

# 구부러진 손가락 끝점 추적을 위한 컬러 영역 보정 알고리즘

## Color Area Correction Algorithm for Tracking Curved Fingertip

강성관\*, 정경용\*\*, 임기욱\*\*\*, 이정현\*\*\*\*

인하대학교 정보공학과\*, 상지대학교 컴퓨터정보공학부\*\*,  
선문대학교 컴퓨터정보공학부\*\*\*, 인하대학교 컴퓨터정보공학부\*\*\*\*

Sung-Kwan Kang(kskk1111@nate.com)\*, Kyung-Yong Chung(dragonhci@hanmail.net)\*\*,  
Kee-Wook Rim(rim@sunmoon.ac.kr)\*\*\*, Jung-Hyun Lee(jhlee@inha.ac.kr)\*\*\*\*

### 요약

기존 손가락 추적을 수행하는데 있어 손가락 끝점을 계산하는 방법 중 가장 일반적인 방법은 먼저 피부색 정보를 추출한다. 블럽 함수의 블럽 컬러링(Blob Coloring) 알고리즘을 통하여 피부 윤곽선을 계산하고, 그 중 가장 최상위 점을 손가락 끝점으로 정한다. 그러나 이 방법은 구부러진 손가락 상태에서 그것의 위치를 측정할 때 실제 손가락 끝이 아닌 잘못된 위치를 탐지하는 문제점을 가지고 있다. 본 논문에서는 구부러진 손가락 끝점 추적을 위한 컬러 영역 보정 알고리즘을 제안하였다. 제안하는 방법은 구부러진 손가락 상태에서 손가락 끝점 추적시 잘못된 곳을 측정하는 문제점을 사용자들의 성향을 통해 미리 예상하고 보정함으로써 성능을 향상시키고자 한다. 제안하는 방법을 개발하여 논리적 타당성과 유효성을 검증하기 위해 실험적인 적용을 시도하고자 한다. 따라서 영상인식에서 서비스의 만족도와 질을 향상시켰다.

■ 중심어 : | 손동작 인식 | 상호작용 | 윤곽선 | 피부색 정보 |

### Abstract

In the field of image processing to track the fingertip much research has been done. The most common way to calculate the fingertip first, to extract color information. Then, it uses Blob Coloring algorithms which are expressed in blob functions the skin contour and calculates. The algorithm from contour decides the highest location with the fingertip. But this method when measuring it location from the finger condition which bents is not the actual fingertip and has the problem which detects the location which goes wrong. This paper proposes the color space correction algorithm to tracks the fingertip which bents. The method which proposes when tracking the fingertip from the finger condition which bents solves the problem which measures the location which goes wrong. Aim of this paper in compliance with the propensity of the users forecasts a problem in advance and corrects with improvement at the time of height boil an efficiency. Ultimately, this paper suggests empirical application to verify the adequacy and the validity with the proposed method. Accordingly, the satisfaction and the quality of services will be improved the image recognition.

■ keyword : | Fingertip Tracking | Interaction | Finger Contour | Skin Color |

"본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음"  
(NIPA-2011-C1090-1131-0004)

