저해상도 손 제스처 영상 인식에 대한 연구

안정호*, 정회원

A Study on Hand Gesture Recognition with Low-Resolution Hand Images

Jung-Ho Ahn*, *Regular Members*

요 약

최근 물리적 디바이스의 도움 없이 사람이 시스템과 인터랙션 할 수 있는 인간 친화적인 인간-기계 인터페이스가 많이 연구되고 있다. 이중 대표적인 것이 본 논문의 주제인 비전기반 제스처인식이다. 본 논문에서 우리는 설정된 가상세계의 객체와의 인터랙션을 위한 손 제스처들을 정의하고 이들을 인식할 수 있는 효과적인 방법론을 제안한다. 먼저, 웹캠으로 촬영된 저해상도 영상에서 사용자의 양손을 검출 및 추적하고, 손 영역을 분할하여 손 실루엣을 추출한다. 우리는 손 검출을 위해, RGB 공간에서 명암에 따라 두 개의 타원형 모델을 이용하여 피부색을 모델링하였으며, 블랍매칭(blob matching) 방법을 이용하여 손 추적을 수행하였다. 우리는 플러드필(floodfill) 알고리즘을 이용해 얻은 손 실루엣의 행/열 모드 검출 및 분석을 통해 Thumb-Up, Palm, Cross 등 세 개의 손 모양을 인식하였다. 그리고 인식된 손 모양과 손 움직임의 콘텍스트를 분석해서 다섯 가지 제스처를 인식할 수 있었다. 제안하는 제스처인식 방법론은 정확한 손 검출을 위해 카메라 앞에 주요 사용자가 한 명 등장한다는 가정을 하고 있으며 많은 실시간 데모를 통해 효율성 및 정확성이 입증되었다.

Key Words : gesture recognition, hand detection and tracking, hand segmentation, silhouette mode analysis, skin color model, floodfill algorithm

ABSTRACT

Recently, many human-friendly communication methods have been studied for human-machine interface(HMI) without using any physical devices. One of them is the vision-based gesture recognition that this paper deals with. In this paper, we define some gestures for interaction with objects in a predefined virtual world, and propose an efficient method to recognize them. For preprocessing, we detect and track the both hands, and extract their silhouettes from the low-resolution hand images captured by a webcam. We modeled skin color by two Gaussian distributions in RGB color space and use blob-matching method to detect and track the hands. Applying the foodfill algorithm we extracted hand silhouettes and recognize the hand shapes of Thumb-Up, Palm and Cross by detecting and analyzing their modes. Then, with analyzing the context of hand movement, we recognized five predefined one-hand or both-hand gestures. Assuming that one main user shows up for accurate hand detection, the proposed gesture recognition method has been proved its efficiency and accuracy in many real-time demos.

I. 서 론

물리적 도구를 사용하지 않고, 인간이 인간을 상대하는 방법으로 제스처나 음성을 통해 기계를 제어하고자 하는 연구가 많이 진행되고 있다. 이는 사람이 복잡한 매뉴얼을 익힘으로써 기계를 사용하던 시대를 지나, 기계가 사람의 소통방식을 학습함으로써 인간 친화적인 인간-기계 인터페이스의 시대가 오고 있음을 의미한다.

본 논문은 사용자가 가상의 사이버 공간의 객체들과 소통하는 프로그램을 제작하기 위한 프로젝트의 일부이다. 설정된 사이버 공간에는 인간과 닮은 사이버 객체들이 존재한다. 세컨드 라이프와 같이, 이들은 스스로 결정하여 다른 사이버 객체들과 인터랙션을 한다. 사용자는 제스처를 통해 이 가상공간에 개입하고자 할 수 있다. 사용자는 지시형 제스처

※ 본 연구는 2012년도 강남대학교 교내연구비 지원에 의한 것임.
* 강남대학교 컴퓨터미디어정보공학부 (jungho@kangnam.ac.kr)
접수일자 : 2014년 1월 26일, 수정완료일자 : 2014년 3월 6일, 최종게재확정일자 : 2014년 3월 7일

(pointing gesture)를 통해 가상공간의 한 객체를 선택하고, 가상 객체의 능력을 향상시키기 위해 다양한 제스처를 통해 객체를 지도할 수 있다. 본 논문에서 정의된 제스처는 이러한 인터랙션을 위한 것이다.

우리는 서론의 뒷부분에서 정의된 제스처의 특징과 용도에 대해 설명하고 관련 연구를 소개한다. 2절에서는 제스처 인식을 위한 저해상도 손 검출 및 추적 방법론을 소개하고, 3절에서 손 모양 인식 및 제스처 인식 방법론을 소개한다. 그리고 4절에서 토의 및 결론을 맺는다.

1. 가상세계와 인터랙션을 위한 제스처 정의

본 논문에서 제안하는 손 제스처 인식 알고리즘은 가상공간의 오브제와의 인터랙션을 위해 현실공간에 있는 사용자의 제스처를 인식하는 것을 목표로 한다. 인터랙션을 위해 정의한 제스처는 PUSH, PULL, POINTING 등 세 가지 명령형 제스처와 가상 오브제(예: 김연아)에게 점프동작을 학습시키기 위한 JUMP 동작이 있다. 이들의 정의, 기능 및 특징은 다음 표와 같다.

