

논문 2011-48SP-3-1

# DI 방식의 대형 멀티터치스크린을 위한 영상처리 알고리즘 설계

( Image Processing Algorithms for DI-method Multi Touch Screen  
Controllers )

강 민 구\*, 정 용 진\*\*

( Min Gu Kang and Yong Jin Jeong )

## 요 약

대형 멀티터치스크린의 구현은 기존의 저항막, 정전용량, 초음파 방식으로는 기술 제약 및 비용 등의 문제로 구현에 어려움이 있어 주로 적외선을 이용하는 방식을 많이 사용한다. 적외선을 사용한 멀티터치스크린은 대형스크린을 구현하기 용이하지만 멀티터치에 기술적인 제한을 갖고 있는 경우가 많다. 이러한 단점을 보완하기 위해 적외선 카메라를 이용한 FTIR(Frustrated Total Internal Reflection), DI(Diffuse Illumination)방식들이 Microsoft의 차세대 유저인터페이스인 Surface를 통해 제안되었다. FTIR이나 DI 방식은 대형스크린의 구현이 쉽고 멀티터치의 개수에 제한을 받지 않는다. 하지만 FTIR은 터치 포인트의 검출은 쉬운 반면에 스크린의 크기와 재질, 적외선 LED 배열을 위한 모듈, 많은 소비전력 등의 단점을 가지고 있고 DI 방식은 구조상의 문제로 터치 검출이 어려운 반면 FTIR이 가지고 있는 단점을 해결할 수 있다. 본 논문에서는 기존에 제안된 DI 방식의 터치 포인트 검출시의 문제점을 해결하기 위해 손가락 외곽선을 이용한 영상처리 알고리즘, 광학 렌즈 왜곡 현상을 효과적으로 보정하기 위한 알고리즘에 대해 연구하였다. 또한 멀티터치의 터치 정확도를 높이기 위한 Calibration 알고리즘과 정확한 제스처 및 정확한 이동을 위한 Tracking 기법을 고안하였다. 연구 결과 DI 방식을 위해 본 논문에서 제안한 영상처리 알고리즘들은 간단하면서도 쉽게 대형 멀티터치스크린 구현을 위한 효과적인 방법이 될 수 있을 것으로 판단된다.

## Abstract

Large-sized multi-touch screen is usually made using infrared rays. That is because it has technical constraints or cost problems to make the screen with the other ways using such as existing resistive overlays, capacitive overlay, or acoustic wave. Using infrared rays to make multi-touch screen is easy, but is likely to have technical limits to be implemented. To make up for these technical problems, two other methods were suggested through Surface project, which is a next generation user-interface concept of Microsoft. One is Frustrated Total Internal Reflection (FTIR) which uses infrared cameras, the other is Diffuse Illumination (DI). FTIR and DI are easy to be implemented in large screens and are not influenced by the number of touch points. Although FTIR method has an advantage in detecting touch-points, it also has lots of disadvantages such as screen size limit, quality of the materials, the module for infrared LED arrays, and high consuming power. On the other hand, DI method has difficulty in detecting touch-points because of it's structural problems but makes it possible to solve the problem of FTIR. In this thesis, we study the algorithms for effectively correcting the distort phenomenon of optical lens, and image processing algorithms in order to solve the touch detecting problem of the original DI method. Moreover, we suggest calibration algorithms for improving the accuracy of multi-touch, and a new tracking technique for accurate movement and gesture of the touch device. To verify our approaches, we implemented a table-based multi touch screen.

**Keywords :** Diffuse Illumination, Multi Touch, Infrared image processing, Large Touch-Screen, Infrared Touch

\* 학생회원, 광운대학교 임베디드 소프트웨어공학과

(Department of Embedded software Engineering, Kwangwoon University)

\*\* 정회원, 광운대학교 전자통신공학과

(Department of Electronic and Communication Engineering, Kwangwoon University)

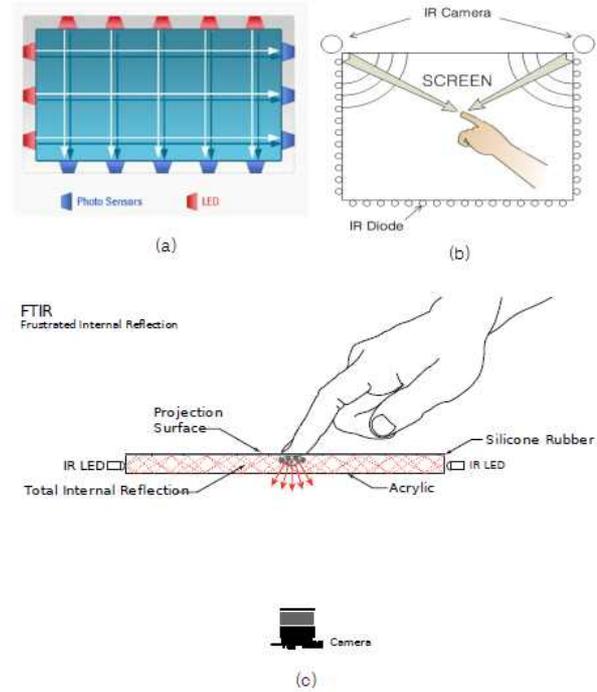
※ 본 논문은 서울시 산학연 협력사업(KU080661) 및 한국연구재단의 기초연구사업(NRF-2010-0014557) 지원으로 이루어졌습니다.

접수일자: 2010년9월13일, 수정완료일:2011년2월15일

### I. 서론

최근 멀티터치스크린은 인간과 컴퓨터 상호작용을 위해 많은 분야에서 연구, 사용되어져 왔다. 컴퓨터를 좀 더 인간에게 쉽고 쓸모 있게 함으로써 인간과 컴퓨터 간의 상호작용을 개선하기 위해 2002년부터 FTIR 방식으로 시작된 멀티터치는 현재까지 많은 분야에서 연구되어 왔으며 현재에는 아이폰의 열풍과 Windows 7의 보급으로 인해 멀티터치에 대한 개발과 사용이 본격적으로 활성화가 되었다. 이러한 추세는 기업, 가정, 개인, 학교 등 수많은 부분에서 멀티터치를 사용한 기기의 필요성이 높아져 전자칠판 및 대형화면을 이용한 광고판, 지하철 정보 시스템, 전자 발권기, 전시회에서 대형 터치스크린 등 다양한 환경에서 대형 멀티터치스크린의 필요성이 요구되고 있다. 멀티터치스크린을 구현하기 위한 방법으로 저항막 방식 (Resistive Overlay), 정전용량 방식 (Capacitive Overlay), 초음파 방식 (Acoustic Wave) 등이 사용되고 있지만 대형 멀티터치스크린의 구현을 위해서는 높은 비용과 스크린 크기의 제한, 멀티터치의 한계 등의 문제점을 가지고 있다. 이러한 문제점을 해결하고 대형스크린에 적용할 수 있는 대안으로 그림 1과 같이 적외선 어레이 (Infrared Array), 적외선 라인 스캔 (Infrared line Scan), FTIR (Frustrated Total Internal Reflection) 등이 사용되고 있으며, 그림 1-a는 적외선 어레이 방식으로 적외선 발광소자와 수광 소자 간의 터치로 인해 차단된 구간을 감지하여 터치를 인식하는 방식이다. 그림 1-b는 적외선 라인 스캔 센서와 적외선 발광 소자를 사용해 터치를 검출하는 적외선 라인스캔 방식이며 그림 1-c는 DI 방식과 유사한 FTIR 방식으로 스크린 측면에서 적외선을 반사시켜 터치 시 오브젝트에서 반사되는 빛을 이용하는 방식이다.<sup>[1]</sup> 그림 1의 멀티터치스크린 방식들은 스크린이 커질수록 외부에 설치되는 적외선 발광 소자의 개수가 비례적으로 증가하게 되어 추가 설계 및 비용이 커지는 문제를 가지고 있으며 적외선 간의 간섭이나 외부의 빛, 멀티터치에 제한을 가지고 있다.

