

Stereo Video-See-Through를 위한 버튼형 인터페이스

최영주, 서용덕
 서강대학교 영상대학원 미디어공학과
 {wildpink, yndk}@sogang.ac.kr

The User Interface of Button Type for Stereo Video-See-Through

Youngju Choi and Yongduek Seo

Department of Media Technology, Graduate School of Media Communications, Sogang University

요 약

본 논문은 두 대의 카메라로부터 들어온 영상을 보여주는 see-through 장치의 사용 환경하에서 일반적인 사용자도 쉽고 편리하게 컴퓨터 시스템 또는 여러 프로세스들을 제어할 수 있는 사용자 인터페이스를 제안한다. 이를 위해 AR기술을 접목하여 영상이 보이는 화면에 가상의 버튼들을 합성하였으며, 화면상에 보이는 손의 위치를 추적하여 손가락에 의한 버튼의 선택 유무를 판단하고 각 상황에 따른 버튼의 색상 변경을 통해 결과를 나타내었다. 사용자는 단순히 화면을 보며 공중에서 손가락을 움직여 버튼을 선택함으로써 관련 작업을 수행할 수 있다.

Abstract

This paper proposes a user interface based on video see-through environment which shows the images via stereo-cameras so that the user can control the computer systems or other various processes easily. We include an AR technology to synthesize virtual buttons; the graphic images are overlaid on the captured frames taken by the camera real-time. We search for the hand position in the frames to judge whether or not the user selects the button. The result of judgment is visualized through changing of the button color. The user can easily interact with the system by selecting the virtual button in the screen with watching the screen and moving her fingers at the air.

키워드: 증강현실, See-Through, 합성, 사람과 컴퓨터간의 상호작용, 평면 투영 변형

Keywords(영문) : AR, See-Through, synthesis, HCI(Human Computer Interaction), Plane Projective Transformation

1. 서론

HCI(Human Computer Interaction)는 사람들이 편리하고 즐겁게 사용할 수 있는 시스템을 개발하는 원리 및 방법을 연구하는 학문이다(ACM, 1992). 즉 HCI는 사람과 상호작용이 가능한 시스템이 사람과 잘 어울려서 주어진 목표를 달성할 수 있도록 이 둘 사이의 상호

작용 방법과 절차를 설계하고, 평가하며, 구현하는 분야라고 정의할 수 있다(Nickerson and Landauer, 1997). HCI 분야는 겨우 20년이 갓 넘은 신생 학문이라고 할 수 있는데 짧은 역사에도 불구하고 최근 들어 HCI에 대한 관심이 매우 높아지고 있다. 이 이유로 두 가지를 보면 아래와 같다.

첫째, 과거에는 사람들이 사용할 수 있는 시스템이 특

정 전문가나 기업 환경에 국한되어 있었다. 그러나 PC가 도입되고 특히 인터넷의 사용이 사회 전체로 확산되면서 이제 컴퓨터는 특정 전문가들의 전유물이 아니라 일반 사람들이 일상 생활에서 사용하는 범용 도구로 탈바꿈하였다. 이에 따라 일반인도 쉽고 즐겁게 사용할 수 있도록 컴퓨터관련 시스템을 만들어 주는 것이 중요하게 되었다.

둘째, 과거에는 컴퓨터 시스템이 큰 방을 하나 가득 차지하였다면, 최근에는 대부분의 시스템이 소형화되고 경량화됨에 따라, 노트북은 물론이고 심지어는 손가락 위에 올려지는 시스템을 비롯해서 더 나아가 머리에 쓰는 형태로 사용자가 바라보는 주변 모습(user's view of his or her surroundings)에 그래픽과 문자를 겹쳐 놓는 "투시 장치" (see-through device) 등이 만들어지고 있다. 물론 see-through device는 현재 일반적으로 상용화가 안되었지만 가까운 미래의 컴퓨터 사용자 인터페이스로서 사용이 급증할 것으로 많은 과학자들이 바라보고 있다. 이와 같이 컴퓨터 시스템이 작아지면서 우리 일상 생활 속에서 사용되는 다양한 가전제품이나 여러 사무용품 속에 스며들게 되었다. 이러한 시스템들을 정보가전(information appliance) 또는 내제시스템(embedded system)이라고 한다. 이러한 것들이 사회에 확산되기 시작하면서 이것들이 사람들과 조화롭게 사용될 수 있도록 하는 HCI의 중요성도 점차 높아지기 시작했다.