접수번호 : #110707-003

접수일자 : 2011년 07월 07일

심사완료일 : 2011년 07월 28일

교신저자 : 정경용, e-mail : dragonhci@hanmail.net

## I. 서론

최근 비디오 영상으로부터 실시간으로 물체를 추적하는 기술에 대한 필요성이 증가하고 있고 많은 분야에서 관련 연구가 진행되고 있다. 물체를 추적하는데 있어서 카메라가 움직이거나 조명과 뒷배경으로 인한 잡영상이 생길 수 있고 또한 물체의 추적이 실패하는 등의 다양한 상황이 발생할 수 있다. 손가락 끝점 추적을 이용한 시각기반 상호작용은 손가락을 지속적으로 따라다니며 그 이동 및 동작에 따라서 이벤트를 발생시켜 주는 인터페이스 방식이다[1][2]. 손가락 끝점을 추적하는 기술은 에어 마우스와 가상 키보드 및 영상 감지 센서를 이용한 디지털 미디어 장치의 동작 등 많은 분야에 응용할 수 있다. 손가락 끝점 추적 기술이 이러한 분야에 적절히 응용되기 위해서는 실시간으로 처리할 수 있을 정도의 적은 계산량으로 처리될 수 있어야 한다. 이러한 필요조건을 동시에 만족하는 알고리즘을 제시하는 일은 상당히 어려운 것으로 이를 해결하기 위한 많은 연구가 있다[3][4]. 닌텐도 Wii와 같은 제품이 인기를 끌면서 특히 무선으로 컴퓨터나 스마트 텔레비전을 조정하는 기술은 많은 사람들의 관심을 사게 되었다. 적외선 센서나 LED 또는 레이저 포인터 등으로 마우스의 역할을 대신하려는 시도 및 연구가 활발히 진행되고 있다. 그러나 이러한 도구들을 이용한 추적은 주변 환경의 영향으로 인한 오인식 또는 추적 실패의 잦은 발생 및 컴퓨터와의 동기화가 맞지 않아서 해결해야 할 부분이 많이 있다[5][6]. 최근에 많이 사용되는 카메라 입력 영상으로부터의 물체를 추적하는 방법으로는 모폴로지, 라벨링, 차영상, CamShift 알고리즘 등 많은 기술이 있다. 이 알고리즘들은 실시간 입력 영상에서 물체나 사람의 추적, 손가락 끝점 추적, 얼굴 추적 등 다양하게 응용되고 있다[7].

제한하는 시스템은 손가락 끝점을 먼저 검출한 후 트래킹을 수행한다[1]. 일반적으로 손가락 끝점 검출은 피부색 정보를 추출한 뒤 블럽 함수의 블럽 컬러링 알고리즘을 통하여 최상위 점을 추출한다[3]. 그러나 이 방법을 적용하여 윤곽선의 최상위 점을 손가락 끝점으로 결정하고 트래킹을 수행하면, 트래킹 시에 사용자가 손

가락을 구부릴 때 정확한 손가락 끝점을 검출하지 못하고 피부색 정보의 가장 최상위 점만을 찾게 되므로 오류가 발생한다.

본 논문에서는 이러한 문제점을 해결하기 위해 손가락 끝점의 예상 범위를 설정하여 그것의 위치를 보장하는 방법을 제안함으로써 사용자의 편리성과 정확성을 높일 수 있도록 하였다.

## II. 관련연구

기존의 손가락 끝점 추적에 있어서 주된 관심사는 배경 영상과 손 영역을 단편화하여 분리해내는 방법과 단편화 된 손 영역에서 손가락들의 윤곽선을 추출해 내는 방법이다. 피부 색상을 이용한 손 영역의 단편화는 손과 손가락 끝점을 찾기 위하여 이용되어지는 중요한 정보이다. 그 방법은 피부색 픽셀의 색상 모델을 기초로 하여 배경 영상과 손영역을 차영상 기법을 이용하여 분리해 낸다. 손이 배경 영상으로부터 단편화된 후에 윤곽선이 추출되어진다[8]. 윤곽선 벡터는 손의 가장자리의 연속적인 좌표를 포함한다. 그 후에 윤곽선 벡터의 처리는 손가락 끝점의 위치를 준다. [그림 1]은 실시간 입력 영상으로부터 색상 단편화를 통한 손영역 탐지 및 손가락 끝점을 찾아내는 과정을 보여준다.

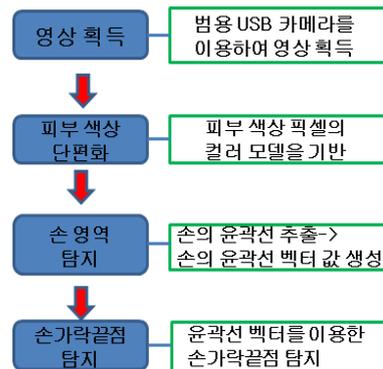


그림 1. 손가락 끝점 찾기

일단 손가락 끝점이 찾아지면 그것을 추적하는 방법은 일반적으로 주어진 어떤 임계값보다 강한 색상의 픽

셀들 중에서  $y$ 좌표값이 가장 큰 곳을 계속 찾아내게 된다. 이 경우에는 모든 픽셀에 대하여 비교 연산을 수행하여야 하고 비교 연산이 많아질수록 연산 속도가 증가하게 되었다. 또한 영상을 얻는 과정에서 피부 색상과 비슷한 잡영이 생기게 되므로 찾아진 손가락 끝점을 잃어버리는 경우가 자주 발생하게 된다.