DI 방식은 내부의 적외선 발광 모듈과 적외선 카메라를 이용해 오브젝트에서 반사되는 빛을 이용해 멀티터치스크린을 구현하는 방법이다. DI 방식을 이용한 멀티터치스크린은 그림 1과 같은 다른 멀티터치스크린 방식에 비해 대형스크린에 유연하게 대처할 수 있다. 또한 적외선의 개수와 상관없이 스크린의 크기를 조절 할



(a) 적외선 어레이 방식 (b) 적외선 라인 스캔 방식  
(c) FTIR(Frustrated Total Internal Reflection) 방식

그림 1. 적외선을 이용한 터치스크린  
Fig. 1. Touch Screen using Infrared.

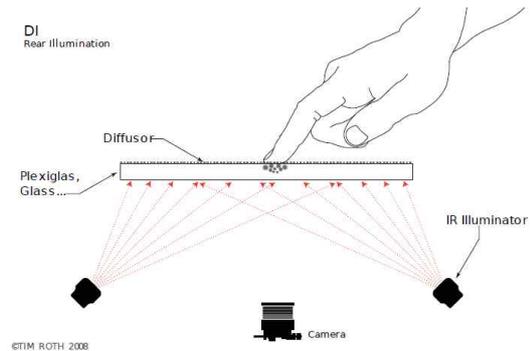


그림 2. DI(Diffuse Illumination) 방식  
Fig. 2. DI(Diffuse Illumination) Method.

수 있으며 카메라의 시야가 확보가 된다면 어떤 스크린의 크기로도 쉽게 조절이 가능한 특징을 가지고 있다. 또한 다른 적외선 터치스크린에 비해 멀티터치 포인트 개수에 제한이 없으며 스크린을 적외선 카메라를 통해 적외선 영상을 입력받아 처리하기 때문에 외부 환경적 요인에 크게 영향을 받지 않고 멀티터치 포인트를 추출할 수 있다.

하지만 다른 대형터치스크린들 방식들의 문제점인 대형 터치스크린의 구현에 있어 큰 장점을 갖는 DI 방식이지만 기존의 알고리즘의 경우 반사된 빛만 사용해

정확한 터치 동작이 어려운 문제점을 가지고 있다. 따라서 주로 FTIR 방식을 이용한 대형 멀티터치스크린이 연구되고 있는 실정이다. 본 논문에서는 기존의 DI 방식의 알고리즘의 단점을 보완하여 간단한 시스템으로 대형 멀티터치스크린을 쉽게 구현하고, 터치 검출 성능을 개선하여 터치 포인트의 개수에 구애받지 않는 영상처리 알고리즘을 설계하였다.

## II. 본 론

### 1. DI(Diffuse Illumination)방식의 터치스크린

적외선을 이용한 DI 방식의 멀티터치스크린은 그림 3과 같이 프로젝터를 이용해 화면에 영사하고 적외선 카메라와 가시광선 차단 필터를 통해 적외선 영상만을 입력받아 영상처리에 사용한다. DI 방식은 그림 2<sup>[1]</sup>와 같이 적외선 빛이 스크린 아래에서 비추며, 터치 시 터치 주변으로 적외선을 반사하는 특징을 가지고 있다. 다른 방식(FTIR, LLP)<sup>[1]</sup>들에 비해 표면 처리가 불필요하며 투명 재료면 사용이 가능해 스크린에 대한 제약이 적고 간단한 설치로 멀티터치 스크린의 동작이 가능한 특징을 가지고 있다. 하지만 기존에 사용되고 있는 DI 방식은 아래에서 비추는 방식으로 인해 오브젝트의 반사되는 부분이 손가락 이외의 부분에서 반사되는 경우가 많고, 터치를 하기 위한 손의 위치가 매우 중요하여 오작동이 일어날 수 있는 문제점을 가지고 있다.

#### 가. DI 방식의 멀티터치스크린 구조

본 논문에서는 DI 방식의 멀티터치스크린을 위한 알

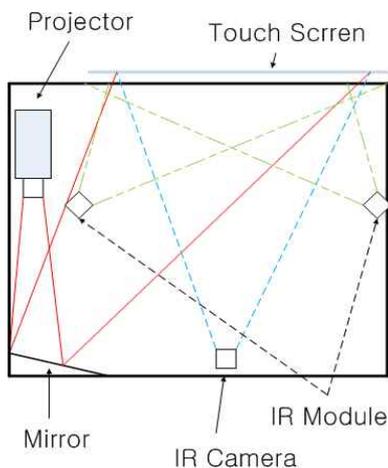


그림 3. 멀티터치스크린  
Fig. 3. Multi-Touch Screen.

고리즘을 설계하기 위해 그림 3과 같은 구조의 시스템을 사용하였다. 1개의 적외선 카메라는 스크린을 바라보며 터치된 손가락에서 반사된 적외선 영상을 입력받으며, 4개의 적외선 발광소자는 손가락의 적외선 반사를 위해 스크린을 비추며 프로젝터는 스크린에 화면을 영사하기 위해 거울을 이용하여 영사한다.

### 2. 멀티터치 영상처리 알고리즘

본 논문에서 제안한 DI 방식의 영상처리 알고리즘의 순서는 다음과 같다.

