location of the fingertip is mapped to a location on the monitor, and how it is both necessary and possible to smooth the path of the fingertip location using a physical model of a mouse pointer. Our method is able to track in real time, yet not absorb a major share of computational resources. The performance of our system shows a great promise that we will be able to use this methodology to control computers in near future.

**Key words** : Human Computer Interface, Human Mouse, Tracking

### 1. 서 론

현재 컴퓨터 사용자는 컴퓨터에 의사를 전달하는 수단으로 키보드, 마우스, 데이터 글로브와 같은 하드웨어를

이용한다. 이와 같은 유선 입력은 사용자의 행동과 장비 설치를 위한 공간상의 제약을 수반한다. 추가적인 하드웨어 없이 사람의 손동작을 컴퓨터의 입력수단으로 사용하면 컴퓨터를 좀 더 사용자 중심으로 만드는 새로운 인간 친화적인 인터페이스가 가능하다. 특히 사람의 손동작 중 검지만을 펴서 표현하는 포인팅 동작은 방향을 지시하기 위하여 가장 빈번히 사용되는 동작이다. 따라서 이와 같은 손동작을 추적하여 윈도우 시스템 사용을 위한 마우스 포인터의 움직임을 제어 할 수 있다면 사용자는 좀 더 자연스럽게 컴퓨터에게 의사를 전달 할 수 있다.

· 본 연구는 한국과학재단 목적기초연구(R01-1999-00339) 지원으로 수행되었음.

† 비 회 원 : 성균관대학교 정보통신공학부  
jiyp@ece.skku.ac.kr

\*\* 종신회원 : 성균관대학교 정보통신공학부 교수  
jihy@ece.skku.ac.kr

논문접수 : 2002년 2월 24일  
심사완료 : 2002년 9월 5일

본 연구는 손의 동작변화로 인하여 발생하는 손 영역의 국부적인 조명변화와 복잡한 주위배경으로부터 안정적으로 손끝 좌표를 추적하고 그 움직임을 사용자의 화면에 자연스럽게 표현하여 윈도우 시스템제어를 위한 입력 값으로 사용하는 것을 목적으로 한다.

마우스 포인터의 움직임을 제어하는 데에는 기본적으로 정확한 손 영역 분할 방법, 그리고 손의 동작변화에 무관하게 손끝 좌표를 안정적으로 검출하고 빠르게 추적하는 기술이 필요하다. 컴퓨터 비전을 기반으로 하여 손영역을 추적하는 기존의 연구들을 살펴보면 계산이 많고 복잡하여 실시간 적용이 어렵거나(손의 3차원 정보를 사용하는 경우[1,2]) 배경에 피부색과 유사한 분포가 있을 때 검출 성능이 현저히 떨어져[3,4] 실제 응용환경에는 적합하지 못하다. 손끝 좌표 추적을 통하여 얻은 결과를 윈도우 시스템제어에 적용하기 위해서는 실시간 추적은 물론 적은 자원의 사용으로 다른 프로그램들과 연동이 가능해야 한다. 본 연구는 피부영역 확률분포의 이동을 따라 얼굴을 추적하는 CAMSHIFT[3]알고리즘을 기반으로 하여 손끝좌표 추적이 적합한 새로운 방법을 제안한다. 사용자의 환경과 손의 동작변화로 인한 국부적인 조명변화에 최적화된 실시간 학습법, 계산 시간과 메모리의 사용을 줄이고 잡음의 영향을 최소화하는 정확한 손 영역 제한 기법, 그리고 빠른 손끝 좌표 검출방법이 본 연구가 제안하는 트래킹 기법의 특징이다.

기존의 연구들을 살펴보면 손끝 좌표가 정확히 검출되었다 하더라도 일반적인 PC환경에서의 적용시 다음과 같은 문제점으로 인하여 마우스 포인터의 움직임을 자연스럽게 표현 할 수 없다[2]. 첫 번째 문제점은 실시간 검지 좌표 검출의 처리속도의 한계로 인하여 얻어지는 좌표의 연속성이 결여되어 마우스 포인터의 움직임이 연결되지 않는다는 점이다. 두 번째 문제점은 트래킹 과정에서 처리속도를 빠르게 하기 위하여 입력영상의 해상도를 스크린의 크기에 비해 작게 설정하기 때문에 발생한다. 카메라 입력영상에서 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. 한편, 휴먼 마우스 시스템에서는 실시간 처리가 필수적인 요건이므로 마우스 포인터의 위치를 결정하기 위한 계산이 빠르고 효율적이어야 한다. 이러한 문제점을 해결하기 위하여 본 연구는 휴먼 마우스 시스템에서 손끝의 이동을 화면상에 자연스럽게 나타내기 위한 방법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2절에서는 제안된 시

스템의 전체 구성을 설명한다. 3절에서는 손끝 좌표 검출 및 추적을 위한 방법을, 4절에서는 이를 통해 얻어진 손끝 좌표를 입력으로 하여 화면상의 마우스 포인터를 자연스럽게 움직이도록 하는 방법에 관하여 기술한다. 5장에서는 제안된 트래킹 기법과 자연스러운 마우스 포인터 움직임 표현 기법을 적용한 실험 결과를 보여준다.

