

벤딩 핑거에 강인한 개선된 핑거팁 위치측정 방법연구

김용훈^o, 남미영, 강성관, 이필규
인하대학교 컴퓨터정보공학과

{mole8201, bk9150}@naver.com, ksk1111@empal.com, pkrhee@inha.ac.kr

An Estimation Method of Location Using Robust Finger Tips for Bending Finger.

Yong Hoon Kim^o, Mi Young Nam, Sung Kwan Kang, Phill Kyu Rhee
Dept. of Computer Engineering & Information, Inha University

요 약

기존 핑거 트래킹(Finger Tracking)을 수행하는데 있어 핑거팁(Finger-tip)을 구하는 방법 중 가장 일반적인 방법은 먼저 피부색 정보(Skin Color)를 추출한다. 그 다음 블럽(Blob) 함수의 블럽 컬러링(Blob Coloring) 알고리즘을 통하여 피부 윤곽선(Skin Contour)을 구하고, 그 중 가장 최상위 점을 핑거팁으로 정한다. 그러나 이 방법은 벤딩 핑거(Bending Finger) 상태에서 핑거팁 위치를 측정할 때 실제 손가락 끝이 아닌 잘못된 위치를 잡는 문제점을 가지고 있다. 본 논문에서 제안하는 방법은 벤딩 핑거 상태에서의 핑거 트래킹시 잘못된 핑거팁을 측정하는 문제점을 사용자들의 성향을 통해 미리 예상하고 보정함으로써 성능을 향상시키고자 한다.

1. 서 론

핑거 트래킹[1]을 이용한 비주얼 인터랙션(Visual Interaction)[2]은 손가락을 지속적으로 따라다니며 그 이동 및 제스처(Gesture)에 따라서 이벤트(Event)를 발생시켜주는 새로운 UI(User Interface)방식이다.

핑거 트래킹은 핑거팁을 먼저 검출한 후 트래킹을 수행한다[1]. 일반적으로 핑거팁 검출은 피부색 정보를 추출한 뒤 블럽함수의 블럽 컬러링 알고리즘을 통하여 최상위 점을 추출한다[3]. 그러나 이 방법을 적용하여 윤곽선(Contour)의 최상위 점을 핑거팁으로 결정하고 트래킹을 수행하면, 트래킹시에 사용자가 손가락을 구부릴 때 정확한 핑거팁을 검출하지 못하고 피부색 정보의 가장 최상위 점만을 찾게 되므로 오류가 발생한다.

본 논문에서는 이러한 문제점을 해결하기 위해 핑거팁을 예상 범위를 설정하여 핑거 팁의 위치를 보정하는 방법을 제안함으로써 사용자의 편리성과 정확성을 높일 수 있도록 하였다.

2. 손 영역의 피부색 정보(Skin Color) 추출

핑거 트래킹을 위해서는 먼저 손 영역을 검출한다[1]. 손 영역은 피부색 정보를 이용하여 추출한다[4]. 피부색 정보를 추출하는 방법은 RGB 컬러(Color)영역 또는 HSV 컬러영역, YCbCr 컬러영역까지 3 개의 컬러 공간을 많이 사용하는 것으로 알려져 있다[5, 6].

RGB 영역은 Red, Green, Blue 의 3 가지 색상 값을 사용하여 나타내는 컬러 공간으로 일반적으로 정규화된(Normalized) 형태로 변경하여 사용한다.

정규화된 RGB 형을 만드는 기본 공식은 다음과 같다.

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

여기서 b 컬러는 어떠한 중요한 정보도 가지고 있지 않기 때문에 보통 생략하고 r 과 g 의 두 가지 컬러를 사용하여 피부색 정보를 추출한다[7].

반면 HSV 영역은 농도, 채도, 그리고 명도에 대한 생각을 기반으로 하여 직관적인 값을 가진 컬러를 제공한다. 농도(Hue)는 붉은색, 초록색, 자주색 그리고 노란색과 같은 지배색을 정의한다. 채도(Saturation)는 밝기에 비례하여 어떤 영역에 대한 색채의 풍부함의

정도를 나타낸다. 또한 명암(Value)은 컬러 휘도와 관련이 있다.