표 1. 제스처의 기능 및 정의

제스처 종류	기능	정의
PUSH	Zoom out	손바닥이 보이는 상태에서 가슴에서 부터 앞으로 민다.
PULL	Zoom in	주먹을 쥔 상태에서 엄지 손가락을 세우고 팔을 앞으로 뻗은 상태에서 뒤로 당긴다.
POINTING	Mouse move	오른손을 주먹 쥔 상태에서 엄지와 검지 손가락을 이용하여 십자모양을 만든다. (화면에 있는 마우스 활성화)
	Mouse click	오른손이 손바닥 모양으로 바뀐다. (마우스 클릭)
JUMP	Teach jump action	두 손을 가슴에 모으고 점프를 할 경우 가상오브제에게 점프동작을 학습시킨 것으로 간주한다.

우리는 위 제스처를 인식하기 위하여 손 검출 및 추적, 손 모양 인식, 제스처 인식 등 세 가지 세부 알고리즘을 개발하였다. 정의된 제스처는 손이 움직이는 경로와 움직임 속에서 취하는 손 모양(posture)에 따라 종류가 구분된다. 이 제스처들을 인식하기 위해서는 양손의 검출 및 추적이 안정적으로 이루어져야 하고, 손이 움직이는 동안 손의 위치와 모양을 지속적으로 인식하여야 한다.

Push와 Pull 제스처에서 당기는 동작과 미는 동작은 사실상 동일하기 때문에 움직인 경로를 통해 이 둘을 구별하긴 어렵다. 이 두 제스처는 양손이 모두 가슴 부분에 위치하고 있을 때 손 모양을 구분하여 인식하였다. 즉, 두 손이 모두 가슴 부분에 위치하였을 때, 손 모양이 손바닥(Palm)이면 Push 제스처, 주먹을 쥔 상태에서 엄지손가락을 세운 모양(Thumb-Up)이 인식되면 Pull 제스처라 인식하였다.

Pointing 제스처는 지시하는 동작을 이용하여 사물을 가리키고 선택하는 기능을 수행한다. 지시형 제스처는 주먹을 쥔 상태에서 검지 손가락을 핀 상태로 물체를 가리키는 것이 일반적이라 할 수 있으나 카메라 한 대를 사용할 경우 깊이 정보를 얻을 수 없기 때문에, 이러한 손 모양을 인식하기 매우 어렵다. 따라서 우리는 주먹을 쥔 상태에서 검지를 세우고 엄지손가락을 옆으로 향하게 한 십자형(Cross) 손 모양을 인식하여 이를 Pointing 제스처로 인식하였다. Cross 손 모양이 인식되면 화면에 놓여 있는 마우스가 활성화되고 마우스가 손을 따라 이동한다. 이 때, 손 모양이 Palm으로 바뀌면 클릭되어 객체를 선택할 수 있다.

Jump 동작은 가상의 오브제에게 점프동작을 학습시키기 위해 고안되었다. 두 손이 가슴부분에 위치한 상태에서 점프동작을 취할 경우 이를 인식한다.

Push와 Pull, Jump 제스처는 양손 제스처이며 Pointing 제스처는 한손 제스처이다. 다음 그림 1은 Palm, Thumb-Up, Cross 손 모양의 대표적인 예를 보여준다. 제안하는 제스처 인식 시스템은 카메라 전방에 한 명의 유저만 등장하고 배경이 매우 복잡하지 않다는 제한 조건을 가진다.

그림 1. Palm, Thumb-Up, Cross 손 모양 정의

2. 관련 연구

손 검출 및 추적과 더불어 손 제스처 인식을 위한 지금까지의 연구는 대부분 계산 복잡도가 높으며, 섬세한 손 모양을 인식하기 위해 고해상도 손 영상을 이용한 방법이 많다[1, 4, 5]. 저해상도 손 영상을 이용하는 경우, 섬세한 손 모양 인식보다는 손의 궤적을 이용한 제스처 인식에 대한 연구가 활발히 진행되었다[2, 3, 8, 9, 10]. 또한 카메라와 더불어 보조도구나 다른 센서를 이용한 손 제스처 인식 방법론도 많이 연구되었다[6, 7].

Fang et.al[1]은 손 검출 및 추적을 위해 Optical Flow, 피부색 컬러 정보를 이용한 Adaboost 방법[11]사용하였고, 고해상도 손 영상에 대해 스케일-공간 정보를 이용하여 손바닥과 손가락을 구별하였다. Bobick과 Davis[2, 3]은 MEI와 MHI에서 Hu 모멘트를 추출한 후, Mahalanobis 거리를 이용하여 몸동작을 인식하였다. Binh et. al[4]은 고해상도 손 영상에서 피부색 모델을 이용하여 손을 검출한 후 개선된 Kalman Filter를 적용하여 손 추적을 수행하였으며, 손 움직임 경로를 Pseudo 2-DHMM를 이용하여 제스처 인식을 수행하였다. Bretzber et. al[5]은 IUV 컬러공간에서 블랍과 리

지(Ridge) 특징을 이용하여 손 모양을 추출하고 Paricle Filtering과 Heirarchical Layered Sampling 방법을 이용하여 손 추적을 수행하였다.

