

http://dx.doi.org/10.7236/JIWIT.2012.12.2.85

JIWIT 2012-2-11

# FTIR 멀티터치 테이블에서 효율적인 사용자 환경 개발

## The Design of Efficient User Environment on the FTIR Multi Touch

박상봉\*, 안중서\*\*

Sang-Bong Park, Jung-Seo Ahn

**요약** 본 논문에서는 FTIR 멀티터치 테이블에서 손가락을 이용하여 기존에 윈도우에서 제공되는 제스처 외에 간편한 방법으로 화면 제어를 할 수 있는 제스처를 개발하였다. 또한, 손가락을 사용하지 않고 적외선 카메라 제어를 통해 모바일 기기를 올려놓아도 인식할 수 있도록 구현하였다. HID 인터페이스가 제공되지 않는 FTIR 멀티터치 테이블의 환경 때문에 기존 블루투스는 연결이 어려웠으나, 본 논문에서 제안한 모바일 기기를 이용한 데이터 전송 방법은 인터페이스의 불편함을 해소하고 데이터 전송을 효과적으로 진행 할 수 있도록 구현하였다. 기존 블루투스 연결은 복잡한 페어링 과정을 지니지만, 본 연구에서는 페어링 과정을 단순화하고, 사용자가 친화적 제스처를 개발하여 특정 기기 연결을 가능토록 구현하였다. 또한, 자체 제작한 FTIR 멀티터치 테이블에서 테스트를 통해 검증하였다.

**Abstract** In this paper, we develop the new gestures of screen control with fingers on the FTIR multi touch table. It also describes recognition of mobile devices on the table using infrared camera. The FTIR multi touch table was inconvenient to the existing Bluetooth connection, because it is not provided with an HID(Human Input Device) interface. The proposed data transmission method using mobile device is to relieve the inconvenience of the data transfer and proceed more effectively. The new gestures and data transmission method is verified by FTIR multi touch table that is implemented for testing.

**Key Words :** FTIR, Multi Touch Table, Bluetooth, Gesture, Mobile devices

### 1. 서론

최근 사회에서 우리는 주변의 많은 터치스크린을 사용하는 매체들에 둘러 싸여있다. 현재 많이 사용하고 있는 스마트폰부터 의료 과학 장비, MP3, 모니터, 텔레비전, 디지털카메라 등은 우리 일상생활에서 많은 양을 차지하고 있다. 터치스크린은 지금도 많은 개발과정을 거치고 있는데 대표적으로 터치스크린을 이용한 테이블이 있다. 멀티터치 테이블은 가정용, 회의용, 전시용 등 여러

가지 방면에서 개발되어지고 있으며 계속적으로 새로운 방식의 제품이 개발되는 중이다. 이 중에서 테이블 방식 제작에 가장 적합하다고 손꼽히고 있는 방식이 FTIR (Frustrated Touch Internal Reflection) 방식의 터치스크린이다. FTIR 방식은 스크린의 상, 하 및 좌, 우 4방면에서 IR LED를 비추게 되고 이 적외선은 상단과 하단의 반사체를 통해 스크린 안에 머물게 된다.

본 논문은 FTIR 멀티터치 테이블에서 손동작 분석과 설계 구조를 고려하여, 멀티 터치 플랫폼에 응용할 수 있

\*정회원, 세명대학교 정보통신학부

\*\*준회원, ㈜다우엑셀리콘

접수일자 2012년 3월 2일, 수정완료 2012년 4월 3일

게재확정일자 2012년 4월 13일

Received: 2 March 2012 / Revised: 3 April 2012 /

Accepted: 13 April 2012

\*\*Corresponding Author: psbcom@semyung.ac.kr

Dept. Information&Communication, Semyung University, Korea

도록 효율적인 제스처를 설계하여 사용자가 손쉽게 화면을 제어할 수 있도록 하고, 모바일 기기를 연결하여 파일 전송 및 컨트롤 하는 방법을 연구하였다. 모바일 기기의 연결은 블루투스 연결을 위해 적외선 카메라를 통해 모바일 기기가 터치스크린 위에 올라왔을 때의 제스처를 개발하고, 페어링 과정을 간소화 시킨 후 자체 제작한 FTIR 멀티터치 테이블에서 테스트를 통해 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 FTIR 멀티터치 테이블의 구조 및 구현 방법에 대해 설명하고, 3장에서는 제안한 제스처 개발 및 모바일 기기를 이용한 파일 전송 과정에 대해 서술한다. 4장에서는 제안한 방법의 실험 및 결과에 대해 설명하고, 5장에서는 결론을 맺는다.<sup>[1, 2]</sup>

## II. FTIR 멀티터치 테이블의 구조 및 구현

### 1. 멀티터치 테이블 구조

FTIR 멀티터치 테이블은 다음과 같은 구조로 설계하였다. 먼저 적외선 LED가 아크릴의 4면에서 적외선을 비추게 한다. 이때 적외선 빛이 새거나 노이즈가 생기는 것을 방지하기 위하여 C형 알루미늄 프레임으로 LED를 보호한다. 적외선 빛은 실리콘 막에 의해서 아크릴을 빠져나가지 못하고 굴절현상을 보이며 난반사가 일어나게 된다. 이때 사람의 손가락 등, 적외선을 반사시킬 수 있는 물체가 트래킹지 표면에 닿게 되면 난반사 하던 적외선이 정반사로 반사시키게 된다. 적외선 카메라는 하단부에서 정반사된 적외선을 촬영하여 터치의 유무와 좌표를 판독하게 된다. 사람의 손가락은 적외선을 반사시키는 반면, 모바일 기기의 일반적인 표면은 적외선을 반사시킬 수 없으므로, 실리콘 케이스를 이용하여 테스트를 진행하였다.