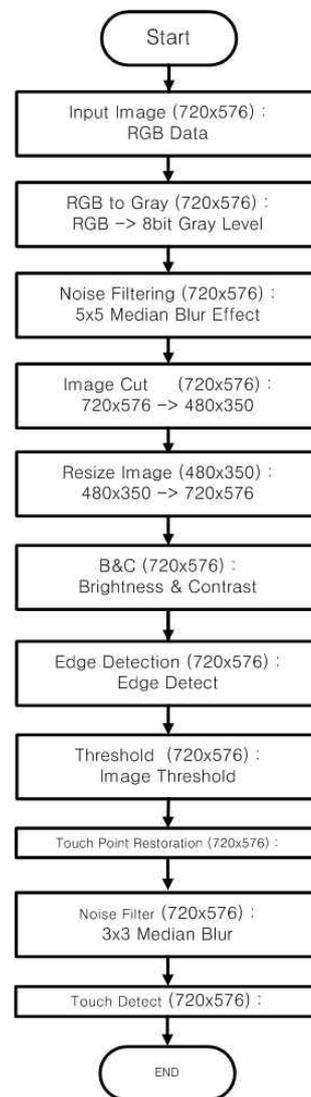


그림 4. 영상처리 알고리즘  
Fig. 4. Image Processing Algorithm.

- (가) RGB to Gray 단계
- (나) 노이즈 제거

- (다) 터치 영역 추출 (Image Cut)
- (라) 영상 확대 (양선형 보간법)
- (마) 밝기, 명암비 조절
- (바) 외곽선 검출 (Edge Detection)
- (사) 터치 검출을 위한 기준 값(Threshold)
- (아) 터치 검출을 위한 터치 포인터 복원
- (자) 노이즈 제거 (3x3 Noise Filtering)
- (차) 터치 검출

(가) RGB to 그레이 변환

본 논문에서 사용된 멀티터치 알고리즘은 적외선 영상을 이용하기 때문에 RGB의 색상정보가 필요하지 않으며 빠른 영상처리를 위해서 8bit 그레이(Gray Level)로 변환하여 영상처리를 하게 된다.



그림 5. 입력 영상 (RGB)  
Fig. 5. Input Image.

(나) 노이즈 제거

DI 방식의 멀티터치 스크린은 기본적으로 적외선의 난반사를 이용해 터치 시 반사되는 적외선을 이용하고 화면을 영사하기 위해 영사지를 사용한다. 적외선 난반사로 인해 영사지의 표면에서 나타나는 적외선이 카메라를 통해 노이즈로 들어오게 되고 이는 소금 후추 노이즈(Salt and pepper noise)라고 불리는 임펄스 노이즈로 나타나게 된다. RGB의 입력 이미지는 나타나지 않지만 Gray로 변환한 이미지에서는 노이즈가 나타나게 되며, 발생하는 임펄스 노이즈는 영상처리에 많은 영향을 줄 수 있으며 터치 포인트 추출에 오작동을 일으킬 수 있는 문제점을 가지고 있다. 이러한 문제점을 효과적으로 제거하기 위해 미디언 블러를 이용한 필터를 사용하여 효과적으로 제거한다. 미디언 필터는 비선형 공간적 필터링 기법으로 입력 영상의 (x, y)좌표 주변 픽셀 값들을 오름 또는 내림차순으로 정렬하여 그

102	67	84
224	189	33
212	163	54

그림 6. 3x3 마스크  
Fig. 6. 3x3 Mask.

중앙에 있는 픽셀 값을 이용하는 노이즈 제거 필터이다. 본 논문에서는 5x5크기의 마스크를 사용하여 자신을 포함한 총 25개의 픽셀 값을 일렬로 늘어 세운 후, 이를 픽셀 값의 크기순으로 정렬한다. 그리고 정렬된 데이터에서 중앙에 있는 픽셀 값의 크기를 사용해 영상의 값을 변경하고 이를 전체 픽셀에 대해서 수행한다. 정렬 알고리즘은 삽입 정렬(insert sort)를 사용한다.

미디언 필터의 3x3 마스크 영역의 값이 그림 6과 같다면 값을 크기순으로 정렬하여 중간 값을 찾게 된다. 중간 값인 102를 찾으면 마스크안의 픽셀을 모두 102로 채우게 된다. 마스크가 씌어진 9개의 값을 읽어 들여 크기가 큰 순서로 정렬한 후 중간크기의 값을 사용하기 때문에 임펄스 노이즈를 제거하는데 효과적인 알고리즘이다. 본 논문에서는 여러 개의 픽셀에 발생한 노이즈를 최대한 효과적으로 제거하기 위해 5x5 크기의 마스크를 사용한다.



그림 7. 미디언 필터 적용 후  
Fig. 7. After apply the Median Filter.

(다) 터치 영역 추출

스크린의 크기에 제약을 적게 받는 DI 방식의 경우 목표 스크린의 사이즈에 따라 터치 영역을 검출하여 영상처리 할 부분만을 추출해 영상사이즈를 조절할 필요



그림 8. 터치 영역  
Fig. 8. Touch area.

가 있다. 터치의 목적과 스크린의 크기에 따라 터치의 영역이 달라지기 때문에 터치영역을 검출하는 작업은 반드시 필요하다. 기존에 제시된 알고리즘은 터치영역을 카메라 전체로 보기 때문에 스크린의 크기를 조절하기 위해서는 카메라의 위치를 조절해야만 했다. 하지만 본 논문에서는 터치 영역을 구별한 후 실제 영상처리가 돼야 하는 부분을 추출하게 된다. 터치 영역 추출은 전체 이미지 영역에서 영사된 부분만을 추출한다. 그림 8 과 같이 실제로 영사되고 터치되는 부분은 카메라의 모든 영역이 아니고 스크린이 영사되는 일부분의 영역이기 때문에 최외각 포인트 4개를 추출하여 실제 터치가 되지 않는 부분은 삭제하게 된다.

(라) 영상 확대 (양선형 보간법)

영상 확대는 실제 터치되는 부분만을 잘라낸 이미지를 영상처리를 하기위한 영상 사이즈로 복원한다. 영상 사이즈가 너무 작을 경우 터치의 정확도와 인식에 문제점이 나타나기 때문에 영상 데이터의 손실을 최소화 하면서 영상 확대를 위해 양선형 보간법을 사용하여 영상 크기를 조절한다. 양선형 보간법은 영상을 확대할 때 생기는 영상의 손실을 최대한 줄여준다. 양선형 보간법은 새롭게 생성된 화소의 값이 네 개의 가장 가까운 화소들에 가중치를 곱한 값으로 할당하는 방법이며 가중치는 선형적으로 결정되어진다. 가장 인접한 이웃 화소의 보간법보다 영상 데이터의 손실을 최대한 줄여 확대한 영상을 얻을 수 있다. 인접 이웃 화소 보간법으로 적용 시에는 정확한 영상의 복원이 되지 않아 멀티터치를 구현하기 위한 알고리즘 사용에는 불충분하다.

그림 9와 같이 양선형 보간법의 식을 통해서 영상 확대를 구현할 수 있다. 실수 좌표의 픽셀 값을 구한 후

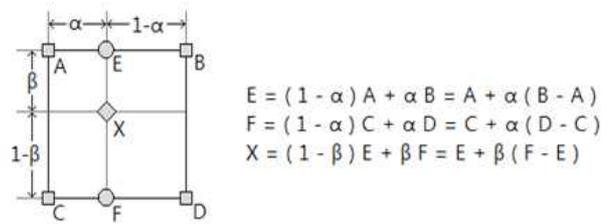


그림 9. 양선형 보간법  
Fig. 9. Bilinear Interpolation.