최근의 많은 연구들이 OpenCV 라이브러리를 기반으로 한 Blob analysis package를 이용하여 입력되는 영상에서 가장 높은 위치에 있는 일정 크기 (1500-19000 pixel)의 피부색(Red 37-60, Green 28-34)덩이(blob)를 찾도록 작성되었다[3]. 이는 색상을 기반으로 하여 연속적으로 새로운 제일 높은 위치한 덩이를 찾아서 갱신하도록 하기 때문에, 피부색과 비슷한 물체가 나타나거나 물체의 위와 아래의 위치가 변경될 경우 손가락 끝점을 잃어버리기 때문에, 효율성이 떨어진다. Mean-Shift 알고리즘의 경우에는 주어진 영역의 전체를 탐색하고 물체에 대한 존재할 가능성이 높은 위치에 대한 수렴 여부를 반복적으로 확인하여 결과를 얻는 방법이다[9]. 이 알고리즘의 정지영상 색 분할 알고리즘은 비디오 영상에서 처리할 수 있도록 확장된 알고리즘이다[11][13].

## II. 손가락 끝점 추적을 위한 컬러 영역 보정

### 1. 손 영역의 피부색 정보 추출

손가락 끝점 추적을 위해서는 먼저 손 영역을 검출한다[1]. 손 영역은 피부색 정보를 이용하여 추출한다[4]. 피부색 정보를 추출하는 방법은 RGB 컬러 영역 또는 HSV 컬러영역, YCbCr 컬러영역까지 3개의 컬러 공간을 많이 사용하는 것으로 알려져 있다[5][6]. RGB 영역은 Red, Green, Blue의 3가지 색상 값을 사용하여 나타내는 컬러 공간으로 일반적으로 정규화 된 형태로 변경하여 사용한다. 정규화 된 RGB형을 만드는 기본 공식은 (식 1)과 같다.

$$\begin{aligned} r &= R/R+G+B \\ g &= G/R+G+B \\ b &= B/R+G+B \end{aligned} \quad (1)$$

(식 1)에서  $b$  컬러는 어떠한 중요한 정보도 가지고 있지 않기 때문에 보통 생략하고  $r$ 과  $g$ 의 두 가지 컬러를 사용하여 피부색 정보를 추출한다[7].

반면 HSV영역은 농도, 채도, 그리고 명도에 대한 생각을 기반으로 하여 직관적인 값을 가진 컬러를 제공하는데, 농도는 붉은색, 초록색, 자주색 그리고 노란색과 같은 지배색을 정의한다. 채도는 밝기에 비례하여 어떤 영역에 대한 색채의 풍부함의 정도를 나타낸다. 또한 명암은 컬러 휘도와 관련이 있다. HSV 공간을 사용하면 각각의 요소가 서로 독립적인 장점이 있다. 트래킹 알고리즘 중에서 CamShift는 RGB공간을 사용하지 않고 HSV 공간을 사용한다. RGB 공간에서 HSV 공간으로 변경 후 Hue를 사용한다.

