구조적 템플렛 매칭에 기반을 둔 실시간 손 추적 및 인식

김송국[1], 배기태[2], 이칠우[3]
전남대학교 컴퓨터정보통신공학과[1 2 3]
{uaini[1], bkt2002[2], leecw[3]}@chonnam.ac.kr

Real-time hand tracking and recognition based on structured template matching

Kim Song Gook[1], Bae Ki Tae[2] , Lee Chil Woo[3]
Dept. of Computer Engineering, Chonnam University[1 2 3]

요 약

본 논문에서는 유비쿼터스 컴퓨팅 오피스 환경에서 가장 직관적인 HCI 수단인 손 제스처를 사용하여 대형 스크린 상의 응용 프로그램들을 쉽게 제어할 수 있는 시스템을 제안한다. 손 제스처는 손 영역의 정보, 손 중심점의 위치 변화값과 손가락 형상을 이용하여 시스템 제어에 필요한 종류들을 미리 정의해 둔다. 먼저 효율적으로 손 영역 획득을 위해 적외선 카메라를 사용하여 연속된 영상을 획득한다. 획득된 영상 프레임으로부터 구조적 템플레이트 매칭 방법을 사용하여 손의 중심(centroid) 및 손가락끝(fingertip)을 검출한다. 인식과정에서는 양손의 Euclidean distance 와 손가락 형상 정보를 이용하여 미리 정의된 제스처와 비교하여 인식을 행한다. 본 논문에서 제안한 비전 기반 hand gesture 제어 시스템은 인간과 컴퓨터의 상호작용을 이해하는데 많은 이점을 제공할 수 있다. 실험 결과를 통해 본 논문에서 제안한 방법의 효율성을 입증한다.

Keyword : Hand tracking, Gesture recognition, HCI, Template matching,

1. 서론

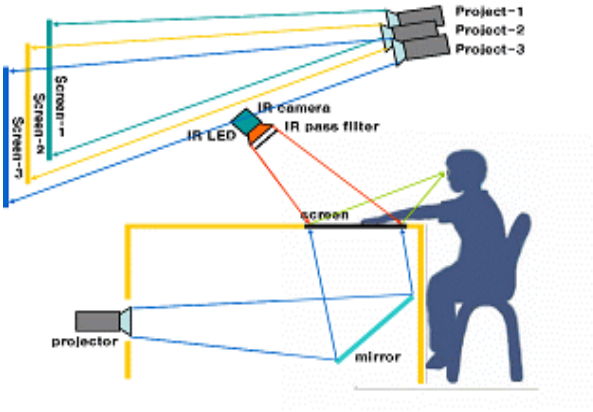
인간과 컴퓨터간의 자연스러운 상호작용을 위하여 시각을 기반으로 한 사용자 의도 및 행위를 인식하기 위한 연구가 활발히 진행되어 왔다. 그 중에서도 손을 이용한 제스처 인식은 시각 기반 인식 분야에서 핵심 기술 분야로 계속 연구되어 왔으며, 이를 이용한 인간과 컴퓨터의 상호작용에 관한 연구가 활발하게 진행되고 있다. 본 논문에서는 오피스 환경의 대형 스크린 상에서 컴퓨터와 가장 쉽게 상호작용할 수 있는 방법으로 손을 이용한 응용프로그램 제어 시스템을 제안한다. 이 분야에서 가장 이른 시도는 Wellner 의 DigitalDesk 에서 보여주고 있다. DigitalDesk 는 CCD 카메라와 빔 프로젝터를 사용하여 구성하였으며, 사용자는 fingertip 을 사용하여 데스크 위에 투영된 application 을 작동시킬수 있다. 현재의 데스크 환경에서의 새로운 시도는 더 많은 application 을 구현하고자 touch 문제를 해결하려는 노력을 보이고 있다. Deitz, P. and Leigh, D. 의 DiamondTouch[1]와 Rekimoto, J.의 SmartSkin[2]은 스크린에 touch 유무 판별 문제를 non-vision 기반 해결책으로 풀어내고 있다. 이 방법들은 fingertip 의 실시간 tracking 을 쉽게 하고자 스크린에 전자적인 장치를 하고 top-down 방식으로 천장에 빔 프로젝터와 카메라를 설치하여 사용하였다. 이 방법의 가장 큰 문제점은 위에서 영상을 투사하기 때문에 손 위에 영상이 투영되어 영상의 왜곡현상 및 손과 영상의 겹침 현상이 일어난다는 것이다. 위에서 언급한 문제들을 고려하

