

Design of OpenCV based Finger Recognition System using binary processing and histogram graph

Yeong-Tae Baek*, Se-Hoon Lee**, Ji-Seong Kim***

Abstract

NUI is a motion interface. It uses the body of the user without the use of HID device such as a mouse and keyboard to control the device. In this paper, we use a Pi Camera and sensors connected to it with small embedded board Raspberry Pi. We are using the OpenCV algorithms optimized for image recognition and computer vision compared with traditional HID equipment and to implement a more human-friendly and intuitive interface NUI devices. comparison operation detects motion, it proposed a more advanced motion sensors and recognition systems fused connected to the Raspberry Pi.

▶ Keyword : OpenCV, Gesture Recognition, Area Detection, NUI

I. Introduction

삶속에 인터넷 연결이 빈번해지고 사용자가 접하게 되는 정보가 더욱 다양하고 방대해짐에 따라 사용자는 습득한 정보를 빠르게 이해하고 활용하는 것이 매우 중요하다. 이러한 정보 전달 및 가공에 있어서 NUI(Natural User Interface) 기술이 더욱 중요해지고 있다. 이 NUI기술은 마우스, 키보드, 펜 등의 HID장치를 이용하지 않고 사용자의 제스처를 인터페이스로 활용하기 때문에 Natural User Interface를 줄여 명명되었다[1]. 정보 획득 및 가공 기술이 복잡해질수록 기술과 사용자 사이의 간격은 점점 넓어지는데, 이러한 간격은 더 직관적이고 쉬운 인터페이스 기술을 통해 해결이 가능하다. 본 논문은 다양한 NUI 응용 시스템의 기반 기술 중 하나인 손 제스처 인식기술을 이용한 NUI응용 인터페이스 개발 및 구현을 제안한다.

II. Related works

1. NUI commercial devices

Fig. 1은 상용화된 제스처 인식 기반 NUI 시스템을 나타내고 있다. 첫 번째로 트랙패드 제스처는 정전식 터치패드로 손가락의 개수, 위치, 좌표, 면적을 계산해 제스처를 인식한 후 인식 결과에 따른 명령을 실행한다. 두 번째로 구글 Soli프로젝트는 터치, 카메라 인식이 아닌 레이더 반사파 원리를 이용해 손의 모양, 위치, 면적을 감지해 제스처 인식 정보로 사용하며, 이는 방해 파동에 민감할 뿐만 아니라 단순 파동을 이용하기 때문에 손이 아닌 물체를 판별하지 못하는 문제가 있다. 또한 정밀한 레이더의 사용을 요구해 NUI장치로서는 상대적으로 부담스러운 가격을 형성한다.

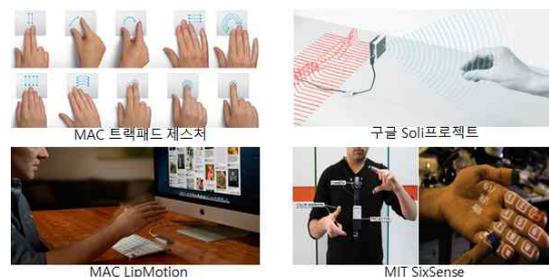


Fig. 1 Similar systems

*First Author: Yeong-Tae Baek, Corresponding Author: Se-Hoon Lee
*Yeong-Tae Baek(hannae@kimpo.ac.kr), Dept. of Multimedia, Kimpo University
**Se-Hoon Lee(seihoon@inhac.ac.kr), Dept. of Computer Systems & Engineering, Inha Technical College
***Ji-Seong Kim(gogovlzk@gmail.com), Dept. of Computer Systems & Engineering, Inha Technical College
• Received: 2016. 02. 05, Revised: 2016. 02. 14, Accepted: 2016. 02. 22.
• This work was supported by KIMPO College's Research Fund.
• This paper will expand the paper ("Gesture Recognition using binary processing and histogram graph with OpenCV") presented at the winter Conference Korea Society of Computer and Information 2016.

