

색상 및 거리 영상에서의 얼굴검출 기반 손 제스처 인식

전훈기, 고재필
 금오공과대학교 컴퓨터공학과
 e-mail: qlfm85@gmail.com, nonzero@kumoh.ac.kr

Face Detection-based Hand Gesture Recognition in Color and Depth Images

Hun-Ki Jeon, Jaepil Ko
 Dept. of Computer Engineering, Kumoh National Institute of Technology

요 약

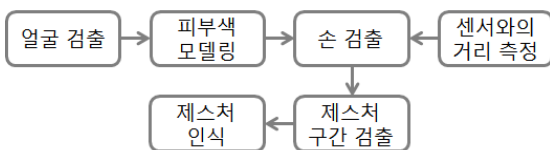
본 논문에서는 얼굴검출을 통한 실시간 피부색 모델링과 거리정보를 결합하여 손 영역을 검출하고 손 움직임에 따른 방향 및 원 제스처 인식을 위한 규칙 기반 인식방법을 제안한다. 기존과는 달리 손 좌표를 사용하는 대신 기존 프레임과 현재 프레임에서의 손 좌표 차이를 이용하여 제스처 구간을 설정하고 자연스러운 제스처 동작에서의 속도변화를 고려할 수 있도록 한다. 실험 데이터는 5명을 대상으로 4방향과 원을 포함하여 총 5가지 제스처를 10회씩 실행하여 획득하였다. 이들 데이터에 대한 인식 실험에서 97%의 인식률을 보였다.

1. 서론

손 제스처 인식은 컴퓨터 비전 분야에서 활발하게 연구되고 있는 분야이다. 최근 Microsoft사의 Kinect가 제공하는 실시간 거리 정보를 이용한 연구가 진행되고 있다.

기존의 손 제스처를 이용한 입력 인터페이스들[1][2]이 많이 연구되고 있지만, 손 영역의 검출하여 손의 모양을 인식하는 포스처 기반 연구가 대부분이다. 또한, 손 움직임을 추적하여 제스처를 인식하는 좌표 기반 방법들[3][4]도 많이 연구되었지만, 대부분 손바닥이나 특정 손모양의 움직임을 한정적으로 추적하고 HMM(Hidden Markov Model)[5]을 이용해 인식한다. 2D 카메라를 이용한 피부색 모델을 기반으로 손 영역을 검출하여 제스처를 인식하는 방법도 있지만 이는 조명에 민감하고 배경과 피부색이 비슷할 경우 손 영역의 검출이 어렵다는 단점이 있다.

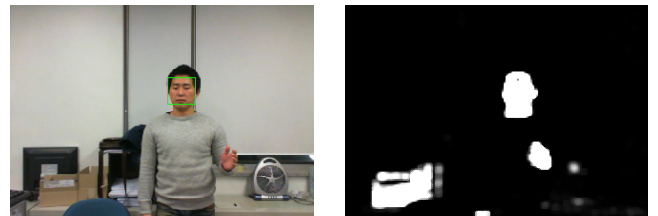
본 논문에서 제안하는 방법은 2D 카메라를 이용한 방법 중 피부색 모델과 거리정보를 결합한 방법이다. 얼굴 검출을 통하여 피부색 검출 및 모델링을 하며, 손 영역 검출을 위해 피부색 모델과 거리정보를 이용한다. 거리 정보의 획득은 Kinect[6]를 이용하며, 획득한 거리 정보를 손 검출에 이용하여 제스처 인식을 한다.



(그림 1) 피부색 모델과 거리 정보를 이용한 제스처 인식

2. 피부색 모델링을 위한 얼굴 검출

본 논문에서는 피부색 모델을 생성하기 위해 얼굴 검출을 이용하였다. 피부색 모델 기반의 방법은 조명에 민감하다는 단점이 있는데 이를 줄이기 위해 제스처 사용 시점의 피부색을 이용하여 모델을 생성하기 위해 얼굴 영역을 검출한다. 얼굴 검출은 Haar-like Features[7]를 이용하였다. 얼굴 검출 영역은 피부색만 있는 것이 아니라 눈썹과 입술과 같은 피부색에 비해 너무 밝거나 어두운 영역도 포함되어 있다. 이러한 피부색 이외의 색을 얼굴 영역에서 제외시킨 후 YCbCr 칼라 모델로 변환하여 피부색 모델을 생성한다. 피부색 모델링에는 Cb, Cr 두 컬러를 사용하며 GMM(Gaussian Mixture Model)[8]을 사용하여 피부색 모델을 생성한다. (그림 2)는 Haar-like Features를 이용한 얼굴 검출 결과 영상을 보여준다.