* 본 연구는 전남대학교 “고품질 전기전자부품 및 시스템 연구센터”의 연구비 지원에 의해 수행되었음.

여 본 논문에서는 <그림 1>과 같이 코팅된 투명 스크린을 이용하고 프로젝터를 밑에 설치 및 투사하여 occlusion 및 왜곡 현상을 제거하였다. 제시한 데스크 시스템의 중요한 요소는 실시간으로 fingertip 을 tracking 하는 비전 기반 방법이라는 것이다. 또한 적외선 카메라를 사용하여 어떤 마커 없이도 실시간으로 fingertip 을 신뢰성 있게 검출할 수 있으며, 스크린 배경영상이 복잡한 경우에도 손쉽게 손 영역만 추출할 수 있는 장점이 있다. 컴퓨터와 상호작용을 하기 위한 시스템에서 손과 손가락 모션의 사용은 일반적으로 두 가지로 분류된다. 하나는 손 형상 및 손가락 형상의 변화를 이용하는 것이고, 또 다른 하나는 손의 움직임 변화를 이용하는 것이다. 이에 본 논문에서는 응용프로그램을 제어하는데 사용하는 명령들로 어려운 제스처를 이용하기 보다는 가급적이면 편하고 쉽게 학습하여 사용할 수 있는 제스처들을 이용하였다. 최종적으로 우리가 제안하는 시스템은 개인 홈페이지를 쉽게 꾸밀 수 있도록 도와주는 핸드제스처를 기반으로 하는 사진 편집 시스템이다. 본 논문에서 제안한 비전 기반 hand gesture 제어 시스템은 인간과 컴퓨터의 상호작용을 이해하는데 많은 이점을 제공할 수 있으며, 최소한의 학습으로 작동할 수 있는 장점이 있음을 확신한다. 본 논문은 다음과 같이 구성된다. 2 장에서는 제안한 시스템의 하드웨어 구성을 설명하고, 3 장에서는 본 논문에서 제안한 핸드 제스처 인식 과정을 설명한다. 4 장에서는 우리가 제안한 방법을 사용하여 응용프로그램 작동 결과를 살펴본다. 마지막으로 5 장에서는 결론 및 향후 연구방향을 제시한다.

2. 하드웨어 구성

우리가 제안한 시스템의 데스크 환경은 <그림1>에서 보는 바와 같이 프로젝터는 데스크 앞쪽에 설치되어 있으며 거울로 반사시켜 사용자가 데스크를 내려다 보면서 작업하도록 하였다. 스크린의