### 2. 적외선 카메라 제어

적외선 카메라는 FTIR 멀티터치 테이블의 하단부에서 반사되는 빛을 촬영한다. 먼저 적외선 카메라를 제어하기 위해 Open CV 라이브러리를 이용하여 적외선 카메라를 제어할 수 있도록 카메라 셋업을 하고 원본 영상을 흑백영상, 이진영상, 침식영상 및 결과영상 4단계로 구분한다. 흑백영상은 적외선 카메라에 들어오는 적은 양의

자외선을 영상에서 제거하는 역할을 한다. 이진영상은 각각의 영상 블록을 모두 0과 1의 값으로 바꿔서 디지털 값으로 변환한다. 침식영상은 가로 세로의 넓이를 측정하여 터치 지점의 중심지점을 찾아내고 마지막으로 결과영상에서 운영체제로 보내기 전 점검을 한다. 찾아낸 터치 지점을 모니터 화면 해상도의 설정치와 비교하여 실제 터치 지점을 X축, Y축 좌표로 찾아내서 운영체제로 전송한다.

### 3. 멀티터치 테이블 전송 절차

본 논문에서 제안된 FTIR 멀티터치 테이블의 모바일 제스처는 리플렉터를 통해 적외선이 반사되는 값을 활용하도록 설계되었다. 개발한 프로그램의 구조는 그림 1과 같다. 여기서 임계점은 최종 결과 화면에서 인식한 범위의 값이 모바일 기기로 판단되는 수치의 넓이일 경우를 판단하여 제스처로 처리할 것인지 데이터 전송 UI를 실행할 것인지에 대한 판단을 하게 된다. 만약 임계점 이상의 값일 경우 데이터 전송 UI를 실행하고 임계점 이하의 값이거나 이미 데이터 전송 UI가 실행되어 있을 경우 마우스 데이터나 제스처 데이터로 사용한다.

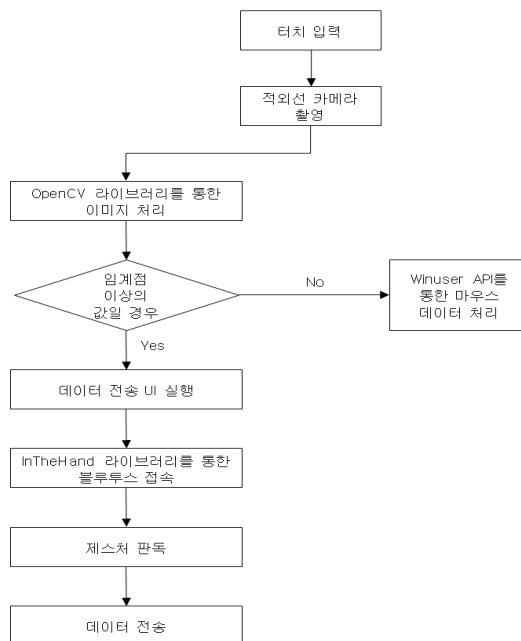


그림 1. FTIR 멀티터치 테이블에서의 데이터 전송 구조  
Fig. 1. Data Structure of FTIR Multi Touch Table

### III. 제스처 개발 및 모바일 기기를 이용한 파일 전송 구현

#### 1. 손가락을 이용한 제스처 개발

본 논문에서는 멀티터치 테이블 환경에서 효율적으로 사용 가능한 제스처를 제안한다. 멀티터치 테이블은 상하좌우 화면의 구분이 사용자가 바라보는 각도에 따라 달라지는 특성과 함께, HID 인터페이스를 제공하지 않아 모든 동작을 터치 입력에만 의존해야 한다는 단점이 있다. 따라서 본 논문에서는 멀티터치 테이블 환경에서 유용하게 사용 가능한 제스처를 개발하였다.

첫 번째로 바탕화면 보기 기능 실행이다. 기존 터치 테이블에서는 바탕화면 보기 기능을 이용하기 위해서는 실행한 모든 윈도우 창의 최소화 버튼을 일일이 손가락으로 눌러주어야 하는 단점이 있었다. 하지만 제안한 제스처를 사용할 경우 단 한 번의 제스처로 바탕화면보기를 실행 할 수 있다.

두 번째는 에어로(Aero)모드 실행이다. HID 인터페이스가 제공되지 않는 멀티터치 테이블은 윈도우 창 하나를 띄워놓게 되면 다른 실행 창을 보기 힘든 점이 있었다. Alt+Tab 기능을 활용할 수 없다는 점 때문에 다수 프로그램을 실행시키기 힘들데 제안한 제스처를 사용할 경우 편리하게 에어로 모드를 통해 실행중인 프로그램을 손쉽게 확인 할 수 있다. 그림 2는 본 논문에서 제안한 제스처의 구동 구조이다. 먼저 멀티터치 테이블 스크린 위에 터치 입력이 들어 오게 되면 적외선 카메라가 이를 촬영한다. 촬영된 데이터를 노이즈 제거 작업을 위한 영상처리 4단계를 거치고 나서 좌표 값을 산출한다. 산출된 좌표 값을 토대로 ID, 터치 좌표, 이동거리의 데이터를 얻어 어떠한 제스처를 실행할지에 대해 판단한 후 제스처를 실행한다.

#### 가. 각 블록별 설명

##### (1) 터치 입력 데이터 처리

터치 입력을 판단하고 좌표를 찾기 위해서는 적외선 카메라의 세팅과 영상 처리가 선행되어야 한다.

본 논문에서는 다중 터치가 들어왔을 경우를 고려하여 구조체와 콜백 함수를 통해 터치입력을 관리하고 dwID로 터치의 데이터를 각각 구분하는 형태로 구성했다. 터치 영상을 확인한 뒤 흑백화, 이진화, 침식화 및 결과영상으로 이어지는 4단계의 영상 처리를 진행하여 잡

음을 제거한다. 이후 결과영상을 토대로 좌표 계산을 하고 이 좌표 값을 구조체에 할당한다. 작업이 모두 끝나면 콜백 함수를 호출하여서 지속적으로 터치 영상 값을 실시간으로 받아들인다. 구조체는 지속적으로 호출되는 콜백 함수에 의해 갱신된 좌표 값과 ID를 가지며 터치 테이블을 보관하게 된다.