Kim et. al[6]은 손가락에 골무를 착용한 사용자에 대해 저해상도 손 영상에 대해 7개의 손 모양 인식 및 5개의 제스처 인식을 수행하였다. 마이크로소프트는 Natal 프로젝트[7]에서 일반 CMOS 카메라와 TOF 방식의 적외선 카메라를 이용하여 키넥트(Kinect) 게임을 개발하였다.

Kirishima et. al[8]은 제스처가 미리 정의되어 있지 않은 상태에서 주요 제스처를 인식할 수 있는 방법론을 개발하였고, Malima et. al[9]는 5개의 손 모양 인식을 이용하여 5개의 로봇 동작을 컨트롤하였다. 이들은 손 중심점을 중심으로 한 여러 동심원의 연결성분을 분석을 통해, 손 모양을 인식 하였다. Yoon et. al[10]은 피부색과 움직임 정보를 이용한 손 검출을 한 후 위치, 각도, 속도 특징을 이용한 HMM을 통해 사용자의 동작을 인식하였다.

Ⅱ. 손 검출 및 추적

우리의 손 검출 알고리즘은 피부색 검출과 움직임 정보에 기반을 둔다. 제스처는 사람의 움직임을 통해 의미를 전달하는 수단이기 때문에, 고정된 카메라로 촬영된 영상에서 움직임이 있는 피부색 화소를 손의 일부로 판단하였다. 그리고 손 추적 시 얼굴의 피부색 영역이 손으로 인식되는 오류를 방지하고, 신체부위를 추정하기 위해 얼굴 검출을 수행하였다. 다음 그림 2는 제안하는 손 검출 및 추적 알고리즘의 도식을 보여준다. 이 절에서 소개하는 손 검출 및 추적 알고리즘은 우리의 이전 결과[13]을 개선한 것이다.

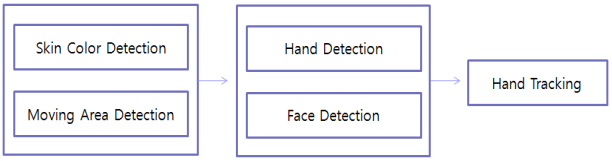


그림 2. 손 검출 및 추적 개요

1. 손 검출을 위한 전처리

제안하는 손 검출 방법론은 피부색 검출에 기반을 둔다. 이를 위해, 타원형 피부색 모델과 배경의 피부색 영역을 제거하기 위한 필터, 피부색을 이용한 얼굴검출 방법론을 소개한다. 우리는 피부색 모델링을 위해 다양한 조명환경에서 100개의 피부 영상조각을 수집하였다. 다음 그림 3은 대표적인 피부색 영상 샘플을 보여준다.

그림 3. 피부색 영상 조각의 예

RGB 공간의 피부색은 조명에 민감하다는 단점이 있는 것으로 알려져 있다. 하지만 다음과 같이 두 개의 타원형 모델을 사용할 경우 다른 색 공간보다 피부색 검출률이 높음을 실험적으로 확인할 수 있었다.

$$\left(\frac{x_R-m_R^L}{s_R^L}\right)^2+\left(\frac{x_R-m_G^L}{s_G^L}\right)^2+\left(\frac{x_R-m_B^L}{s_B^L}\right)^2<T_L \qquad (1)$$

$$\left(\frac{x_R-m_R^D}{s_R^D}\right)^2+\left(\frac{x_R-m_G^D}{s_G^D}\right)^2+\left(\frac{x_R-m_B^D}{s_B^D}\right)^2<T_D \qquad (2)$$

여기서 기호 L과 D는 각각 밝음과 어두움을 의미한다. m_R^L, m_G^L, m_B^L (m_R^D, m_G^D, m_B^D)는 밝은(어두운) 피부색의 R, G, B 채널의 평균값이고, s_R^L, s_G^L, s_B^L (s_R^D, s_G^D, s_B^D)는 밝은(어두운) 피부색의 R, G, B 채널의 표준편차이다. 밝은 피부색과 어두운 피부색은 3차원 RGB 공간에서 K 평균 군집화(K means clustering) 알고리즘을 이용해 분리하였다.

피부색은 인간의 피부가 아닌 영역에서도 검출될 수 있는 자연에 존재하는 색이다. 우리는 배경에 존재하는 피부색 영역을 필터링하기 위해, 위 조건 (1), (2)를 만족하는 피부색 화소들에 대해 연결성분석(Connected Component Analysis)을 수행하여 연결성분의 외곽박스(bounding box)를 검출하고, 다음 조건 (3)을 만족하는 연결성분을 제거하였다.

$$w\times h<T_1,\ w/h>T_2,\ h/w>T_3 \qquad (3)$$

여기서 w, h는 연결성분 외곽박스의 가로, 세로의 길이이다. 조건 (3)을 만족하는 영역은 너무 작거나 가로-세로의 비가 지나치게 크거나 작은 영역으로, 사람이 아닌 배경에 존재하는 피부색 영역의 대표적인 특징이다. 실험에서 임계치는 $T_1=100$, $T_2=3$, $T_3=3$으로 설정하였다.

얼굴 검출을 위해 알려진 정교한 방법론들이 많이 알려져 있으나, 실시간 처리를 위해 피부색 영역의 크기를 이용한 간단한 방법으로 얼굴 영역을 검출하였다. 소개하는 얼굴 검출 방법론은 주요 제스처 행위자가 한 명일 경우에 유효하다.