Table 1. Similar systems comparison table

	트랙패드	Soli	LipMotion	SixSense	본 연구
제작사	애플	구글	애플	Mit	
접촉/비접촉	접촉	비 접촉	비 접촉	간접접촉	비 접촉
가격	비교적 저렴	고가	고가	비교적 저렴	비교적 저렴
적용 기기	PC	PC	PC	웨어러블	PC
인식 감도	보통	높음	보통	낮음	보통
인식 장치	정전식 패드	레이더	카메라+레이더	카메라	카메라+센서

세 번째로 LipMotion은 적외선 광을 이용해 손의 반사광을 인식해 손의 제스처를 감지한다. 이는 NUI장치에 프로세싱 기능이 없이 센서만 존재하며, 제스처 인식 및 처리를 사용자 PC가 전담하기 때문에 많은 리소스를 요구한다는 문제가 있다. 네 번째로 SixSense는 앞에서 설명한 NUI장치보다 더욱 직관적인 형태로, 사용자가 직접 착용하고 사용할 수 있는 컴퓨팅 디바이스이다. 소형 빔 프로젝터와 카메라를 이용해 사용자의 제스처를 감지하며, 사용자의 제스처 인식 정보를 스마트폰 또는 PC로 전송해 명령을 수행한다. 그러나 웨어러블 장치이지만 아직 휴대할 수 있는 수준의 소형화 단계에 도달하지 못해 PC에 종속적인 NUI장치와 비교해 경쟁력이 상대적으로 떨어진다. Table 1은 앞에서 설명한 네 가지 시스템과 본 논문에서 연구한 시스템의 특징을 비교해 나타내고 있다.

2. Typical recognition algorithm



Fig. 2 Typical object recognition algorithms

Table 2. Algorithm description table

알고리즘	설명
이진화 피부영역 검출연산	사람의 피부 색 추정치로 신체 일부분을 마스크 하여 손을 검출 및 판단하며, 검출한 마스크에서 가장 큰 픽셀 덩어리를 손바닥 중심으로 가정하고 연산한다. 이미 손을 검출한 후 손가락의 개수만 파악하기 때문에 제스처 인식 속도가 가장 빠르며, 주위 환경 변화에 영향이 적다.
히스토그램 비교연산	영상의 각 영역별 색상 값을 그래프로 나타낸 후 2개 이상의 그래프를 비교하여 유사도를 도출한다. 이미지 비교연산 시 다른 알고리즘과 비교하여 연산 속도가 빠르다
템플릿 매칭	가장 빈번하게 사용되는 매칭 알고리즘이며, 관심 대상을 도형으로 치환한 후 비교영상에서 관심 대상과 유사한 도형을 찾는다. 이는 1:1 크기의 영상 보다는 큰 영상에서 작은 대상을 찾아낼 때 사용한다.
피처 매칭	비교할 영상의 모서리, 특별한 색 등을 기준으로 특징을 연산한 후 비교영상에 같은 연산을 하여 특징을 추출해 비교 연산한다. 연산속도가 상대적으로 느린 대신 비교이미지의 회전 시에도 강력한 검출 효과를 얻을 수 있다.

Fig. 2와 Table 2는 제스처 인식을 위한 대표적 물체 인식 알고리즘 4가지를 나타내고 있다. 본 논문에서는 빠른 연산을 통해 평균값을 추출하여 제스처를 인식하기 위해 비교적 연산 속도가 빠른 이진화 피부영역 검출연산 알고리즘과 히스토그램 비교연산 알고리즘을 사용해 제스처 인식 시스템을 구현하고자 한다.

III. Object recognition algorithms

1. Skin area detection finger counting method

영상에서 피부영역 검출 방식으로 손을 검출하기 위해서 피부영역과 피부영역 이외의 영역을 이진화 영상에서 제거하기 위해 사람의 피부색과 유사한 색 값을 추출하여 이진화 마스크 과정을 통해 손을 검출해야 한다. 이 때 RGB 색 공간에서 마스크 할 경우 영상에서 비치는 손의 조도에 따라 가변 하는 손의 색을 판별할 수 없는 문제가 발생한다. 따라서 본 논문에서는 이 문제를 해결하기 위해 RGB 정보를 인코딩한 색 정보인 YCbCr 색 공간에서 마스크연산을 수행한다.