(그림 2) 얼굴 검출 및 피부색 검출 (a) 얼굴 검출 (b) 피부색 검출

3. 피부색 모델과 거리정보를 이용한 손 검출

얼굴 검출을 통해 생성된 피부색 모델을 이용하면 피

부색 영역을 효과적으로 검출할 수 있으나, (그림 2)에서 보듯이 피부색 모델만으로는 배경에 나타난 피부색 영역을 손 영역과 구분할 수 없다. Kinect는 거리뿐만 아니라 관절의 위치도 제공한다. 그러나 이는 관절위치검출 연산 시간으로 인해 프레임 처리 속도를 떨어뜨려 실시간 처리에는 적합하지 않으므로 본 논문에서는 관절위치 정보사용은 배제하고 거리정보만을 사용한다.

사람이 제스처를 할 경우 손이 얼굴보다 앞에서 움직인다는 가정 하에 거리정보를 이용하여 아래 수식과 같이 손 영역을 정의한다.

$$\begin{cases} \text{손 영역} & \text{if } 0 < D_{face} - D_{face} \times t \\ \text{배경} & \text{else} \end{cases} \quad (2)$$

여기서, D_{face} 는 Kinect 센서와 얼굴 간의 거리를 나타내며 실험을 통해 t 값을 0.1로 사용하여 손 영역을 추출하였다.

손의 중심점을 추출하기 위해 검출된 손 영역을 이진화 하여 2차원의 축에 대하여 각각 P_x , P_y 로 프로젝션을 실시한 후 손의 중심점 x' , y' 을 아래의 수식과 같이 구한다.

$$x' = \sum_{x=1}^w (P_y(x))x / N \quad (3)$$

$$y' = \sum_{y=1}^h (P_x(y))y / N \quad (4)$$

여기서, w , h 는 이미지의 가로, 세로 크기이며 N 은 각 x , y 축으로 프로젝션 시킨 좌표축의 합이다.

4. 제스처 인식

제스처 인식을 위해서 움직인 구간 설정은 매우 중요한 부분이며 이 구간을 어떻게 설정하느냐에 따라 인식률에 큰 변화가 생긴다. 자연스러운 제스처 동작 시 프레임 간의 손의 움직임 차이는 증가하다가 감소한다고 정의한다. 손의 움직임 차이는 이전 프레임과 현재 프레임의 손 영역의 중심 좌표를 가지고 계산한다. 획득된 좌표 차이 값을 이용해 제스처의 시작과 끝을 판별하고, 제스처의 인식 정보로 활용한다.

4.1 움직임 구간 설정

손의 움직임이 발생할 경우 크기가 N 인 큐에 이전 프레임의 손 중심 좌표와 현재 프레임의 손 중심 좌표의 차이 값을 삽입한다. 큐에 모인 좌표의 차이 값이 증가하여 일정 이상의 분산이 되면 제스처의 시작으로 간주하고, 움직임이 줄어들어 분산이 일정 이하가 되면 제스처의 끝으로 간주한다. 이때, 제스처 시작과 끝 사이의 좌표 차이 목록을 다음과 같이 β_x , β_y 라 정의 하고 이를 대상으로 제스

처 인식을 수행한다.

$$\beta_x = (x_1, x_2, \dots, x_n) \quad (5)$$

$$\beta_y = (y_1, y_2, \dots, y_n) \quad (6)$$

여기서, n 은 손이 움직인 구간의 프레임 수다.

