

색상 조건 검사와 차영상을 이용한 레이저 포인터의 좌표 검출

이두희, 김 윤, 최창열
강원대학교 컴퓨터정보통신공학과
e-mail: getback9@kangwon.ac.kr, yooni@kangwon.ac.kr,
cychoi@kangwon.ac.kr

A Laser Pointer Detection Algorithm Based on Conditional Test in Color Model and Differential Image

Doo-Hee Lee, Yoon Kim, Chang-Yeol Choi
Dept. of Computer Science and Engineering, Kangwon National University

요 약

최근 고성능 모바일 단말기와 다양한 콘텐츠가 등장하면서 유비쿼터스 환경에서의 사용자 인터페이스에 대한 관심이 높아지고 있다. 특히 모바일 프로젝트는 장소의 제한을 받지 않고 큰 화면을 다른 사람과 공유할 수 있는 장점이 있는 반면에 단말기를 직접 제어해야 하는 불편함이 있다. 본 논문에서는 모바일 환경에서 카메라를 통해 입력된 영상 정보만으로 사용자가 스크린에 비추는 레이저 포인터를 실시간으로 검출하는 알고리즘을 제안한다. 제안하는 알고리즘은 색상 감지와 움직임 감지로 나뉜다. 단일 프레임에서 영상 성분의 평균을 이용한 조건을 검사하여 레이저 포인터 색상 영역을 검출하고, 인접한 프레임과 현재 프레임과의 차를 구하며 그 차이가 임계값보다 큰 영역을 움직임 영역으로 검출한다. 마지막으로 색상 검출 영역과 움직임 검출 영역을 동시에 만족하는 영역을 최종적으로 레이저 포인터 영역으로 인식한다. 본 기법은 영상 정보만 사용하기 때문에 센서나 불필요한 장비를 착용할 필요가 없고 영상 성분 평균을 이용하므로 프로젝터 성능에 따른 조도의 변화에 강건하여 효과적인 레이저 포인터 검출이 가능하다. 실험결과는 주변 조명의 밝기에 따라 차이가 있지만 대부분 80% 이상의 검출률과 16% 미만의 오검출률의 성능으로 나타났고, 이 같은 결과는 사용자의 주관적인 만족을 보장하였다.

Keyword: 레이저 포인터 검출(Laser Pointer Detection), 사용자 인터페이스(User Interface)

1. 서론

최근 모바일 단말기의 성능 발전과 유비쿼터스 컴퓨팅의 보편화에 따라 사용자 인터페이스에 대한 연구가 활발히 이루어지고 있다. 유비쿼터스 환경에서는 사용자가 네트워크나 컴퓨터를 의식하지 않고 장소에 상관없이 자유롭게 이용할 수 있다. 또한 사용자 인터페이스는 사용자와 시스템 사이의 의사소통이 가능하게 해주는 목적으로 만들어진 물리적, 가상적 매개체를 뜻한다. 특히 제스처 기반 사용자 인터페이스는 사용자와 미디어 콘텐츠간의 자연스러운 상호작용을 지원한다.

제스처 기반 사용자 인터페이스는 접촉식과 비접촉식으로 구분된다. 사용자가 직접 센서나 장치를 착용하는 접촉식에는 마그네틱 센서와 자이로 센서 등을 이용한 데이터 글로브[1][2]와 모션 트랙커[3] 등이 있고, 주로 영상정보를 이용하는 비접촉식에는 마커를 부착시키고 시각기술을 이용하여 추적하는 방법[3]과, LED나 적외선 등의 마커를 이용한 액티브 마커 등이 연구되었다.[4][5] 하지만 기존의 제스처 기반 사용자 인터페이스는 고가의 장치를

사용하거나 사용자가 부자연스러운 장비를 착용해야 한다는 단점이 있고, 센서를 사용할 경우 시스템과 센서간의 통신도 정의해야 한다.