<그림1> 하드웨어 구성도

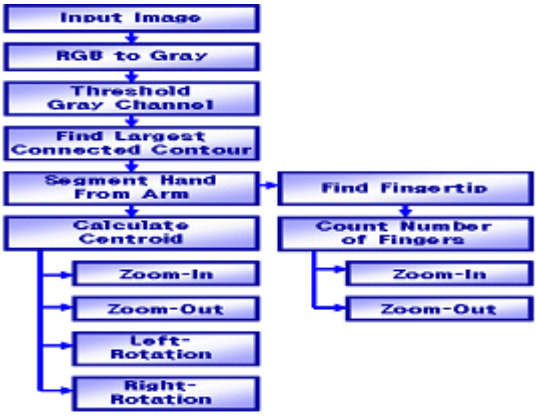
<그림 2> 데스크 인터페이스 시스템

넓이는 120cm × 90cm 이며, 두께는 약 7mm 정도로서 일반 통유리에 필름을 코팅하여 제작하였다. 스크린의 크기가 매우 크기 때문에 스크린과 프로젝터간의 거리가 가까우면 투영되는 영상이 작아지므로 직접 스크린에 투영하지 못하고, 왜곡을 줄이기 위해 최대한 얇은 거울을 사용하여 반사시켜 스크린에 투영시켰다. 실험에 사용된 PC 는 Intel Pentium4 3.0GHz CPU, 1024MB RAM 의 사양을 가지고 있으며 1 대는 테이블의 스크린과, 2 대는 테이블 앞의 wall display 에 연결되어 있다. 스크린 위의 손은 천장 위에 설치된 IR pass filter 를 부착한 적외선 카메라를 이용하여 실시간으로 검출하였다. 적외선 카메라는 320×240 화소의 영상을 초당 30 프레임 속도로 동영상을 전송한다. 적외선 LED 가 포함된 적외선 카메라는 스크린에 수직으로 설치했을 경우 스크린에 적외선 LED 가 반사되어 손 영역 검출에 어려움을 초래하므로 약 45 도 각도로 비스듬히 설치하였다.

<그림 2>는 우리가 최종적으로 구현할 wall display 와 table-top display 간의 상호작용, 스크린에 투사된 objects 와 스크린 위의 physical objects 간의 상호작용을 통하여 인간과 컴퓨터가 상호작용하는데 한층 더 효율적이며 쉽게 다가갈 수 있는 전체적인 시스템 환경을 보여준다.

3. 핸드 제스처 인식 과정

우리가 제안한 제스처 인식 시스템은 인텔사의 Image Processing Libraries (IPL)과 Open Computer Vision library (OPENCV)를 이용하여 마이크로 소프트 C++ 컴파일러로서 개발하였다. 이 라이브러리는 특별한 영상처리 기능을 포함하고 있어서 효율적으로 계산을 수행할 수 있게 해준다. 최근 Table-top Display 상에서 인터페이스를 하는데 있어서 non-비전 기반 방법을 사용하는데, 우리는 비전 기반 방법만을 사용하여 hand 와 fingertip 을 검출하고 tracking 할 것이다. 본 논문의 제스처 인식 시스템은 사용자의 손의 움직임과 손가락 형상을 인식하여 응용프로그램을 제어하는 기능을 가지고 있다. 또한 입력되는 영상을 정확히 데스크 위의 스크린 크기에 맞추어서 가로축, 세로축으로 정확한 위치를 측정하도록 하기 위해 카메라 칼리브레이션 처리를 하였다. <그림 3>은 본 시스템에서 제안하는 알고리즘의 흐름도를 보여주고 있다.