YCbCr 컬러모델은 색상 신호가 아니라 휘도(luminance) Y 와 색차 신호 Cb, Cr 에 기반한 색 표현 방식이다. 인간의 눈이 색상보다 밝기에 더 민감하기 때문에 색차 신호를 이용해서 처리하기 위한 모델이다.

본 논문에서는 세가지 컬러 방식 중 광원에 대한 표면방향의 변화에 불변하기 때문에 정규화된 RGB 컬러 방식[8]을 사용하였고, 피부색 영역은 Red 컬러값을 30~60, Green 컬러값을 25~35 사이로 실험에 의해 설정하였다.

3. 블러프 컬러링

블러프 컬러링 알고리즘은 정규화된 RGB 컬러모델을 이용하여 피부색 영역을 추출하고, 추출된 영역을 이진화한 후 픽셀의 총합을 계산한다[4].

블러프 컬러링을 이용한 CT 영상에서 간 영역 자동 추출 연구에 따르면 블러프 컬러링 알고리즘은 다음과 같다[3].

```

알고리즘 1 : 블러프 컬러링 알고리즘
Given binary Image
Let the initial Color[k] = 0
Scan the original image from left to right and top to bottom(raster Scan)
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
        Color[C] = Color[L]
Repeat
    
```

블러프 컬러링 알고리즘은 다른 컬러가 같은 영역인 부분으로 발견될 때마다 같은 컬러로 변경해주는 반복적인 작업을 하여 각 영역을 나누어 주는 역할을 한다.

4. 밴딩 핑거에 따른 핑거팁 문제점

핑거팁은 손가락의 끝을 나타내는 것이다. 정규화된 RGB 컬러 방식을 통하여 피부색 정보를 측정된 뒤, 블러프함수를 통하여 실제 손의 영역을 산출할 수 있음을 보았다[4].

핑거팁은 일반적으로 블러프함수를 통해 얻은 영역 중 최상위에 있는 점으로 결정된다. 그러나 이 방법은 핑거팁을 정하는데 있어서 밴딩 핑거시에는 문제가 된다. 사용자들의 성향을 보면 웹캠의 위치와 깊이 정보가 없는 2D 영상을 사용한다는 두 가지의 조건으로 인하여 핑거팁을 쉽게 정하지 못한다.

첫 번째로 일반 PC 사용자가 쓰는 웹캠의 위치는 모니터의 위다. 그걸 바라보는 일반적인 사람들은 모니터를 높은 위치에서 내려다 보지는 않는다. 그러므로 웹캠은 당연히 사용자를 정면으로 향하거나 약간 아래로 내려다보게 된다. 사용자가 그 위치에서 웹캠에 보이게 손을 들어올릴 때, 손은 당연히 위를 향하고, 손가락 또한 위를 향하게 되므로 최상위에 있는 손가락을 핑거팁으로 하는 것이 적합하다고 볼 수 있다.

두 번째로 웹캠은 깊이 정보가 없는 2D 영상을 사용한다. 블러프함수를 통해 나온 최상위 점을 핑거팁으로 하지 않고 최상위 점 외의 다른 점을 핑거팁으로 결정하면, 그 점이 실제 어떤 손가락의 끝인지 아닌지 확실히 알 수 없다. 블러프함수는 손가락 끝만을 잡는 함수가 아니라, 정규화된 RGB 를 통하여 구해진 피부색 정보를 블러프 컬러링 알고리즘을 통하여 피부로 인식되는 영역을 전반적으로 찾아내는 함수이기 때문이다.

그림 1 은 밴딩 핑거시 잘못된 핑거팁을 측정하는 모습을 보여준다. [그림 1]에서 보이는 바와 같이 실제 핑거팁을 잡지 못하고 손가락 위쪽 마디를 추출하고 있다.