최근 등장한 모바일 프로젝트는 장소의 제한을 받지 않고 원하는 장면을 다양한 공간에서 많은 사람과 공유할 수 있는 장점이 있다. 하지만 프로젝터를 이용한 큰 화면 출력에도 불구하고, 사용자가 여전히 단말기 자체를 제어하여 입력해야 하는 단점이 있다. 본 연구에서는 모바일 프로젝트에서 영사한 화면에 사용자가 레이저 포인터를 비추고 이를 카메라를 통해 입력받은 후 좌표를 검출하여 이러한 단점을 해결하는 기법을 제안한다. 관련 연구로는, 카메라 추적 기반의 레이저 포인터 시스템이 최초로 제안되었지만[6] 영상처리 부분은 구체적으로 다루지 않았다. [7]에서는 레이저 포인터의 검출과정을 두 단계로 나누는데 가장 밝은 R값을 찾고 구역을 지정하여 좌표와 구역의 상관관계를 이용하였다. 이 경우는 스크린에 비추고 있는 대상 중에서 R값이 높을 경우 오검출이 우려되고 조도와 광원에 강건하지 못할 수 있다. 또한 최근의 연구 [8]은

팬/틸트/줌 카메라를 이용하여 레이저 포인터를 추적하는 시스템으로 3D 공간의 특징을 이용하고, 자체 제작된 카메라 시스템을 사용하기 때문에 일반적인 실험 환경으로 보기 힘들다.

본 논문에서는 모바일 프로젝터에 장착된 카메라를 통해 촬영되는 순수한 영상 정보만으로 사용자의 레이저 포인터를 검출하는 알고리즘을 제안한다. 먼저 색상 검출 정보와 움직임 검출 정보를 이용하여 이 두 가지 검출을 모두 만족하는 픽셀의 좌표를 사용자 제스처로 판단한다. 색상 검출은 단일 프레임에서 RGB 성분의 평균을 구하고 각 픽셀단위로 평균값과 여러 조건을 검사하여 레이저 포인터 여부를 판단한다. 또 움직임 검출은 인접한 프레임과 현재 영상과의 차이를 구하고 임계값으로 설정된 범위를 벗어난 영역을 움직임 영역으로 판단한다. 최종적인 레이저 포인터 검출 판단은 색상 검출 영역과 움직임 검출 영역에 모두 만족한 경우이다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 알고리즘을 기술하고 3장에서는 실험 결과를 통해 그 성능을 확인하며 최종적으로 4장에서 결론을 맺는다.

2. 제안하는 알고리즘

2.1 색상 검출

본 논문에서는 실제 사용자가 붉은색 레이저 포인터를 사용한다고 가정하였다. 일반적인 사용자 마커와는 달리 레이저 포인터의 경우 빛을 스크린에 실제로 쏘는 형태이기 때문에 사용자가 눈으로 관찰하는 경우와 카메라를 통해 입력되는 영상과는 많은 차이가 있다. 영상은 카메라를 통해 입력된 레이저 포인터의 영상 가운데 부분이 흰색에 가까운 픽셀을 포함하는 특징을 가진다.

2.1.1 평균 성분 값 계산

카메라로 촬영된 입력 영상은 RGB컬러로 시스템에 전달된다. 레이저 포인터 검출을 위해서는 단일 프레임에서의 RGB 각각의 평균 성분 값을 구하며 그 식은 (1)과 같다.

$$\begin{aligned} R_{mean} &= \frac{1}{K} \sum_{i=1}^w \sum_{j=1}^h R(i,j), \\ G_{mean} &= \frac{1}{K} \sum_{i=1}^w \sum_{j=1}^h G(i,j), \\ B_{mean} &= \frac{1}{K} \sum_{i=1}^w \sum_{j=1}^h B(i,j). \end{aligned} \quad (1)$$

여기서 K 는 단일 프레임에서 모든 픽셀의 개수이며 (i,j) 는 프레임에서의 픽셀 위치를 나타낸다.

2.1.2 조건 검사

평균 성분 값과 각각의 픽셀의 성분 값을 이용하여 조건을 검사한다. 조건 검사는 RGB 컬러 공간에서 레이저

포인터 내부 픽셀은 대체로 흰색에 가까우므로 다른 RGB 평균 성분 값보다 크를 이용한다. 또한 주변 조명이나 조도에 따라 입력되는 레이저 포인터의 RGB 수치는 다양하게 변할 수 있으므로 영상 평균과 비교한다. 이 조건식은 (2)와 같이 정의된다.

$$F(x,y) = \begin{cases} 1, & \text{if } R(x,y) > R_{mean}, \\ & G(x,y) > G_{mean}, \\ & B(x,y) > B_{mean} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