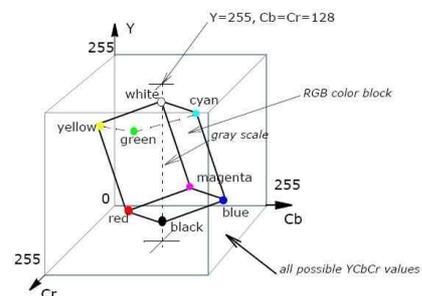


Fig. 3 Relationship between RGB and YCbCr[2]

Fig. 3에 따르면 빛의 3원색을 정보로 이용하는 RGB 색 공간과는 달리 YCbCr 색 공간에서는 휘도와 색차 값을 정보로서 이용하게 되며, Y는 휘도, Cb와 Cr는 색차 성분을 갖고 있다. 사람의 피부색을 조사하기 위해 RGB 색 공간에서 영상에 나타난 피부색의 최대, 최소값을 조사한 후, Fig. 4의 RGB-YCrCb 변환 식에 따라 색 정보를 YCbCr 색 공간으로 변환해 준다. 영상의 조도에 따라 변하는 휘도 Y 값을 배제한 후 Cr Cb 색차 값만을 통해 피부색을 조명에 상관없이 검출해 낼 수 있다.

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B + 0$$

$$C_b = -0.169 \times R - 0.331 \times G + 0.499 \times B + 128$$

$$C_r = 0.499 \times R - 0.418 \times G - 0.0813 \times B + 128$$

$$R = \text{clamp}(Y + 1.402 \times (C_r - 128))$$

$$G = \text{clamp}(Y - 0.344 \times (C_b - 128) - 0.714 \times (C_r - 128))$$

$$B = \text{clamp}(Y + 1.772 \times (C_b - 128))$$

Fig. 4 RGB-YCrCb conversion formula[3]

위 식을 통해 YCrCb 색 공간에서의 빛에 따라 가변 할 수 있는 가변 치 범위의 피부색을 도출할 수 있으며, 이에 따라 본 논문에서 도출해낸 피부색 범위는 Cr 최소값 128, 최댓값 170, Cb 최소값 73, 최댓값 158이다. 위 과정을 통해 도출해낸 정보를 토대로 피부색을 판별해 마스킹 할 수 있지만 피부색 판별 과정에서 피부색과 유사한 영역이 노이즈로서 나타날 수 있다는 문제점이 있다. 이러한 문제를 해결하기 위해 본 논문에서는 모폴로지 침식 연산을 이용해 노이즈를 최소화하며, 경계선을 단순화한다.

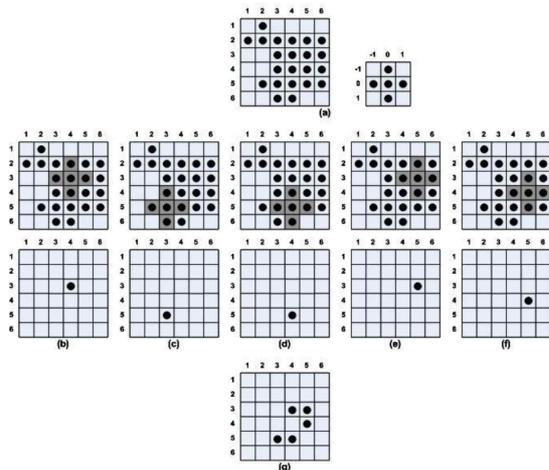


Fig. 5 Morphology erosion operation[4]

Fig. 5에 표현된 모폴로지 침식연산은 테두리 영역부터 안으로 픽셀을 소거해 규모가 작은 픽셀집단을 제거할 수 있으며, 이는 직접적으로 노이즈를 최소화 할 수 있는 수단이 된다. 또한 경계선 단순화 작업을 통해 손가락영역을 비교적 정확하게 구분할 수 있는 장점도 있다.

Algorithm 1 Get Hand Mask

```

Begin
  minCr=128; maxCr=170; minCb=73; maxCb=158;
  cvtColor(image, YCrCb, CV_BGR2YCrCb);
  vector<Mat> planes; split(YCrCb, planes);
  mask(image.size(), CV_8U, Scalar(0));
  nr=image.rows; nc=image.cols;
  for(i=0; i<nr; i++){
    CrPlane=planes[1].ptr<uchar>(i);
    CbPlane=planes[2].ptr<uchar>(i);
    for(j=0; j<nc; j++){
      if((minCr<CrPlane[j])&&(CrPlane[j]<maxCr)&&(minCb<CbPlane[j])&&(CbPlane[j]<maxCb))
        mask.at<uchar>(i, j)=255;
    }
  }
  erode(mask, mask, Mat(3, 3, CV_8U, Scalar(1)), Point(-1, -1), 2);
End
    
```

Fig. 6은 모폴로지 침식연산 이전과 이후를 비교하고 있다. 위와 같이 검출된 손 영상에서 손가락의 개수를 인식하고 제스처 정보로 활용하기 위해 손의 중심을 찾는 과정이 필요하다. Algorithm 2의 거리변환 행렬연산을 통해 가장 큰 픽셀집단의 중심으로부터 픽셀집단의 가장자리까지의 가장 짧은 거리를 구할 수 있는데, 이 값은 손바닥을 둘러싸는 원의 반지름으로 간주할 수 있으며, 이 반지름으로 손바닥의 중심을 도출해낼 수 있다[5].

