

# 실시간 인식기술을 이용한 가상입력시스템 설계 및 구현

김용수\*, 한판암\*\*

\*거창전문대학 컴퓨터정보시스템과

\*\*경남대학교 컴퓨터공학부

e-mail:yskim@kc.ac.kr

## Implementation and Design of Virtual Input System Using Realtime Recognition

Yong-Soo Kim\*, Pan-Am Han\*\*

\*GeoChang Provincial College Dept Of Computer & Information System

\*\*Kyungnam University Division of Computer Engineering

### 요 약

사용자 인터페이스 환경을 위한 경제적인 입력 방법으로 핸드 인식기술이 널리 이용되고 있다. 그러나 대부분의 기법들은 고정된 배경과 정해진 패턴만을 인식하는 제한점을 가진다. 이러한 제한점은 수행환경의 제약과 다양한 응용분야에 적용하기 어렵다는 문제점을 가진다. 본 논문에서는 배경의 제약 없이 핸드 영역정보와 위치정보를 추출하여 입력 장치를 구동시키는 가상입력시스템을 제안한다. 본 연구는 개방 환경에서 사람의 입력을 요구하는 공학이나 전시관의 키오스크 시스템, 각종 조정장치 및 산업용 시스템분야에 경제적인 가상입력시스템으로 사용될 수 있을 것이다.

### 1. 서 론

인간과 컴퓨터의 의사소통을 위한 기본적인 입력 인터페이스는 주로 키보드나 마우스를 사용하고 있다. 하지만, 제한된 내부 공간이 아닌 열린 공간(박물관, 공항, 전시관 등)에서 키보드, 마우스를 이용한 컴퓨터 조작은 매우 불편하다. 물론 대체방법으로 터치스크린을 사용할 수 있지만, 외부로 돌출된 환경에서 많은 사용자들의 잘못된 조작으로 파손할 우려가 많이 있다. 따라서 열린 공간에서 사용자들의 컴퓨터 조작을 편하게 하면서 안정적인 새로운 사용자 입력 시스템이 요구 된다. 현재는 인간과 컴퓨터간의 상호작용( HCI : Human Computer Interaction)을 위한 다양한 연구가 활발하게 진행되고 있다. 대표적인 예로, 인간의 움직임 정보를 이용하여 컴퓨터를 쉽게 조작할 수 있는 인터페이스 기술을 들 수 있다[1,2]. 이러한 인간 행동 기반의 인터페이스를 위한 방법은 전자 센서를 이용하거나, HMD(Head Mounted Display)를 이용한 방법이 있다[3]. 하지만 이런 방법은 고가의 장비를 요구하며, 사용자가 별도의 장치를 신체에 부착해야 하는 불편함을 가진다. 최근에는 USB 캠을 이용하여 신체에 별도의 장비를 부착하지 않는 저가형의 인식시스템에 대한 연구가 널리 진행되고 있다[4,5]. 핸드 제스처 연구는 손의 움직임 정보와 패턴정보를 주로 이용한다. 손의 움직임 정보를 이용하여 화면의 위치정보를 지정할 수 있으며, 손의 패턴정보를 이용하면 다양한 기능을 부여할 수 있다. 하지만 기존의 핸드 제스처를 이용한 방법은 정해진 패턴에 맞는 기능은 수행할 수 있지만 다양하게 변하는 패턴은 인식하기 어렵다.

본 논문은 컴퓨터 비전 기법을 기반으로 실시간 핸드

인식을 이용한 저가형 휴먼 인터페이스 시스템을 구현하였다. 본 시스템은 실시간 영상에서 피부색 영역을 추출하여 배경과 피부색을 분리하고 차 영상기법과 모션 처리기법을 이용하여 손을 인식하게 된다. 인식된 손의 위치정보와 움직임 정보를 계산하여 키보드와 마우스의 기능을 수행 하도록 한다. 본 논문의 구성은 다음과 같다. 2장에서는 제안 시스템을 구현하기 위한 배경 기술에 대해 알아본다. 3장에서는 구현한 가상입력 시스템의 기본 구조를 파악하고, 실시간 영상에서 손 인식 및 특징 추출을 위한 방법을 설명한다. 4장에서는 구현한 시스템의 실험 및 결과를 보여주고 마지막으로 5장에서는 결론을 맺는다.