2. 전체 시스템 개요

그림 1의 전체 시스템은 크게 세 부분으로 나누어 볼 수 있다. 첫 번째로 사용자의 피부색 및 환경에 관한 정보와 손의 동작변화로 인한 피부 영역의 국부적인 조명변화에 관한 정보를 실시간으로 학습하는 과정이 있다. 두 번째로 학습과정을 통해 생성된 2차원 룩업테이블을 사용하여 피부영역의 확률 분포를 얻어 손의 이동을 추적하고 손 영역의 크기와 방향정보를 이용하여 관심영역을 제한하며 손끝 좌표를 빠르게 검출하는 트래킹 과정(점선 표시 영역)이 있다. 세 번째로는 두 과정을 거쳐 얻어진 불연속적인 손끝 좌표를 부드러운 마우스 포인터의 움직임으로 표현하기 위한 부분이 있다.

실시간 학습과정에서는 확률적인 모델링을 통하여 사용자의 피부색과 조명환경에 최적화된 2차원(hue, saturation) 룩업테이블을 구성한다. 학습과정 후 룩업테이블을 통해 피부영역의 확률 분포를 얻고 이 분포가 이동하는 방향을 따라 관심영역을 이동한다. 관심 영역은 손의 크기와 방향 변화를 고려하여 결정된다. 손끝좌표는 분할된 손 영역의 무게 중심 좌표와 손의 주축을 이용하여 빠르고 정확하게 얻을 수 있다. 이렇게 얻어진 검지 좌표를 입력으로 하여 마우스 포인터의 움직임을 표현한다. 마우스 포인터의 이동거리는 프레임간 손끝좌표의 이동거리와 카메라 입력영상의 해상도와 화면의

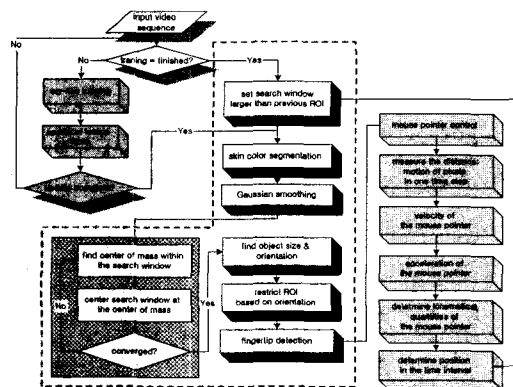


그림 1 전체 시스템 흐름도

해상도차이를 고려하여 결정한다. 이를 위해 손끝 좌표를 따라 움직이는 마우스 포인터의 운동 방정식을 정의하고 마우스 포인터의 위치를 결정한다. 이러한 과정을 통하여 입력좌표가 불연속적인 경우에도 자연스러운 움직임 나타내는 마우스 포인터를 표현할 수 있다.

### 3. 손끝 좌표 추적 기법

손끝 좌표 추적을 위하여 얼굴의 중심좌표 추적을 위해 제안된바있는 CAMSHIFT 알고리즘에서 피부영역검출과 관심영역 제한을 위한 방법을 개선하여 적용하였다. 복잡한 배경의 영향을 줄이는 피부영역 검출방법과 손의 다양한 동작변화를 따라 최적화되는 관심영역 설정을 통해 손 영역 추적의 정확성을 높였다. 이 절에서는 본 연구를 통해 제안된 빠르고 정확한 손끝좌표 추적을 위하여 사용된 여러 알고리즘에 관하여 자세히 다룬다. 피부영역 검출을 위한 실시간 피부영역 학습방법과 검출된 사용자의 손 영역에 최적화된 관심영역 설정 기법 그리고 빠른 손끝좌표 계산법을 설명한다.

#### 3.1 실시간 피부영역 학습

손동작을 분석하고 추적하기 위해서는 기본적으로 정확한 손 영역 분할을 위한 알고리즘이 필수적이다. 그러나 사람의 피부는 인종 혹은 개인의 특성에 따라 컬러 성분의 분포가 모두 동일하지는 않다. 따라서 다양한 피부에 대한 많은 데이터베이스를 사용하여 최적화된 컬러 분포를 얻어내는 사전학습이 필요하다. 그러나 사전학습과정을 통하여 사용자마다 다른 조명환경과 손의 다양한 동작 변화에 따라 달라지는 조명의 영향으로 인한 피부색 변화정보를 모두 학습하기에는 데이터의 양과 학습시간에 한계가 있다. 따라서 각 사용자의 환경과 손의 동작변화로 인한 국부적인 조명변화를 실시간으로 학습하는 방법을 사용한다. 이 방법은 사용자 각각에 최적화된 학습모델을 얻을 수 있는 강점이 있으며 손의 동작 변화에 따른 컬러분포의 변화 정보를 포함하여 조명의 변화에 능동적으로 대처할 수 있다. 그림 2는 손의 동작 변화에 따라 달라지는 조명의 영향으로 인하여 피

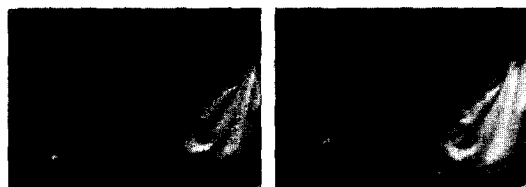


그림 2 손의 동작변화에 따른 피부색 변화정보 학습을 위한 실시간 학습과정

부색이 다르게 나타나는 것을 보이고 있다. 사용자는 이러한 피부색 변화 정보 학습을 위해 화면에 표시된 사각형의 내부에 피부영역이 포함되도록 손을 위치시키고 학습과정동안 손동작을 변화시켜 이에 따른 피부색 변화 정보를 학습하게 된다.