인접한 픽셀 A, B, C, D, 소수 부분인  $\alpha, \beta$ 를 가지고 계산하면 보간된 좌표를 구할 수 있다.

계산된 좌표를 통해 이미지 크기 변환한다. 그림 10 과 같이 잘라낸 영상을 원본 영상의 해상도로 크기를 조절한다.

(마) 밝기, 명암비 조절

터치된 부분의 외곽선 검출을 명확히 하기 위한 위해 영상을 밝기와 명암비를 조절한다. 명암비를 조절해 터치가 된 부분을 외곽선을 조금 더 명확히 하고 밝기 조절을 통해서 전체 영상의 밝기를 조절하게 된다. 밝기나 명암이 너무 높거나 조건에 맞지 않을 경우 이후 진행될 외곽선 검출에서 손가락 이외의 부분에서 잘못된 외곽선 검출을 하게 된다.

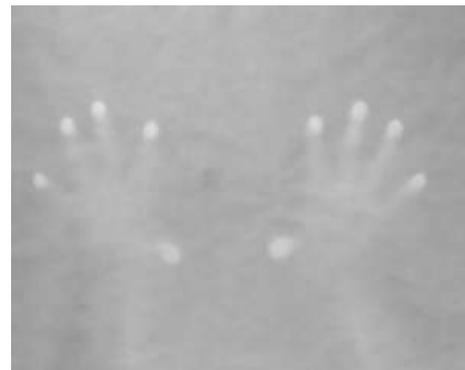


그림 10. 영상 확대  
Fig. 10. Image enlarged.

(바) 외곽선 검출 (Edge Detection)

외곽선 검출은 손가락에서 터치된 부분을 정확히 검출 하기위해 사용된다. DI 방식의 멀티터치스크린 알고리즘을 제안한 NuiGroup의 알고리즘은 손가락 터치가 있어서 많은 문제점을 가지고 있다.<sup>[2]</sup> FTIR 방식과 DI 방식의 동작 방법이 비슷하기 때문에 FTIR의 알고리즘과 DI 방식의 알고리즘이 비슷하다. 기존의 제안된 알

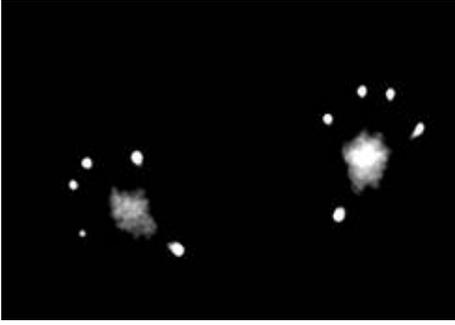


그림 11. NuiGroup DI 알고리즘

Fig. 11. NuiGroup DI Algorithm.

고리즘은 터치된 부분의 밝기의 정보로 터치된 부분이라는 것을 판단하기 때문에 그림 11과 같이 손가락 터치된 점 이외에 손바닥 부분에서 반사되는 빛을 터치 후보가 될 수 있다는 것을 확인할 수 있다. 이 경우 FTIR은 스크린 측면을 통해 적외선 빛을 비추기 때문에 손가락 이외의 부분에서의 터치가 쉽게 일어나지 않지만 DI 방식은 스크린 아래에서 비추기 때문에 그림 11과 같은 문제가 나타난다. 기존의 DI 방식은 반사되는 빛을 이용하여 터치 검출을 위한 기준 값을 적용해 터치 검출을 하기 때문에 다음과 같은 문제점이 나타난다.

- (1) 다중 터치 시에 사람에 따라 손바닥의 빛의 양이 다르기 때문에 손바닥 높이에 따라 터치의 오작동이 매우 크다.
- (2) 손목이나 밝은 색의 옷으로 인한 터치 오작동의 문제가 있다.
- (3) 외부 밝기에 민감하다.

본 논문에서는 문제점을 해결하기 위한 방법으로 외곽선 검출을 사용해 최종 터치 후보인 손가락 부분의 터치만을 검출하였으며 외곽선 검출은 소벨 외곽선 검출 알고리즘(Sobel Edge Detect Algorithm)을 사용한다. 소벨 외곽선(Edge) 알고리즘은 1차 미분연산을 사용하여 외곽선을 검출하며 X, Y 축으로 각각 한 번씩 미분을 하게 된다.

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \times A \quad (1)$$

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times A \quad (2)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (3)$$

소벨 외곽선 검출 연산은 수직방향의 기울기와 수평방향 기울기를 구한 후 기울기의 크기를 구함으로써 얻어질 수 있다. 수식-1, 수식-2에서 A는 소스 이미지를 나타내고 Gx와 Gy는 각각 수직 방향과 수평 방향의 기울기를 나타낸다. Gx와 Gy를 이용하여 에지의 크기인 G를 수식-3과 같이 얻을 수 있다.

식-3을 통해 처리된 외곽선 검출은 그림 12와 같이 처리된다.

밝기의 정보만을 가지고 터치 포인트 후보를 만드는 알고리즘과 달리 외곽선 검출은 밝기에 민감하지 않으며 터치 포인트 검출 할 경우 손바닥이나 손목에서 검출 될 수 있는 문제점을 해결할 수 있고 빛의 영향을 최대한 적게 받을 수 있어 오작동의 문제를 최대한 억제 한다.

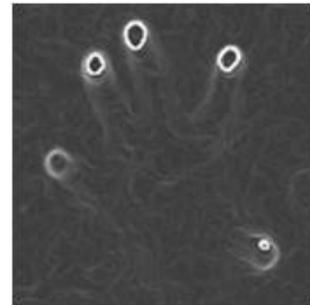


그림 12. 외곽선 검출

Fig. 12. Edge Detection.

(사) 터치 검출을 위한 기준 값 (Threshold)

외곽선 검출된 영상을 통해 손가락 터치 이외에 검출된 부분을 제거하는 부분이다. 터치 검출을 위한 기준 값을 사용해 기준 값 이하인 경우 픽셀 값을 0으로 바꾸고 손가락의 부분만을 검출하게 된다. 적용한 영상은



그림 13. 터치 검출을 위한 기준 값 적용 영상

Fig. 13. Threshold Image.

그림 13과 같다. 적용결과 손가락 터치 부분의 외곽선만 검출되며 손가락이 이어지는 부분에서는 픽셀 값이 비슷하기 때문에 검출이 되지 않는다.

(아) 터치 검출을 위한 터치 포인터 복원

검출된 터치 후보 점들은 터치 포인터 검출을 위해 손가락 터치의 모습으로 복원을 하게 된다. 그림 14와 같이 검출된 외곽선의 데이터를 갖고 손가락 터치의 모습으로 복원한다. 터치 포인트 복원을 통해 정확한 터치 포인터를 추출할 수 있다.