<그림 3> 제안된 알고리즘의 흐름도

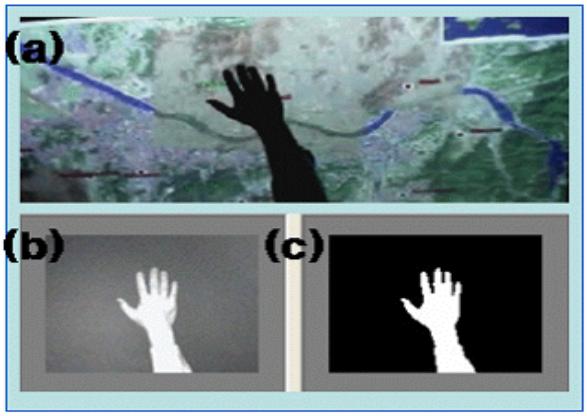
3-1. 손 영역의 추출

보통 피부색을 이용한 세그먼테이션이나 배경 차분 방법을 사용하는 경우에는 조명이 동적으로 변하거나 복잡한 배경 영상을 가지고 있는 경우에는 효과적으로 손 영역을 추출하지 못한다. 그러나 우리가 제안한 시스템에서는 빔 프로젝터가 투사한 영상에 피부색이 있거나 또는 배경이 복잡한 경우에도 IR pass filter 를 부착한 적외선 카메라와 적외선 LED 를 사용하여 효율적으로 손 영역을 추출할 수 있다. <그림 4>는 복잡한 배경 영상을 가진 스크린 위의 손 사진(a)와 적외선 카메라를 통해서 들어온 영상(b), 입력영상을 이진화 시킨 영상(c)을 보여주고 있다. 먼저 적외선 카메라를 통해서 실시간으로 영상이 입력되면, BGR 기반의 컬러영상을 Gray 영상으로 바꾸어준다. 변경된 Gray 영상은 식(1)을 이용하여 임계값 미만의 픽셀은 0 으로 임계값 이상의 픽셀은 255 값으로 하는 이진화 처리를 거친다. 이진화 처리를 한 후 잡음을 제거하기 위하여 식(2)와 같이 opening, closing 연산을 적용한다.

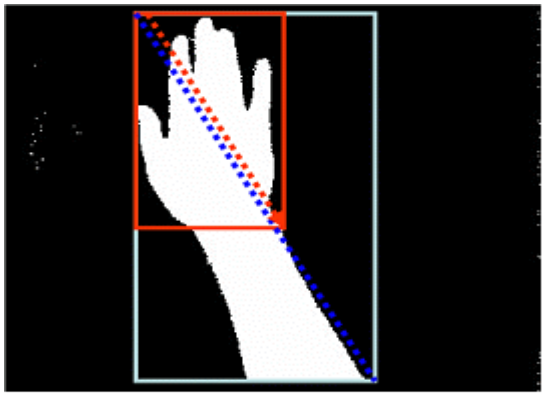
$$P'(x,y) = \begin{cases} 255 & P(x,y) \geq T \\ 0 & P(x,y) < T \end{cases} \qquad \text{식(1)}$$

$$f \circ b = (f \ominus b) \oplus b$$
$$f \bullet b = (f \oplus b) \ominus b \qquad \text{식(2)}$$

영상에서 잡음을 제거한 후 가장 큰 영역을 추출하기 위해 labeling 과정을 수행한다. 다음으로 적외선 카메라를 통해 입력된 영상에서는 <그림 5>와 같이 손을 포함한 팔 영역 전부가 추출되었으므로 손 영역만을 추출하기 위하여 추출된 팔 영역의 윈도우에서 좌측상단의 시작점을 기준으로 윈도우대각선 모서리를 연결하여 일정 픽셀(60pixel)을 대각선으로 내려오는 지점까지를 잘라서 윈도우를 생성하면 손 영역만 추출되게 된다. 이 방법은 손이 좌, 우로 회전되는 경우에도 어느 정도 효율적인 결과를 얻을 수 있다. 본 논문에서는 적외선 카메라로부터 스크린 위의 손 까지는



<그림 4> 스크린 영상과 카메라 입력 영상


<그림 5> 팔 영역에 손 추출

항상 일정한 거리를 유지하고 있으며 손은 의도적으로 하지 않는 이상 좌,우로 심하게 꺾이지 않으며 거의 앞쪽에 존재하는 것으로 가정한다.