이러한 관점에서 본 논문에서는 새로운 컴퓨터 사용자 인터페이스로 관심을 받고 있는 see-through device의 버튼형 인터페이스 방식을 제안하고 구현하였다. 이것은 증강현실(AR: Augmented Reality)을 기반으로 한 새로운 사용자 인터페이스가 되겠다. 여기서 AR이란 사용자의 지각 인식(user's sensory perceptions)에 가상정보(virtual information)를 추가(adding)하는 컴퓨터 화면장치(computer display)를 말한다.

제안하는 사용자 인터페이스 방법은 두 대의 카메라가 장착된 see through device를 사람이 착용시 적용되는 기술로써, 이 장치를 통해 보이는 화면 상에 가상의 button을 합성시킨다. 이렇게 합성된 button을 사람은 마치 실제 공중상에 버튼이 있다고 느낄 수 있게 되며, 이를 통해 사용자가 버튼을 선택하는 행위를 함으로써

그에 대응되는 이벤트를 실행시키게 되는 것이다. 본 논문에서는 버튼을 선택하여 발생하는 이벤트로 그 버튼의 번호를 출력하면서, 버튼이 선택되었음을 시각적으로 표현하기 위해 버튼의 색상을 변화시켰다.

또한 두 대의 카메라가 장착이 되어있기 때문에 실제 사람의 눈이 쌍으로 존재함으로써 사물에 대한 거리감 즉 깊이(depth) 정보를 이용하는 것과 같은 현상을 유도할 수 있다. 따라서 한대의 카메라를 사용한 것에 비해 사용자가 좀더 정확하고 사실적인 사용 감을 느낄 수 있게 된다. 두 대를 이용한 장치의 구조는 아래 그림과 같다. 여기서 화살표는 사람의 손가락을 대신 표현한 것이다.

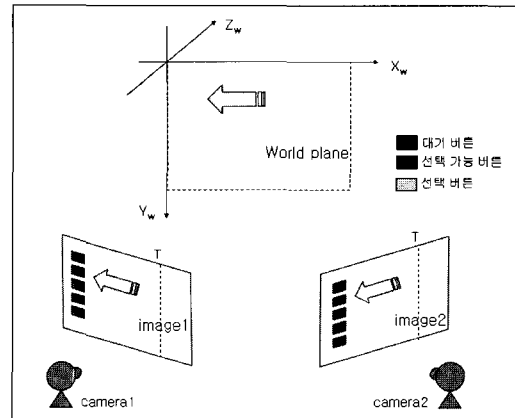


그림 1. 실험 환경의 구조

사용자가 두 대의 카메라를 통해 얻은 실세계(world coordinate)의 영상(world plane)을 캡처하고 시스템은 얻어진 이미지 위에 가상의 빨간색의 버튼을 그리게 된다. 만약 특정 버튼 위에 직접 손가락이 위치하지는 않았지만 그 버튼이 포함된 행에 손가락이 존재할 경우 해당하는 버튼의 색상을 파란색으로 변하게 하여 그 버튼이 차후 실제로 선택 가능한 후보 버튼임을 알리며, 버튼 위로 손가락이 위치하게 되면 그 버튼은 선택 버튼 상태가 되어 녹색으로 변하게 된다. 버튼 위에 존재하던 손을 떼어내면 후보 버튼의 상태가 되어 파란색으로 변하던가 아니면 대기 버튼의 상태가 되어 빨간색으로 변하게 된다. 이로써 사용자는 자신이 어떤 버튼을 선택하고 있는지를 확인할 수 있게 된다.