식 (2)의 조건에서 각각의 픽셀에 대해 평균 화소 값과 비교하여 다른 성분보다 밝은 지를 검사한다. $F(x,y)$ 는 이진영상으로 레이저 포인터 후보로 판별되는 픽셀이며 $R(x,y)$, $G(x,y)$, $B(x,y)$ 는 화소의 R, G, B 값을 각각 나타낸다. 이와 같은 조건검사를 수행하는 이유는 전체적으로 어둡거나 밝은 영상이 입력되었을 때 레이저 포인터도 상대적으로 RGB 화소 값의 증감이 있는데 이러한 변화에도 적용하기 위함이다.

또한 입력 영상의 패턴을 분석한 결과, 레이저 포인터의 가장자리 픽셀이 순서대로 R, B, G의 순서로 분포하는 특징이 있었다. 이러한 특징을 이용하면 잡음을 효과적으로 제거할 수 있으며, 이와 같은 조건식은 (3)과 같이 정의된다.

$$F'(x,y) = \begin{cases} 1, & \text{if } R(x,y) > B(x,y) > G(x,y), \\ & F(x,y) = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

식 (3)에서 얻어진 레이저 포인터 색상 후보는 실제 레이저 포인터의 일부분만 검출할 가능성이 있으므로 3×3 Dilation을 적용하여 잡음으로 오인할 수 있는 가능성을 방지한다. 식 (4)는 이와 같은 조건식을 나타내며, 여기서 \oplus 는 Dilation을 나타내는 기호이며 SE 는 3×3 Structuring Element를 말한다.

$$P(x,y) = F'(x,y) \oplus SE. \quad (4)$$

2.2 움직임 검출

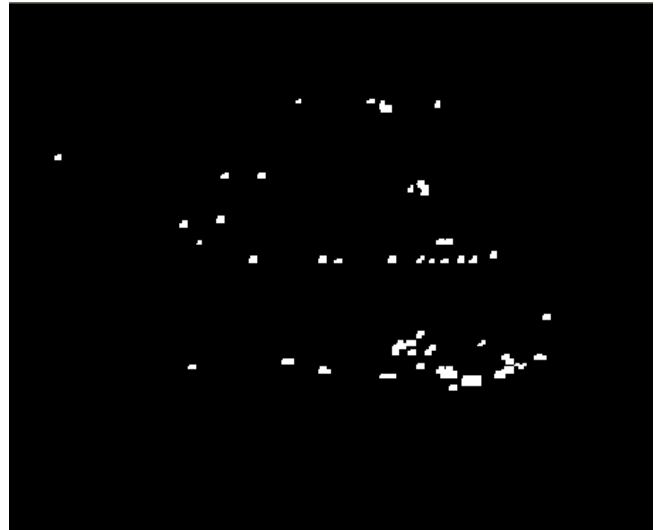
움직임 검출은 이전 영상과 현재 영상과의 차를 이용하며, 식 (5)와 같다.

$$M(x,y) = \begin{cases} 1, & \text{if } |Cur(x,y) - Ref(x,y)| > thr \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

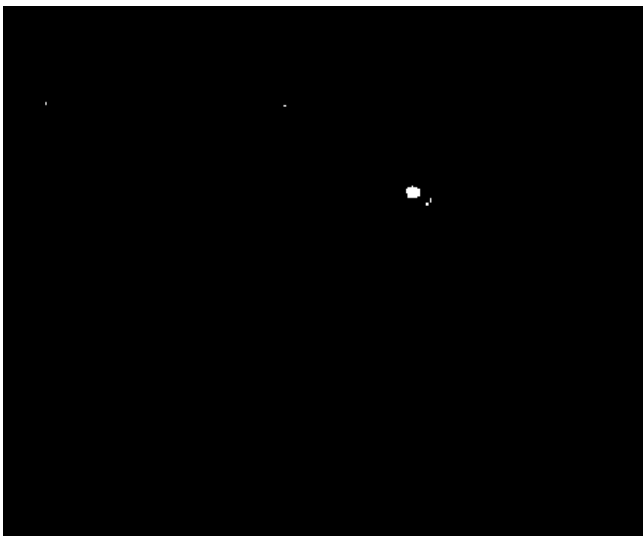
여기서 $Cur(x,y)$ 와 $Ref(x,y)$ 는 현재 프레임과 이전 프레임의 RGB 평균값을 나타낸다. 또한 thr 는 임계치로써 이전 프레임과 현재 프레임의 RGB 평균값의 차이를 의미하며 검출의 민감도를 조절하는 역할을 한다. 실험을 통해 임계치는 30으로 설정하였다. 이 조건 검사는 고정적