조명변화의 영향을 줄이기 위하여 입력 RGB 컬러 모델을 조명변화에 상대적으로 강인한 HSV 컬러 모델로 변환한다. "사람의 피부는 인종과 관계없이 같은 color(hue)를 가진다[3]"는 기존의 연구보고가 있었다. 그러나 hue값만을 사용하여 피부를 검출하는 경우 사용자의 주변 환경에서 흔히 발견될 수 있는 붉은 계통이나 피부와 유사한 색을 가진 배경요소의 영향을 크게 받으므로 사용자의 피부영역만을 검출하기 어렵다. 그림 3은 피부영역의 hue정보를 기반으로 피부영역을 검출하고 추적하는 CAMSHIFT 알고리즘을 손 영역 추적에 적용한 결과이다. 십자를 이루는 두 선과 그 교점은 각각 피부영역으로 판단된 범위와 그 중점을 나타내는데, 배경의 영향을 받아 손 영역을 검출하지 못함을 보였다. (a)는 사용자의 옷의 일부에 붉은색이 포함된 경우 붉은색을 모두 피부 영역으로 인식한 경우이며 (b)는 배경에 있는 피부색과 유사한 색을 가진 책장의 영향을 받아 손 영역을 전혀 검출하지 못하는 결과를 나타내고 있다. 이러한 문제점 해결을 위해 본 연구에서는 정확한 피부영역 검출을 위한 실시간 학습방법과 주변 잡음에 영향을 줄이는 최적화된 영역 제한 방법을 제안하였다.



(a) 사용자가 붉은 계통의 옷을 입은 경우의 입력영상(좌)과 검출결과(우)



(b) 배경에 피부색과 유사한 색의 기구가 있는 경우의 입력영상(좌)과 검출결과(우)

그림 3 CAMSHIFT 알고리즘을 적용한 피부영역 분할 결과

학습과정을 통해 물체 고유의 색을 나타내는 hue와 고유의 색에 대한 채도를 나타내는 saturation을 각각 0~255 사이의 값으로 정규화하여 히스토그램을 생성한다. 피부영역 검출을 위한 룩업테이블은 히스토그램을 기반으로 하여 식 (1)을 통해 얻어진 확률 값으로 구성한다. 2차원 룩업테이블은 hue와 saturation 두 축으로 구성되며  $L(h, s)$ 는 룩업테이블의  $(h, s)$  위치의 값을 나타낸다. 피부영역 분할과정에서는 이 룩업테이블의 값을 적용하여 입력영상의 픽셀이 피부에 속할 확률을 그레이 스케일로 나타내는 피부확률분포 영상을 얻는다.

$$L(h, s) = \frac{H(h, s)}{\max_{i, j=0}^{255} H(i, j)} \quad (1)$$

$H(h, s)$ 는 히스토그램에서 hue의 인덱스  $h$ 와 saturation의 인덱스  $s$ 의 값을 나타낸다.

### 3.2 최적화된 손 영역 제한

전체 영상 중 손 영역만을 효과적으로 추출하면 계산을 줄이고 배경의 영향을 최소화 할 수 있다. 본 연구에서는 정확한 손 영역 제한을 위하여 CAMSHIFT[3] 알고리즘을 개선하여 적용하였다. CAMSHIFT 알고리즘은 영역제한을 위해 현재 프레임의 관심영역내부에 존재하는 추적하고자 하는 영역의 크기에 비례하여 관심영역의 크기를 결정한다. 이 경우 관심영역에 잡음이 포함되면 그림 4에서와 같이 관심영역의 크기가 점점 커져 더욱 많은 잡음의 영향을 받게 된다.

그림 4를 통해 연속적인 세 개의 프레임 (a), (b), (c)에서 배경에 있는 피부색과 유사한 가구의 일부가 관심영역내부에 포함되는 경우 관심영역의 범위가 점진적으로 확대되어 추적중인 손 영역의 정보를 잃는 것을 볼 수 있다. 뿐만 아니라 방향의 변화가 자유로운 손을 추적하는 경우 방향변화에 따라 관심영역의 형태를 고려하지 않으면 손의 일부를 포함하지 못하거나 손 영역 전체를 포함하기 위해 필요이상으로 큰 관심영역을 설정하게 된다.

효율적인 손 영역 제한을 위해서는 단순한 계산으로 손의 빠른 이동을 추적할 수 있어야 한다. 그리고 손 영역의 크기 변화에 따라 제한영역의 크기가 변화되어야 하고, 손의 자유로운 방향변화를 고려하여 사각형으로 표현되는 제한영역의 가로 세로 비율이 결정되어야 한다. 본 연구에서는 이러한 조건을 만족하는 영역제한 방법을 제안하여 트래킹 알고리즘에 적용하였다. 그림 5는 손끝 좌표 추적을 위한 트래킹 알고리즘의 흐름을 나타내며 단계별로 아래와 같이 설명할 수 있다.

1. 초기 검색영역의 위치와 크기를 정한다.
2. 검색영역 내에서 학습과정을 통하여 얻어진 2차원 룩업테이블을 사용하여 컬러 확률 분포를 계산한다.
3. 검색영역 내에서 중점의 위치를 찾는다.
4. 3번에서 찾은 중점을 중심으로 검색영역을 이동한다.
5. 중점의 이동이 정해진 임계값 이하가 될 때까지 3번과 4번 과정을 반복한다.
6. 수렴된 검색영역에서 피부영역의 크기와 방향을 이용하여 최적화된 관심영역을 제한한다.
7. 손 영역의 방향을 사용하여 손끝좌표를 계산한다.
8. 현재의 관심영역에서 2차원 이동거리와 손 영역의 크기 변화를 고려한 영역을 2번 영역의 크기로 한다.
9. 현재의 관심영역에서 손 영역의 크기 변화만을 고려한 영역을 3번 영역의 크기로 한다.
10. 2번의 과정으로 돌아간다.