3-2. 템플릿 매칭을 이용한 fingertip 검출 및 추적

추출한 손 영역에서 fingertip 을 검출하기 위해서 <그림 6>과 같이 template matching 방법을 사용한다.[3] 우리가 찾고자 하는 fingertip 을 템플릿으로 정의하고 목표 영상인 손 영역에서 템플릿 매칭을 수행한다. 비교 방법은 주로 두 영상의 유사도를 계산하는 상관계수를 사용한다. 즉, 유사도가 높을수록 객체일 확률이 높다는 것을 이용하여 fingertip 위치를 찾는다. 유사도 측정 방법으로 유클리디언 거리를 이용하는 방법, 상관도(correlation)를 이용하는 방법, 상관계수를 이용하는 방법이 있으나 본 논문에서는 상관계수 방법을 사용했으며 상관계수의 범위는 -1 에서 1 까지이다. 상관계수 값이 -1 이나 1 에 가까워질수록 상관성이 커지며, 0 에 가까워질수록 상관성이 없어진다. 특히, 템플릿 매칭에서는 1 에 가까울수록 객체와 유사하다. fingertip 에 해당되는 템플릿을 가지고 3 장에서 구한 손의 윈도우 안에서 차례대로 template matching 을 수행하면서, 상관계수의 최소, 최대 값이 나온 위치를 찾는다. Fingertip 에 해당하는 후보자들 중에서 가장 높은 값을 갖는 것을 fingertip 으로 인식하고 이를 계속적으로 tracking 한다. <그림 7>은 template matching 방법을 이용해서 fingertip 을 찾은 결과를 보여주고 있다.


<그림 6> fingertip template

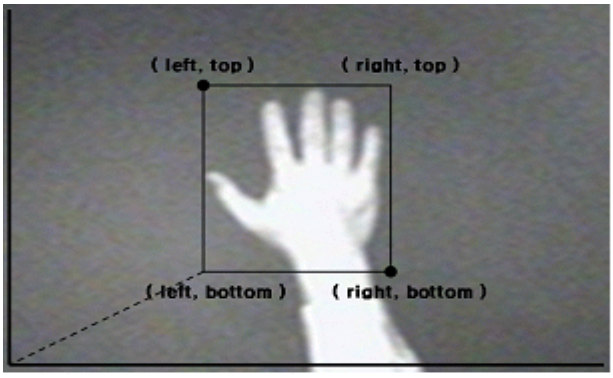
<그림 7> fingertip detection

3-3. 두 손의 중심점 찾기 및 거리와 각도 계산

손의 중심을 찾는 목적은 두 손의 움직임의 변화를 이용하여 사진 편집 툴의 불러온 사진을 zoom-in, zoom-out, left-rotation, right-rotation 하기 위함이다. 손의 중심은 무게 중심을 이용하여 검출하고, 계속적으로 추적한다. 3-1 에서 찾은 손 영역에서 손의 중심은 <그림 8>과 같이 window 를 생성함으로써 보다 쉽게 구할 수 있다. 비록 손이 좌우로 회전되는 상황에서는 정확하게 손의 중심을 찾지 못하는 경우도 있으나, 의도적인 경우를 제외하고 손이 좌우로 많이 회전되지는 않으며, 두 손을 이용한 명령 제스처 경우는 정확하게

손의 centroid 를 찾지는 않아도 된다. 두 손을 이용한 제어에서는 왼손과 오른손이 서로 교차하지는 않는다고 가정을 하였으며 <그림 8>과 같이 보이는 영상에서 식(3)의해서 손의 중심을 찾게 된다. 또한 두 손의 Euclidean Distance 와 두 손의 각도를 <그림 9>에 보이는 것처럼 식(4)에 의해서 구하게 된다.

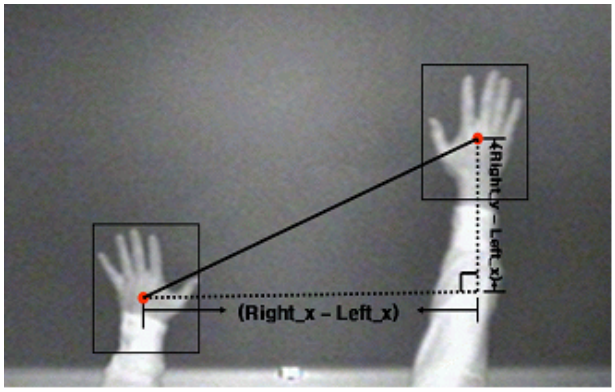
$$hand\_x = (rect.right - rect.left) \times 1/2 + rect.left$$
$$hand\_y = (rect.top - rect.bottom) \times 1/2 + rect.bottom \quad \text{식(3)}$$