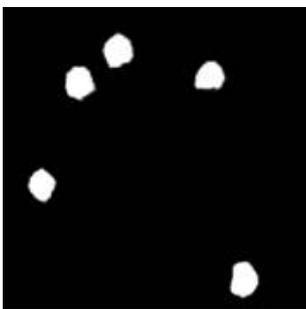


그림 14. 포인터 복원  
Fig. 14. Pointer Restore.

(자) 노이즈 제거 (3x3 Noise Filtering)

최종 포인터 복원이 이루어지면 3x3 마스크를 갖는 미디안 블러(Median Blur)를 이용해 노이즈를 제거한다. 외곽선 검출에서 나올 수 있는 잡음에 대한 노이즈와 복원된 터치 추출 포인트를 매끄럽게(smooth)하게 만드는 역할을 한다.

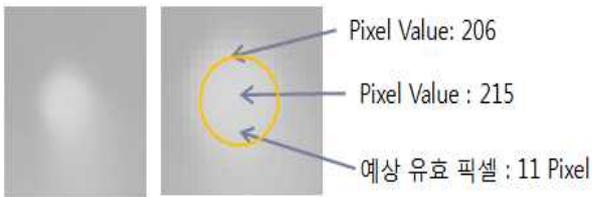


그림 15. 유효 터치 포인트  
Fig. 15. Effective touch points.

(차) 터치 검출

최종 이미지가 만들어지면 터치 후보 점들에 대한 터치유무를 판단하게 된다. 그림 15는 검지의 기준으로 터치를 크기를 측정하였으며 측정한 결과 영상 크기 기준 최저 11 픽셀부터 최대 18픽셀 까지를 실제터치로 인식을 하도록 한다.

터치 추출 알고리즘은 순차적으로 터치된 부분의 영



그림 16. 터치 검출  
Fig. 16. Touch Detection.

상을 검색하여 터치 후보가 터치 조건에 만족할 경우 터치로 인식해 그 때의 좌표(x, y)를 추출 하게 된다. 전체 영상이미지를 통해 터치 포인트를 추출하기 때문에 터치 포인터가 터치 조건에 만족할 경우 터치의 개수에 상관없이 터치 좌표추출이 가능하다. 추출된 좌표는 터치 좌표 보정을 통해 만들어진 수식과 보정 값을 통하여 실제 화면 해상도(1024x768)에 맞게 계산되어 최종 좌표를 통해 터치가 동작하게 된다. 그림 16은 카메라 좌표를 검출한 결과이다.

3. 멀티터치를 위한 보정 알고리즘

정확한 터치 포인트를 검출할 수 있도록 터치 포인터 보정이 필요하며 이는 터치 좌표 보정(Calibration)과 좌표 추적(Touch Tracking)을 통해 가능하다. 좌표 보정은 구역별로 각 포인트의 정보를 가지고 있어 각 구역별로 정확한 동작을 보장하게 되고, 좌표 추적은 멀티터치에 ID를 부여하여 터치가 위치가 변경되거나 원하지 않는 제스처로 인식 하는 것을 방지한다.

(1) 터치 좌표 보정 (Calibration)

터치 좌표 보정은 광학 카메라를 이용하기 때문에 생기는 렌즈 왜곡을 보완하고 터치 포인터의 정확도를 높이기 위해 사용된다. 렌즈의 왜곡을 효과적으로 해결하기 위해 터치 좌표 보정을 통해 그림 17과 같이 1024x768의 해상도에서 각 포인트별로 영역을 나눠 터치 포인트들의 사이의 보정 값을 측정하여 실제 좌표 연산에 사용하게 된다.

각 포인트의 실제 좌표와 영상 좌표가 매칭이 되면 각 좌표 구간 마다 보정계수 연산을 하게 된다.

$$SectionA_x = \frac{X_2 - X_1}{256} \tag{4}$$

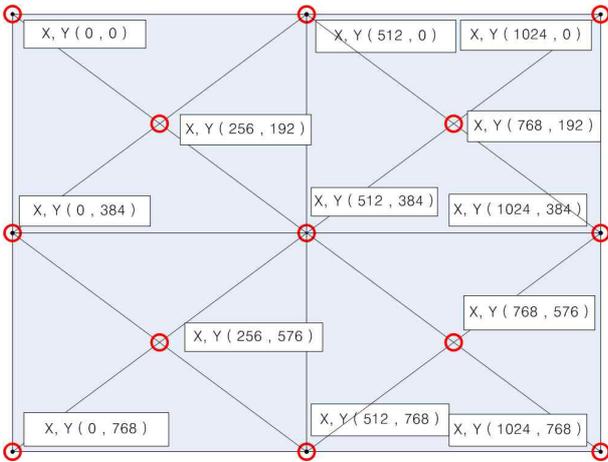


그림 17. 터치 좌표 보정

Fig. 17. Calibration.

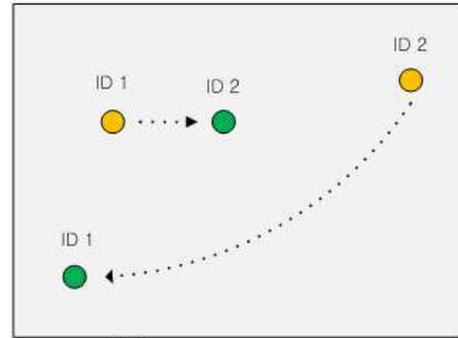
$$SectionA_y = \frac{Y_2 - Y_1}{384} \quad (5)$$

식에서 A는 구간 번호를 타나내며 SectionA일 경우에서의 X축과 Y축의 보정계수는 식-4, 식-5를 통해 계산할 수 있다. 터치 좌표 보정 포인트들의 좌표 사이의 값을 실제 해상도로 나눈 값을 가지고 각 구간마다 보정 계수를 측정할 수 있다. 보정 계수는 실제 좌표의 해상도로 나누어 실제 각 필셀마다 고유의 보정 계수 값을 갖게 된다. 실수형태로 연산되며 나눗셈을 통해 실제 좌표를 추출 할 수 있다.