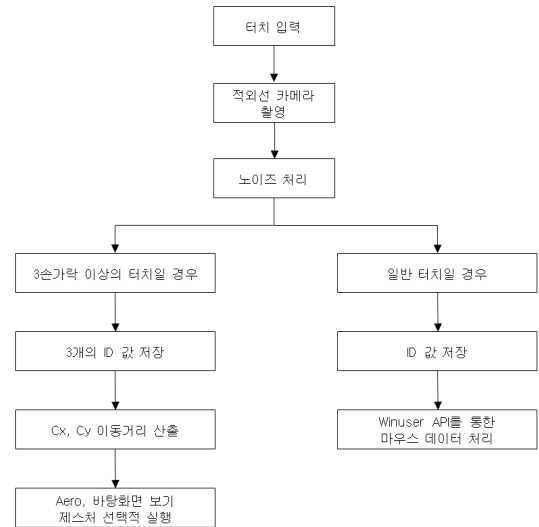


그림 2. 제안한 제스처 구조  
Fig. 2. The proposed gesture structure

##### (2) 좌표 추출

좌표 추출은 촬영된 최종 영상의 적외선 영역을 가로, 세로의 가상의 사각형을 그린 뒤, 정 중앙의 점을 반환하도록 구현하였다. 본 논문에서는 tracking Point 라는 임의 함수를 구현하여 진행하였다. tracking Point 함수는 이미지를 입력 받는 것으로부터 시작된다. 이때 이미지의 수를 카운트하며, 영역의 넓이도 체크한다. 이후 CvMem Storage에 영역을 저장한다. cvFind Contours에서 영역을 찾는데, 영역의 넓이와 라인의 경계를 찾는다. 각각의 영역 넓이는 po[countarea].x 와 .y에 저장하며 checkmax 값과 비교하여 중심좌표를 계산하고 좌표를 검출한다.

##### (3) 적외선 캠 영상 처리

적외선 캠으로부터 촬영된 영상은 총 4단계의 보정작업을 거친다. 첫 번째로 이미지를 흑백영상화 하여 터치 입력을 보다 뚜렷하게 만든다. 두 번째로 흑백영상 이미지는 다시 이진영상으로 보정하여 1과 0의 영상으로 바

꾸고 세 번째로 침식영상으로 보정작업을 한다. 침식영상은 노이즈 값을 모두 깎아 내림으로써 보다 정확한 터치 입력을 판독하게 된다. 마지막으로 결과 영상에는 터치 입력이 확인 된 부분에 사각형의 가상 테두리를 만들어서 입력의 중심점을 찾는다. 이때 다소 작은 입력 크기가 발생 할 경우 노이즈로 처리하여 터치 신호를 무시하고 가장 큰 신호부터 터치 ID를 배정한다. ID가 배정되면 미리 선언한 구조체 안에 현재의 좌표와 입력 넓이 등을 저장한다.<sup>[3]</sup>

(4) 마우스 데이터 처리

영상처리가 끝나면 구조체에 좌표와 dwID 등 필요한 정보를 입력한다. 이 정보들을 SetCursor Pos를 통해 WinuserAPI에서 제공하는 마우스 제어 부분으로 보내어 제어한다. 이때 마우스의 좌표는 해상도와 비교하여 스크린상의 실제 터치 지점과 비교하여 추출한다.

제스처 구현을 위한 손가락의 이동 거리에 대한 측정 은 구조체 안에서 cxContact, cyContact 에 저장하여 측정한다. 일반적으로 OS에서 제스처를 처리하기 위해 WM\_Touch와 WM\_Gesture 를 이용하여 처리하지만 본 논문에서는 cxContact와 cyContact를 이용하여 이동거리를 측정하고, 이 데이터를 통해 제스처를 직접 실행시키는 방식으로 구현하였다.

나. 제스처 실행

영상에서 만약 터치 입력이 3개로 이루어졌다면, dwID 와 처음 시작된 좌표 값을 저장한다. 이후 3개의 dwID 가 소멸되는 것을 확인하는 것과 동시에 끝난 좌표를 저장하여 시작좌표 - 종료좌표 값을 계산한다. 이 값이 만약 x축에 한하여 기준치를 넘어서게 되면 에어로 모드를 실행, y축 좌표에 한하여 기준치를 넘어서게 되면 모든 창 최소화 모드로 실행한다.

그림 3은 제안한 제스처를 실행 할 때 손의 모양이며, 그림 4는 실행을 위한 일부 소스 코드이다. 그림 3의 첫 번째 그림은 3손가락으로 왼쪽에서 오른쪽으로 슬라이드 시키면 Aero모드로 전환되는 제스처를 실행한다. 현재 작업중인 여러 창들 중에 손쉽게 원하는 창으로 이동이 가능하다. 오른쪽 그림과 같이 3개의 손가락을 위에서 아래로 드래그하면, 다중의 활성화된 윈도우 창이 모두 최소화 된다. 이것은 우리가 프로그램을 여러개 실행하였을 때, 일일이 마우스로 하나씩 최소화를 시키지 않아도, 모

두 한꺼번에 처리가 가능하다. 프로그램이 종료된 것은 아니며, 작업 표시줄에서 해당 프로그램들을 확인할 수 있다.