4.2 위, 아래, 왼쪽, 오른쪽 인식

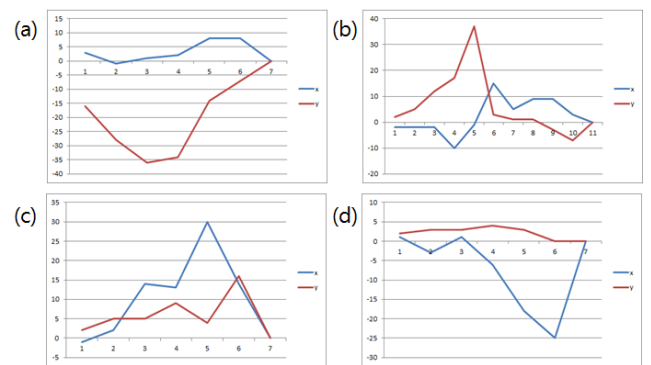
위, 아래, 왼쪽, 오른쪽 인식은 손의 중심점 좌표의 차이 값을 이용한다. 손의 움직임의 가로축과 세로축의 β_x , β_y 의 분산의 비율을 계산하여 판별한다. (7)의 수식은 위, 아래, (8)의 수식은 왼쪽, 오른쪽 판별에 사용된다.

$$\sigma^2(\beta_x) \geq \theta_1 \quad \text{AND} \quad (\sigma^2(\beta_x) / \sigma^2(\beta_y)) \geq \theta_2 \quad (7)$$

$$\sigma^2(\beta_y) \geq \theta_1 \quad \text{AND} \quad (\sigma^2(\beta_y) / \sigma^2(\beta_x)) \geq \theta_2 \quad (8)$$

여기서, $\sigma^2(\beta_x)$ 와 $\sigma^2(\beta_y)$ 는 각 축에 대한 분산을 나타내고, θ_1 , θ_2 는 분산 비율의 임계값을 나타낸다. 수식 (7), (8)을 각각 만족할 경우 각 α_x , β_y 집합의 전체 좌표 합이 부호가 +이면 아래(7), 왼쪽(8)로 인식하고 -이면 위(7), 오른쪽(8)으로 인식한다.

(그림 3)은 제스처 별 x , y 좌표의 차이 값을 그래프로 나타낸 것이다. 파란색이 x , 붉은색이 y 를 나타낸 것인데 (a), (b)에서는 y 좌표의 차이 값의 변화가 심하고, (c), (d)에서는 x 좌표의 차이 값의 변화가 심하다는 것을 확인할 수 있다.



(그림 3) 제스처 별 좌표 차이 값 그래프 (a)위 제스처 (b)아래 제스처 (c) 오른쪽 제스처 (d)왼쪽 제스처

4.3 원 인식

원 인식은 제스처 구간의 프레임 수를 n 이라 할 때, n 이 일정 값 이상일 때 인식한다. 원 인식의 경우 좌표 차이를 사용할 수 없으므로 손 중심 좌표 값을 그대로 사용한다. 반지름 정보를 아래와 같이 구해 수식(11)을 만족하면 원으로 인식한다.

$$x_c = \frac{\sum_{i=1}^n x_i}{n}, y_c = \frac{\sum_{i=1}^n y_i}{n} \quad (9)$$

$$R = \left\{ \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2}, \dots, \sqrt{(x_n - x_c)^2 + (y_n - y_c)^2} \right\} \quad (10)$$

$$\bar{R} \geq 30 \text{ AND } \sigma_R^2 < 15.0 \quad (11)$$

여기서, x, y 는 각각의 좌표 값을 나타낸다.

