

깊이 센서를 이용한 손 재활 시스템
 박혜란 이동우 김만배
 강원대학교 IT대학 컴퓨터정보통신공학과
 hyeran08@kangwon.ac.kr

Hand Rehabilitation System Using a Depth Sensor
 Hyeran Park, Dongwoo Lee and Manbae Kim
 Dept. of Computer and Communications Engineering, Kangwon National University

요약

별개의 컨트롤러 없이 사용자의 신체만을 이용하여 다양한 게임과 엔터테인먼트를 경험할 수 있는 키넥트에 대한 관심이 높아지고 있다. 최근 재활 치료를 필요로 하는 환자가 늘어남에 따라 본 논문에서는 운동 장애를 가진 환자들이나 노인들이 고가의 장비 또는 다른 사람의 도움 없이 키넥트와 컴퓨터만을 이용하여 손 재활 운동을 할 수 있도록 하는 것을 목적으로 한다. 키넥트 영상으로부터 손 영역을 찾고, 영역의 윤곽선을 추출 한다. 이 때 손가락 중심점을 찾아 손가락과 손바닥 영역을 구분해 준다. 손가락의 개수를 확인하기 위해서 손의 중심점과 끝 점을 찾은 후 두 점을 연결함으로써 손가락의 개수를 확인할 수 있고, 실시간으로 손의 움직임을 감지하도록 한다.

1. 서론

키보드나 마우스 등의 복잡하고 간접적인 제어 장치에서 몸동작과 손동작 등의 단순하면서도 직관적인 제어 방법으로서의 연구가 점차 증가하는 추세이다 [1, 2]. 이는 일반 사용자에게 현실감과 몰입감을 주어 편리함과 흥미를 줄 수 있으며 어린이와 노인에게는 직관적인 제어기기 사용법으로 좀 더 쉽게 장치를 제어할 수 있게 한다. 키넥트는 컨트롤러 없이 이용자의 신체를 이용하여 게임과 엔터테인먼트를 경험할 수 있는 깊이 센서로, 사용하기 매우 쉽고 새로운 사용자 인터페이스를 제공한다 [3]. 최근 키넥트 열풍 속에 다양한 방면으로의 활용이 각광받고 있다. 따라서 본 논문에서는 나날이 증가되어 가고 있는 운동 장애 환자들이나 노인들의 재활을 위해 고가의 장비나 다른 사람의 도움 없이 현재 많이 보급 되고 있는 키넥트를 이용함으로써 실시간으로 손의 움직임 동작을 검출한 후에, 손가락 개수를 구하는 기법을 제안한다.

본 논문의 구성은 다음과 같다. 서론에 이어 2절에서는 전체적인 흐름도를 소개한다. 3절에서 손가락 개수 검출방법, 4절에서는 실험결과를 설명하고 5절에서 결론을 맺는다.

2. 전체 흐름도

그림 1은 키넥트 영상으로부터 손 영역을 찾아 손가락의 개수를 찾아내는 시스템의 전체 흐름을 보여주는 블록도이다. 키넥트에서 획득한 깊이 영상으로부터 손 영역을 검출한다. 검출한 손 영역의 윤곽선을 찾아 낸 후 손의 중심점과 손 끝 점을 추출한다. 추출 된 중심점과 손 끝점을 연결한 선을 구하고, 이로부터 손가락의 개수를 계산한다.

3. 제안 방법

3.1 손 영역 detection

키넥트 깊이 영상에서 전경 객체는 color로, 배경은 gray scale로 인식되어진다. 따라서 영상에서 color 영역만 추출 한 뒤, 임계값 T_1 를

주어 (예: $T_1=70$) T_1 이하인 영역은 픽셀값 0으로 바꾸어 손 영역을 분리하였다. 그림 2는 키넥트 영상으로부터 추출된 손 영역을 보여준다. 또한 손 영역 픽셀은 임의의 상수값을 할당하였다.