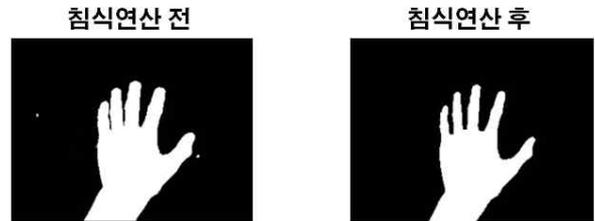


Fig. 6 Morphological Erosion Operation

Algorithm 2에서 도출한 손바닥의 중심점과 반지름을 이용해 손가락의 개수를 알아내 제스처 인식 정보로 사용해야한다. Fig. 7에 따라 중심점 기준으로 약 radius * 2 의 크기로 원을 그린 후 원의 경계면을 지나는 픽셀집단의 개수를 세어 손가락의 개수를 알 수 있다. Algorithm 3을 이용하면 원의 외곽을 따라 이동하면서 이진화 픽셀 값이 1로 바뀌는 지점을 체크해 손가락의 개수를 알 수 있다.

Algorithm 2 Distance Transform Matrix

```

Begin
  distanceTransform(mask, dst, CV_DIST_L2, 5);
  minMaxIdx(dst, NULL, &radius, NULL, maxIdx, mask);
End
    
```

Algorithm 3 findContours Operating

```

Begin
  cImg(mask.size(), CV_8U, Scalar(0));
  circle(cImg, center, radius*scale, Scalar(255));
  findContours(dmg, contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE);
  fingerCount=0;
  for(int i=1; i<contours[0].size(); i++){
    p1=contours[0][i-1];
    p2=contours[0][i];
    if(mask.at<uchar>(p1.y, p1.x)==0 && mask.at<uchar>(p2.y, p2.x)>1)
      fingerCount++;
  }
End
    
```

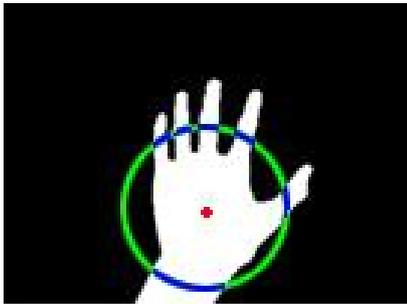


Fig. 7 Count Finger

2. Comparison arithmetic method of the histogram

피부영역 검출 방식으로 제스처를 체크할 경우 손가락의 개수로 제스처를 인식하므로 손의 모양을 통해 제스처를 인식하는 것은 불가능하다. 이러한 문제를 해결하기 위해 본 논문에서는 히스토그램 비교연산을 사용한다. Fig. 8은 히스토그램 그래프를 나타내고 있다. 히스토그램이란 영상의 픽셀 값의 분포를 그래프로서 표현한 것인데, 그래프의 세로축은 픽셀 값의 빈도를 나타내며, 가로 축은 영상의 픽셀 값을 나타낸다. 이 때 가로 축의 픽셀 값이 RGB색 공간을 바탕으로 이루어진다면 비교적 많은 연산 시간이 소요 되므로 처리속도가 늦어질 수 있다. 따라서 색을 단순화 시킨 HSV 색 공간의 이미지로 변환한 후 히스토그램 연산을 시행한다. Fig. 9는 RGB색 공간에서 HSV색 공간으로 변환한 이미지를 나타내고 있다[6].