### 2. 배경기술

손 인식을 위한 방법은 칼라 영상처리, 패턴 모델링, 차 영상 기법 등 세 가지로 구분된다. 첫 번째로 칼라영상을 이용하는 방법은 RGB, HSV등의 칼라모델을 이용하여 피부색과 유사한 칼라를 찾아서 손 영역을 인식하게 된다. 이 방법은 빛에 따라서 색상의 변하는 요인이 있다. 최근에는 다양한 칼라모델을 이용하여 빛의 제약 없이 피부색 정보를 이용하는 방법들이 연구되고 있다[4,6]. 두 번째 방법은 손의 패턴을 모델링 하여 비교하는 방법으로 그레이 영상으로 충분히 검출이 가능하지만 다양한 패턴을 모델링 하여야 하는 불편함이 있다[7]. 세 번째 방법은 움직이는 영상에서 이전 프레임과 현재 프레임의 차 영상을 찾아서 이용하여 방법이다. 이 방법은 배경의 조명과 움직임에 민감하여 많은 필터링 작업이 요구된다.

본 논문에서는 다양한 배경을 가진 실시간 영상에서 피부색과 유사한 칼라의 영역을 그대로 유지하고 나머지 영역

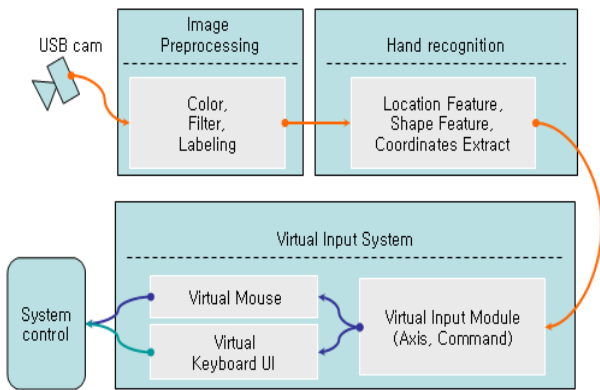
은 제거한다. 피부색으로 선택된 영역을 손 후보 영역으로 판단하고 차 영상 기법을 이용하여 모션 히스토리를 생성한다. 모션 히스토리는 이전 프레임의 영상이 보관 된다. 모션 히스토리의 정보를 이용하여 움직임이 있는 영역을 분리하여 최종적으로 손의 위치를 판단하게 된다.

### 3. 가상입력 시스템

손 인식을 이용한 가상입력 시스템은 기존의 입력 장치로 사용되고 있는 키보드와 마우스를 대신하는 시스템이다. 시스템에 장착된 USB캠의 영상에서 손을 인식하여 가상입력 시스템을 조작할 수 있도록 한다. 본 장에서는 구현된 시스템의 세부구조에 대해 기술한다.

#### 3.1 가상입력 시스템의 구조

가상입력 시스템은 (그림 1)과 같이 영상 입력 및 전처리부, 손 인식부와 가상 입력처리로 되어있다. 가상입력 시스템은 먼저 USB캠을 통하여 입력되는 영상을 전처리하게 된다. 전 처리과정은 영상에서 손을 인식하기 위한 첫 번째 단계로 칼라변환을 수행하고 피부색과 유사한 색은 남기고 나머지 색상은 제거를 한다. 피부색과 유사한 색상에서 잡음제거를 위하여 이진영상으로 변환하고 임계값을 지정하여 임계값을 벗어나는 영역은 필터처리하여 부럽게 한다. 영상을 필터처리 후 존재하는 영상은 후보 손 영역으로 간주한다. 후보 손 영역을 검색하는 것으로 영상의 전처리단계는 끝내고, 처리한 영상은 다음의 손 인식 단계로 진행된다.