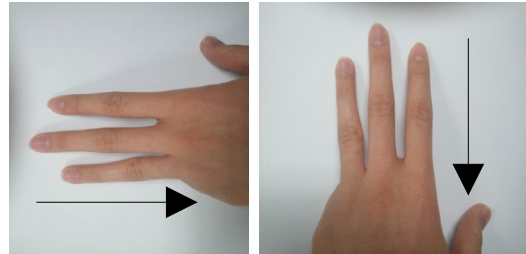


그림 3. 제스처 동작 방법  
Fig. 3. The gesture action method

```
void trackingPoint(IplImage* img)
{
    countarea = 0;
    double area = 0;

    CvMemStorage* contour_storage=cvvCreateMemStorage(0);
    CvSeq* contour =0;

    cvFindContours(img, contour_storage, &contour, sizeof(CvContour),
        CV_RETR_CCOMP, CV_CHAIN_APPROX_SIMPLE);
    for(maxArea = 0; contour !=0; contour = contour->h_next) {
        area = cvContourArea(contour, CV_WHOLE_SEQ);
        CvRect rect = cvContourBoundingRect(contour);
        if(area > 25)
        {
            po[countarea].x = (rect.x + (rect.width/2));
            po[countarea].y = (rect.y + (rect.height/2));
            if(area>maxArea) {
                maxArea = area;
                maxvalue = countarea;
            }
        }
    }
}
```

```
if(touchToggle == 0)
{
    touchToggle = 1;
    for(int i=0; i<4; i++)
        stop[i] = po[i];
}
if(checkmax)
{
    SetCursorPos(1279-(po[0].x*x),767-(po[0].y+y));
    for(int i=0; i<4; i++)
        ltpo[i] = po[i];
    for(int i=0; i<4; i++) {
        if(stpo[i].y - ltpo[i].y >=30)
            WinExec("C:\\\\User\\Default\\AppData\\Roaming\\Microsoft\\Internet Explorer\\Quick Launch\\Shows Desktop",
                SW_SHOWNORMAL);
        if(stpo[i].x - ltpo[i].x >=30)
            WinExec("C:\\\\User\\Default\\AppData\\Roaming\\Microsoft\\Internet Explorer\\Quick Launch\\Windows Switcher",
                SW_SHOWNORMAL);
    }
}
else
{
    SetCurSorPos(1279-(po[maxvalue].x*x), 767-(po[maxvalue].y+y));
    tepo = po[maxvalue];
}
}
```

그림 4. 구현된 일부 소스 코드  
Fig. 4. The implemented source code

먼저, 터치테이블에서 터치를 인식하면, 각 좌표값이 계산된다. 그 다음에는 처음 터치를 인식하였을 때, 좌표값과, 마지막 인식된 좌표값을 비교한다. 두 개의 좌표값의 차이가 임계치 이상 나오면, 특정 제스처를 수행한다. 여기서 사용된 임계치 값은 30을 사용하였다. 일반적인 모니터 해상도가 1280 \* 768를 기반으로 구현하였는데, 터치 테이블은 상당히 크기 때문에 30이상의 값만 들어오면 모션이 인식된다. 우리가 사용하는 기본 모니터는 세로가 768개의 픽셀로 되어 있을 때, 약 200정도의 모션 길이가 인식되면 제스처가 실행된다. 제스처가 수행되면, 윈도우의 구동 명령어와 서로 연결을 시킨다. 그리고 최종적으로 결과값을 확인한다.

## 2. 모바일 기기로의 제스처 구현 및 파일 전송

FTIR 멀티 터치 테이블에서 먼저 모바일 기기를 인식하고 파일 전송을 하기 위해서는 특정 제스처를 구현해야 한다. 테이블 위에 모바일 기기를 올려놓으면 테이블에 모바일 기기 아이콘이 생성되고, 테이블에서 터치로 특정 파일을 선택하여 생성된 아이콘으로 가져다 놓으면 블루투스가 연결되어 모바일 기기으로 파일이 전송되도록 구현하였다.

### 가. 데이터 전송 UI

적외선 카메라로부터 판단된 결과 값에 의해 실행된 UI는 블루투스를 연결하고 데이터를 전송하기 위한 목적으로 설계하였다. 기본적인 UI 구성은 연결된 모바일 디바이스의 이름과 아이콘을 화면에 출력하기 위한 프레임과 사용자가 전송할 수 있는 데이터를 테이블 형태로 구성하여 올려놓은 프레임, 전송 가능한 파일을 자유롭게 이동시키고 회전시킬 수 있는 아이콘으로 구성되어 있다. 화면에서 사용자가 전송 가능한 오브젝트는 vcf로 이루어진 명함 파일로 모바일 친화적 전송이 가능하며 별도의 저장작업을 거치지 않더라도 명함파일을 전송하게 되면 수신한 모바일 기기에 자동으로 저장되는 구조로 이루어져 있다.<sup>[4]</sup> 다른 전송 가능한 파일로는 이미지 파일이 있다. 이미지 파일은 사용자가 선택한 사진을 자신의 모바일 기기로 이동시킬 수 있다. 데이터 전송 UI에서는 테이블 화면에 구성되어 있는 다수의 명함파일과 사진들 중 자신이 원하는 사진이나 명함파일을 드래그하여 자신의 모바일 기기로 전송시킬 수도 있고 여러 사진을 회전시키며 다양한 각도에서 사진을 볼 수 있도록 구현하였

다. 직사각형 형태의 테이블에 사용자가 어느 쪽에서 위치하여 테이블을 바라보느냐에 따라 상대적으로 시점이 변하기 때문에 멀티터치 테이블에서는 테이블의 상하좌우에 대한 명확한 규정을 내리기가 힘들다. 이러한 특징 때문에 본 논문에서는 데이터 전송 UI에서 출력되어있는 사진을 터치하여 움직이면서 회전시킬 수 있도록 구현하였다.

### 나. 블루투스 페어링 데이터 전송

일반적으로 스마트폰과의 블루투스 연결 과정은 페어링이라는 전용 인증 절차를 거쳐야만 연결이 된다. 이 과정은 스마트폰에 사용자가 비밀번호를 입력하고 연결하고자 하는 기기에 같은 번호를 적음으로써 보안 인증을 하는 과정으로 보안 신뢰도는 매우 높다고 평가되지만 실제 연결에는 다소 시간이 걸린다. 본 논문에서는 페어링 과정의 간소화를 위하여 새로운 방식의 제스처를 제안한다.