#### 3.2.1 관심영역의 위치 결정

연속적인 입력영상에서 관심영역의 위치를 결정하기 위하여 영역 내에서 중점의 위치가 변하는 방향을 따라 관심영역을 이동하는 과정을 반복한다. 우선 1차원을 고려해보면  $x$  방향으로의 이동은 다음과 같이 표현 할 수 있다. 관심영역  $W$ 의 중점은 손 영역내의 임의의 한 점  $P_c$ 이다. 이 윈도우 내에서 크기가  $S$ 인 손 영역의  $x$  방향 중점  $\bar{P}_x(W)$ 는 식 (2)를 통해 구할 수 있다.

$$\bar{P}_x(W) = \frac{1}{S} \sum_{x \in W} P_x \quad (2)$$

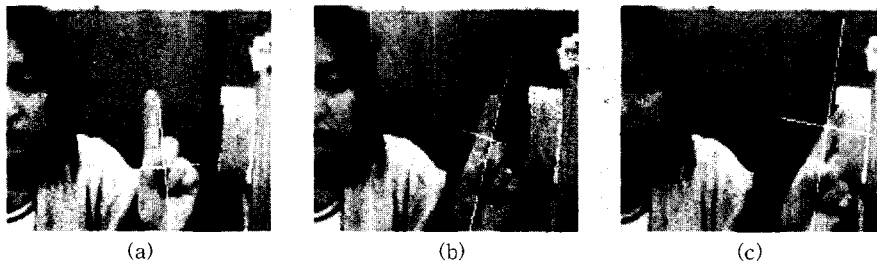


그림 4 CAMSHIFT 알고리즘을 적용한 관심영역 설정

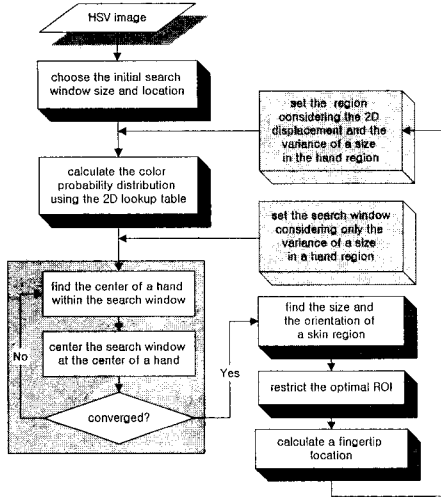


그림 5 트래킹 알고리즘 흐름도

$$\bar{P}_x(W) - P_c \cong 0 \quad (3)$$

식 (3)을 만족하지 않으면  $\bar{P}_x(W)$ 가  $W$ 의 중점( $P_c$ )이 되도록 관심영역을 이동하고 다시  $\bar{P}_x(W)$ 를 구하는 과정을 반복한다. 관심영역의  $y$  방향 이동 역시 같은 방법으로 결정한다.

### 3.2.2 관심영역의 크기 결정

관심영역의 위치는 손 영역의 중점을 따라 이동하므로 손 영역의 주위에 잡음이 존재하는 경우 손의 움직임을 잘 따라갈 수 없다. 따라서 훈련과정을 통하여 정확한 2차원 피부영역 특업테이블을 얻는 것이 중요하다. 또한 손 영역 분할과정에서 얻은 피부영역 확률분포에서 임계값 이하의 작은 확률을 가지는 픽셀은 피부영역으로 간주하지 않음으로써 주변의 잡음으로 인하여 발생하는 영향을 최대한 줄인다. 그러나 이러한 과정을 통하여 부분적으로 피부영역의 정보를 잃는 문제(false negative)가 발생한다. 따라서 본 연구에서는 피부영역 추출과정에서 손 영역의 정보가 부분적으로 손실된 경우에도 효율적으로 관심영역을 제한하고 손끝의 좌표를 찾을 수 있는 방법을 제안하고 적용하였다.

관심영역의 크기는 손 영역의 크기와 방향을 고려하여 결정한다. 손 영역의 크기와 방향은 다음과 같이 영상의 모멘트 정보를 사용하여 구할 수 있다[6]. 빠른 연산과 잡음제거를 위하여 피부 확률 분포 영상에 임계값을 적용하여 이진영상을 얻는다.  $B(x, y)$ 가 영상의  $x, y$  위치에서 픽셀의 값을 나타낼 때 영상의 0차, 1차, 2차

모멘트는 아래와 같이 정의된다.

$$\begin{aligned} M_{00} &= \sum_x \sum_y B(x, y), & M_{11} &= \sum_x \sum_y xyB(x, y) \\ M_{10} &= \sum_x \sum_y xB(x, y), & M_{01} &= \sum_x \sum_y yB(x, y) \\ M_{20} &= \sum_x \sum_y x^2B(x, y), & M_{02} &= \sum_x \sum_y y^2B(x, y) \end{aligned} \quad (4)$$

관심영역의 이동을 위해 사용되는 손 영역의 중점  $x_c, y_c$ 는 모멘트를 사용하여 다음과 같이 표현할 수 있다.