CamShift 알고리즘을 처리하기 위해서는 일단 RGB 공간으로 이루어진 입력 영상 정보를 HSV 공간으로 변환하여 Hue 색상 정보만 사용할 수 있도록 한다. 얻어진 Hue정보로부터 1D 히스토그램을 계산하고 2D 확률 분포를 얻는다. 얻어진 분포에서 포인트의 중심점, 장축, 단축 및 각도를 계산 한다. YCbCr 컬러모델은 색상 신호가 아니라 휘도  $Y$ 와 색차 신호  $C_b$ ,  $C_r$ 에 기반한 색 표현 방식이다.  $Y$ 성분은 밝기 정보를 포함한다.  $C_b$ 와  $C_r$  성분은 실제 색상에 대한 정보를 가지고 있으며, 특히  $C_b$ 는 파란색에서 밝기 성분을 뺀 정보를 가지고 있고  $C_r$ 은 빨간색에서 밝기 성분을 뺀 정보를 가지고 있다. 이 모델에서는 영상의 그레이스케일 정보와 색상 정보를 따로 분리하여 부호화한다. 흑백 TV에서는 YCbCr 성분 중에서  $Y$  성분만을 이용하여 화면에 영상을 표현한다. [그림 2]는 RGB 색상공간을 나타낸다. 일반적으로 색상 성분은 밝기 성분에 비하여 사람이 덜 민감하게 반응하기 때문에  $C_b$ 와  $C_r$  성분을 방송에서 송신할 때에는  $Y$  성분보다 적은 비트를 할당한다. 보통은  $C_b$ 와  $C_r$  성분은  $Y$  성분의 반에 해당하는 정보만을 사용한다. YCbCr 성분을 사용하면 전송할 신호의 양을 줄일 수 있다는 장점도 있다. 인간의 눈이 색상보다 밝기에 더 민감하기 때문에 색차 신호를 이용해서 처리하기 위한 모델이다. 본 논문에서는 세 가지 컬러 방식 중 광원에 대한 표면방향의 변화에 불변하기 때문에 정규화 된 RGB 컬러 방식을 사용하였다.

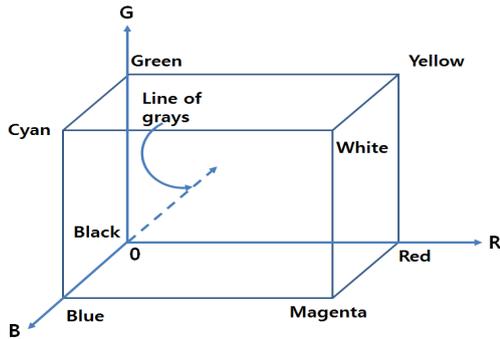


그림 2. RGB 색상 공간

RGB 색 모델은 CRT 모니터에서 컬러 영상을 표현하기 위해 만들어진 색 모델이다. RGB 색 모델은 [그림 2]와 같이 정육면체 형태의 모델로 표현된다. 각 축은 빨간색, 녹색, 파란색을 표현하며, 모든 색상은 각 R, G, B 색상 성분의 선형 조합으로 표현할 수 있다. 제안하는 보정 시스템에서는 RGB 컬러 공간을 사용하여 컬러 영상 영역 분할 방법을 이용한다. 이미지 영역 분할이란 각 비슷한 화소값끼리 모아 영역으로 만드는 것으로서, 입력 영상에 먼저 손 영역을 탐색한 후에, 배경과 손을 분리하기 위해 사용한다. 밝기의 특성을 이용하여 손가락 영역을 최종적으로 추출하기 위해 블러프 컬러링 기법을 사용하였다. 기존 블러프 컬러링의 연산 속도를 개선하기 위하여 보정된 알고리즘을 사용하였으며 실험한 결과로 손가락 영역 추출의 성공률 95%를 얻었다.

실제 컴퓨터에서 사용하는 R, G, B 성분의 값의 범위는 각각 [0, 255]를 갖는다. 그러므로 컴퓨터에서 표현할 수 있는 색의 가짓수는  $256^3=16,777,216$  개가 된다. 그러므로 트루컬러 영상은 대략 천육백만 가지 색을 표현할 수 있다고 표현 한다. 제안하는 보정 시스템에서는 손 영역을 탐지하기 위하여 피부색 영역은 Red 컬러값을 30~60, Green 컬러값을 25~35사이로 실험에 의해 설정하였다[8][9].

## 2. 블러프 컬러링 알고리즘

블러프 컬러링 알고리즘은 정규화 된 RGB 컬러모델을 이용하여 피부색 영역을 추출하고, 추출된 영역을 이진

화한 후 픽셀의 총합을 계산한다[4]. 블러프 컬러링을 이용한 CT 영상에서 각 영역 자동 추출 연구에 따르면 블러프 컬러링 알고리즘은 [표 1]과 같다[3]. 이 알고리즘은 입력 영상이 임계값이나 다른 방법을 사용하여 영역의 번호로 분할되면, 영상 안에 있는 연결된 구성요소의 각각을 식별할 수 있다. 이 방법은 입력 영상안의 특정 객체의 픽셀들이 자신 또는 그 이웃들의 특정 색상으로 이루어진 영역 안에 있다는 사실을 기초로 한다. 따라서 알고리즘은 다른 컬러가 같은 영역인 부분으로 발견될 때마다 같은 컬러로 변경해주는 반복적인 작업을 하여 각 영역을 나누어 주는 역할을 한다.