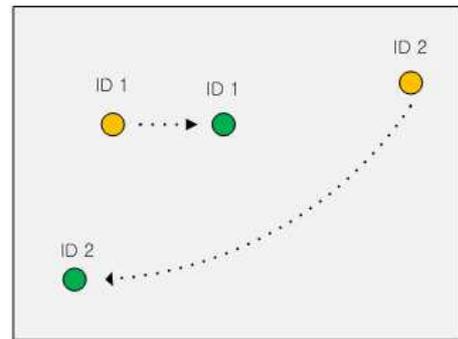
$$Axis_x = \frac{|ImagePoint_x - Calibration_{x0}|}{SectionA_x} + \alpha \quad (6)$$

$$Axis_y = \frac{|ImagePoint_y - Calibration_{y0}|}{SectionA_y} + \beta \quad (7)$$

보정 계수는 터치 좌표 보정이 끝난 후 첫 터치가 입력되기 전 각 구간별로 연산을 하게 된다. 연산이 끝나면 터치가 일어날 때마다 보정계수를 통해 터치 포인트의 실제 좌표를 계산하게 된다. 식-6과 식-7은 실제 좌표 x, y를 구하기 위한 식이다. 실제 좌표 x를 구하기 위해서는 Image Point<sub>x</sub>(카메라 터치 좌표)가 입력이 되면 터치 좌표 보정의 첫 X좌표를 뺀 후 보정 계수로 나누어 주게 된다. 실제 좌표 y를 구하기 위한 방법도 x 좌표를 구하는 방법과 같다. 식-6과 식-7에서의 α, β는 보정계수의 오차가 생기는 부분에 대한 추가적인 보정 부분이다. 이는 실제 터치 시 측정된 오차를 기반으로 정한다.



(a) Touch Tracking Off



(b) Touch Tracking On

그림 18. 터치 추적의 예

Fig. 18. Touch Tracking Example.

(2) 터치 추적 (Touch Tracking)

싱글터치인 경우 영상으로부터 들어오는 신호의 ID는 한 개이기 때문에 터치 포인트를 추적 할 필요가 없다. 하지만 멀티터치의 경우 2개 이상의 터치 포인트가 영상을 통해 입력되고 서로 x, y좌표에 대해 겹치는 부분이 생길 경우 터치 ID가 변하는 경우가 생긴다. 이는 멀티터치 제스처에 있어서 원하는 동작을 하지 못하는



그림 19. DI 방식의 멀티터치스크린

Fig. 19. DI method Multi-Touch Screen.

경우가 생길 수 있어 이러한 문제점을 해결하기 위해 터치 추적 기술이 필요하며 이를 사용해 각각의 터치가 독립적으로 사용 될 수 있도록 한다.

터치 추적 알고리즘은 멀티터치 포인트들의 ID를 부여하여 동작을 추적하게 된다. 실제 터치 영역에서 동일한 x 좌표를 향해 이동할 경우 x 좌표를 서로 지나쳐가면서 x좌표에 대해 서로 위치가 바뀌는 경우가 생기게 되는데 이를 전 x좌표와 y좌표의 값을 추적하여 멀티터치 포인트들의 ID를 유지할 수 있도록 한다. 그림 18과 같이 터치 추적을 사용하기 전에는 터치의 이동시 터치 ID가 변경되는 경우가 생긴다. 이러한 문제는 터치 제스처를 감지하고 동작하는데 있어서 오동작의 원인이 된다. 따라서 각 터치 포인트를 추적하는 터치 추적 알고리즘을 사용하여 각 터치 포인트가 고유의 ID를 잃어버리지 않고 동작할 수 있도록 한다.

### III. 성능 분석 및 결과

알고리즘을 검증 및 분석을 하기 위해 그림 19와 같이 DI 방식 멀티터치 테이블을 구성하여 멀티터치의 성능을 확인하였다. 멀티터치스크린의 성능은 멀티터치의 정확한 동작, 좌표의 오차, 동작속도로 확인할 수 있다. 멀티터치의 정확한 동작은 전체 스크린에 대해서 터치 인식이 정확히 되는지 실제 터치를 통해 확인할 수 있으며 좌표의 오차 성능은 터치 좌표 보정을 진행한 후 검출되는 좌표를 좌표 오차 검증 프로그램을 통해 오차를 측정하였다. 멀티터치 검증 동작속도는 알고리즘 별로 수행시간을 측정하였다.

스크린 크기는 적외선 카메라의 시야각을 고려하여 30인치 정도의 크기를 갖는 스크린으로 구성하였으며 화면 해상도는 1024x768을 갖는 DLP방식의 프로젝터를 사용하여 영사하였다. 적외선 카메라는 일반 카메라에 가시광선 차단 필터를 사용하여 적외선 카메라를 구성하고 터치 성능을 최대한 높이기 위해 적외선 발광 모듈을 4개를 사용하였으며, 소프트웨어 검증을 위해 Athlon X2 2.16GHz 기반의 PC에서 Visual Studio 2008로 작성된 프로그램을 통해 검증하였다.

#### 1. 소프트웨어 검증

DI 방식의 멀티터치 스크린을 구현하기 위해 소프트웨어 환경을 그림 21과 같이 구성하였다. 적외선 카메라의 아날로그 데이터를 PC에서 입력받기 위해 RCA

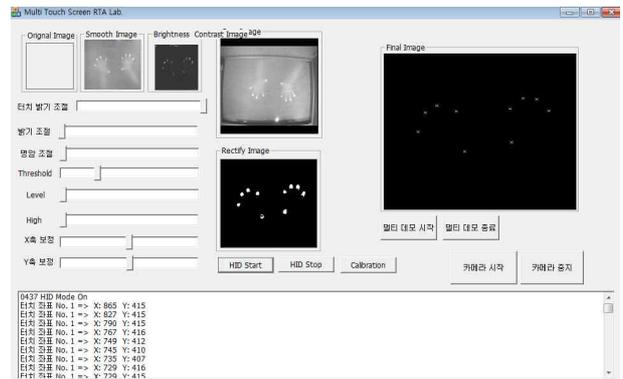


그림 20. 멀티터치 검증 소프트웨어  
Fig. 20. Multi-Touch Software.

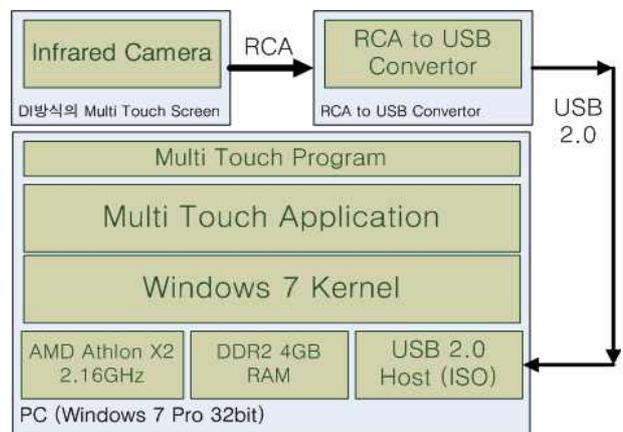


그림 21. 소프트웨어 검증 환경  
Fig. 21. Software Implementation.

to USB Converter를 사용해 USB 2.0으로 PC에 적외선 카메라를 연결하고 연결된 카메라는 720 x 576 사이즈의 영상을 사용하였다. 그림 20의 멀티터치 검증 프로그램은 Visual Studio 2008을 기반으로 USB로부터 들어오는 카메라 영상을 입력을 받아 그림 4와 같이 알고리즘을 통해 멀티터치 포인트를 추출하고 그림 16과 같이 터치 포인트의 제한이 없이 터치 포인트 추출이 가능한 것을 확인할 수 있다.