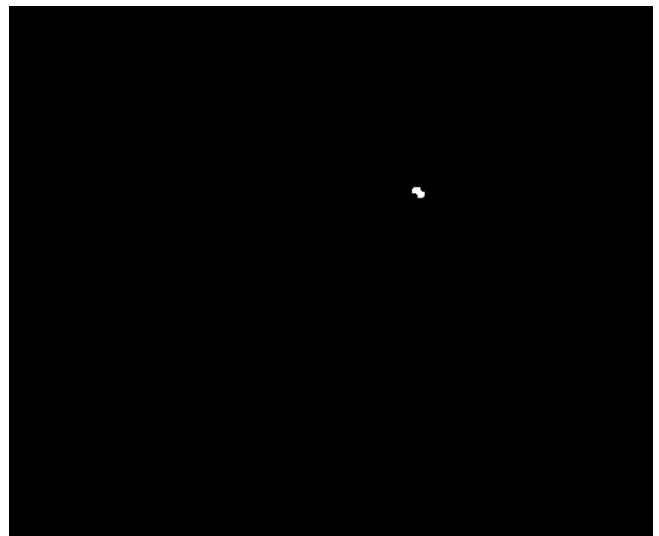
(a) 입력 영상의 예.



(b) 색상 검출 영상 $P(x,y)$ 의 예



(c) 움직임 영상 $M(x,y)$ 의 예.



(d) 결과 영상 $Pointer(x,y)$ 의 예

그림 1. 레이저 포인터 검출 예.

Fig. 1. Example of Laser Pointer Detection Image.

으로 위치하지 않는 레이저 포인터의 특성을 반영하며 움직임 발생 픽셀은 1, 그 외 픽셀은 0으로 이진화한다. 여기서 0은 배경 영역, 1은 움직임 영역이다.

2.3 최종 레이저 포인터 검출

최종 레이저 포인터 검출은 입력 영상 중에서 움직임 영역과 색상 영역이 모두 1이고, 이 좌표의 주변 픽셀이 경계값 이상의 크기일 때 최종 좌표로 인식하게 되며 그 식은 (6)과 같다.

$$Pointer(x,y) = \begin{cases} 1, & \text{if } P(x,y)=1, \\ & M(x,y)=1. \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

식 (4)에서 구한 $P(x,y)$ 와 식 (5)에서 구한 $M(x,y)$ 의

값이 모두 1을 만족하는 $Pointer(x,y)$ 를 레이저 포인터가 위치한 최종 좌표로 인식하고, 모든 일련의 과정은 그림 1과 같다. 그림 1의 (a)는 카메라에 입력되는 영상이다. 그림 1의 (b)와 (c)는 각각 입력된 영상에서 색상 검출과 움직임 검출 결과를 나타내며 최종적으로 포인터로 인식하는 영역은 그림 1의 (d)와 같다.

3. 실험 결과

영상의 레이저 포인터 발생 유무를 다르게 하여 설정하고 해당 알고리즘이 얼마만큼의 프레임을 검출 혹은 오검출 하는지를 실험하였다.

표 1. 실험에 사용한 조명의 구성 목록.

Table 1. Lighting list using in an experiment

항목	어두움	보통	밝음
RGB 평균	131~140	141~150	151~160
프레임 수	1000	1000	1000

테스트에는 320x240의 크기와 초당 15프레임을 갖는 영상을 입력하였고 Microsoft의 VX3000 USB Webcam을 사용하였다. 표 1은 실험 환경의 조명에 대한 정보를 나타낸다. 표 1에서 나타나듯이 전체 프레임에 대한 RGB평균값을 이용하여 조명 환경을 변화시킴으로써 프로젝트의 성능에 따라 존재할 수 있는 전체적인 밝기의 차이를 고려하였고 각 조명마다 5회씩 실험하고 평균을 구하였다. 실험 결과는 그림 2와 같다.