표 1. 블러프 컬러링 알고리즘

```

Given binary Image
Let the initial Color[k] = 0
Step 1.
Scan the original image from left to right and top to bottom(raster Scan) using L-shape template

For k = ImageSize
If Image[C] = 255 then
    If Image[U] = 255 and Image[L] = 0 then
        Color[C] = Color[U]
    If Image[U] = 0 and Image[L] = 255 then
        Color[C] = Color[L]
    If Image[U] = 0 and Image[L] = 0 then
        Color[C] = K++, new Color
    If Image[U] = 255 and Image[L] = 255 then
        If Color[U] is not equal Color[L] then
            ReColoringTable process
            Color[C] = Color[L]
        repeat
Step 2.
For count = k
    Color[count]=ReColoring[Color[count]]
repeat
    
```

## 3. 구부러진 손가락에서 손가락 끝점의 문제점

손가락의 끝점은 찾아진 손 영역으로부터 가장 높은 곳의 위치를 나타내는 것이다. 정규화 된 RGB 컬러 방식을 통하여 피부색 정보를 측정한 뒤, 블러프 함수를 통하여 실제 손의 영역을 산출할 수 있다[4].

손가락 끝점은 일반적으로 블러프 함수를 통해 얻은 영역 중 최상위에 있는 점으로 결정된다. 그러나 이 방법은 손가락 끝점을 정하는데 있어서 손가락이 구부러져 있을 때는 문제가 된다. 사용자들의 성향을 보면 웹캠의 위치와 깊이 정보가 없는 2D 영상을 사용한다는 두 가지의 조건으로 인하여 손가락 끝점을 쉽게 정하지 못

한다. 첫 번째로 일반 컴퓨터 사용자가 쓰는 웹캠의 위치는 모니터의 위다. 그걸 바라보는 일반적인 사람들은 모니터를 높은 위치에서 내려다보지는 않는다. 그러므로 웹캠은 당연히 사용자를 정면으로 향하거나 약간 아래로 내려다보게 된다. 사용자가 그 위치에서 웹캠에 보이게 손을 들어 올릴 때, 손은 당연히 위를 향하고, 손가락 또한 위를 향하게 되므로 최상위에 있는 손가락을 손가락 끝점으로 하는 것이 적합하다고 볼 수 있다. 두 번째로 웹캠은 깊이 정보가 없는 2D 영상을 사용한다. 블러프 함수를 통해 나온 최상위 점을 손가락 끝점으로 하지 않고 최상위 점 외의 다른 점을 손가락 끝점으로 결정하면, 그 점이 실제 어떤 손가락의 끝인지 아닌지 확실히 알 수 없다. 블러프 함수는 손가락 끝만을 잡는 함수가 아니라, 정규화 된 RGB를 통하여 구해진 피부색 정보를 블러프 컬러링 알고리즘을 통하여 피부로 인식되는 영역을 전반적으로 찾아내는 함수이기 때문이다 [10]. [그림 3]은 구부러진 손가락이 있을 때는 잘못된 손가락 끝점을 추정하는 모습을 보여준다. [그림 3]에서 보이는 바와 같이 실제 손가락 끝점을 탐지하지 못하고 손가락 아래쪽 마디를 추출하고 있다.