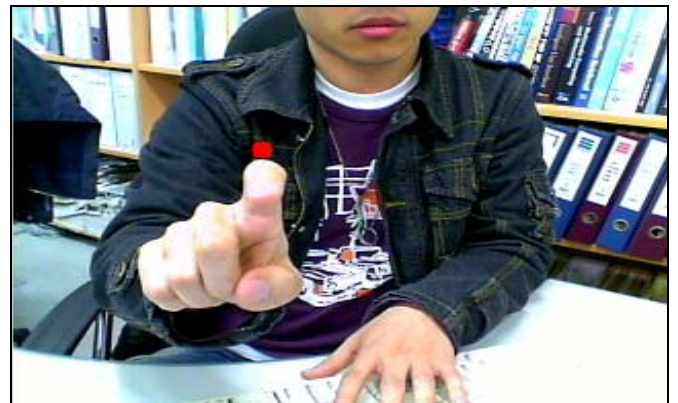


그림 1 밴딩 핑거에 따른 핑거팁의 문제점

5. 밴딩 핑거상태의 핑거팁 보정

그림 1 에서 보듯이 핑거팁에 따른 문제점은 사실상 2D 영상을 이용하는 웹캠으로 해결하기에는 많은 무리가 따른다. 그에 따른 문제를 해결하기 위해 일반적으로 2 대 이상의 웹캠을 사용하는 방법이 가장 많이 제시되고 있다. 그러나 두 대 이상의 카메라를 사용한다는 것은 비용적인 측면에 있어서의 부담을 줄 수 있을 뿐만 아니라 카메라의 위치를 어디에 둘 것인가 하는 문제에 있어서도 쉽지 정할 수 없기 때문에 여러 대의 카메라를 사용하는 것이 그러한 문제의 완전한 해결책이라고 할 수 없다.

본 논문에서는 이러한 문제점에 대해 단 한대의 카메라를 사용하여 사용자들의 경향을 통해 그 문제점을 일정 부분 보정하는 방법을 제안한다.

실제로 일반 사용자들은 핑거 트래킹 사용시에 손가락을 펴서 사용하거나 구부린다 하여도 일정각도 이상을 구부리진 않는다. 그 범위는 손가락을 구부리는 각도는 완전히 폼 때를 180° 완전히 구부렸을 때를 0° 라고 했을 때 0~180° 까지이다.

그 중 90° 부터 180° 까지는 실제로 2D 영상만을 처리하는 카메라가 핑거팁을 잡을 수 있지만 0~90° 까지는 인식할 수 없으므로 해결해야 할 영역은 0~90° 까지의 손가락을 구부렸을 때이다. 이 부분을 해결할 수 있다면 핑거팁 문제를 일정부분 해결했다고 볼 수 있다.

본 논문에서는 이를 해결하기 위하여 0~90° 사이에 손가락이 들어온다면 그 부분을 그 영역의 중심인 45° 에 있다고 가정한다. 0~90° 까지 핑거팁의 이동범위는 매우 작으므로 그것을 약 45° 에 위치한다고 예상범위를 설정하여도 오차범위가 매우 적다

그림 2 를 보면 보정 알고리즘을 적용하여 밴딩 핑거에 따른 핑거팁 문제점을 개선하였음을 볼 수 있다.



그림 2 밴딩 핑거의 문제점 보정

6. 실험 및 평가

본 논문에서 제안하는 알고리즘의 성능을 평가하기 위한 실험은 AMD 64 X2 Dual Core Processor 3800+칩이 장착된 PC 에서 수행하였다. 운영체제로는 Windows XP Professional 을 사용하였고 컴파일러로는 Visual C++ 6.0 을 사용하였다. 실험에 사용한 데이터는 320*240 의 해상도를 가지는 웹캠을 통해 입력받았다. 성능 평가를 위해 기존 알고리즘과 밴딩 핑거상태의 보정 알고리즘을 동일한 환경에서 각각 50 번을 반복적으로 수행하였다[그림 3]. 가로축은 각각의 실험횟수를 나타내고 세로축은 실험횟수에 따른 성공횟수를 나타낸다. 그림 3 에서와 같이 실험결과 밴딩 핑거상태의 기존 알고리즘을 사용하였을 시에는 정확한 핑거팁을 측정하지 못하였으나, 보정 알고리즘을 사용하였을 시에는 약 65%정도는 정확한 핑거팁을 측정할 수 있음을 알 수 있었다.

```

알고리즘 2 : 밴딩 핑거상태의 핑거팁 보정 알고리즘

declare
    FT : Finger-tip
    FL : Length of Most High Finger
begin
    Function BlobColoring()
    If Blob->MaxHeight > FL/2
        FT = Blob->MaxHeight
    Else
        FT = FL/4
end.
    
```