우리는 필터링이 안 된 피부색 연결성분 중, 영상의 가장 윗부분에 나타나는 일정 사이즈 이상의 영역을 얼굴이라 지정하였다. 아래 그림 7은 얼굴 검출의 예를 보여 준다. 우리는 얼굴을 검출함으로써, 손 움직임 영역을 설정할 수 있어 안정적인 손 추적을 가능케 할 수 있었고, 가슴영역을 추정할 수 있어 양손 제스처 인식률을 향상시킬 수 있었다.

2. 손 검출 및 분할

우리는 손 검출을 위해 이동 중인 피부색 블랍(덩어리)을 검출하고, 이 블랍이 설정된 손 검출 영역에 속하는 경우 제스처인식을 위한 손이 검출된 것으로 인식한다. 그림 4는 손 검출 및 추적 알고리즘의 개요를 보여 준다.

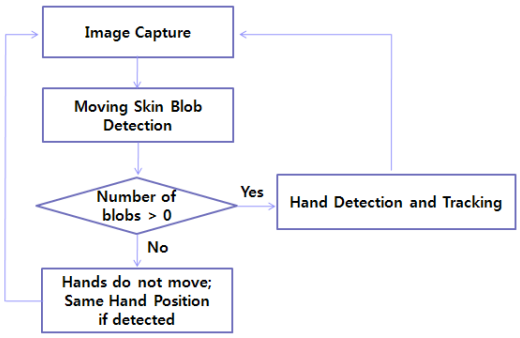


그림 4. 손 검출 및 추적 개요

이동중인 피부색 블랍 검출은 차영상과 피부색 픽셀 검출 및 손 복원을 통해 이루어진다. 카메라가 고정되어 있을 때, 두 프레임의 차이가 발생한 영역은 객체의 움직임이 발생한 영역으로 볼 수 있다. 작은 움직임을 무시하고 주요한 움직임만을 검출하기 위해 다음 식 (4)를 이용하여 연속된 두 프레임의 차영상에서 임계치 이상의 값 차이가 발생한 픽셀들을 검출하였다.

$$|I_t(p) - I_{t-1}(p)| > T \quad (4)$$

여기서 $I_t(p)$는 시간 t에서 픽셀 p의 밝기값을 의미한다. T는 미리 정해 놓은 임계치로 실험에서 30으로 설정하였다.

피부색 컬러값을 가지는 픽셀들 중 식 (4)를 만족하는 픽셀들을 찾는다. 이 픽셀들은 이동 중인 손의 일부라 할 수 있다. 우리는 이 픽셀들을 포함하는 손 영역을 플러드필(floodfill) 알고리즘을 이용하여 복원한다. 즉, 이 픽셀들 각각을 시드(seed)로 하여, 시드와 컬러값의 차이가 적은 주변 픽셀들을 차례로 검출한다. 다음 그림 5는 이 과정을 순서대로 보여 준다.

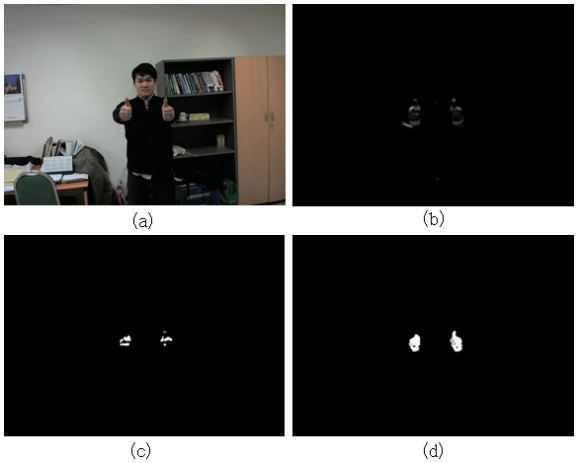


그림 5. 손 검출 및 분할 과정. (a)원영상, (b)차영상, (c)차영상에서 검출된 피부색 영역, (d)손 영역 복원(분할) 결과

다음 그림 6은 제안한 손 검출 알고리즘을 이용하여 분할된 Pull, Push, Mouse_Move, Mouse_Click 손 모양의 실루엣들을 보여준다.

그림 6. 분할된 손 영역의 예

사용자와 가상세계의 인터페이스 시작 프로토콜은 사용자가 차렷 자세를 일정시간 지속한 후 손을 올리는 것으로 정하였다. 제스처를 수행하기 위해 손을 올릴 때, 손이 처음 등장하는 위치는 화면의 아래 부분이라 예상할 수 있다. 이러한 이유로 우리는 다음 그림 7의 하얀색 점선 박스와 같이, 검출된 얼굴박스 중심점의 x좌표를 기준으로 왼쪽과 오른쪽에 오른손과 왼손의 초기 검출 영역을 설정하였다. 각 박스의 크기는, 얼굴박스의 높이를 L이라 할 때, 폭이 $1.5L$이고 높이가 $3L$이다.