5. 실험 및 결과

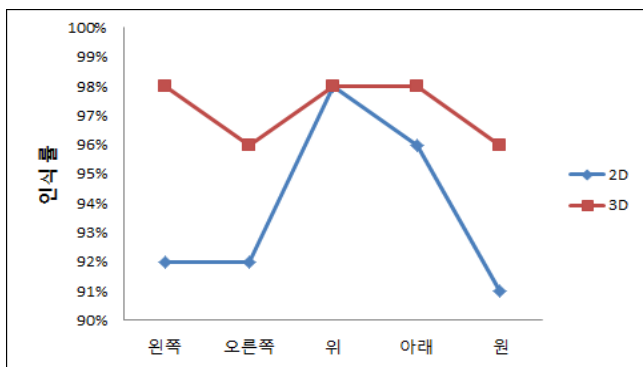
실험은 Windows 7 환경의 Intel Core2 Duo 3.0GHz, 메모리 3G, Kinect와 Visual C++ 2010에 OpenCV 2.2를 설치하여 진행하였다. 총 5명의 실험자를 대상으로 5가지 제스처를 각각 10회씩 실험하여 결과를 얻었고, 인식률은 97%로 나타났다. <표 1>은 실험자 별 제스처 인식 수와 인식률을 나타내고 있다.

<표 1> 실험자별 제스처 인식 수 및 인식률

	왼쪽	오른쪽	위	아래	원
실험자1	10	10	9	10	10
실험자2	10	9	10	10	9
실험자3	9	10	10	10	10
실험자4	10	10	10	9	9
실험자5	10	9	10	10	10
인식률	98%	96%	98%	98%	96%

여기서 실험자별 차이는 크게 없는 것으로 보인다.

비교를 위해 거리 정보를 사용하지 않고 2D 카메라를 이용하여 동일한 조건에서 Microsoft LifeCam Cinema를 사용하여 실험하였다. (그림 4)는 제스처별 2D와 3D 카메라의 인식률을 비교한 그래프이다.



(그림 4) 2D 및 3D 카메라를 사용한 인식률

여기서, 2D 카메라를 이용한 방법은 위 제스처를 제외한 나머지 제스처들은 3D 카메라를 이용한 방법보다 낮은 인식률을 보였고, 전체 인식률은 94%로 3D 카메라를 이용한 방법보다 3%정도 낮다.

6. 결론

본 논문에서는 기존 방법들의 문제점 해결을 위해 열 굴 검출을 통해 피부색 모델과 거리정보를 이용해 손 검출을 하였다. 원 제스처 인식은 손 중심점 좌표를 이용하였고, 방향 제스처는 기존과는 달리 손 중심점 좌표 차이 값을 이용하여 자연스러운 제스처 동작에서의 속도변화를 고려하였다. 실험에서는 손 좌표 차이 값을 이용한 방법이 복잡한 학습 대신 간단한 규칙만으로 제스처를 인식할 수 있음을 보였다. 거리정보를 이용한 방법은 기존 피부색 모델만을 이용한 방법보다 높은 인식률을 보였다.

Acknowledgment

본 논문은 중소기업청에서 지원하는 2012년도 산학연공동 기술개발사업(No.C0033001)의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

- [1] K. Oka, Y. Sato, and H. Koike, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems," Proc. of the 5th Intl. Conf. on AFGR, pp 429-434, 2002.
- [2] J. Pablo Wachs, M. Kolsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," Communications of the ACM, Vol. 54, Issue 2, pp.60-71, 2011.
- [3] Zhong Yang, Yi Li, Weidong Chen, and Yang Zheng, "Dynamic hand gesture recognition using hidden markov models," Proc. of the 7th Intl. Conf. on IEEE, pp.360-365, 2012.
- [4] Xiaoyan Wang, Ming Xia, Huiwen Cai, Yong Gao, and Carlo Cattani, "Hidden-markov-models-based dynamic hand gesture recognition," Mathematical Problems in Engineering, Vol. 2012, 2012.
- [5] Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang, Toshiaki Ejima, "Hand gesture recognition using hidden Markov models," IEEE Intl. Conf. on Systems, Man, and Cybernetics, Vol.5, pp.4232-4235, 1997.
- [6] Wikipedia. Kinect-Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/Kinect>, 2011.
- [7] P. Viola and M. Jones, "Robust real-time object detection," Intl. J. of Computer Vision, Vol. 57, no.2, pp. 137-154, 2004.
- [8] Haque, M., Murshed, M. and Paul, M, "On Stable Dynamic Background Generation Technique Using Gaussian Mixture Models for Robust Object Detection", Proc. of 5th IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance, pp. 41-48, 2008.