<그림 8> 손의 centroid 찾기

$$\theta = \mathrm{Tan}^{-1}\frac{(Right.hand\_y - Left.hand\_y)}{(Right.hand\_x - Left.hand\_x)} \quad \text{식(4)}$$

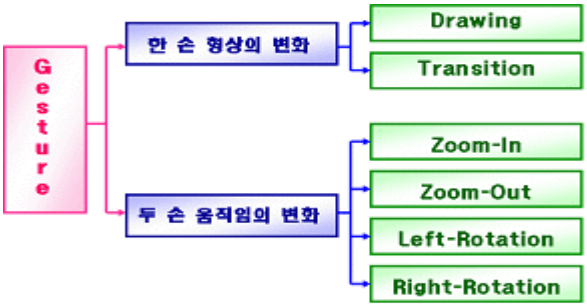
$$\sqrt{\{(Right.hand\_x - Left.hand\_x)^2 + (Right.hand\_y - Left.hand\_y)^2\}}$$



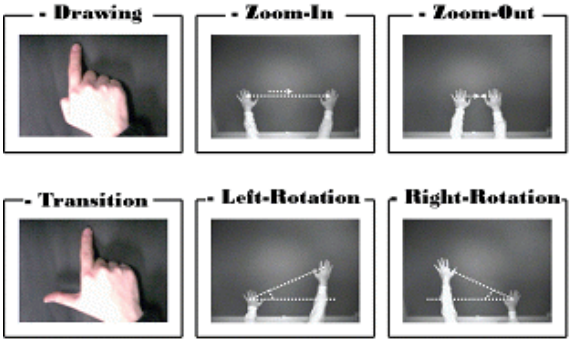
<그림 9> 두 손의 거리 및 각도 계산

3-4. 제스처 인식

터치 센서 없이 프리 핸드 Drawing 을 가능하게 하는 데서 가장 중요한 문제는 drawing 명령과 transitional 명령의 확실한 구별이 필요하다는 것이다.[4] 예를 들면 사용자가 데스크 위에서 그리는 중인지 아니면 다음 누르는 곳의 시작 지점으로 fingertip 을 재배치시키는지의 구별이 필요하다는 것이다. 우리가 제안한 시스템에서 핸드 제스처는 <그림 10>과 같이 두 분류로 나누어서 생각할 수 있다. 한 손으로 제어하는 경우는 <그림 11>에서 보는 바와 같이 Drawing 명령이 활성화된 상태와 비활성화된 상태로 나눌 수 있다. Drawing 상태로의 변환은 엄지손가락과 집게 손가락을 둘 다 펼친 상태에서 집는 형상을 취함으로써 명령을 수행하게 된다. 이때 집게 손가락의 fingertip 은 실시간으로 tracking 되어야 하기 때문에 엄지손가락을 오므리는 것만으로 Drawing 상태로 변환하도록 하였다. 위와 같이 제스처를 만든 이유는 마우스를 클릭하는 이벤트가 손가락으로 가상의 물체를 집는 것과 유사하다고 생각하기 때문이다.