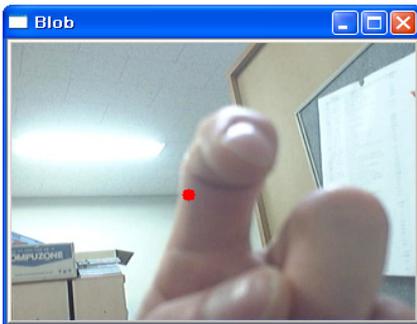


그림 3. 잘못된 손가락 끝점을 추정한 결과

#### 4. 구부러진 손가락 끝점의 위치 보정

블러프 컬러링 알고리즘은 90°부터 180°까지는 실제로 2D 영상만을 처리하는 카메라가 손가락 끝점을 탐지할 수 있지만 0~90°까지는 인식할 수 없으므로 해결해야 할 영역은 0~90°까지의 손가락을 구부렸을 때이다. 이 부분을 해결할 수 있다면 손가락 끝점 추적 문제를 일정부분 해결했다고 볼 수 있다. 또한 이 알고리즘은 다

른 Color가 같은 영역인 부분으로 발견될 때 마다 같은 Color로 변경해 주어야 하는 반복 작업 때문에 많은 연산 시간이 소요된다. 본 논문에서는 이를 해결하기 위하여 0~90°사이 손가락이 들어온다면 그 부분을 그 영역의 중심인 45°에 있다고 가정한다. 0~90°까지 손가락 끝점의 이동범위는 작으므로 45°에 위치한다고 예상범위를 설정하여도 오차범위가 매우 적다. [표 2]는 손가락 끝점 위치 보정 알고리즘을 나타낸다.

표 2. 손가락 끝점 위치 보정 알고리즘

손가락 끝점 위치 보정 알고리즘
1. 영상을 좌상단부터 우하단 순으로 주사한다.
2. 현재 화소가 전경화소일 경우
(a) 상단 화소와 왼쪽 화소 중 하나만 라벨링 되어 있으면 그 라벨을 할당한다.
(b) 상단 화소와 왼쪽 화소가 동일한 라벨일 경우, 그 라벨을 할당한다.
(c) 상단 화소와 왼쪽 화소가 다른 라벨일 경우, 상단 화소의 라벨을 할당하고 두 라벨을 동치 테이블에 동치 라벨로 등록한다.
(d) 둘 다 라벨링 안되어 있을 경우는 새로운 라벨을 할당하고 이 라벨을 동치 테이블에 등록한다.
3. 주사가 완료될 때까지 단계 2를 수행한다
4. 동치 테이블에 등록되어 있는 각 동치 라벨 중에서 가장 작은 라벨 값을 선택한다.
5. 영상을 주사하며 각 라벨을 동치 테이블의 가장 작은 라벨값으로 치환한다.

[그림 4]는 구부러진 손가락의 문제점을 보정하여 측정하는 모습을 나타낸다. [그림 4]에서 보정 알고리즘을 적용하여 구부러진 손가락에 따른 손가락 끝점 탐지의 문제점을 개선하였음을 볼 수 있다.

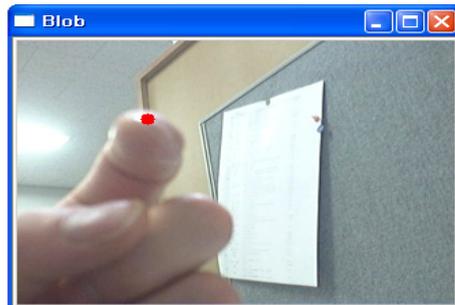


그림 4. 구부러진 손가락의 문제점을 보정한 결과

#### IV. 실험 결과

본 논문에서 제안하는 알고리즘의 성능을 평가하기 위한 실험은 AMD 64 X2 Dual Core Processor 3800+ 칩이 장착된 컴퓨터에서 수행하였다. 운영체제로는 Windows XP Professional을 사용하였고 컴파일러로는 Visual C++ 2010을 사용하였다. 실험에 사용한 데이터는 320×240의 해상도를 가지는 웹캠을 통해 입력받았다. 성능 평가를 위해 기존 알고리즘과 구부러진 손가락 상태의 보정 알고리즘을 동일한 환경에서 각각 50번을 반복적으로 수행하였다.