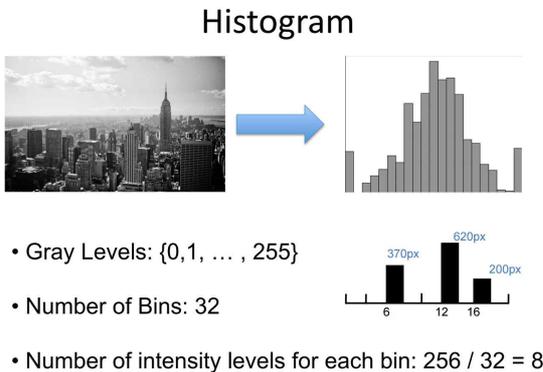


Fig. 8 Histogram Graph[7]



Fig. 9 RGB Color To HSV Color

영상에서 추출해낸 히스토그램 값을 비교 대상의 히스토그램 값과 비교하여 두 영상의 유사도를 측정할 수 있다. 따라서 히스토그램 비교연산을 바탕으로 이전에 기억해 둔 손의 모양 히스토그램을 카메라에 비치는 손의 모양과 비교하여 현재 손의 제스처가 어떤 형태를 나타내는지 분석해낼 수 있다. Fig. 10은 히스토그램 비교연산 결과를 나타낸 것이다. 왼쪽 영상은 비교대상 영상이고 오른쪽 영상은 현재 카메라에 비치는 영상을 나타낸다. 커맨드 창에 출력된 비교 연산 결과는 Table 3과 같다.

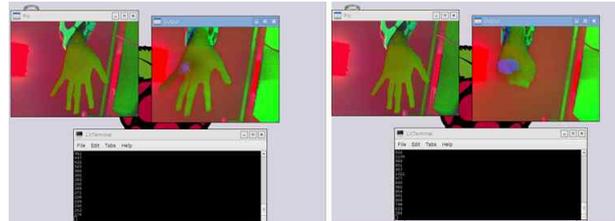


Fig. 10 Histogram Matching

Table 3의 결과에 따르면 유사 영상은 히스토그램 비교연산 결과 값이 약 450 이하를 나타내며, 비 유사 영상은 결과 값이 약 750 이상을 나타낸다. 따라서 위 실험을 통해 히스토그램 비교연산으로 유사 영상과 비 유사 영상을 판별할 수 있음을 알 수 있다.

Table 3. Histogram Matching Result

회차	유사이미지	비 유사이미지
1	362	822
2	437	1134
3	422	983
4	403	851
5	389	957
6	300	1021
7	282	977
8	230	830
9	268	982
10	271	864
11	228	891
12	239	804
13	230	758
14	262	623
15	274	784

IV. Implementation of the system

1. System Overview

Fig. 11은 본 논문에서 제안하는 전체 시스템 구성을 나타내고 있다. 라즈베리파이에 장착된 파이 카메라와 ADC에 연결된 IR 거리센서 3개가 제스처 인식을 담당하며, Windows OS PC가 최종 결과를 사용자에게 출력하는 서버 역할을 한다. 이 때 통신은 라즈베리 파이와 PC는 UDP 프로토콜로 통신해 서버-클라이언트 관계를 유지한다.

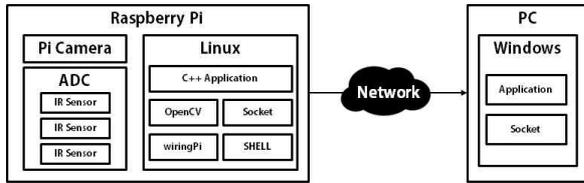


Fig. 11 Sequence Diagram

처음 프로그램을 실행하게 되면 먼저 라즈베리파이가 연결된 외부 장치를 사용할 수 있는지 기본적인 체크를 한 후 PC서버에 접속한다. Fig. 12는 PC와 라즈베리파이의 사용 준비를 마친 후 PC가 서버 역할 그리고 라즈베리파이가 클라이언트 역할을 시작하는 시점에서의 시스템 흐름과 구조를 표현한 다이어그램이다. 서버는 클라이언트의 연결을 감지하면 사용자의 설정 정보를 클라이언트로 전송하는데, 이 설정정보는 어떤 방식으로 손을 검출한 후 제스처를 인식할 것인지에 대한 사용자 설정정보이다. 이 정보는 피부검출 손가락 카운팅 방식 또는 히스토그램 비교연산 방식 둘 중 하나를 의미한다. 초기화를 마쳤다면 클라이언트는 연산 대기상태로 진입 후 물체 접촉 인터럽트 발생을 기다린다. 이후 손을 감지해 인터럽트가 발생했다면 OpenCV를 이용해 피부영역 검출 손가락 카운팅 방식 또는 히스토그램 비교연산을 통해 제스처를 감지한다. 감지한 제스처 처리결과를 1초 동안의 통계를 통해 제스처를 판단한 후 판단 결과를 서버로 전송해 지정된 명령을 실행한다.