본 논문에서는 프로세스의 효율을 높이기 위해 각 카메라에서 얻어진 이미지 상에 손의 중심위치가 가로축의 T 위치 이상에 존재 할 때에만 손 영역 추출 이후의 일련의 작업을 수행하도록 하였다. 작업의 전체적인 구성은 아래 그림처럼 크게 전처리구간과 대기구간, 그리고 메인처리 구간으로 이루어져 있다.

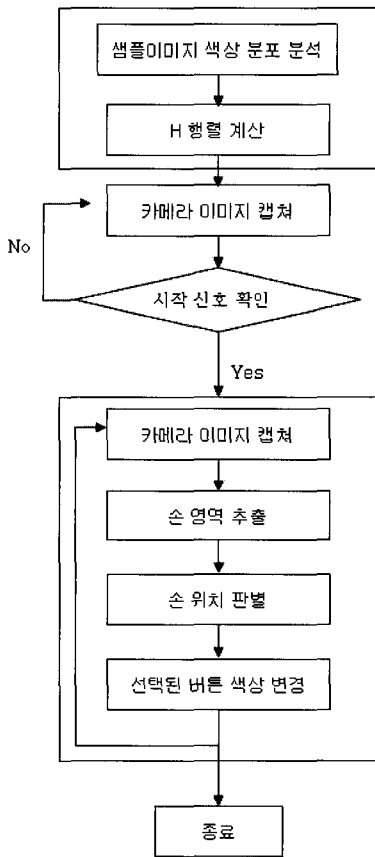


그림 2. 작업 순서도

일반적으로 카메라마다 색상을 표현할 때 같은 색상이라도 약간의 차이를 보이게 되며, 주변 빛에 의한 영향을 많이 받게 된다. 이렇게 되면 시간과 장소에 따라 같은 살색이라도 카메라에서 받아들이는 색상 정보가 틀려져 손 영역을 찾는데 어려움이 생긴다. 이런 문제를 고려하여 본 논문에서는 전처리 과정으로 실제 카메라에서 찍은 손의 이미지를 샘플로 얻어 그것의

평균값과 분산 값을 구했다. 이 값들은 각 카메라에 맞는 손 색상의 특성을 나타내는 것으로 프로그램을 수행하면서 카메라로부터 얻은 이미지에서 손 영역을 찾는데 기준이 되는 데이터가 된다.

다음으로 실 세계 좌표계와 각 카메라에서 얻은 이미지의 좌표계간의 변환 관계를 나타내는 Homography matrix를 구하게 된다. 이 변환행렬을 통해 world plane 상에 존재하는 가상의 버튼을 각각의 카메라에서 얻은 이미지상으로 투영시킴으로써 실 세계 상의 특정 위치에 버튼이 위치하는 것처럼 보이게끔 이미지와 버튼을 합성시킬 수 있게 된다.

대기 구간에서는 실시간으로 카메라의 영상을 캡처하고 그 이미지를 분석하여 시작 신호를 확인한다. 본 논문에서는 손 영역을 나타내는 블록의 중심이 이미지의 중심직사각형 영역에 존재하는지를 판단함으로써 메인 프로세스의 시작을 결정한다. 블록의 중심이 직사각형 안에 존재하게 되면 메인 프로세스를 시작하고 그렇지 않으면 다음 이미지를 캡처하고 다시 시작 신호를 확인한다. 중심직사각형 영역 안으로 블록의 중심이 들어올 때까지 위 작업을 반복 수행하면서 대기상태로 존재한다.