우리는 이동 중인 피부색 블랍이 왼손 또는 오른손 검출 영역에 처음 나타날 때, 검출 영역에 속한 가장 큰 블랍을 해당 손이라 검출한다. 검출되지 않은 손에 대해서는 검출이 될 때까지 매 프레임 검출여부를 체크한다. 참고로 제안하는 손 검출 및 추적 알고리즘은 사용자가 제스처 수행 중 한 손 또는 양 손을 내렸다가 올리는 과정을 불규칙하게 여러 번 반복하더라도 해당 손을 검출 및 추적할 수 있다.

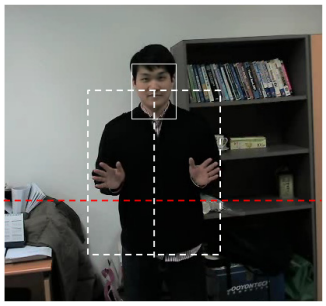


그림 7. 손 검출 영역과 제스처 영역

제안하는 손 제스처 인식 시스템은 한 번만 사용자의 손을 검출하고, 다음 프레임부터는 손 추적 알고리즘을 이용하여 손의 위치와 모양을 인식한다. 다음은 손 추적 알고리즘에 대해 설명한다.

3. 손 추적

우리는 오른손과 왼손을 개별적으로 검출 및 추적을 수행한다. 만약 손의 움직임이 검출되지 않은 경우 해당 손 검출 영역에 대해 검출 과정만 반복한다.

일반적으로 객체 검출은 추적보다 더 많은 시간을 소요하기 때문에, 객체가 검출된 다음 프레임부터는 검출된 객체

영역을 추적한다. 하지만 손이라는 객체는 형태의 변화가 매우 크고 빠르게 움직일 수 있기 때문에 추적하기 어려운 대상이다. 우리는 2절에서 제안한 손 검출 방법론이 매우 빠르고 안정적이기 때문에 검출에 기반을 둔 추적 방식을 제안한다.

사람들은 의미 없는 동작을 많이 한다. 이를 가비지 제스처(garbage gesture)라 하는데, 이러한 동작은 의미 있는 제스처로 오인될 가능성이 높으며 손 추적 시 오류를 범할 가능성이 높다. 이러한 오류를 방지하기 위해, 의미 없는 손동작이 빈번한 허리 아래 부분을 제외한 위 영역을 제스처 영역으로 설정하였다. 손 추적 및 제스처 인식은 설정된 제스처 영역 안에서만 수행되며 손이 이 영역을 벗어날 경우, 손 검출부터 다시 수행하였다.

위 그림 7의 빨간색 점선과 같이, 제스처 영역은 검출된 얼굴 박스의 높이를 L이라 할 때, 얼굴 박스의 중심점으로부터 아래로 $2L$ 만큼 떨어진 지점의 가로방향 선의 윗부분으로 설정하였다.

손 추적이란 이전 프레임에서 손의 위치가 결정되었을 때, 다음 프레임에서 해당 손의 위치를 결정하는 것이다. 우리는 제스처 영역 안에서 검출된 이동 피부색 블랍을 이전 프레임의 가장 가까운 손으로 대응시키는 블랍 매칭을 수행하였다. 만약 이전 프레임에 손이 추적되었지만 현재 프레임에서 이동 피부색 블랍이 검출되지 않은 경우, 손이 잠시 멈춰있는 상태로 간주하고 이전 손 위치를 현재 손 위치로 간주하였다.

노이즈에 강인한 손 추적을 위해 손 사이즈 필터를 고안하였다. 연속된 두 프레임에서 손의 크기가 많이 변하지 않는다는 가정 하에, 이전 프레임의 손의 크기와 비슷한 크기를 가지는 피부색 블랍을 현재 프레임의 해당 손으로 선택하였다. 이 사이즈 필터링으로 인해 몸통에서 발생할 수 있는 작은 피부색 블랍을 손으로 오인하는 오류를 방지할 수 있다. 다음 표 2는 손 추적 과정을 의사코드로 정리한 것이다.

표 2. 손 추적 알고리즘

```
For each previously detected hand blob H = Left, Right,
    find blob B satisfying |H|-E < |B|< |H|+E
        and being closest to H
    if dist(B, H) < T
        identify B as the hand
    else
        H is staying
```

Ⅲ. 손 모양 및 제스처 인식

우리는 Mouse_Move, Mouse_Click, Pull, Push 등 4개의 손 제스처와 Jump 동작을 인식하고자 한다. 아래 표 3은 제스처들의 정의와 특징을 요약한 것이다.

표 3. 제스처의 특징

손 개수	제스처		손 위치	손 모양
한손 제스처	Pointing	Mouse_Move	임의의 위치	Cross
		Mouse_Click		Palm
양손 제스처	Pull		가슴 부분	Thumb-Up
	Push			Palm
	Jump			N/A

한 손을 이용하는 지시형 제스처는 화면에 놓인 객체를 지시(선택)하고 이 객체를 원하는 곳으로 이동시키기 위한 것이다. 우리는 화면에 마우스를 위치시키고 사용자가 이 마우스를 컨트롤하여 사물을 선택할 수 있게 함으로써, 손가락으로 지시 또는 선택할 때 발생하는 모호함을 줄였다.