Fig. 12. Sequence Diagram

Fig. 13은 OpenCV 연산 시작 시 제스처 인식 과정 흐름도로써, 라즈베리파이가 인지한 영상을 처리하는 방식을 나타낸다. 센서와 연동해 불필요한 연산을 줄이며, 사용자 설정에 의한 분기연산으로서 피부검출 손가락 카운팅 방식 또는 히스토그램 비교 연산 방식으로 연산해 사용자가 선택한 방식으로 결과를 서버로 전송할 수 있다.

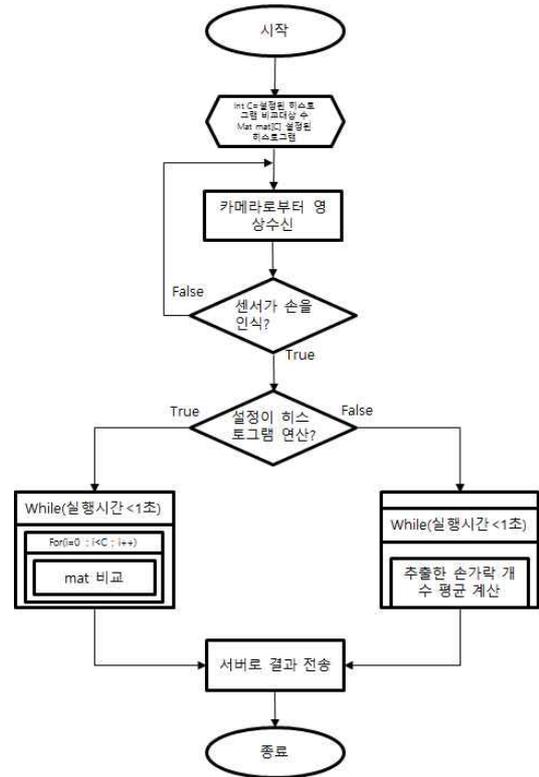


Fig. 13 Gesture recognition process flow diagram

2. Using Infrared distance sensors

3개의 적외선 거리센서를 통해 손의 접촉과 두드림 및 손의 이동을 감지하여 제스처 인식을 보조한다. Algorithm 4에서 센서의 거리 감지를 통해 사용자의 손이 카메라 밖 영역에서 좌-우로 이동하였는지를 감지한 후 결과를 서버로 전송한다.

Algorithm 4. Left and right motion detection

```

Begin
if(read_mcp3208_adc(0)>700){
st = clock();
while(clock()-st<1500){
if(read_mcp3208_adc(1)>700){
strcpy(msg, "[TYPE=MOVE:VAL=LEFT:]");
write(sock,msg,strlen(msg));
delay(600);
}}}
if(read_mcp3208_adc(1)>700){
st = clock();
while(clock()-st<1500){
if(read_mcp3208_adc(0)>700){
strcpy(msg, "[TYPE=MOVE:VAL=RIGHT:]");
write(sock,msg,strlen(msg));
delay(600);
}}}
}}
End
    
```

3. Histogram gesture recognition

사용자 설정정보가 히스토그램 비교연산으로 설정되었다면 비교할 제스처가 표현된 영상을 로드한 후 HSV색 공간으로 변환 후 각 비교 대상 영상의 히스토그램 그래프를 생성까지의 초기화를 마친 다음 손이 감지되길 기다린다. 손을 카메라에 비추어 인터럽트가 발생한다면 현재 피카카메라에 비친 제스처를 히스토그램 그래프화 하여 초기화한 히스토그램과 비교한다. 비교 결과가 Table 1의 결과를 참조하여 450 이하의 결과 값이 나온다면 비교대상 영상과 일치한다고 판단 후 서버에 일치하는 영상번호를 전송한다.

4. Skin area detection finger counting

사용자 설정정보가 피부 영역 검출 손가락 카운팅 방식으로 설정되었다면 손에 대한 인터럽트가 발생하였을 때 피카카메라에 비치는 영상을 RGB 색 공간에서 YCbCr 색 공간으로 인코딩한다. 이 후 휘도값을 제외한 색차 값을 통해 손 영역을 검출해낸 후 2진화 마스크 처리한 다음 거리변환 행렬을 통해 손바닥의 반지름과 중심을 찾아낸다. 이진화 마스크 결과에서 손바닥 반지름*1.5의 원의 테두리를 따라 연산하며 결과가 0에서 1로 바뀌는 지점을 체크하여 손가락의 개수를 카운팅한 후 서버로 손가락 개수를 전송한다.