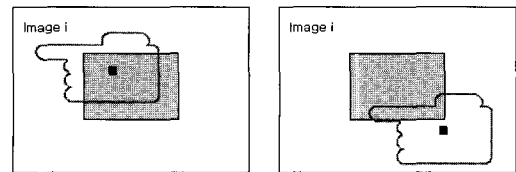


그림 3. 메인 프로세스의 시작(왼쪽) 과 대기(오른쪽) 상태의 예

손 블록의 중심이 직사각형에 위치하게 되어 메인 프로세스가 시작되게 되면 카메라 캡처가 수행되고 이를 통해 얻은 이미지에서 손의 영역을 추출하게 된다. 추출 작업을 통해 얻게 된 손 블록은 녹색, 배경은 검정으로 설정하였다. 다음 작업은 추출된 손의 블록을 통해 실제로 손가락이 가리키는 버튼의 위치를 판단 하는 것이다. 이 작업을 통해 버튼들이 대기버튼 상태인지 또는 후보 버튼 상태인지 아니면 선택버튼 상태인지를 판단하고 각 상황에 따라 버튼의 색상을 변경한다. 이렇게 하나의 캡처된 이미지에 대한 메인 작업을 수행하며 종료 이벤트가 발생하기 전까지 메인 프로세스를 반복 수행한다.

각 과정에 대한 더 자세한 설명은 다음 장부터 이루

어진다.

2. 평면 투영 변형 (가상 버튼 그리기)

카메라로 얻은 image1, image2에 가상의 button을 그리기 위해서는 실제 world plane과 각 image plane들 간의 변형 관계를 알아야 한다. 아래 그림과 같이 우리가 합성하고자 하는 button은 실제 world 상에서 특정 plane에 존재하게 된다. 즉 그 world plane을 이미지에 투영 시킴으로써 world plane에 존재하는 영상들과 가상의 버튼이 image plane에 상이 맞춰지게 된다고 볼 수 있는 것이다. 결국 두 plane들 간에는 Homogeneous Transformation이 존재함을 알 수 있다.

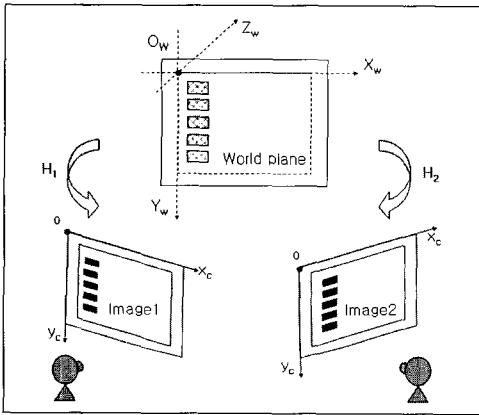


그림 4. world plane과 image plane들간의 기하학적 관계

이 변형관계를 설명하는 H matrix를 구하기 위해 본 논문에서는 전처리 과정으로 실제로 실 세계의 특정 위치에 그림5 와 같은 검정색과 흰색의 직사각형으로 이루어진 패턴 이미지(Image_w)를 설치하고 두 대의 카메라를 그림 5와 같이 설치하여 각 카메라에서 얻은 image plane들과 world plane간의 H matrix를 구하였다.

world point와 그에 대응되는 이미지상의 point간의 관계는 아래와 같이 이루어 진다.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

식1)

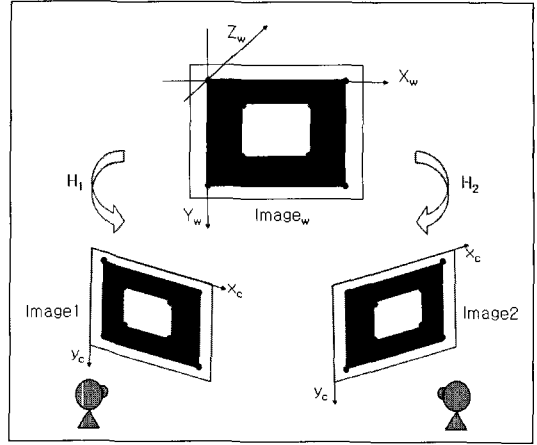


그림 5. 패턴 적용시의 기하학적 관계

여기서 x, y 는 world plane 상에 좌표 값을 말하며, x', y' 는 image plane 상의 좌표 값을 말한다.