사용자가 손을 들어, Cross 손 모양 취하면 마우스가 활성화되고, 마우스는 사용자의 손과 같은 방향으로 이동한다. 그리고 사용자의 손 모양이 Cross에서 Palm으로 바뀌면, 클릭 이벤트가 발생하여 마우스 위치에 있는 객체가 선택된다. 사용자는 손 모양을 다시 Cross로 바꾸어 선택된 객체가 손을 따라 이동하게 만들 수 있다. 그리고 원하는 위치에서 손 모양을 Palm으로 바꾸면, 이동 중인 객체는 그 위치에 놓이게 된다.

양손 제스처는 화면 확대를 위한 Pull 제스처, 화면 축소를 위한 Push 제스처, 가상공간의 객체에게 점프 동작을 가르치기 위한 Jump 제스처가 있다. 두 손이 모두 가슴 부분에 위치하면, 이 세 가지 제스처 중 어느 하나가 인식될 수 있다. 두 손이 가슴에 위치한 상태에서 두 손의 모양을 Thumb-Up(Palm)으로 하고 양 손을 앞으로 폈다가 당기면 Pull(Push) 제스처로 인식된다. 두 손이 가슴에 놓인 상태에서 머리의 위치가 올라가기 시작하여 특정 높이 이상으로 올라간 후 내려오게 되면 Jump 동작으로 인식한다.

1. 손 모양 인식

인식해야할 손 모양의 종류는 Palm, Thumb-Up, Cross, Else이다. Else는 Palm, Thumb-Up, Cross가 아닌 상태를 말한다. 우리는 손을 분할한 영상에서 행과 열의 모드(mode)의 개수로 이 네 가지 손 모양을 구분하였다. 여기서 모드는 국부적 최댓값(local maximum)을 의미한다.

그림 8. 손 분할 영상

위 그림 8는 Thumb-Up, Palm, Cross의 손 분할 영상을

보여 준다. 우리는 이 손 모양을 구별하기 위해 다음 두 가지 특징을 추출하였다.

- 열 모드 개수: 영상의 열별로 영상 위쪽 1/4 부분의 흰색 픽셀의 개수를 나타내는 히스토그램의 모드의 수
- 행 모드 개수: 영상의 행별로 영상 오른쪽 1/4 부분의 흰색 픽셀의 개수를 나타내는 히스토그램의 모드의 수

손 분할 영상의 크기는, 카메라와 손사이의 거리에 따라 달라질 수 있기 때문에, 40×60으로 정규화하였다.

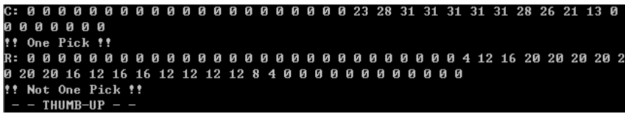


그림 9. Thumb-Up 손 영상의 히스토그램 및 인식 결과

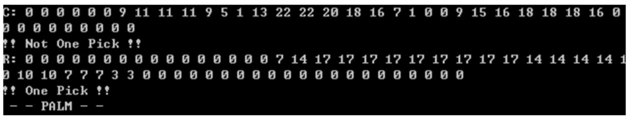


그림 10. Palm 손 영상의 히스토그램 및 인식 결과

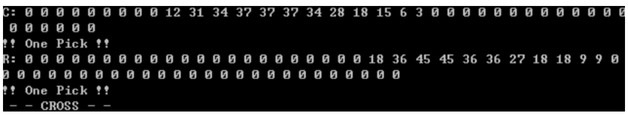


그림 11. Cross 손 영상의 히스토그램 및 인식 결과

위 그림 9, 10, 11은, 그림 8의 Thumb-Up, Palm과 Cross 손 분할 영상의 행/열 히스토그램 및 인식 결과를 보여준다. 'C:' 는 위 1/4부분의 열 히스토그램을 나타내고 'R:' 은 오른쪽 1/4 부분의 행 히스토그램을 보여준다.

우리는 히스토그램으로부터 빈(bin)의 값이 증가하다 감소하는 순간을 포착하여 히스토그램의 모드의 개수를 조사하였다. 그러나 히스토그램의 범위(support)가 너무 긴 경우 하나 이상의 모드를 가지는 것으로 간주하였다. 손등과 같이 Else 손 모양에서 이러한 경우가 많이 발생한다. 다음 표 4는 주어진 히스토그램의 모드가 한 개인지 체크하는 함수의 의사코드이다.

표 4. 모드 검출 의사코드

```
bool Is_One_Mode( unsinged hist[], int RANGE_TOL )
{
        range = find_range(hist);
        if (range > RANGE_TOL)
                return false;
        else
                num_pick = pick_count(hist);
        if (num_pick > 1)
                return false;
        else
                return true;
}
```

손 모양의 행/열 히스토그램의 모드 개수의 특징은 다음 표 5와 같다.

표 5. 손 모양의 행/열 히스토그램 모드 개수 정의

손 모양	열 모드 개수	행 모드 개수
Thumb-Up	One	More than one
Palm	More than one	One
Cross	One	One
Else	More than one	More than one

모드 검출은 높은 정확성을 보이나 손의 기울어짐 정도에 따라 잘못 인식되는 경우가 있다. 이러한 오류에 대해 강인하고자 보팅(Voting) 필터링을 고안하였다. 보팅 필터링이란, 크기가 N인 큐(Queue)에 최근 N개 프레임의 손 모양(type)을 기록하고, 이 중 다수(majority) 손 모양을 현재 프레임의 손 모양으로 결정하는 것이다. 이 필터링은 안정적인 손 모양 인식에 큰 도움을 준다. 다음 표 6은 손 모양 인식 알고리즘의 의사코드를 보여 준다. 여기서 PALM, THUMB_UP, CROSS, ELSE는 미리 정의된 int형 상수이다.