터치 포인트의 오차를 확인하기 위해 16구간을 나누어 터치 포인트의 오차를 확인 하였다. 멀티터치의 정확도를 높이기 위해서 카메라의 위치에 따른 터치 좌표 보정이 중요하기 때문에 터치 좌표 보정을 통해 1차 보정연산이 끝난 후 터치 포인트 추출 알고리즘을 사용하여 터치를 동작시키고 각 터치 포인트마다 오차를 측정 한 결과를 표1에 정리하였다. 표1에서 볼 수 있듯이 렌즈 왜곡에 의한 오차를 확인 할 수 있었다. 총 10회의 터치를 통해 평균오차를 측정하였으며 터치 오차 위치는 각 구간별 꼭짓점을 기준으로 하였다. x좌표는 오른

표 1. 터치 포인트 구간별 오차

Fig. 1. Touch Point error value in section.

x : 0	x : 9	x : -46	x : -54
y : 9	y : 8	y : 10	y : 12
x : 15	x : 12	x : -43	x : -54
y : 4	y : 7	y : 15	y : 17
x : 10	x : 15	x : -44	x : -48
y : 5	y : 7	y : 22	y : +22
x : 3	x : 1	x : -3	x : -8
y : 10	y : 14	y : -2	y : 15

표 2. 터치 포인터 인식 범위

Fig. 2. Touch pointer detect range.

	Range	인식률
Min	11 Pixel 이상	100%
Max	18 Pixel 이하	100%

쪽 구간으로 갈수록 오차가 심해 졌으며 y좌표는 아래의 구간으로 갈수록 오차가 생기는 것을 확인 할 수 있다. 이는 전체 렌즈 시야각에서 실제 터치가 일어나는 부분만을 검출하여 사용했기 때문에 전체 광학 렌즈의 일부구간에서 나타나는 왜곡과 같이 나타난다. 이를 보정하기 위해 보정 값을 정하여 각 구간을 보정하였으며 보정결과 전체 평균 오차 값이 5픽셀 이하로 감소하는 것을 확인할 수 있었다. 추가 보정 값은 식-6, 식-7에 추가적용 하였다. 추가 보정 값이 필요한 이유는 렌즈 왜곡률이 카메라 영상에 터치 영역의 위치에 따라서 다르기 때문에 추가 보정 값을 통해 정확한 터치 포인터를 맞춰줘야 한다. 보정 된 좌표를 검출한 후 실제 해상도인 1024x768사이즈로 좌표연산을 하게 된다.

본 논문에서 제안한 멀티터치 알고리즘은 터치 개수에 상관없이 동일한 동작 속도를 갖는다. 터치 포인터 검출 알고리즘은 한 프레임 마다 표 2와 같은 기준에 만족하는 터치의 개수를 찾게 되는데 찾아진 터치 포인터 후보들은 각각의 ID를 부여받게 되고 터치 추적을 통해 고유의 동작을 하게 된다. 터치 포인터의 기준에 맞을 경우 표2에서와 같이 터치 인식은 전 구간에서 100%의 동작하는 것을 확인할 수 있다.

전체 적인 동작 시간을 알아보기 위해 본 논문에서 제한한 DI 방식 알고리즘의 영상처리의 수행시간을 측정하였다. 표 3과 같이 전체 수행 시간은 총 307ms, 약 3.25fps의 동작 시간을 갖는다. 각 알고리즘의 수행시간에서 가장 오랜 시간이 걸리는 부분은 노이즈 필터 부분이다. 이는 전체 픽셀에 대해 정렬 연산을 하기 때문

표 3. 전체 영상처리 동작시간

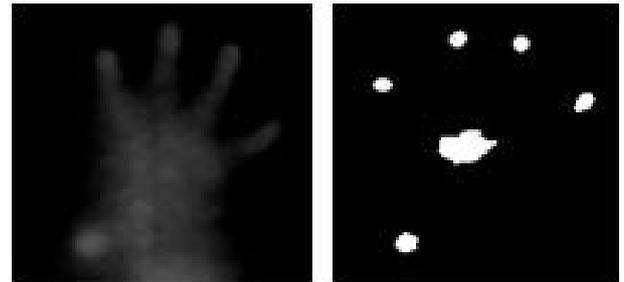
Table 3. All image processing work time.

알고리즘	수행시간
RGB to Gray	14ms
Noise Filter	208ms
5x5 Median Blur	
Image Cut & Resize	23ms
Brightness & Contrast	10ms
Edge Detect	26ms
Threshold	5ms
Touch Point Restoration & Noise Filter	15ms
Touch Detect	6ms
<b>영상처리 최종시간</b>	<b>307ms</b>

표 4. 터치스크린 비교

Table 4. Touch-Screen Comparison.

	적외선 어레이	적외선 라인스캔 <sup>[2]</sup>	DI 방식
터치 개수	1 ~ 2개	1 ~ 2 개	No Limit
대형화	Bad	Good	Excellent
외부영향	Good	Bad	Excellent
구현방법	Difficult	Easy	Easy
소비전력	Bad	Excellent	Good



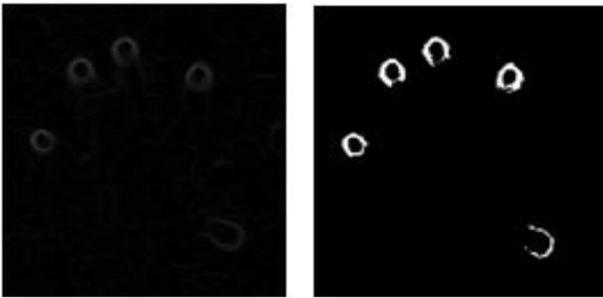
(a) (b)

그림 22 이전 DI 방식의 알고리즘<sup>[4]</sup>

Fig. 22. Before DI method algorithm.

에 시간이 가장 오래 걸리는 부분이다.

기존에 제안된 DI 방식의 알고리즘은 영상의 밝기와 명암비, 외부의 빛에 매우 민감하게 반응하였다. 영상의 밝기가 상대적으로 높을 경우 터치의 감도는 높아지지만 손바닥이나 손목, 외부에서 들어오는 빛들에 의해 오작동이 있었다. 그림 22-(a)<sup>[4]</sup>와 같이 손가락이외의 부분에서도 반사되는 빛이 손 전체에 있는 것을 확인할 수 있다. 이는 정밀한 터치 검출을 위한 기준 값을 적용한다 하더라도 그림 22-(b)<sup>[4]</sup>의 결과처럼 원하지 않는 부분의 터치가 생기게 된다. 하지만 본 논문에서 제안한 DI 방식의 멀티터치 알고리즘은 그림 23과 같이



(a) (b)

그림 23. 최적화된 DI 방식의 알고리즘  
Fig. 23. The DI method is optimized algorithm.