<그림 10> 제스처의 분류



<그림 11> 미리 정의된 제스처 종류들

Drawing 과 transition 상태의 구분은 fingertip 검출을 위해 template matching 을 수행할 때 손등에서 fingertip 까지의 중심축을 이용하여 구한다. 이 중심축을 기준으로 손 영역 윈도우의 좌측 에지까지의 간격이 임계값(18pixel) 이상 차이가 나면 엄지손가락 펴진 상태 즉, transitional 상태로 인식하여 fingertip 을 재배치 시키기 위해 움직이는 상태로 본다. 만약 간격이 임계값(18pixel)보다 작다면 drawing 상태로서 마우스 클릭 이벤트를 발생시킨다. 두 손을 이용하여 제어하는 경우에는 두 손의 움직임의 변화값으로 두 centroid 거리의 차와 각도를 이용한다. 즉, 계산된 각도가 15 도 이하이고 Euclidean distance 의 값이 120 픽셀 이상이면 Zoom-In 동작을 수행하고, 80 픽셀 이하이면 Zoom-Out 동작을 수행한다. 또한 이와 더불어 두 손의 centroid 거리가 100 픽셀 이상이고, 각도가 30 도 이상일 경우에는 좌회전을 수행한다. 반대로 각도가 -30 도 이상일 경우에는 우회전을 수행하도록 하였다. 이 때 단지 두 손의 위치 값만을 다루며, 손이 움직이는 속도는 계산하지 아니한다.

4. 응용 프로그램 작동

우리가 제안하는 응용 프로그램은 사용자가 편안하고 효율적으로 사진을 꾸밀 수 있도록 도와주는 툴이다. 이 툴에서 제공하는 서비스로는 손가락을 이용한 글씨쓰기 및 기존에 저장되어 있는 클립아트를 이용한 사진틀 씌우기 등 이다. 마우스에서 사용되는 기능으로는 마우스의 이동 외에 마우스 왼쪽 버튼을 이용한 클릭, 더블클릭과 오른쪽 버튼의 클릭 및 휠의 회전이 사용된다. 여기서 우리는 주로 사용하는 왼쪽 버튼의 클릭과 마우스 이동의 이벤트 핸들을 손 제스처를 이용하여 명령으로 구현한 것이다.

5. 결론 및 연구방향

본 논문에서 제안한 시스템은 테이블 위의 대형 스크린 위에서 응용프로그램을 손을 사용하여 제어하도록 한 것이었다. 프로그램을 조작하는데 있어서 가장 중요한 것은 정확하게 fingertip 을 검출 및 추적하는 것이었으나, 간혹 잘못 추출하는 경우가 발생했다. 따라서, template matching 이외의 방법을 이용하여 보다 강건하게 fingertip 을 추출하는 방법을 고려할 것이다. 또한 실시간으로 손을 tracking 하는 방법에 있어서도 일단 손을 인식하게 되면 두 손이 겹치는 경우가 있더라도 정확하게 계속적으로 tracking 하도록 하기 위해서 Kalman filter 법을 사용할 계획이다. 더 복잡하고 보다 나은 응응 프로그램을 쉽고 효율적으로 작동시키기 위해서는 스크린의 손가락 touch 유무 판별이 매우 중요하다. 우리는 touch 유무 판별 문제를 비전 기반 방법을 사용하여 풀기 보다는 전도성 투명 필름을 사용하여 해결할 것이다. 또한, 인간과 컴퓨터의 상호작용뿐만 아니라 스크린 속의 displayed objects 와 스크린 위의 physical objects 간의 상호작용도 고려할 것이다. 마지막으로 데스크 환경의 최대 장점은 서로 얼굴을 마주보고 협력하여 작업을 수행할 수 있다는 것이다. 앞으로 수행되는 연구에서는 위에서 언급한 세 가지 사항을 중점에 두고 생각하고, 구현할 것이다.

< 참고 논문 >

[1] Deitz, P. and Leigh, D. (2001). DiamondTouch: A Multi-User Touch Technology. *In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST) 2000*, pp. 219-226.

[2] Rekimoto, J. (2002). SmartSkin: An Infrastructure for freehand manipulation on interactive surfaces. *In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI) 2002*, pp. 113-120.

[3] K. Oka, Y.Sato, and H. Koike, "Real-Time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition* (FG 2002), IEEE CS Press, 2002, pp. 429-434.

[4] Zhenyao Mo, J.P.Lewis, Ulrich Neumann "SmartCanvas: a gesture-driven intelligent drawing desk system " *In Proceedings of the 10th international conference on Intelligent user interfaces table of contents* (2005), pp. 239-243