스마트폰을 FTIR 멀티터치 테이블 위에 올려놓는 제스처를 통해 사용자의 블루투스 접속에 대한 요구를 감지하고 연결 허가를 함으로써 연결 과정을 간소화시키고 효율적으로 데이터를 전송할 수 있게 된다. 블루투스로 파일을 전송하기 위하여 OBEX 프로토콜을 사용하였고, 모바일 기기가 테이블 위에 올라오게 되면 전용 UI를 통해 디바이스 정보를 찾는다. 디바이스 정보를 찾은 뒤 연결된 모바일 기기를 통해 전송하고자 하는 파일을 드래그 앤 드롭 제스처를 통해 전송을 실행하게 되면 OBEX 프로토콜을 통해 데이터를 전송하고 이 데이터를 모바일 기기에서 확인할 수 있도록 구현하였다. 현재 구현한 전송 UI에서 전송 가능한 파일은 jpg, vcf 등이 있으며 jpg는 사진파일이며, vcf는 모바일 기기에 직접 저장 가능한 전화번호, 이메일, 이름 및 개인 프로필 사진 등을 이용할 수 있도록 제작된 vCard이다. vcf는 전송 후 사용자가 별도의 설정을 하지 않더라도 모바일 기기에 자동으로 전화번호부에 등록되어 손쉽게 서로의 명함과 연락처를 교환 가능하다.

### 다. 각 블록별 설명

#### (1) UI 제스처 개발

데이터 전송 UI가 실행되면서 사용자는 자신이 전송하고자 하는 파일을 터치하여 드래그가 가능하도록 설계하였다. 사용자가 원하는 파일을 선택하여 자신의 모바일

일 기기 일람이 올라와 있는 곳으로 아이টে을 드래그하여 전송을 실행하게된 순간, Drop 이벤트가 발생된다. 이벤트가 발생되면 사용자가 선택한 파일의 속성 정보를 전송 탭으로 넘겨서 전송한다. 데이터 전송 UI에서의 제스처는 크게 3가지로 구분하였다.

첫 번째로 커서 이벤트이다. 커서 이벤트는 사용자가 어떠한 파일도 선택하지 않고 UI의 백그라운드를 터치하였을 경우에 실행된다. 이벤트가 실행되면 마우스 커서를 터치 입력 지점의 좌표로 이동시킨다. 이 제스처는 적외선 카메라 제어 프로그램으로부터 넘어온 마우스 좌표 값을 WinuserAPI를 통해 제어하고, UI에서는 터치의 지점에 전송 가능한 이미지 또는 vcf 명함 파일이 존재하는지를 판단, 파일이 존재하지 않는다면 커서 이벤트로 넘어간다. 두 번째로 드래그 앤 드롭 이벤트이다. UI는 파일을 전송하지 않더라도 기본적으로 사용자가 명함과 사진을 자유롭게 움직이고 회전시킬 수 있도록 구현하였다. 사용자가 파일을 블루투스로 연결된 자신의 모바일 기기로 전송하지 않을 경우 선택된 파일은 사용자의 터치 지점을 따라다니도록 구현하였다.

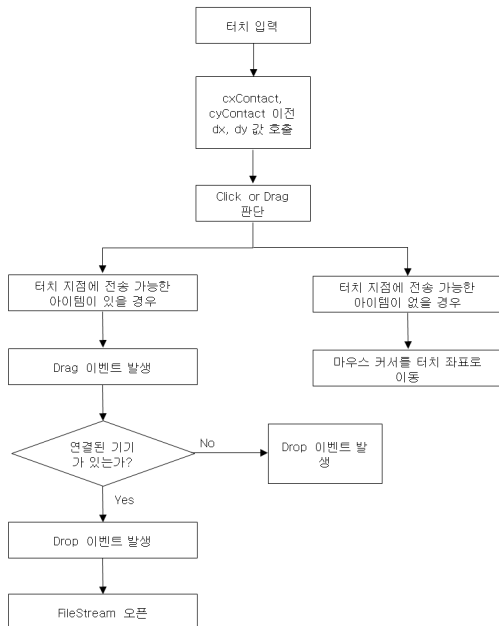


그림 5. 데이터 전송 UI의 상호 연결 관계  
Fig. 5. The interconnection of data transfer UI

세 번째로 드롭 이벤트이다. UI에 블루투스로 모바일 기기가 연결될 경우 우측 화면에 연결된 모바일 기기의

디바이스네임과 함께 모바일 기기 사진이 올라온다. 이때 사용자가 자신이 원하는 파일을 모바일기기의 사진 쪽으로 드래그 앤 드롭하게 되면 UI는 해당 파일을 선택한 모바일기기로 전송하게 된다. 그림 5는 데이터 전송 UI의 제스처 상호 연결 관계를 나타낸다. 블루투스 전송을 위한 UI를 실행시키기 위해서는 checkmax 값과 비교하여 핸드폰의 범위 값이라고 판단될 경우 마우스의 위치는 마지막 저장 지점으로 고정하고 데이터 전송 UI를 WinExec를 통해 실행시킨다. 만약 빛의 넓이가 checkmax값보다 작다면 손가락을 이용한 터치라고 판단하고 SetCursorPos를 통해 좌표값을 산출한다. 데이터 전송을 위한 UI는 실행과 즉시 블루투스 탐색을 실행하여 모바일기기와의 페어링을 실행한다.<sup>[4, 5]</sup>

(2) 모바일 디바이스 연결 및 데이터 전송

블루투스에 연결하는 과정은 장치 탐색, 연결 및 통신의 3가지 순서를 거친다. 일반적으로 블루투스는 wm장치에 추가된 직렬 장치이기 때문에 직렬 통신으로 구현하는 게 가장 기본적인 방법이다. 하지만 그 방법으로는 접근하기가 불편하기 때문에 소켓통신으로 구현 가능한 api를 이용하여 구현하였다. 블루투스는 실시간 탐색 상태로 지속적으로 새로운 기기 연결을 제공한다. 블루투스를 연결하는 과정은 그림 6과 같다.