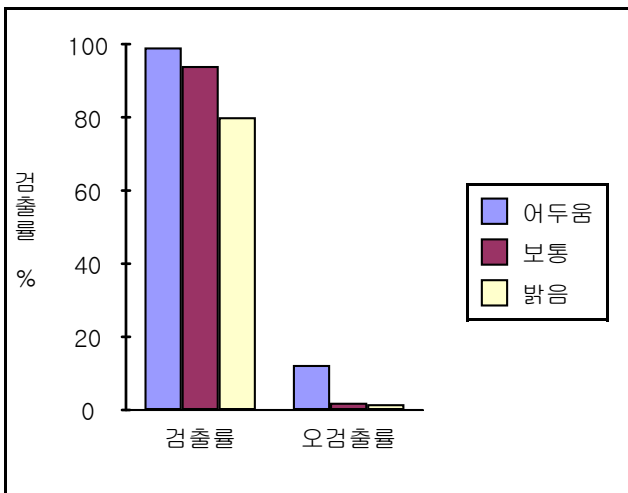


그림 2. 조명에 따른 검출 성능 비교.

Fig. 2. Performance Compare about Light.

그림 2에서 나타나듯이 조명의 밝기에 상관없이 레이저 포인터를 인식하는 검출률은 80%이상의 성능을 유지했고, 레이저 포인터가 없을 때 오검출하는 경우도 16% 미만이었다. 특히 실험을 진행하면서 빛의 반사와 영사되는 화면 상태에 따라 많은 성능 차이가 있었지만 대부분의 경우에는 사용자가 주관적인 만족을 느낄 만한 성능을 보장했다.

3. 결론

본 논문에서는 RGB 컬러 모델에서의 평균 성분과 인접한 프레임과의 차영상을 이용하는 방법으로 색상과 움직임 검출하였다. 특히 색상 감지 알고리즘은 영사되는 스크린에 따른 조도의 변화에 강건하고, 움직임 검출 또한 이전 프레임과의 변화량을 고려하기 때문에 새롭게 변화하는 구간을 효과적으로 검출한다. 위와 같은 레이저 포인터 검출 알고리즘을 통해 사용자가 사용하는 레이저 포인

터 좌표를 실시간으로 탐지 가능하며, 이를 통해 유비쿼터스 컴퓨팅에 상응하는 다양한 사용자 인터페이스를 기대할 수 있다. 향후에는 다양한 환경에서도 높은 검출률을 제공하기 위한 검출 알고리즘의 개발을 추후 연구를 통해 제시하고자 한다.

참고문헌

- [1] D. J. Sturman and D. Zeltzer, "A Survey of Glove-based Input," IEEE Computer Graphics and Applications, Vol. 14, No. 1, pp. 30-3, 1994.
- [2] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A Hand Gesture Interface Device," in Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface, pp. 189-192, 1987.
- [3] G. B. Guerra-Filho, "Optical Motion Capture: Theory and Implementation," Journal of Theoretical and Applied Informatics (RITA), Vol. 12, No. 2, pp. 61-89, 2005.
- [4] K. Kurihara, S. Hoshino, K. Yamane, and Y. Nakamura, "Optical Motion Capture System with Pan-Tilt Camera Tracking and Realtime Data Processing," in Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Vol. 2, pp. 1241-1248, 2002.
- [5] H. Chen, G. Qian, and J. James, "An Autonomous Dance Scoring System using Marker-based Motion Capture," in Proceedings of IEEE 7th Workshop on Multimedia Signal Processing, pp. 1-4, 2005.
- [6] C. Kirstein and H. Müller, "Interaction with a Projection Screen Using a Camera-tracked Laser Pointer", in Proceedings of The International Conference on Multimedia Modeling, pp.191-192, 1998.
- [7] Dan R. Olsen Jr. and Travis Nielsen, "Laser Pointer Interaction," CHI 2001, Volume No. 3, Issue No. 2001.
- [8] Kurz, D., Häntsch, F., Grosse, M., Schiewe, A. and Bimber, O., "Laser Pointer Tracking in Projector-Augmented Architectural Environments," IEEE International Symposium on Mixed and Augmented Reality (ISMAR'07), pp. 19-26, 2007.