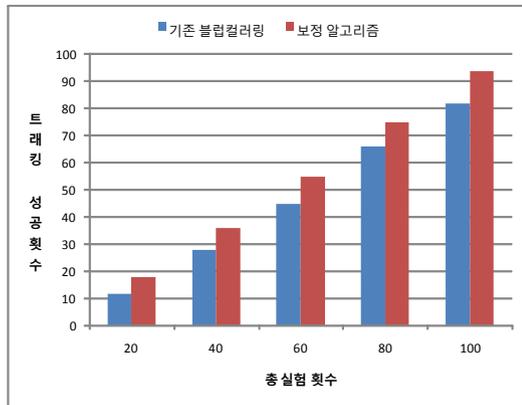


그림 5. 시스템의 성능 평가 결과

[그림 5]는 제안하는 알고리즘의 성능평가 결과를 나타낸다. [그림 4]의 실험 결과를 보면 구부러진 손가락 상태에서 기존의 블립 컬러링 알고리즘을 사용하였을 시에는 정확한 손가락 끝점을 50% 이상 탐지하지 못하였으나, 보정된 블립 컬러링 알고리즘을 적용하였을 시에는 약 70% 정도는 정확한 손가락 끝점을 추적하는 것을 알 수 있다.

#### V. 결론

본 논문에서는 구부러진 손가락 상태에서의 손가락 끝점을 탐지하는 문제에 있어 위치의 오차범위가 작다는 것을 인식하고 중간값을 결정함으로써 손가락 끝점

추적 시스템을 보완하였다. 앞으로의 연구 방향은 연구에서와 같이 2대 이상의 웹캠을 사용하여 보다 정확하게 손가락 끝점 추적의 문제를 해결하는 방법을 찾아내는 것이다. 일반 PC에서는 그 문제가 웹캠의 위치를 잘 설정해주면 충분히 해결할 수 있지만, 손가락 끝점은 일반 PC에서보다 향후 임베디드 환경에서 더 많이 사용될 것으로 보여진다. 임베디드 환경에 있어서는 일반 PC에서보다 좀 더 제한적이다. 웹캠의 위치만 하여도 쉽게 결정할 수 있는 문제가 아니기 때문에 여러가지 환경제약에 따라 PC 환경에서는 쉽게 해결할 수 있던 문제에 있어서는 접근하기 매우 어렵다. 따라서 이에 대한 해결방안이 제시되어야 할 것이다. 또한 실제 임베디드 시스템 환경에서의 보다 검증된 실험이 필요하다.

#### 참고 문헌

- [1] T. Brown and R. C. Thomas, "Finger Tracking for the Digital Desk," In Proc. of the Australasian User Interface Conference, pp.11-16, 2000.
- [2] A. M. Bowden and R. Bowden, "View-based Location and Tracking of Body Parts for Visual Interaction," In BMVA Symposium on Spatiotemporal Image Processing, 2004.
- [3] 임옥현, 김진철, 박성미, 이배호, "블립 컬러 링을 이용한 CT 영상에서 간 영역 자동 추출", 한국정보과학회, 제31권, 제2호, pp.760-762, 2004.
- [4] S. Tsuruoka, A. Kinoshita, T. Wakabayashi, and Y. M. Ishida, "Extraction of Hand Region and Spectification of Fingertips from Color Image," In Proc. of International Conference Virtual Systems and Multimedia, pp.206-211, 1997.
- [5] R. Hota, V. Venkoparao, and S. Bedros, "Face Detection by using Skin Color Model based on One Class Classifier," In Proc. of International Conference on Information Technology, pp.15-16, 2006.
- [6] F. Javier, Toledo-Moreo, J. J. Martinez-Alvarez,