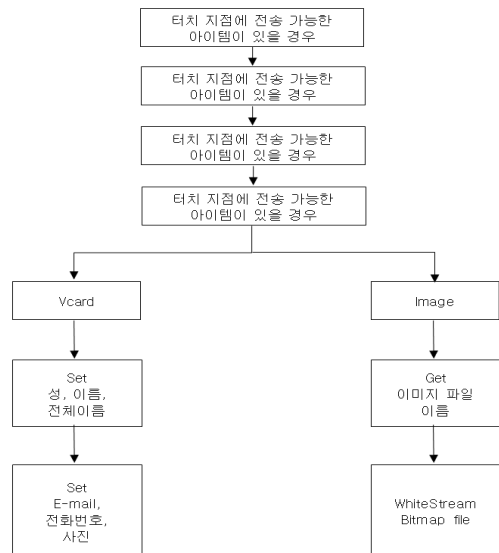


그림 6. 데이터 전송 과정 블록도  
Fig. 6. The block diagram of data transfer process

또한, 모바일 기기를 탐색하는 과정은 다음과 같다. 아래 그림 7의 영상과 같이 만약 기준치보다 넘어서는 터치 입력이 들어올 경우 모바일기기의 터치 입력이 활성화 되었다고 판단하고, 데이터 전송을 위한 UI를 실행하도록 설계하였다. 모바일 기기가 결과 영상에 잠깐 경우 dwID를 고정하고 WinExec를 통해 UI를 실행한다. 모바일 기기로 판단된 dwID 이외의 터치 입력은 다시 원래대로 SetCursorPos를 통해 마우스 데이터로써의 역할을 함으로써 모바일기기의 터치 입력이 기존 터치 입력을 방해하지 않도록 처리했다. 모바일 기기의 임계치는 checkmax라는 값이 비교치를 위한 변수를 선언하고 촬영된 영상의 흰색 음영의 넓이 값을 구하여 dwID 별로 비교하는 방식으로 구현하였다. 모바일 기기가 인식되면, 파일 전송을 위해 파일 속성과 파일명, 콘텐츠 타입을 받아서 WriteToStream을 통해 전송을 완료하게 된다. WriteToStream에서는 파일이름, 콘텐츠 타입을 받아 전송하며, 전송을 위한 콘텐츠들은 사용자가 이용하기 쉽게 구성하기 위하여 xaml로 명함모양의 디자인을 하였다. 사용자가 디자인 된 파일을 전송하고자 할 때 드롭 이벤트를 발생시키며 전송이 진행된다. 블루투스 탐색을 위하여 UI에서는 실행 즉시 기존에 연결되어 있던 블루투스를 모두 Disposable 한다. 이어서 블루투스 디바이스 인포를 탐색하여 정보를 획득하는 과정을 거치게 된다. 획득한 디바이스 정보는 디바이스의 이름 정보, 연결 정보, 디바이스의 주소 및 확인의 4가지이다. 이 과정을 거쳐 블루투스에 연결하고 데이터 전송 준비를 마치게 된다. 만약 테이블 위에 올라온 기기, 혹은 주변 기기들 중 블루투스 연결이 가능한 기기가 없다면 전송 UI는 기기의 연결을 대기하는 상태로 들어간다.

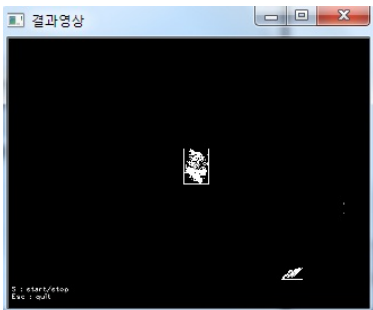


그림 7. 모바일 기기를 촬영한 적외선 카메라 영상  
Fig. 7. Infrared camera images taken on mobile devices

vCard를 전송하기 위한 vcf OBEX 전송부를 설계하였다. 설계된 vcf는 Name, Phone Number, E-mail 및 Picture의 속성으로 구성되어 있으며 Contents Type을 WriteToStream에 전송한다. 파일 전송이 시작되기 전에 완성된 vcf 파일은 FirstName + LastName + “.vcf”의 구조로 파일 이름이 생성되며 WriteToStream을 오버라이드하여 전송이 진행된다. 전송에 앞서 System.IO.StreamWriter를 선언하고 전송방식을 선택한다. 본 논문에서는 전송방식을 ASCII로 설정하여 두었다. WriteToStream을 통해 파일을 전송하기 위하여 vcf의 속성인 Name, PhoneNumber, E-mail, Picture를 NewWrite를 통해 올린다. 이미지 파일인 Picture의 경우 어떠한 속성을 가진 파일인지 확인 후 JPEG 인코더와 BitmapFrame 둘 중 하나로 선택하여 라인에 올린다. 모든 작업이 끝나면 ObexItem : DependencyObject를 통해 모바일 기기로 데이터를 전송한다.

### (3) 이미지 파일 전송

사용자가 데이터 전송 UI에서 이미지파일을 드래그하여 자신의 모바일 기기로 드롭시킬 경우 이미지 파일 전송 이벤트가 발생된다. 이벤트가 발생되면 선택된 이미지 파일의 정보를 전달하고, 이미지 전송 오브젝트에서 전송을 진행하도록 구현하였다.

본 논문에서는 이미지 파일을 전송하기 위해 InTheHand 라이브러리를 이용하였다. InTheHand 라이브러리를 통해 전송하기 전에 먼저 이미지 파일의 인코딩 형태에 따라 파일을 전송하는 과정이 다르다. 이 과정은 vcf파일 전송과 정 중 이미지 파일을 보내는 과정과 유사하다. 이미지 파일 전송과정은 그림 8의 구조로 진행된다. 먼저 이미지 파일의 속성을 통해 인코딩 된 형태를 파악하고, JPEG와 파일 형태로 확인이 되면 각각의 인코더를 통해 메모리에 저장후 전송하도록 구현하였다. JPEG 이미지 파일은 JPEG 인코더를 통해서 이미지를 읽어 들이고 메모리 버퍼를 선언 한 뒤 저장하도록 구현하였으며, DependencyProperty. Register의 ObexImage를 통해 전송한다.