(그림 1) 가상입력 시스템의 구조

두 번째 단계는 전처리 과정을 거친 후보 손 영역에서 차 영상 기법과 모션 처리 방법을 이용하여 손 인식을 마치면 손의 위치정보와 모양정보를 계산한다. 손의 위치정보는 가상 입력 장치를 제어하는 기초 데이터로 캠상의 좌표와 화면상의 좌표를 서로 맵핑 할 수 있는 데이터가 된다. 그리고 손의 모양정보는 추가적인 기능을 수행하는 명령어로 응용이 가능하다.

세 번째로 가상입력 시스템은 손인식부에서 계산된 데이터를 전달받아 가상입력 모듈이 처리를 하게 된다. 가상입력 모듈은 전달된 데이터를 이용하여 마우스 위치를 이

동하거나 클릭 동작을 수행하도록 시스템에 마우스 제어 신호를 보내게 된다. 마우스 제어 신호는 표준 마우스 신호형태로 운영체제에 전달되고 키보드의 기능을 수행하기 위한 키보드의 신호는 Windows XP의 기본프로그램인 화상키보드 통하여 표준 키보드 신호로 전달된다.

따라서 본 시스템은 간단한 USB캠의 영상처리 및 손 인식을 기법을 응용하여 표준 키보드, 마우스 장치의 기능을 대신하는 가상입력 시스템이다. 다음은 손 인식을 위한 다양한 처리 기법을 파악하고 본 시스템에 적용된 컴퓨터 비전 기반의 손 인식을 위한 피부색 처리 기술에 대하여 자세히 소개한다.

#### 3.2 컴퓨터 비전 기반의 피부색 처리

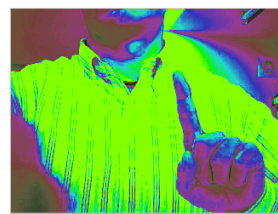
본 논문에서는 칼라 영상의 다양한 배경과 빛의 변화에 대한 제약을 없애기 위한 방법으로 영상의 기본 칼라모델 (RGB모델)을 HSV칼라모델로 변환하여 사용하였다. 칼라 변환 수식은 아래 (식 1)를 참조하였다. HSV칼라모델은 3개의 채널로 구성된다[8].

$$\begin{aligned}
 &H \in [0, 360], S, V, R, G, B \in [0, 1] \\
 &MAX = \max(R, G, B), MIN = \min(R, G, B) \\
 &H = \begin{cases} \text{undefined,} & \text{if } MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \text{ and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \text{ and } G < B \text{ (식 1)} \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases} \\
 &S = \begin{cases} 0, & \text{if } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{otherwise} \end{cases} \\
 &V = MAX
 \end{aligned}$$

(그림 2)는 각 채널의 칼라 정보를 피부색 구분이 용이하도록 변환한 영상이다. H(Hue)채널은 빨강, 파랑, 노랑 등의 기본 색상정보를 갖고 있으며, S(Saturation)채널은 색상의 채색에 관한 정보를 갖고 있다. 그리고 V(value)채널은 칼라의 밝기에 대한 정보를 갖고 있다. 본 논문에서는 피부색 처리를 위해서 H채널 정보를 이용하였다.



a) h-channel                      b) s-channel



c) v-channel

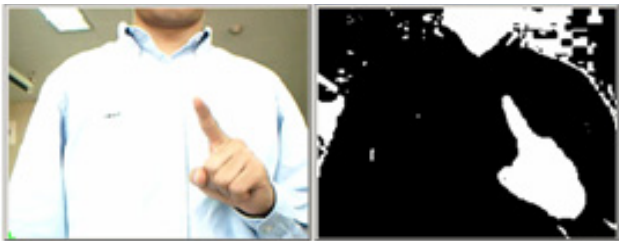
(그림 2) HSV 칼라모델

H채널은 기본 색상을 갖고 있기 때문에 피부색과 유사한 색상을 선택 할 수 있다. 하지만 빛의 밝기가 매우 높은 경우에는 고유한 색상 정보에 변화를 주어 흰색으로 변하게 된다.

입력영상의 H채널 정보를 가지고 피부색 경계 값으로 피부색과 유사한 색상은 H채널의 최대값으로 지정하고 나머지 색상은 0으로 지정하여 제거한다. 여기서 최대값을 갖는 픽셀은 후보 손 영역으로 정한다.