이 행렬식을 풀면 다음과 같이 정리 할 수 있게 되며,

$$x' = \frac{u}{w} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad \text{식2-1)}$$

$$y' = \frac{v}{w} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \quad \text{식2-2)}$$

위 식들을 $Ah=0$ 꼴로 구하려 하는 h 값들에 대한 행렬의 곱으로 정리한다. H행렬의 마지막 요소인 h_{33} 을 1로 만들고 H행렬을 h_{33} 로 나누어주게 되면 총 모르는 parameter의 개수는 8개가되고 한 점당 2개의 방정식이 나오므로 결국 최소한 4개의 대응되는 특징 점만 존재하면 H행렬을 구할 수 있게 된다. 본 논문에서는 총 8개의 특징 점을 사용하여 H행렬을 구하였다. 따라서 $Ah=0$ 은 아래 식5)와 같은 형태가 된다. 여기서 A 행렬은 $(i * 2 * 9)$ 의 크기를 가지며, i 는 사용된 점의 개수이다.

3-1. 샘플 데이터 분석

$$\underbrace{\begin{bmatrix} x_1 y_1 1 0 0 0 & -x_1 x_1 - x_1 y_1 - x_1 \\ 0 0 0 x_1 y_1 1 & -y_1 x_1 - y_1 y_1 - y_1 \\ x_2 y_2 1 0 0 0 & -x_2 x_2 - x_2 y_2 - x_2 \\ 0 0 0 x_2 y_2 1 & -y_2 x_2 - y_2 y_2 - y_2 \\ \dots \\ x_i y_i 1 0 0 0 & -x_i x_i - x_i y_i - x_i \\ 0 0 0 x_i y_i 1 & -y_i x_i - y_i y_i - y_i \end{bmatrix}}_A \underbrace{\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}}_h = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{식3)}$$

일반적으로 A 행렬이 row와 column의 크기가 같지 않은 $(m \times n)$ 크기의 행렬일 경우 A 행렬을 SVD (singular value decomposition)함으로써 $\|h\|=1$ 을 만족하면서 $\|Ah\|$ 를 최소화 시키는 h값을 구하게 된다.

$$\text{SVD: } A = UDV^T \quad \text{식4)}$$

이때 V의 마지막 column이 $\|Ah\|$ 를 최소화 시키는 eigenvector가 되고 결국 찾고자 하는 h값이 된다.

위에서 설명한 방법으로 두 카메라의 image plane과 world plane간의 H1과 H2를 구한다. 이렇게 구한 H행렬들을 식2-1)과 식2-2)에 적용해 월드좌표상에 존재하는 가상의 버튼들을 카메라에서 얻은 이미지에 투영시켜 그릴 수 있다.

3. 손 영역 추적

손 영역을 트래킹 하는 작업은 크게 2개의 모듈로 구분할 수 있다. 먼저 손 영역이라고 판단할만한 기준 데이터를 뽑아내는 샘플데이터 분석과, 둘째로 기준 데이터를 바탕으로 주어진 이미지 내에서 손 영역을 찾는 손 영역 추출과정이다. 하지만 실제로 손 영역 추출 과정은 좀더 세분화하여 손 색상 분포를 갖는 부분과 그 외의 배경부분으로 이진화하는 추출(subtraction) 과정과 추출된 흑백이미지의 흰색영역에 대해 blob화 시키는 CCI작업 과정으로 나누어 볼 수 있다.

샘플 데이터는 아래 그림과 같이 $W \times H$ 의 직사각형의 꼴로 주어진다. 직사각형 안에는 손 영역만이 존재해야 한다.