- and J. M. Ferrandez-Vicente, "Hand-based Interface for Augmented Reality," In Proc. of IEEE Symposium on Field-Programmable Custom Computing Machines, pp.291-292, 2007.
- [7] F. Boussaid, D Chai, and A. Bouzerdoum, "On-Chip Skin Detection for Color CMOS Imagers," In Proc. of International Conference on MEMS, NSNO and Smart System, pp.357-361, 2003.
- [8] B. A. Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A. P. Sheth, I. B. Arpinar, A. Joshi, and T. Finin, "Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection," In Proc. of the International Conference on World Wide Web, 2006.
- [9] K. Oka, Y. Satoand, and H. Koike, "Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems," In Proc. of International Conference on Automatic Face and Gesture Recognition, pp.411-416, 2002.
- [10] H. Fillbrandt, S. Akyol, and K. F. Kraiss, "Extraction of 3D Hand Shape and Posture from Image Sequences for Sign Language Recognition," IEEE International Workshop on Analysis and Modeling of Faces and Gestures, Vol.17, No.10, pp.181-186, 2003.
- [11] W. Z. Gang, and Z. D. Fang1, "Application of Visualization Technology in Spatial Data Mining," Computer Engineering, Vol.33, No.18, pp.67-71, 2009.
- [12] Z. Jin. Visualization for Information Retrieval, Springer, 2008.
- [13] A. Jaimes and N. Sebe, "Multimodal Human Computer Interaction: A Survey," Computer Vision and Image Understanding, Vol.108, Issue1-2, pp.116-134, 2007.
- [14] M. Cabral, M. Zuffo, S. Ghiretti, O. Belloc, L. Nomura, M. Nagamura, F. Andrade, R. Faria, and L Ferraz, "An Experience using X3D for Virtual Cultural Heritage," In Proc. of the Int. Conference on 3D Web Technology, pp.161-164, 2007.
- [15] K. Y. Jung and J. H. Lee, "User Preference Mining through Hybrid Collaborative Filtering and Content-based Filtering in Recommendation System," IEICE Tran. on Information and Systems, Vol.E87-D, No.12, pp.2781-2790, 2004.
- [16] P. A. Chirita, A. Damian, W. Nejdi and W. Siberski, "Search Strategies for Scientific Collaboration Networks," In Proc. of the ACM Workshop on Information Retrieval in Peer-to-Peer Networks, 2005.

#### 저 자 소 개

##### 강 성 관(Sung-Kwan Kang)

정회원



- 2001년 2월 : 인하대학교 컴퓨터 공학부(공학사)
- 2005년 8월 : 인하대학교 정보통신공학과(공학석사)
- 2005년 9월 ~ 현재 : 인하대학교 정보공학과(박사과정)

• 2011년 2월 ~ 현재 : 인하대학교 정보통신처 애플리케이션 매니저

<관심분야> : 영상처리, 패턴인식, 컴퓨터 비전, HCI

##### 정 경 용(Kyung-Yong Chung)

정회원



- 2000년 2월 : 인하대학교 전자계산공학과(공학사)
- 2002년 2월 : 인하대학교 컴퓨터 정보공학과(공학석사)
- 2005년 8월 : 인하대학교 컴퓨터 정보공학과(공학박사)

- 2005년 9월 ~ 2006년 2월 : 한세대학교 IT학부 교수
  - 2006년 3월 ~ 현재 : 상지대학교 컴퓨터정보공학부 교수
- <관심분야> : IT융합기술, HCI, 상황인식, USN

**임 기 욱(Kee-Wook Rim)**

정회원



- 1977년 2월 : 인하대학교 전자공학(공학사)
  - 1987년 2월 : 한양대학교 전자계산학(공학석사)
  - 1994년 8월 : 인하대학교 전자계산학(공학박사)
  - 1977년 ~ 1983년 : 한국전자기술연구소 선임연구원
  - 1983년 ~ 1988년 : 한국전자통신연구소 연구실장
  - 1989년 ~ 1996년 : 한국전자통신연구원 시스템연구부장
  - 2001년 ~ 2003년 : 한국전자통신연구원 컴퓨터소프트웨어 연구소장
  - 2000년 ~ 현재 : 선문대학교 컴퓨터정보학부 교수
- <관심분야> : 실시간데이터베이스시스템, 운영체제

**이 정 현(Jung-Hyun Lee)**

정회원



- 1977년 2월 : 인하대학교 전자과(공학사)
  - 1980년 9월 : 인하대학교 전자공학(공학석사)
  - 1988년 2월 : 인하대학교 전자공학(공학박사)
  - 1979년 ~ 1981년 : 한국전자기술 연구소 시스템 연구원
  - 1984년 ~ 1989년 : 경기대학교 전자계산학과 교수
  - 1989년 1월 ~ 현재 : 인하대학교 컴퓨터공학부 교수
- <관심분야> : IT융합기술, HCI, 상황인식, USN