기존 방식의 문제였던 터치 오작동의 원인을 효과적으로 해결하였으며 외부 빛에 민감하게 반응하지 않는 특성을 갖는 멀티터치스크린을 설계하였다.

### V. 결 론

기존의 DI 방식의 멀티터치스크린은 설치가 간단하며 멀티터치 포인터 개수에 제한이 없는 특징을 가지고 있다. 하지만 내부의 적외선 빛과 외부의 빛에 민감하여 오동작이 많다. 실제로 DI 방식의 대형 터치스크린보다 DI 방식과 유사한 FTIR 방식이 주로 사용되고 있다. 하지만 FTIR 방식은 스크린이 커질수록 스크린을 감싸는 적외선 LED의 개수가 많아져 전력소모가 크게 증가하고 중앙 부분에서 적외선 감쇄연상도 나타나는 단점이 있다.

기존의 DI 방식의 영상처리 알고리즘은 터치 시 손에서 반사되는 빛을 이용해 터치포인트를 추출하기 때문에 손에서 반사되는 빛은 사람의 손의 밝기의 차이, 외부의 환경의 밝기, 옷의 색까지 많은 부분에서 영향을 받는다. 본 논문에서는 오작동의 영향을 받지 않는 DI 방식의 대형멀티터치 스크린을 구현을 위해 정확한 동작과 기존의 문제점을 해결한 알고리즘을 통해 효과적으로 해결하였다.

제시한 알고리즘은 밝기의 정보만을 보고 판단하지 않고 외곽선을 통해 전체 손가락의 형태를 추출하고 이를 터치를 인식할 수 있도록 터치 후보를 복원하는 방법, 터치 검출 알고리즘, 터치 추적 등을 통해 기존의 DI 방식의 문제점들을 해결하였다. NuiGroup에서 제안한 DI 알고리즘 그림 22<sup>[4]</sup>와 Microsoft Surface<sup>[1]</sup>에서 제안한 DI 알고리즘 그림 24<sup>[1]</sup>는 복잡한 오브젝트 인식 부분이 필요하지만 본 논문에서 제시한 알고리즘은 외

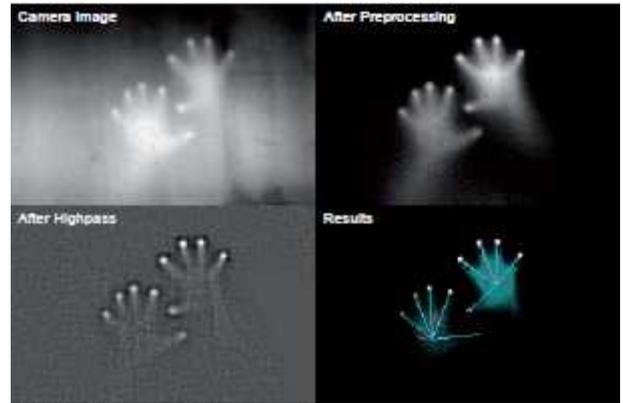


그림 24. DI 트래킹 파이프라인 샘플 이미지  
Fig. 24. Sample Images of the DI Tracking Pipeline.



그림 25. 터치 어플리케이션  
Fig. 25. Touch Application.

곽선의 검출을 통해 손가락을 인식 하기 때문에 알고리즘이 비교적 간단하며, 오브젝트의 밝기에 민감하지 않고 사람 손의 밝기 차이가 있더라도 정확한 손가락 검출이 가능하다.

본 논문에서 제안한 DI 방식의 알고리즘은 멀티터치 스크린으로써의 오브젝트의 검출, 멀티터치 포인트 개수, 터치 정확도에서 만족한 성능을 보였다. 기존의 다른 방식들의 멀티터치에 비교하여 멀티터치의 인식률이 매우 뛰어나며 터치 개수가 많아도 일정한 성능을 확인할 수 있었다. 하지만 멀티터치 포인트를 검출하기 위한 속도는 소프트웨어 검증에서 부족함을 보였으며 대형 터치스크린으로 사용되기에는 연구에 사용된 적외선 카메라의 렌즈 시야각 제한과 프로젝터 렌즈의 시야각의 제한으로 스크린 크기에 한계가 있었다. 향후 하드웨어로 구현이 되고 단초점 렌즈를 사용한다면 HID 규격에 만족하는 고속의 대형 멀티터치스크린을 쉽게 구현할 수 있을 것으로 판단된다.

### 참 고 문 헌

[1] Johannes Schoning, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook,

- Markus Lchtefeld, Nima Motamedi, Laurence Muller, Patrick Olivier, Tim Roth, Ulrich von Zadow. Multi Touch Surfaces : A Technical Guide, Technical Report TUM-I0833
- [2] 정성완, 이현오, 정용진, “광학 영상 기법을 이용한 멀티 터치 스크린 컨트롤러 개발”, 전자공학회 학술대회, 344-347쪽, 전북대학교, 대한민국, 2009년 5월.
- [3] 정성완, 권오준, 정용진, “스테레오 비전을 이용한 저전력 멀티 터치스크린 컨트롤러의 설계”, 전자공학회 논문지, 제 47권 SD편, 제 2호, 68-76쪽, 2010년 2월.
- [4] Nuigroup. TouchLib  
<http://nuigroup.com/touchlib>. (온라인 문헌 참고 2010. 5.2)
- [5] Jorg Edelman, Andreas Schilling, THE DABR - A MULTITOUCH SYSTEM FOR INTUITIVE 3D SCENE NAVIGATION, IEEE, 3, 2010.03
- [6] 김송국(Song-Gook Kim), 이철우(Chil-Woo Lee) 멀티터치를 위한 테이블-탑 디스플레이 기술 동향, 한국콘텐츠학회, 84-91, 2009.02
- [7] A.D Wilson, “TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction,” In Proceedings of the 6th International Conference on Multimodal Interface, pp.66-76, 2004
- [8] A. Wilson, “PlayAnywhere : A Compact Interactive Tabletop Projection-Vision System”, In Proceedings of User Interface Software and Technology, pp.83-92, 2005.

---

 저 자 소 개
 

---



강민구(학생회원)  
 2009년 건양대학교 전자정보공학과 학사 졸업.  
 2011년 광운대학교 임베디드 소프트웨어공학과 석사 졸업 예정.

<주관심분야 : 임베디드 시스템, 신호처리, 영상처리, SoC 설계>



정용진(정회원)  
 1983년 서울대학교 제어계측공학과 학사  
 1983년~1989년 한국전자통신연구원  
 1995년 미국 UMASS 전자전산공학과 석사/박사

1995년 삼성전자 반도체 수석연구원  
 1999년~현재 광운대학교 전자통신공학과 교수  
 <주관심분야 : 무선통신, 정보보호, SoC 설계, 영상처리 및 인식, 임베디드 시스템>