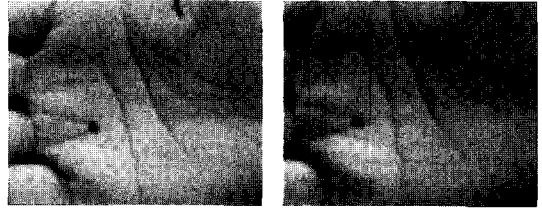


그림 7. 샘플 이미지 데이터: 왼쪽<153x122>, 오른쪽<153x122>

샘플 데이터가 주어지면 데이터의 색상 분포도를 구하기 위해 데이터의 평균과 분산을 구한다. 일반적으로 RGB color space는 3개의 채널에 모두 색상의 밝기 정보를 포함하고 있기 때문에 빛(illumination)의 영향을 많이 받는다. 따라서 색상, 채도, 명도를 따로 표현하는 HSV space로 변경하여 명도는 고려하지 않고 H, S 에 대해서만 고려하도록 한다. 즉 색상과 채도에 대한 샘플 이미지의 평균(hm, sm)과 표준편차($h\sigma, s\sigma$)을 구하고, 손 영역을 판단하는 기준 데이터로 한다. 샘플 손 색상의 분포가 정규 분포를 갖는다는 가정하에 각 이미지의 H와 S에 대해 아래와 같은 신뢰도를 갖는 픽셀이 손의 색상을 갖는다고 판단한다.

$$\left| \frac{hm^i - h_{x,y}^i}{h\sigma^i} \right| < 2.5 \cap \left| \frac{sm^i - s_{x,y}^i}{s\sigma^i} \right| < 2.5 \quad i = \{0, 1\} \quad \text{식5)}$$

3-2. 손 영역 추출

손 영역을 추출하는 프로세스는 크게 2단계로 나누어진다. 먼저 샘플데이터를 통해 얻은 평균과 분산을 이용해 주어진 이미지 상에서 손 색상영역에 속하는 픽셀과 그렇지 않은 픽셀과의 분리를 수행한다. 이때 손의 색상에 속하는 픽셀(pixel)은 흰색 값을 가지며, 그렇지 않은 부분은 검정색을 가진다. 대체적으로 아래 그림8과

같이 정확하게 손의 부분만 추출되는 것이 아니라 배경 부분에서 손의 색상과 유사한 부분까지도 함께 손의 영역이라고 추출됨을 알 수 있다.



그림 8. 손 색상 부분을 추출한 흑백이미지

이러한 잘못된 손 영역의 선택을 제거 하기 위해 다음 작업을 수행하게 된다.

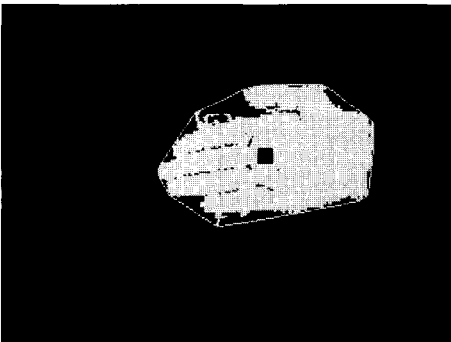


그림 9. 그림 8)을 CCL 작업을 통해 얻은 손 블롭

둘째로 수행되는 것은 첫 번째 작업 수행 후 얻은 흑백 이미지에 CCL(Connected Component Labeling)을 적용하는 작업이다. 일반적으로 흑백 이미지의 흰색 부분 중, CCL 작업을 수행 했을 때 실제 손의 영역이 잘못 선택된 배경 영역들에 비해 상대적으로 가장 큰 사이즈의 블롭을 구성될 것이라는 것은 직관적으로 알 수 있다. 따라서 이 특징을 이용해 손의 블롭을 찾게 된다. CCL작업을 통해 얻은 블롭들 중 영역이 특정 사이즈 이상이 되는 것을 찾고 그 중 가장 큰 영역을 갖는 블롭을 손이라고 판명한다. 물론 나머지 블롭들은 잘못된 선택 영역이라고 판단하고 제거하게 된다. 손 영역이라

고 판단된 블롭은 그림9)와 같이 녹색으로 표현한다. 그림 9에서 파랑 직사각형은 이 손 블롭의 센터를 나타낸다.

4. 버튼 선택

본 장에서는 손의 블롭이 추출된 상태에서 그 블롭이 실제로 어떤 버튼을 선택하였는지를 판단하는 과정을 설명한다. 우선 카메라의 개수가 하나일 때 수행 방법을 소개한 후 2대일 경우 확장되어 추가되는 부분을 설명 하겠다.