표 6. 손 모양 인식 의사코드

```
  int Hand_Shape_Recognition( HandShapeQueue )
  {
    HandShapeQueue.push(Current_Hand_Shape(
 row_hist[], column_hist[]));
        return HandShapeQueue.voting();
  }
int Current_Hand_Shape( unsigned row_hist[], unsigned
   column_hist[] )
{
        bool is_One_In_Column = Is_One_Mode( row_hist );
        bool is_One_In_Row = Is_One_Mode( column_hist );

        if (is_One_In_Column) {
                if (is_One_In_Row) return CROSS;
                else               return THUMP_UP;
        }
        else {
                if(is_One_In_Row) return PALM;
                else              return ELSE;
        }
  }
```

2. 제스처 인식

인식하고자 하는 제스처의 유형은 표 2와 같이 손 모양 뿐 아니라, 손 개수와 손 위치에 따라 다르게 구분되어진다. 이러한 손 개수와 손 위치 정보를 콘텍스트 정보라 부른다. 여기서 가슴 영역은 그림 7의 흰색 점선과 같이, 얼굴박스의 높이를 L 이라 할 때, 얼굴 박스 하단 1/2지점부터 너비가 $3L$ 이고 높이가 $3L/2$ 인 사각형 영역으로 설정하였다.

Jump 제스처는 두 손이 가슴에 위치한 상태에서 위로 점

프하는 동작으로 정의하였다. 우리는 점프 시 높이의 변화 정도를 알아보기 위해 얼굴 중심 위치의 변화를 측정하였다. 아래 그래프는 한 학습데이터의 세 번의 점프에 대한 얼굴 중심의 높이 변화를 보여 준다.

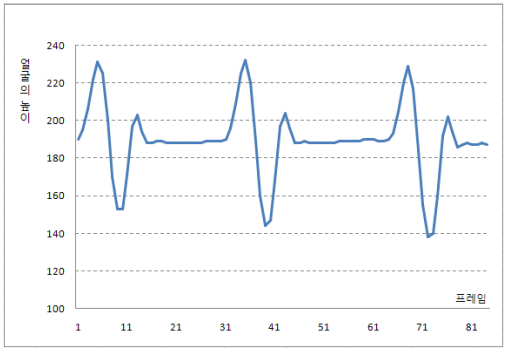


그림 12. 점프 시 얼굴 높이 변화 예

그림 12의 y축은 얼굴 박스 상단의 y 좌표로, 최고점에 도달할 때 얼굴의 높이 값은 최솟값이 된다. 위 그래프에서 8, 39, 72번 프레임에서 최고점에 도달하였다. 초기 정지상태의 값이 190이었고, 최고점에서의 y 좌표는 각각 153, 144, 138 이었다. 우리는 점프에 대한 학습데이터 분석을 통해, y 좌표가 초기 정지상태의 85% 이하가 될 경우 점프 동작으로 인식하였다. 다음 표 7은 제스처 인식에 대한 의사코드를 보여준다.

표 7. 제스처 인식 의사코드

```
int Gesture_Recognition( )
{
    if(Two_Hands_On_The_Chest)
    { // 양손 제스처
        if(face_y_position <
0.85*avg_face_y_coordinate)
            return JUMP;
        else
            if(Hand_Posture == THUMB_UP)
                return PULL;
            else if(Hand_Posture == PALM)
                return PUSH;
            else
                return NULL;
    }
    else if(Right_Hand_in_Gesture_Tracking_Area)
    { // 한손 제스처
        if(Right_Hand_Posture == CROSS)
            return MOUSE_MOVE;
        else if(Right_Hand_Posture == PALM)
            return MOUSE_CLICK;
        else
            return NULL;
    }
}
```

Ⅳ. 토의 및 결론

우리는 본 논문에서 효과적인 손 검출 및 추적 방법론과 손 모양 인식에 기반을 둔 제스처 인식 방법론을 제안하였다. 제안한 손 검출 및 추적 방법론은 움직임이 발생한 부분 주변의 피부색 영역이 제스처를 수행중인 손이라는 생각에 기초하였다. 플러드필 알고리즘을 효과적으로 적용하여 손 영역을 분할하여 얻어진 손 실루엣에서 모드 분석을 통해 Thumb -Up, Palm, Cross, Else 등 네 가지 손 모양을 구분하였다. 손 모양과 더불어, 손의 위치 및 개수의 콘텍스트 정보를 인식하여 정의된 제스처들을 인식할 수 있었다. 또한, 쉽게 재현(reproduction)할 수 있도록 알고리즘의 의사코드를 많이 제시하였다.