$$x_c = \frac{M_{10}}{M_{00}}, \quad y_c = \frac{M_{01}}{M_{00}} \quad (5)$$

관심영역 윈도우의 가로와 세로의 길이를 결정하기 위해서는 우선 손 영역의 크기와 방향을 알아야한다. 손 영역의 크기는 앞서 정의된 0차 모멘트와 같다. 2차원 영상 내에서 손 영역의 방향(주축의 방향)  $\theta$ 는 아래와 같이 모멘트 정보를 사용하여 나타낼 수 있다[7].

$$\begin{aligned} a &= \frac{M_{20}}{M_{00}} - x_c^2, & b &= 2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right), & c &= \frac{M_{02}}{M_{00}} - y_c^2 \\ \theta &= \frac{\arctan(b, (a-c))}{2} \end{aligned} \quad (6) \quad (7)$$

손 영역의 크기와 방향정보를 사용하여 다음과 같이 관심영역 윈도우의 크기를 결정한다. 2차원에서의 손 영역 크기는 0차 모멘트의 제곱근을 사용하여 대략적인 1차원 길이로 표현할 수 있다. 그러나 포인팅 동작에서 측정된 2차원 손 영역의 크기를 가로와 세로의 비율을 고려하지 않고 단순히 제곱근을 사용한 1차원 길이로 표현하게 되면 손 영역 전체를 포함하는 관심영역을 구할 수 없게 된다. 뿐만 아니라 손의 자유로운 방향변화를 따라 관심영역의 가로와 세로의 길이는 연속적으로 바뀌어야 한다. 이러한 조건을 고려하여 다음과 같이 길이 조절을 위한 인자( $s_x, s_y$ )를 곱한다.

$$R_x = s_x \sqrt{M_{00}}, \quad R_y = s_y \sqrt{M_{00}} \quad (8)$$

$$s_x = \cos|\theta| + 1, \quad s_y = \sin|\theta| + 1, \quad \max(s_x, s_y) + 0.1 \quad (9)$$

손의 방향을 나타내는 좌표계가 그림 6과 같으므로 가로축과 세로축 방향 각각의 길이 조절 인자는  $\cos$ 과  $\sin$  함수를 사용하여 표현할 수 있다. 포인팅 동작에서 손 영역을 포함하는 영역의 단축과 장축 길이의 비율은 통계적으로 1:2.1을 넘지 않는다. 따라서 길이 조절인자의 값은 식 (9)에서와 같이 손의 방향을 고려하여 1과 2.1사이의 값으로 결정된다.

관심영역 윈도우의 좌표는 손의 방향에 따라  $x$ 축 방향 관심영역 길이  $R_x$ 와  $y$ 축 방향 관심영역 길이  $R_y$ 를 손의 중점을 기준으로 그림 7과 같이 배치하여 결정한다.

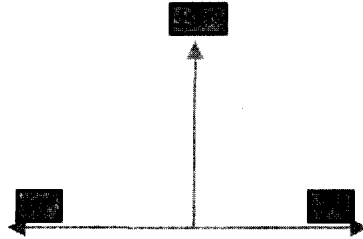
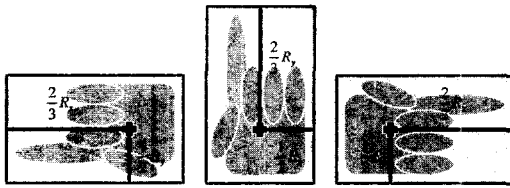


그림 6 손의 방향을 나타내는 좌표계



(a)  $0^\circ < \theta < 45^\circ$  (b)  $|\theta| \geq 45^\circ$  (c)  $-45^\circ < \theta < 0^\circ$

그림 7 손 영역의 방향과 크기를 고려한 관심영역 윈도우 결정

3.3 손끝 좌표 검출

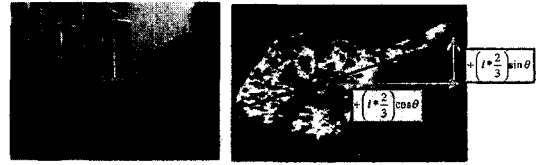
피부영역 검출과정을 통하여 손의 형태를 정확히 얻을 수 있다고 가정하는 경우 손끝의 좌표를 찾기 위하여 손의 경계[8]를 사용하거나 손가락의 무게중심을 이용[9]하여 찾는 연구들이 발표된바 있다. 그러나 실시간 처리를 목적으로 복잡한 계산을 줄이거나 앞서 언급한 바와 같이 잡음을 줄이는 과정으로 인하여 손의 모든 부분을 정확히 추출하기는 어렵다. 본 연구에서는 손 영역 정보의 손실이 있는 경우에도 손끝의 좌표를 찾을 수 있는 방법을 제안하고 적용하였다.

마우스 포인터의 이동을 결정하는 손끝의 좌표는 검지의 첫 번째 마디 영역 내부에 존재하는 한 점이라고 가정한다. 식 (10)을 이용하여 손 영역의 대체적인 길이  $l$  을 계산한 후 손 영역의 중점을 기준으로 그림 8과 같이 조정하여 결정한다.

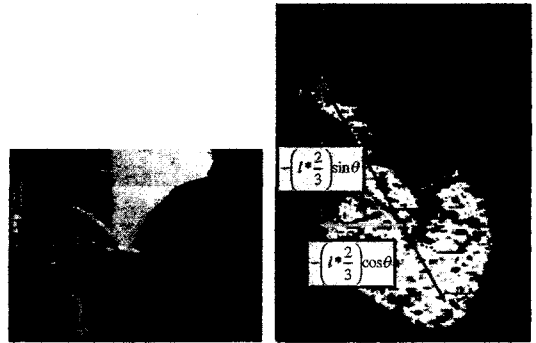
$$l = \frac{\sqrt{(a+c) + \sqrt{b^2 + (a-c)^2}}}{2} \quad (10)$$

이때  $a, b, c$ 는 식 (6)과 같다.