5. Gesture command

본 논문에서 구현한 제스처 인식 프로그램은 Fig. 14에서 의미하는 4개의 제스처 표본을 대상으로 명령을 수행하도록 설계되었으며, 명령수행 단계에서 서버는 패킷을 전송받아 Win32 API를 이용한 외부 프로그램 실행 및 제어를 통해 사용자가 본 논문에서 개발한 NUI장치를 이용해 PC를 제어할 수 있도록 구성하였다.

제스처	OK	주먹	손바닥 세움	손 좌우이동
이미지				
명령수행	IExplore 실행	IExplore 종료	설정된 문서 실행	문서 페이지 넘김

Fig. 14 Gesture recognition process flow diagram

V. Conclusions

본 논문은 많은 정보를 보다 효율적으로 처리하기 위해 기존 HID장비를 대체할 수 있는 NUI장치를 개발하는데 목적을 두었으며, 라즈베리파이와 거리센서, 피카카메라를 이용하여 손가락의 개수 및 모양을 판별해 PC 서버로 명령을 전달하여 손 인식 NUI장치를 개발하고자 하였다. 손 인식에는 적외선 거리 센서와 OpenCV라이브러리를 사용해 구현하였으며, 손의 최초 감지는 센서를, 손의 인지는 색 정보 인코딩, 모폴로지 연산, 히스토그램 그래프 등을 사용하여 피부 영역 검출 손가락 카운팅 방식과 히스토그램 비교연산 두 가지로 제스처를 인식하는 방식을 나누어 사용자가 선택할 수 있게끔 설계하였다. 본 논문에서는 상대적으로 양호한 조건에서 실험하였으며, 향후 이를 실험환경과 다른 열악한 환경에서 이용하기 위해서는 보다 정교하게 개선된 이진화 처리, 영상비교알고리즘이 필요할 것으로 사료된다.

REFERENCE

- [1] Gwang-hyung Lee, Dong-kyoo Shin, Don-gil Shin, "NUI/NUX framework based on intuitive hand motion" Korea Network Information Society pp .12-13, Jun. 2014
- [2] Intel , "Color Models" Intel Developer Zone , <https://software.intel.com/en-us/node/503873>
- [3] Logic Devices, "RGB to YCbCr Report" Logic Devices pp .2, Jan 2001, <http://www.logicdevices.com/support/appnotes/1f3370r gb.pdf>
- [4] Jong-Hong Sin, Sun-Bong Jang, In-Ho Ji, "IT CookBook - Introduction to Digital Image Processing" ,Hanbit Media pp C-11, Jan. 2008
- [5] Ju-Hwang Kim, jeong-hun Park, Jong-seo Jeon, Min-Goo Kang, Seung-Hyun Lee, "Recognition of Hand Interface for PC-Mouse using OpenCV" Korea Network Information Society pp .303-306, 2009
- [6] Einführungspraktikum Computer Vision, "Digital Image Processing" R.C Gonzalez & R.E Woods , pp. 7, 2008
- [7] Fabrizio Dini, giuseppe lisanti, "ImageProcessing with OpenCV" PPM2010 Seminar, pp. 11, 2010

Authors



Yeong-Tae Baek received the B.S. degree in Computer Science from Inha University, Korea, in 1989 and M.S. and Ph.D degrees in Computer Science from Inha University, Korea, in 1993 and 2002, respectively. He is currently a Professor in the Department of Multimedia at Kimpo University. He is interested in multimedia contents, IoT platform, and mobile system.



Ji-Sung Kim is currently studying in the Department of Computer & Engineering at Inha Technical College. He is interested in embedded software, IoT security, and NUI.



Se-Hoon Lee received the B.S. degree in Computer Science from Inha University, Korea, in 1985 and M.S. and Ph.D degrees in Computer Science from Inha University, Korea, in 1987 and 1996, respectively. From 1987 to 1990, he worked as a information analyst officer in the Computing Center, at Marine Corp. Dr. Lee joined the faculty of the Department of Computer Engineering at Inha Technical College, Incheon, Korea, in 1993. He is currently a Professor in the Department of Computer Systems & Engineering at Inha Technical College. He is interested in embedded software, IoT platform, and mobile system.