후보 손 영역은 잡음제거를 위해서 그레이 영상으로 변환하여 스무딩 필터를 적용하였다. 그리고 잡음이 제거된 영상은 계산된 임계치(t=30)를 이용하여 이진화 영상으로 변환하였다. (그림 3)은 원본 영상과 후보 손 영역을 전처리 과정을 거쳐서 최종적으로 이진화한 영상이다.

다음 내용은 전처리 단계의 후보 손 영역을 이용한 손 인식 및 특징 값을 추출에 관한 설명을 한다.



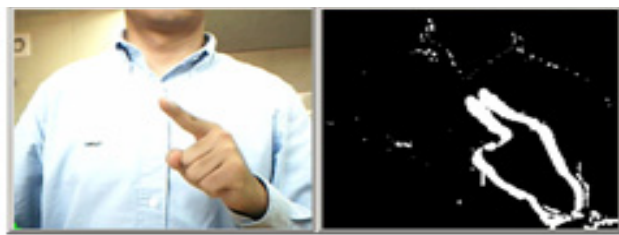
a) 원 영상                      b) 이진화 영상  
(그림 3) 후보 손 영역 영상

3.3 손 인식 및 특징 값 추출

본 시스템에 구현된 손 인식 기법은 후보 손 영역을 기반으로 하여 차 영상 기법과 모션 템플레이트(Motion Template), 모션 세그먼트(Motion segmentation)방법을 이용하여 움직이는 손을 인식 하였다[9].

$$DiffIm = Im_N - Im_{N-1} \quad (식 2)$$

차 영상 기법은 (식 2)과 같이 이전 프레임(Im<sub>N-1</sub>)과 현재프레임(Im<sub>N</sub>)의 영상에 대한 차이 값을 계산 하는 것으로 (그림 4)는 실시간 영상에서 차 영상을 추출하여 이진화 영상으로 변환시킨 결과 화면이다.



a) 원 영상                      b) 차 영상  
(그림 4) 차 영상 처리

차 영상 결과는 손의 움직임 분석을 위해서 모션 템플레이트 방법을 이용하는데 차 영상정보와 시스템의 현재 시간정보 등을 모션 히스토리 벡터에 저장을 하게 된다. 다음으로 모션 세그멘테이션은 (그림 5)과 같이 모션 히스토리 벡터에 저장된 차 영상 정보를 이용하여 움직임이 있는 객체를 따로 분리하고 각각의 객체에 대한 위치정보를 계산한다.



a) 모션 히스토리 벡터                      b) 모션 세그멘테이션  
(그림 5) 모션 세그멘테이션

마지막으로 손 인식을 위해서 분리된 객체들 중 객체의 영역이 경계값(t=200)이상인 객체를 손으로 판단한다. 그리고 인식된 손의 위치값을 가상입력 시스템으로 전달하게 된다.

4. 실험 및 결과

본 논문에서 구현한 시스템의 개발환경은 C언어를 기반으로 하였으며 기본적인 이미지 처리를 위해서 인텔의 공개된 영상처리 라이브러리를(OpenCV v1.0)[10] 이용하였다. 시스템의 구동환경은 펜티엄IV 3.0GHz, 시스템 메모리 1GB, 영상 입력 장치는 칼라를 지원하는 USB캠을 사용하였다. 입력 영상의 크기는 320\*240, PC의 화면 해상도는 1024\*768을 기준으로 하였다.

가상입력 시스템의 실행화면 우측하단에 미리보기 화면이 표시되고 화면의 중앙에 커서가 위치하도록 하였다. 그리고 가상 키보드 입력을 위해서 windows XP에서 기본적으로 제공되는 화상키보드를 실행되도록 하였다.

사용자는 마우스를 대신하여 화면의 커서를 보면서 손을 움직이면 이동거리에 따라 PC화면의 커서가 움직이게 된다. 마우스의 클릭 동작은 움직임을 멈추고 3초간 지연이 생기면 PC의 현재 화면 위치에서 클릭 동작을 수행하게 된다.