일반적으로 손가락 끝이 버튼에 위치할 때 그 버튼에 닿은 손끝이 블롭의 영역 내에서 가장 긴 길이를 갖는다. 이 특징을 이용해서 손이 버튼을 선택하려고 할 때 일차적으로 그 블롭에서 가장 긴 길이를 갖는 행 축을 찾아 그 축에 속하는 버튼을 후보버튼으로 설정한다.

블롭의 긴 축을 찾기 위해서는 이미지의 y축을 따라 각 행에 해당하는 블롭 내의 픽셀 개수를 counting하는 작업을 수행한다. 이렇게 y 축을 따라 counting된 픽셀 개수가 가장 큰 위치, 즉 행의 길이가 블롭 내에서 가장 긴 y축의 위치를 찾고 그 y값 부분에 속하는 버튼을 선택하게 된다.

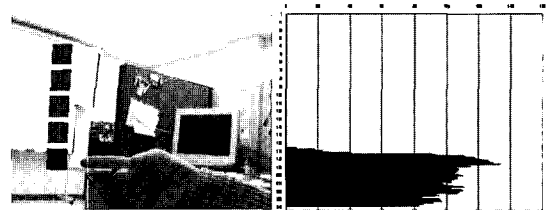


그림 10. 후보버튼 상태와 이때의 블롭의 Histogram

다음으로 하는 것이 실제로 손가락이 버튼 위에 존재하는지를 판단하는 것이다. 이것은 앞에서 이미 선택된 후보버튼의 영역을 스캔 하면서 그 속에 blob의 색상, 즉 녹색이 존재하는지를 확인함으로써 쉽게 해결할 수 있다. 만약 그 버튼 영역에 녹색이 존재한다면 손가락이 버튼 위에 위치하는 것임으로 버튼을 선택했다고 보고 버튼의 색상을 녹색으로 표시한다.

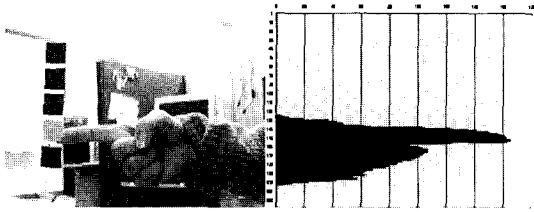


그림 11. 선택버튼 상태와 이때의 블롭의 Histogram

이 것을 2개의 카메라로 확대 시키게 되면 먼저 후보버튼을 각 카메라로 얻은 이미지들에서 찾고 양쪽에서 얻은 후보버튼의 번호가 같다면 양쪽의 해당 버튼 색상을 파랑으로 바꾼다. 다음으로 실제 버튼 위에 손이 존재하는지를 판단하는 것 역시 양쪽의 동일한 버튼에 대해 녹색이 존재하는지 확인함으로써 행할 수 있다. 만약 동시에 녹색이 있다면 하나의 버튼을 선택한 것이 됨으로 그 버튼을 양 쪽 화면에서 녹색으로 변경시켜준다.

5. 실험 및 결과

본 논문에서 사용된 두 대의 카메라는 Logitech Quickcam pro5000 이고 해상도는 320x240이다. 실험은 오피스 내에서 수행되었으며 실시간으로 작동된다. CCL 작업은 openCV 라이브러리를 이용하였다.