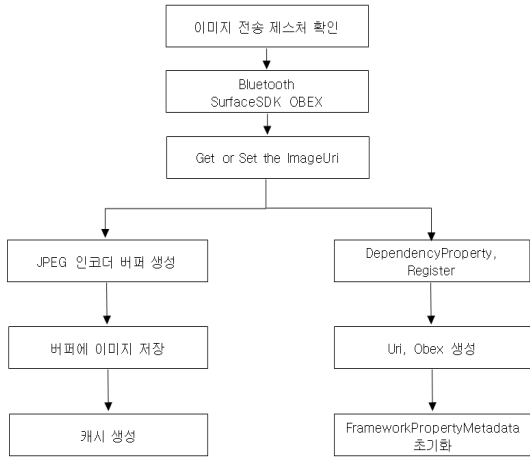


그림 8. 이미지 파일 전송 과정  
Fig. 8. The image file transfer process

## IV. 실험 및 결과

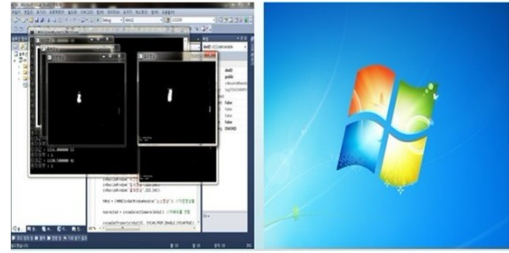
### 1. 손가락을 이용한 제스처 개발 결과

#### 가. 적외선 카메라 제어 구현 결과

적외선 카메라는 리플렉터로부터 반사된 적외선을 촬영하여 4단계의 영상처리 과정을 거친 뒤 터치 지점의 중심점을 찾는다. 이 과정을 통해 사용자의 실제 터치 지점과 화면 해상도를 비례공식에 대입하여 좌표 값을 계산, WinuserAPI를 통해 제어 한다. 그림 9는 세 개의 손가락을 인식한 흑백영상과 잡음을 제거한 결과 영상이다.

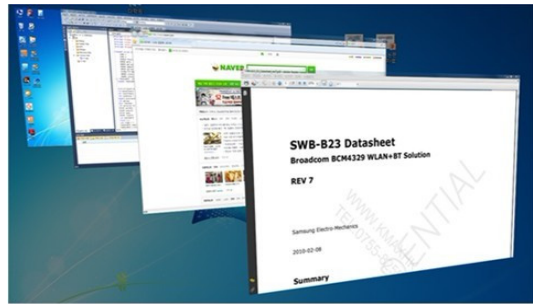


그림 9. 노이즈 제거 전과 후의 흑백 영상  
Fig. 9. Black and white images of noise removal



다수의 윈도우 창 활성화      제스처 실행 후 모든 화면 최소화

그림 10. 활성화된 모든 프로그램의 최소화 결과  
Fig. 10. Minimize results of all active programs



제스처 실행후 Aero 모드로 화면 전환

그림 11. 프로그램을 에어로 모드로 화면 전환 결과  
Fig. 11. Aero mode transitions result

#### 나. 손가락 제스처 실행 후 화면 결과

그림 10의 왼쪽 화면과 같이 다수의 창이 활성화 되어 있을 경우 손가락 세 개를 세로로 드래그 다운하면, 그림 10의 오른쪽 화면과 같이 모든 창이 최소화 되어 화면에 보이지 않게 된다.

그림 11은 손가락을 왼쪽에서 오른쪽으로 드래그하면 모든 화면이 3D 화면처럼 보여진다. 이때, 원하는 화면으로 손쉽게 전환가능하다.

### 2. 모바일 기기의 연결 및 파일 전송 결과

#### 가. 적외선 카메라 제어 구현 결과

그림 12는 결과영상의 영역 탐색과 좌표 값 실행 결과이다. 왼쪽 그림과 같이 터치 입력이 시작되면 적외선 카메라는 결과 영상에서 보듯이 적외선 탐색 영역을 가상의 사각형을 그려서 영역과 중심점을 찾게 된다. 영역의 중심부위는 작은 원을 하나 그려서 터치 지점을 표시한다. 영역의 넓이를 계산하여 일정 넓이 이상의 범위가 검출 되면 핸드폰 터치 입력으로 판단하고 오른쪽 화면에서, 좌표 검출 및 동작여부 확인에서 동작 여부 값을 1로



증가시키고 UI를 실행하도록 구성하였다. 터치 입력 넓이가 만약 기준치 이하일 경우 손가락이라고 판단하며 이에 따른 마우스 좌표값을 검출한다. 이 과정을 통해 사용자의 실제 터치 지점과 화면 해상도를 비례공식에 대입하여 좌표 값을 계산, Winuser API를 통해 컨트롤을 한다. 산출된 좌표값은 윈도우에 의해 직접 컨트롤 되며 Surface SDK를 통해 제스처 구현을 하였다. 기본적으로 cxContactx 와 cyContact에 의해 터치의 이동 거리를 측정하도록 구현되었다.

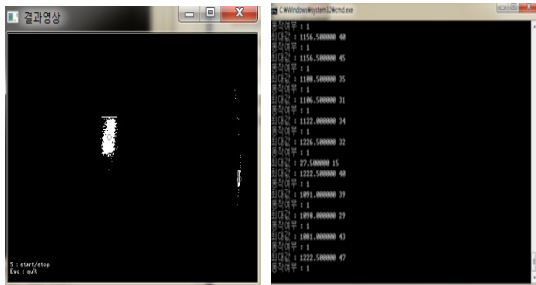


그림 12. 좌표값 산출과 동작 여부 판단 결과 영상  
Fig. 12. The result image of coordinate calculation and confirmation of operating or not

#### 나. 블루투스 전송 구현 결과

그림 13은 블루투스 전송을 위한 UI 구성 화면이며, 그림 14는 블루투스로 모바일기기에 전송한 뒤의 결과 화면이다. HID 기기가 제공되지 않는 멀티터치 테이블은 블루투스 페어링 과정을 진행하기 상당히 어려운 부분이 있다. 이러한 문제를 모바일 기기를 올려놓는 것으로 손쉽게 접속을 시도할 수 있다. 화면에 보이는 명함과 그림 파일을 우측에 떠있는 자신의 모바일 기기로 드래그 앤 드롭하여 데이터를 전송할 수 있다.