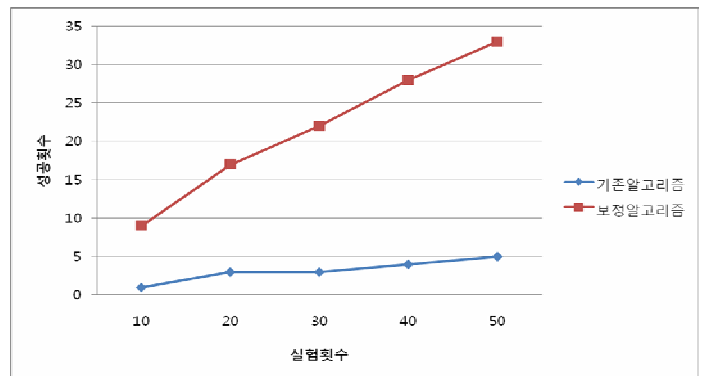


그림 3 밴딩 핑거시 기존 알고리즘과 보정 알고리즘 성능평가 비교

7. 결론 및 향후 연구 방향

본 논문에서는 밴딩 핑거 상태에서의 핑거팁 문제에 있어 위치의 오차범위가 작다는 것을 인식하고 중간값을 잡음으로써 핑거 트래킹을 보완하였다.

앞으로의 연구 방향은 연구에서와 같이 2 대 이상의 웹캠을 사용하여 보다 정확하게 핑거팁의 문제를 해결하는 방법을 찾아내는 것이다[9]. 일반 PC 에서는 그 문제가 웹캠의 위치를 잘 설정해주면 충분히 해결할 수 있지만, 핑거팁은 일반 PC 에서보다 향후 임베디드 환경에서 더 많이 사용될 것으로 보여진다. 임베디드 환경에 있어서는 일반 PC 에서보다 좀 더 제한적이다. 쉽게 말해, 웹캠의 위치만 하여도 쉽게 결정할 수 있는 문제가 아니기 때문에 여러 가지 환경제약에 따라 일반 PC 환경에서는 쉽게 해결할 수 있던 문제에 있어서도 접근하기 매우 어렵다. 때문에 그에 대한 해결방안이 제시되어야 할 것이다.

또한 본 논문에서 제안한 방법에 따른 실제 임베디드 시스템 환경에서의 보다 검증된 실험이 필요하다.

참고문헌

[1] T Brown and R Thomas, "Finger Tracking for the Digital Desk," AUIC 2000 , User Interface Conference, pp. 11-16, 2000.

[2] A. Micilotta and R. Bowden, "View-based Location and Tracking of Body Parts for Visual Interaction," CVSSP, SEPS, University of Surrey, Guildford, UK, 2004.

[3] 임옥현, 김진철, 박성미, 이배호, "블럽 컬러 링을 이용한 CT 영상에서 간 영역 자동 추출," 한국정보과학회 제 31 권, 제 2 호, pp. 760-762, 2004.

[4] S. Tsuruoka , A. Kinoshita, T. Wakabayashi, Y Miyake and M. Ishida, "Extraction of hand region and spectification of finger tips from color image," VSMM '97.Proceedings, International Conference on 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," 9th International Conference on Information Technology (ICIT'06), pp. 15-16, 2006.

[6] F. Javier Toledo-Moreo, J. Javier Martinez-Alvarez and J. Manuel Ferrandez-Vicente, "Hand-based Interface for Augmented Reality," 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines(FCCM 2007), pp. 291-292, 2007.

[7] F. Boussaid, D Chai and A. Bouzerdoum, "On-Chip Skin Detection for Color CMOS Imagers," MEMS, NSNO and Smart System, 2003 Proceedings, International Conference on 20-23 July 2003, pp. 357-361, 2003.

[8] W. Sharbek and A. Koschan, "Colour Image Segmentation - A Survey," Technical Report 94-32, Technical University of Berlin, Department of Computer Science, 1994

[9] K. Oka, Y. Sato and H. Koike, "Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems," 2002 Proceedings, Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 411-416, 2002.