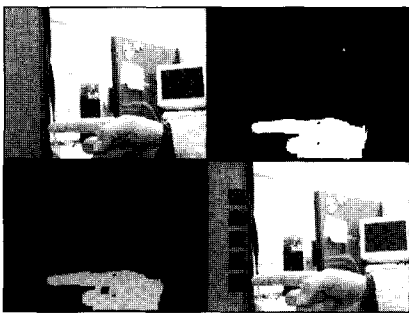


그림 12. 한 이미지에 대한 처리 결과

그림12)는 4개의 320x240이미지를 합쳐서 한 장의 이미지로 만든 것으로 왼쪽 위의 이미지는 카메라에서 캡춰된 원본 이미지이며 오른쪽 위의 이미지는 원본 이미지에서 손 영역만을 추출하여 이진화한 이미지이다. 왼쪽

쪽 아래 이미지는 이진화한 이미지에서 CCL작업을 수행한 후 얻은 손 블롭 이미지이며 마지막으로 오른쪽 아래 이미지는 가상의 버튼을 실제 이미지와 합성시킨 이미지이며 현재 손가락의 동작에 따라 버튼의 상태를 나타내고 있는 이미지이다.

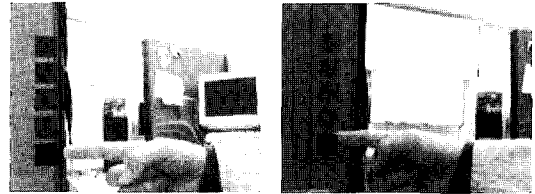


그림13. 후보 버튼 상태의 결과 이미지

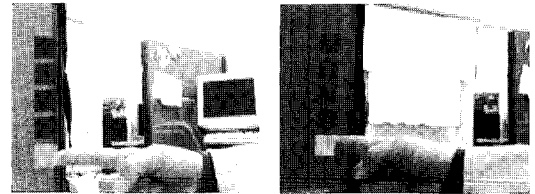


그림 14. 선택 버튼 상태의 결과 이미지

그림13)은 오른쪽 카메라에서는 손가락이 버튼 위에 있으나 왼쪽 카메라에서는 손가락이 버튼 위에 위치하지 않았으므로 선택 버튼이 아닌 후보 버튼상태로 나타난다. 그림14)는 양쪽 카메라에서 모두 손가락이 버튼 위에 존재하므로 선택 버튼으로 상태를 나타내고 있다.

실험 결과 손 영역의 추출이 빛과 카메라의 특성에 크게 영향을 받지 않고 잘 추출됨을 볼 수 있었다. 또한 빠른 수행 처리로 인해 사용에 용이하며 큰 어려움 없이 실시간으로 작동함을 알 수 있었다.

6. 결론 및 추후 과제

본 논문은 see-through 장치의 사용 환경하에서 사용자의 장치 사용에 편리함과 효율성을 제공하기 위한 방법으로 AR 기술을 접목한 버튼형 사용자 인터페이스를 제안하고 있다. 이 방법은 공중에 존재하는 손의 위치를 인식하고 그 손이 가상의 버튼에 가하는 행위에 따라 상호작용적인 결과를 제공한다. 즉 손가락이 특정 가상의 버튼을 선택할 때 그 버튼이 선택되었음을 버튼

의 색상을 변경함으로써 가시적으로 제공하고 있다. 이 방법은 사용자가 단순히 손을 공중에서 이동시키면서 see-through 장치를 통해 어떤 시스템들을 제어하고 실행시킬 수 있는 인터페이스를 제공한다. 따라서 사용이 용이하고 직관적이어서 일반인이 사용하는데 편리함을 제공한다.

추후 과제로는, 현재 손과 버튼이 일직선 상에 존재한다는 가정하에 이 시스템이 작동됨으로 손동장에 있어서 약간의 제약이 존재하는데 이 제약을 제거함으로써 좀더 자유로운 손동작을 통해서도 버튼을 선택하는데 어려움이 없도록 하는 것이다.

7. 참고 문헌

- [1] Richard Hartley and Andrew Zisserman, Multiple View Geometry, p24-64, 2003.
- [2] Rafael C. Gonzalez Richard E. Woods, Digital Image Processing, p282-344, 2001
- [3] R.Jain, R. Kasturi and B.Schunck, Machine Vision
- [4] B.K.P.Horn, Robot Vision
- [5] 김진우, Human Computer Interaction 개론 : 사람과 컴퓨터의 어울림
- [6] Intel Open Computer Vision Library, <http://sourceforge.net/projects/opencvlibrary>