손 영역의 주축은 손 영역에 속하는 모든 점들과의 거리가 최소가 되는 축을 의미한다. 포인팅 동작의 특성상 주축의 방향은 검지의 영향을 받게 되므로 그림 8과 같이 손 영역의 중점을 지나  $\theta$ 의 방향으로 길이가  $l$  인 직선을 그리면 직선의 한 끝점이 손끝으로 정의한 검지의 한 마디 내에 속하게 된다.



(a)  $\theta < 0$ 인 경우 손끝 좌표



(b)  $\theta > 0$ 인 경우 손끝 좌표

그림 8 손 영역의 방향에 따른 손끝 좌표 계산

4. 마우스 포인터 움직임 표현 기법

휴먼 마우스 시스템에서는 트래킹 과정을 통해 얻어지는 손끝 좌표의 변화가 화면상의 마우스 포인터의 움직임으로 표현된다. 사용자가 마우스 포인터의 움직임을 자연스럽게 느낄 수 있도록 하기 위해서는 몇 가지 조건들을 만족해야 한다. 우선 사용자의 손동작처럼 스크린상의 마우스 포인터 역시 끊어지지 않고 부드럽게 이어져야 한다. 또한 사용자의 손이 움직이지 않을 때 마우스 포인터 역시 움직임 없이 멈추어 있어야 한다. 마우스 포인터는 사용자의 움직임이 시작되었을 때 그 움직임에 빠르고 정확하게 응답할 수 있어야 하며 그 위치는 항상 사용자의 손 방향을 따라 움직여야 하고 현재 손의 위치를 지나치지 않아야 한다[10].

사용자가 마우스 포인터의 움직임을 제어하기 위하여 손을 움직일 때 사용자는 공간상의 손의 위치나 카메라를 통해 보이는 영상내의 손의 위치를 고려하여 움직이는 것이 아니라 화면상에 보이는 마우스 포인터의 위치를 기준으로 하여 손의 움직임을 결정하게 된다. 따라서 마우스 포인터가 항상 현재 손끝의 위치를 따라잡기 위해 움직이도록 현재 손끝좌표의 위치  $H_t$ 와 이전프레임을 통해 결정된 현재 마우스 포인터의 위치  $X_{t-1}$  사이의 거리를 이용하여 마우스 포인터의 이동 방향과 거리를 나타내는 힘  $F_t$ 를 결정한다.

$$D_i = H_i - X_{i-1} \quad (11)$$

$$F_i = f(D_i) \quad (12)$$

이러한 과정을 통하여 사용자의 화면보다 상대적으로 작은 입력영상 내의 움직임은 큰 화면에 비례하는 한 좌표로 표현[9]하는 경우 발생하는 그림 9와 같은 문제를 해결할 수 있다. 그림 9는 입력영상과 화면의 해상도 차이로 인하여 입력영상내의 마우스 포인터의 위치를 해상도 비례관계만을 고려하여 화면에 표현하는 경우 특정 위치에 마우스 포인터를 둘 수 없는 문제를 나타내고 있다. 실제 응용 환경에서는 해상도의 차이가 더욱 크므로 마우스 포인터가 위치할 수 없는 부분이 증가하게 된다.

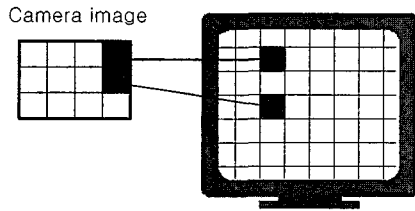


그림 9 입력영상과 화면의 해상도 차로 인하여 발생하는 문제

사용자가 특정위치를 지칭하기 위하여 손의 이동을 멈추거나 손의 이동 중에 발생하는 손의 작은 떨림이 마우스 포인터의 움직임으로 표현되지 않도록 손끝좌표의 이동이 임계값보다 작은 경우의 힘  $F_{min} = 0$ 으로 설정한다. 이때 입력영상의  $x$ 축과  $y$ 축의 해상도가 같지 않으므로 임계값은 화면 해상도의 1%로 설정한다.

결정된 힘에 의하여 이동하는 마우스 포인터의 자연스러운 움직임을 위한 마우스 포인터의 위치 결정법은 다음과 같다. 화면상의 마우스 포인터는 질량과 위치, 속도를 가지는 물리적 객체인 사용자의 손을 따라 움직이게 된다. 마우스 포인터 역시 손과 유사하게 질량  $m$ , 위치  $x$ 와 속도  $v$ 를 가지는 물체로 모델링 하여 그 운동을 표현한다. 그리고 손끝 좌표 검출 처리속도의 한계로 인하여 발생하는 좌표의 불연속성문제 해결을 위하여 프레임간에 존재하는  $\Delta t$ 의 시간간격 내에서 마우스 포인터의 위치를 결정한다.