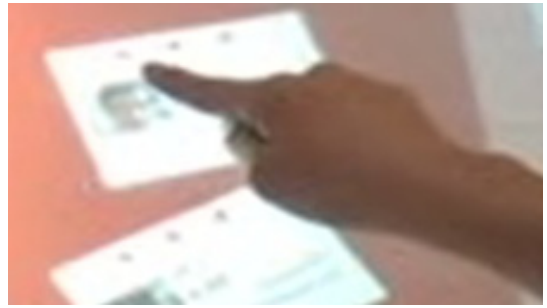


그림 13. 멀티터치 테이블에서의 데이터 전송 UI  
Fig. 13. The data transfer UI of multi touch table

데이터를 전송하게 되면 자신의 모바일 기기에서 확인 버튼을 한번 누르는 것으로 데이터는 전송이 시작되며 전송된 파일은 /bluetooth 폴더에서 직접 확인할 수 있다. vcf 파일의 경우에는 별도의 확인 작업과 추가 작업을 하지 않더라도 자신의 모바일 기기의 주소록에 자동으로 저장된다.

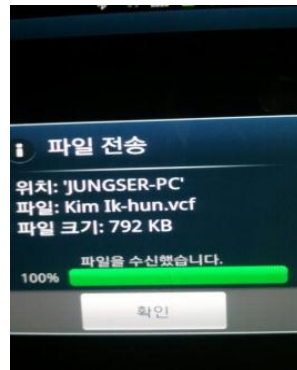


그림 14. 모바일 기기의 실제 데이터 수신 화면  
Fig. 14. The real data receive image of mobile devices

## V. 결론

기존에 개발된 제스처는 대화면 터치스크린 화면에서는 그 실효성이 부족한 것이 사실이다. 기본적으로 구성되어있는 7가지 제스처 이외에 응용동작을 통한 제스처 모두 소형 화면을 기준으로만 제시되어있다. 본 논문에서 제시한 제스처는 키보드와 마우스 등 HID 인터페이스 장치가 지원되지 않는 환경 속에서 터치스크린만으로 조작을 해야하는 불편함을 일정부분 해소시켰다. 윈도우 운영체제에서 사용 가능한 새로운 터치 제스처를

개발하여 손쉽게 화면을 제어하였고 이로 인해 사용자는 큰 불편함 없이 대화면의 터치스크린을 자유자재로 제어할 수 있게 되었다.

또한, 블루투스를 통한 기기간의 데이터 전송은 점점 그 활용 폭을 넓혀가고 있지만 보안을 너무 의식하여 복잡한 숫자로 이루어져 있는 비밀번호를 넣어주어야만 연결이 되는 등 사용자가 이용하기에는 복잡한 절차들이 산재해 있다. 본 논문에서는 FTIR 멀티터치 테이블 환경에서 모바일 기기를 테이블 위에 올려놓는 행동을 하나의 전송을 원한다는 사용자 인증형식으로 받아들이고, 보다 사용자들이 쉽게 데이터를 주고 받을 수 있는 환경을 구성하였다. 블루투스 인증 방식의 하나인 페어링 과정을 사용자가 테이블 위에 핸드폰을 올려놓는 것으로 대체하고 자동으로 실행되는 UI를 통해서 원하는 데이터를 손쉽게 전송할 수 있도록 설계하였다. 본 논문에서 제안한 방식을 이용하게될 경우 블루투스 연결을 위해 별도의 프로그램을 실행시키지 않고 손쉽게 사용자는 블루투스를 이용할 수 있다. 추후 FTIR 방식에서 모바일 기기를 올려놓아도 정반사 현상을 일으킬 수 있는 실리콘 케이스에 실리콘을 통해 개인 모바일 기기의 정보를 담은 기술이 확보될 시에는 블루투스 연결 또한 획기적으로 효율적 이용이 가능해지며, 모바일 기기 뿐만 아니라 블루투스가 지원되는 카메라, 노트북 등 다양한 기기를 연동할 수 있게 된다. 기술의 적용 범위를 조금 더 확장해 가정집에서 사용될 멀티터치 테이블로 확대하면 블루투스 연동을 통해 게임, TV 등 다양한 콘텐츠를 컨트롤 하는 중추가 될 것으로 기대된다.

## 참 고 문 헌

- [1] Minha Yang, "UI Development for FTIR Multi-touch Screen ", Korea Digital Design Council, pp.335-345, 2009
- [2] Jinbok Kim, Munkyu Lee, "User authentication using touch positions in a touch-screen interface", KIISC, Vol 21, pp. 135~141, 2011
- [3] MinGu Kang, YongJin Jeong, "Image Processing Algorithms for DI-method Multi Touch Screen Controllers", IEEK, Vol 48, SP, pp. 1~12, 2011
- [4] Jongyun Yeo, Dongbyeong Kang, Sunghun Shin, Gumin Jeong, "Implementation of Android SmartPresenter Using Bluetooth-Based multi-Links Services", KIPS, pp.707~708, 2011
- [5] Byungdon Kong, Jungsang Min, Rohae Myung, "Menu design for touch screen interface with regard to task characteristics and cognitive processes", HCI Korea, pp.763~769, 2009
- [6] Samkweon Oh, Geunduk Park, Byoungkuk Kim "Performance Comparison of the Recognition Methods of a Touched Area on a Touch-Screen Panel for Embedded Systems", KAIS, pp. 2334-2340, 2009

※ 본 논문은 2009학년도 세명대학교 지원에 의한 연구임

## 저자 소개

### 박 상 봉(정회원)



- 1985년 : 광운대 전자재료공학과 학사 졸업
- 1987년 : 고려대 전자공학 석사 졸업
- 1992년 : 고려대 전자공학과 공학 박사 졸업
- 1992년 ~ 1999년 : 삼성전자 선임 연구원

• 1999년 ~ 현재 : 세명대학교 정보통신학과 부교수  
 • 2000년 ~ 현재 : @lab(주) Digital 설계팀 기술고문  
 <주관심분야 : HID chip design >

### 안 증 서(준회원)



- 2010년 : 세명대 정보통신학과 학사 졸업
- 2012년 : 세명대 전산정보 석사 졸업
- 2012년 ~ 현재 : (주)다우엑실리콘 엔지니어

<주관심분야 : Multi-Touch Display, TFT Signal Control>