(그림 6) 가상입력 시스템의 실험

(그림 6)은 가상입력 시스템을 실행한 화면으로 인터넷 검색 엔진에서 검색어를 입력하는 실험을 하고 있다.

실험 결과는 실시간 영상에서 손을 인식하여 원하는 위치로 마우스 조작이 가능하였다. 하지만 일부 과도한 조명이나 피부색과 유사한 영상이 있을 경우에 마우스 위치가 불규칙하게 변하였다. 실험 결과에서 환경1은 정상 조도에서 제한된 배경을 대상으로 실험한 결과이고, 환경2는 정상 조도보다 10% 밝은 환경에서 제한된 배경을 대상으로 실험한 결과이다. 그리고 환경3은 정상 조도에서 배경에 피부색과 유사한 영역을 임의로 설정하여 실험한 결과이고, 마지막으로 환경4는 정상 조도보다 10% 밝은 환경에서 임의적인 피부색 성분을 많을 때의 실험결과이다.

<표 1> 영상인식 결과

실험환경	인식율
환경1	96%
환경2	88%
환경3	84%
환경4	82%

## 5. 결 론

본 논문에서는 움직이는 영상의 손을 인식하여 가상입력이 가능하도록 시스템을 디자인 하였다. 그리고 인간과 컴퓨터간의 인터페이스를 위하여 신체의 장치 부착이 없이 손으로 자연스러운 컴퓨터의 조작이 가능하며, 저렴한 비용으로 가상 입력시스템을 구현 하였다. 본 시스템은 제한된 사무실이나 책상의 공간이 아닌 박물관, 공항, 교육 전시관, 산업현장 등의 열린 공간에서 별도의 입력 장치 없이 컴퓨터를 조작할 수 있도록 한다. 향후 과제로 조명의 밝기에 따른 유동적인 피부색 모델 선정과 손의 모양 패턴을 확장하여 가상 입력 시스템의 기능과 명령을 추가 하면 다양한 응용이 가능할 것으로 본다.

## 참고문헌

[1] F. K. Quek, T. Mysliwiec and M. Zhao, "FingerMouse: A Freehand Pointing Interface," Proc. Int'l Conf. on Automatic Face and Gesture Recognition, pp.372-377, 1995.

[2] Ho-Sub Yoon, Byung-Woo Min, Jung Soh Young-lae Bae, HyunSeung Yang, "Human Computer Interface for Gesture-Based Editing System," Proc. of the 10th Int'l Conf. on Image Analysis and Processing, pp.969, 1999.

[3] 이의철, 박강령, "눈동자 시선 추적에 의한 3차원 1인칭 슈팅 게임", 한국정보처리학회 논문지 B, Vol 12-B No.04, pp.465-472, 2005.

[4] K. Imagawa, S. Lu and S. Igi, "Color-Based Hands Tracking System for Sign Language Recognition,"

Proc. of IEEE Int'l Conf. on Automatic Face and Gesture Recognition, pp.151-156, 1996.

[5] 권경수, 김상호, 장재식, 김항준, "제스처 기반의 HTPC 인터페이스", 한국정보과학회 '04 학술표논문집 가을(II), pp.715-717, 2004.

[6] 양환석, 정혜원, "멀티미디어 시스템을 위한 영상내의 손 인식에 관한 연구", 한국콘텐츠학회 논문지 '05 Vol.5 No.2, pp.267-274, 2005.

[7] 한영환, "움직임 정보를 이용한 제스처 인식 시스템", 정보처리학회 논문지B 10권 4호, pp.473-478, 2003.

[8] HSV color space, wikipedia, [http://en.wikipedia.org/wiki/HSV\\_color\\_space](http://en.wikipedia.org/wiki/HSV_color_space)

[9] J. Davis, G. Bradski, "Real-time Motion Template Gradients Using Intel CVLib," IEEE Visualization '99, pp.437-440, 1999.

[10] 강동중, 하종은, Visual C++을 이용한 디지털 영상처리, 사이텍미디어, pp.247, 2003.