우선 1차원운동을 가정하여 보면, 마우스 포인터의 위치를 하나의 변수로 표현 할 수 있다. 마우스 포인터의 속도는 일정한 시간간격을 가지는 프레임 사이에서 손 끝 좌표의 이동거리로 나타내어 질 수 있다. 마우스 포인터의 움직임을 2차원 운동으로 표현하는 것은 이 절

의 끝 부분에서 기술한다. 마우스 포인터의 순간적인 위치  $x(t)$ , 속도  $v(t)$ , 가속도  $a(t)$ 는 다음과 같이 정의할 수 있다.

$$v(t) = \frac{dx(t)}{dt}, \quad a(t) = \frac{dv(t)}{dt} \quad (13)$$

마우스 포인터의 운동은 다음과 같이 위치에 관한 이차 미분 방정식으로 표현 될 수 있다.

$$\frac{d^2x(t)}{dt^2} = \frac{F}{m} \quad (14)$$

$\Delta t$ 를 연속적인 프레임 간의 시간간격,  $a_n, v_n, x_n$ 은 각각 시간  $t_n = t_0 + n\Delta t$ 에서  $a, v, x$ 라고 할 때 운동방정식 (14)는 Euler method를 사용하여 다음과 같이 표현할 수 있다.

$$v_{n+1} = v_n + a_n\Delta t \quad (15)$$

$$x_{n+1} = x_n + v_n\Delta t \quad (16)$$

식 (14)의 해는 여러 방법을 사용하여 구할 수 있다.  $x_{n+1}$ 의 위치가  $v_{n+1}$ 을 통하여 결정되도록 하기 위하여 작은 시간간격 즉,  $\Delta t$ 를 작은 값이라고 이라고 가정하면 식 (16)은 새로이 갱신된  $v_{n+1}$ 을 이용하여 다음과 같이 표현 할 수 있다(modified Euler method: Euler-Cromer)[11].

$$x_{n+1} = x_n + v_{n+1}\Delta t \quad (17)$$

사용자가 인지하는 연속적인 움직임을 위해서는 최소한 초당 20프레임 이상의 처리가 필요하다. 트래킹 시스템에서 이러한 처리속도를 지원하지 못하는 경우에는 다음과 같은 방법으로 마우스 포인터의 부드러운 움직임을 표현할 수 있다. 입력되는 각 프레임에는 손끝좌표 추적을 위하여 손 영역 분할, 관심영역 설정, 손끝좌표 계산 등의 여러 알고리즘들이 적용되므로 처리되는 프레임 사이에는 시간간격  $\Delta t$ 가 존재한다. 이때  $\Delta t/k = 0.03(s)$ 를 만족하는  $k$ 를 사용하여 자연스러운 마우스 포인터의 움직임을 보장할 수 있다. 트래킹을 위한 프레임 처리 시간 사이에서  $1/k$ 에 비례하는  $v_{n+1}$ 을 현재 마우스 포인터의 위치에 더하여 다음 단계의 위치를 나타낸다. 이러한 과정을 통해 마우스 포인터의 움직임은 손이 천천히 움직이는 경우 이동거리가 짧게 표현되어 서서히 이동하는 부드러운 움직임이 표현되고 반대로 손이 빨리 움직이는 경우 이동거리는 길게 표현되어  $\Delta t$ 의 시간 내에 손끝좌표의 위치에 정확히 도달하게 된다. 실제로는 스크린상의 마우스 포인터의 운동은 2차원으로 표현되어야 한다. 2차원에서의 운동은 각 차원  $x, y$ 의 움직임을 1차원의 운동방정식으로 고려함으로써 쉽게 확장하여 생각 할 수 있다.

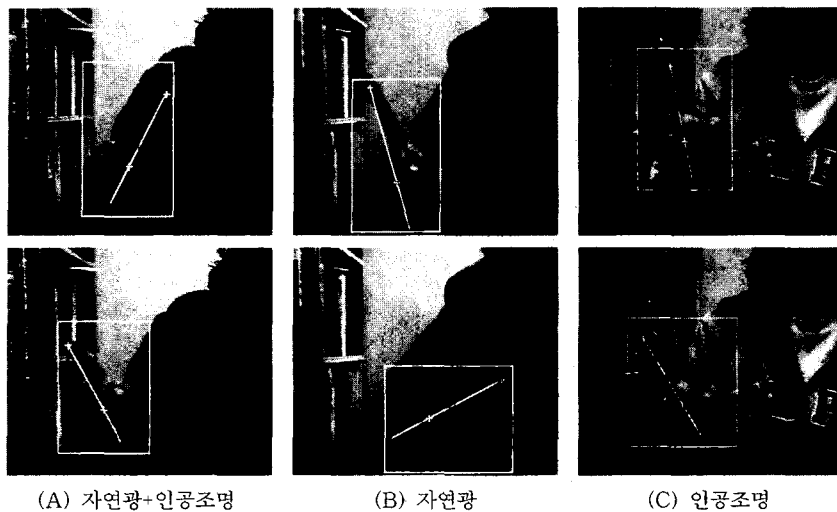


그림 10 손끝좌표 검출결과

## 5. 실험결과

실험을 통하여 본 논문에서 제안한 기법을 적용한 손끝좌표 추적결과와 마우스 포인터 이동 궤적을 보였다.

그림 10은 조명환경이 다른 상황에서 크기와 방향변화에 따른 적합한 관심영역 제한을 통해 손끝좌표를 구하는 결과를 나타낸다. (A)는 자연광과 인공조명의 영향을 받는 상황에서 붉은 옷을 입고 실험한 경우이다. (B)는 인공조명을 제거하고 자연광의 영향만을 받는 경우의 실험결과이며 (C)는 자연광을 차단하고 인공조명만 존재하는 경우에 해당한다. 각각의 배경에는 모두 피부색과 유사한 가구와 다양한 색의 책들이 존재한다. 그림 10을 통하여 제안한 방법이 손의 다양한 방향변화에 따라 적절하게 관심영역을 설정하고 복잡한 배경에서도 손끝좌표를 잘 검출하고 있음을 알 수 있다.