특정 어플리케이션에 적용하기 위해 개발된 제스처인식 알고리즘은 공인된 DB에서 성능을 평가하기 어려울 뿐 아니라 다른 알고리즘과 성능을 비교하기 어렵다[1, 4, 5, 6, 7]. 인식하고자 하는 제스처의 종류나 카메라 환경이 다르기 때문이다. 이와 같은 이유로, 공인된 DB에서 제안한 제스처 인식 알고리즘의 실험결과를 제시하지 못한 점을 아쉽게 생각한다.

제안한 제스처인식 방법론은 정확한 손 검출을 위해 다음과 같은 제약조건을 가진다.

첫째, 제안한 손 검출 알고리즘은 피부색 검출에 기반을 두고 있기 때문에 조명 조건 및 인종에 따라 달라지는 모든 피부색을 검출하지 못한다. 하지만, 형광등이 켜져 있는 일반 강의실 및 실험실의 조명에서 높은 검출 성능을 보임을 확인할 수 있었다. 피부색 검출 성능을 높이기 위해 적응적인 피부색 모델에 대한 연구를 진행하고 있다. 얼굴검출기[11, 12]를 통해 얼굴영역을 검출한 후, 이 얼굴영역의 컬러 값으로부터 피부색을 추출하여 이를 기존 피부색 모델과 함께 다시 모델링하는 방법(transfer learning)을 연구 중에 있다.

둘째, 우리는 배경이 복잡하지 않고 사용자가 피부색과 유사한 옷을 착용하지 않음을 가정하고 있다. 영상 차가 발생한 부분에 있는 피부색 부분을 손으로 인식하기 때문에, 피부색 옷이 이동하는 경우 손으로 오인될 가능성이 높다.

셋째, 우리는 한 명의 사용자만 프로그램을 제어한다고 가정하고 있다. 이러한 가정 하에서 피부색 블랍의 크기만으로 얼굴을 검출한 후, 허리, 가슴 등의 인체 부위를 추정할 수 있었다.

제안한 손 추적 방법론과 인식 방법론은 간단하고 효과적이라는 장점이 있다. 위의 세 가지 제약조건 하에서, 제안한 알고리즘은 100%에 가까운 인식 성능을 보임을 많은 데모를 통해 확인할 수 있었다.

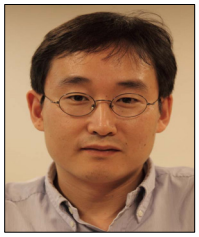
참 고 문 헌

[1] Y. Fang, K. Wang, J. Cheng and H. Lu, "A real-time hand

gesture recognition method", IEEE International Conference on Multimedia and Expo, pp. 995-998, 2007.
[2] A. F. Bobick, and J. W. Davis, "The Recognition of Human Movement Using Temporal Templates", IEEE Transactions on Pattern Recognition and Machine Learning, Vol. 23, No. 3, 2001.
[3] J.W. Davis, "Hierarchical Motion History Images for Recognizing Human Motion", IEEE workshop on Detection and Recognition of Events in Video, pp.39-46, 2001.
[4] N. D. Binh, E. Shuichi and T. Ejima, "Real-Time Hand Tracking and Gesture Recognition System", GVIP Conference, 2005.
[5] L. Bretzber, I. Laptev, Tony Lindeberg, "Hand Gesture Recognition using Mulit-Scale Colour Features, Hierarchical Models and Partical Filtering", IEEE Interntional Conference on Automatic Face and Gesture Recognition, pp.423-428. 2002.
[6] H. Kim, G. Albuquerque, S. Havemann, D. W. Fellner, "Tangible 3D: Hand Gesture Interaction for Immersive 3D Modeling", IPT & EGVE Workshop, 2005.
[7] Microsoft Project Natal, http://research.microsoft .com/apps/video/default.aspx?id=144455 .
[8] T. Kirishima, K. Sato, K., Chihara, "Real-Time Gesture Recognition by Learning and Selective Control of Visual Interest Points", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 3, pp.351-364, 2005.
[9] A. Malima, E. Ozgur, M. Cetin, "A Fast Algorithm for Vision-based Hand Gesture Recognition for Robot Control", IEEE Conference on Signal Processing and Communications Applications, 2006.
[10] H.-S. Yoon, J. Soh, Y. J. Bae, H. S. Yang, "Hand gesture recognition using combined features of location, angle and velocity", Pattern Recognition, Vol. 34, pp. 1491-1501, 2001.
[11] P. Viola and M. Jones, "Robust Real-time Face Detection", International Journal of Computer Vision, vol.57, No. 2, pp. 137-154, 2001.
[12] X. Xiong and F. D. Torre, "Supervised Descent Method and its Applications to Face Alignment", IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[13] J.-H. Ahn and J.-H. Kim, “A Stable Hand Tracking Method by Skin Color Blob Matching”, Pacific Science Review, Vol.12, No.2, pp. 146-151, 2010.

저자

안 정 호(Jung-Ho Ahn)



・1996년 2월 : 연세대학교 수학과 (이학사)
・1998년 9월 : 연세대학교 수학과 (이학석사)
・2001년 12월 : Texas A&M University 통계학과 (이학석사)
・2006년 8월 : 연세대학교 컴퓨터과학과 (공학박사)
・2007년 3월～현재 : 강남대학교 컴퓨터미디어정보공학부 교수
<주 관심분야> : 패턴인식, 얼굴인식, Deep Learning, 제스처 인식, 컴퓨터비전