표 1 손끝좌표 추적의 정확도

	# total frames	# detected frames	accuracy (%)
자연광+인공조명(A)	400	373	93.25
자연광(B)	400	368	92
인공조명(C)	400	387	96.75
계	1200	1128	94

표 1을 통하여 손끝 좌표의 검출 결과가 주위 조명 환경의 변화에 크게 영향을 받지 않는 것을 볼 수 있다. 본 연구를 통해 제안된 기법은 사용자의 환경에 최적화된 실시간 학습법과 효율적인 손끝 좌표 검출알고리즘

을 사용하여 손끝좌표 추적에 좋은 성능을 보이고 있다.

마우스 포인터의 자연스러운 움직임 표현을 위해서는 트래킹 과정을 통하여 최소한 5Hz이상[8] 손끝 좌표를 얻을 수 있어야 한다는 연구 보고가 있다. 그림 11은 트래킹 결과 손끝좌표의 값이 6Hz로 얻어진 경우의 입력 좌표와 본 연구를 통해 제안한 기법을 적용한 ( $k=5$ ) 마우스 포인터의 이동 궤적을 표시하여 비교한다.



(a) 입력 좌표(6HZ) (b) 제안된 기법 적용결과

그림 11 마우스 포인터 이동 궤적 표현 예

실험을 통하여 제안된 방법을 적용하면 연속성이 결여된 입력좌표들을 보다 자연스러운 경로의 움직임으로 표현할 수 있음을 보였다.

## 6. 결론

본 연구는 사용자의 손끝좌표를 추적하여 기존의 마우스 입력을 대체하는 휴먼마우스 시스템을 위한 손끝 추적 방법을 제안하였다. 그리고 이를 통해 얻어진 불



연속적인 좌표들을 마우스 포인터의 부드러운 움직임으로 표현하는 기법을 제안하였다.

제안한 방법을 적용하여 빠르고 정확하게 손끝좌표를 추적할 수 있을 뿐 아니라 결과로써 얻어지는 좌표의 입력이 불연속적인 경우에도 마우스 포인터의 움직임을 자연스럽게 표현할 수 있음을 확인하였다. 본 연구를 통해 제안한 기법을 적용하면 보다 효율적으로 사용자 친화적인 휴먼 마우스 시스템을 구현할 수 있다.

### 참 고 문 헌

- [ 1 ] C. Jennings, "Robust finger tracking with multiple cameras" International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp. 152-160, 1999.
- [ 2 ] J. Reh and T. Kanade, "Visual Analysis of High DOF Articulated Object with Application to Hand Tracking," CMU Tech. Report CMU-CS-95-138, Carnegie Mellon University, April, 1993.
- [ 3 ] Gary R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," Intel Technology Journal Q2, 1998.
- [ 4 ] R. Kjeldsen and J. Kender, "Towards the use of Gesture in Traditional User Interfaces," International Conference on Automatic Face and Gesture Recognition, pp.151-156, 1996.
- [ 5 ] R. Kjeldsen and J. Kender "Interaction with On-Screen Objects using Visual Gesture Recognition" IEEE Conference on Computer Vision and Pattern Recognition, pp. 788-793, 1997.
- [ 6 ] W. T. Freeman, K. Tanaka, J. Ohta and K. Kyuma, "Computer vision for computer games" Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp. 100 -105, 1996.
- [ 7 ] B. K. P. Horn, "Robot vision," MIT Press, 1986.
- [ 8 ] Andrew Wu, Mubarak Shah and N. da Vitoria Lobo, "A Virtual 3D Blackboard: 3D Finger Tracking using a Single Camera" Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition, pp. 536-543, 2000.
- [ 9 ] 김유호, 김종선, 박지영 "휴먼 마우스 구현을 위한 효율적인 검지 좌표 검출 및 추적", HCI 학술대회, 한국정보과학회 10권 1호, pp800-805, 2001.
- [ 10 ] R. Kjeldsen, "Visual Interpretation of Hand Gestures as a Practical Interface Modality" PhD Thesis, Department of Computer Science, Columbia University, October 1996.
- [ 11 ] Harvey Gould and Jan Tobochnik "An Introduction to Computer Simulation Methods" Addison Wesley Publishing Company, 1996.



박 지 영

2000년 성균관대학교 전기전자 및 컴퓨터공학부(학사). 2002년 성균관대학교 전기전자 및 컴퓨터공학부(석사). 2002년 ~ 현재 성균관대학교 정보통신공학부 박사과정. 관심분야는 컴퓨터비전, 영상처리



이 준 호

1985년 서울대학교 전기공학부(학사). 1987년 The Pennsylvania State University, Department of Electrical Engineering (석사). 1994년 Purdue University, School of Electrical and Computer Engineering(공학박사). 1994년 ~ 1995년 University of California, Riverside 연구원. 1995년 ~ 1997년 한국과학기술연구원 선임연구원. 1997년 ~ 2001년 성균관대학교 전기전자컴퓨터공학부 조교수. 2001년 ~ 현재 성균관대학교 정보통신공학부 부교수. 관심분야는 컴퓨터비전 기반 HCI