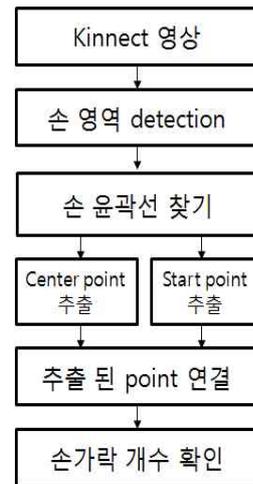


그림 1. 제안 방법의 전체 흐름도



(a) 키넥트 영상 (b) 추출된 손 영역
 그림 2. 키넥트 영상으로부터 손 영역 추출

3.2 손 윤곽선 찾기

손 영역을 더욱 효과적으로 보이기 위해 영역의 윤곽선을 추출한다. 손 영역과 다른 배경의 픽셀 값을 비교하여 픽셀 값이

변하는 픽셀 값을 255로 변환한다. 영상의 픽셀 좌표를 (x_i, y_i) 라 하면, 수평 및 수직으로 스캔하면서 윤곽선 픽셀을 탐색한다. 즉, (x_{i-1}, y_i) 의 픽셀 값과 (x_i, y_i) 값의 픽셀 값을 비교하고 값이 같지 않을 때 윤곽선 처리를 해 준다. 이 때 더욱 정확한 윤곽선을 나타 내 주기 위해 (x_i, y_{i-1}) 과 (x_i, y_i) 의 픽셀 값도 비교하여 같은 방식으로 처리하여 준다.



그림 3. 추출된 손의 윤곽선 (white로 표시됨)

손가락 영역과 손바닥 영역을 구분하기 위해 손가락 영역에서의 각 손가락 사이에 중심선을 그려 주었다. 손가락의 수평길이와 손바닥의 수평길이는 차이가 있기 때문에 임계값 T_2 를 주어 T_2 이하인 영역은 손가락으로 간주하였다.



그림 4. 손가락 중심선 추출 영상

그림 4에서 수평선에서 찾아지는 윤곽선 포인트의 좌측 경계픽셀이 a이고, 우측 경계픽셀이 b이면, a와 b 사이의 거리 S는 다음과 같이 계산된다.

$$S = \|b - a\| \quad (1)$$

3.3 중심점과 손 끝점 추출

손가락의 개수를 확인하기 위해서는 손의 중심점과 손 끝점을 찾아 이어 주는 것이 필요하다. 따라서 추출 된 손 영역에서 중심점과 손 끝점을 찾는다. 3.2절에서 추출된 윤곽선의 x와 y좌표 값을 산술 평균하여 중심점 $C=(M_x, M_y)$ 를 구한다.

$$M_x = \frac{1}{N} \sum_{i=1}^N x_i \quad M_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (2)$$

또한 추출된 손가락 중심선을 기준으로 각 손가락마다 제일 처음으로 detection 되는 중심 픽셀의 좌표를 손 끝점으로 지정한다.

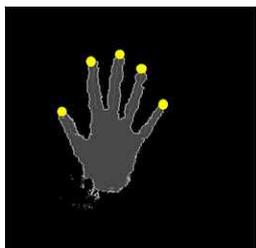


그림 5. 손 끝, 중심 point 추출

3.4 손가락 개수 계산

손 끝점이 정확한 위치에 존재하지 않을 경우 손가락의 개수에서 오류가 발생한다. 예를 들어 그림 6의 원은 잘못 구해진 손 끝점을 보여준다.

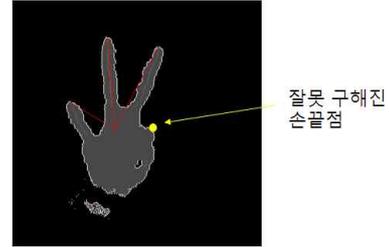


그림 6. 잘못 구해진 손 끝점

따라서 이 오류를 줄이기 위하여 추출된 중심점과 손 끝점의 거리의 표준 편차를 구하여 특정 범위 내에 존재하는 선의 개수만 이용한다.

$$\bar{d} = \frac{1}{N} \sum_{i=1}^N d_i \quad (3)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (d_i - \bar{d})^2}{N}} \quad (4)$$

여기서 d_i 는 중심점과 손 끝점의 거리, \bar{d} 은 d_i 의 평균, σ 는 d_i 의 표준편차이다. 또한 다음 조건식을 만족하는 선을 사용한다.

$$\bar{d} - \lambda\sigma \leq d_i \leq \bar{d} + \lambda\sigma \quad (5)$$

일반적으로 λ 는 confidential interval (CI)을 가리키며, 신뢰도 측정에 사용된다. 실험에서는 $\lambda=2$ 을 사용하였다.

4. 실험결과

키넥트 기기에 손을 인식 시킨 후 손가락의 개수를 달리하면 손의 중심점과 손 끝점이 인식 된다. 찾아진 점들 사이에 선을 그려 현재 인식 되고 있는 손가락의 개수가 몇 개인지 알 수 있도록 하였다. 따라서 손 운동이 필요한 사용자가 키넥트 기기 앞에서 손동작을 달리 하면 손가락이 몇 개 펴져 있는지 알아냄으로써 효과적인 운동을 할 수 있도록 도와준다. 제안방법의 손가락 검출 결과 영상은 그림 7과 같다.

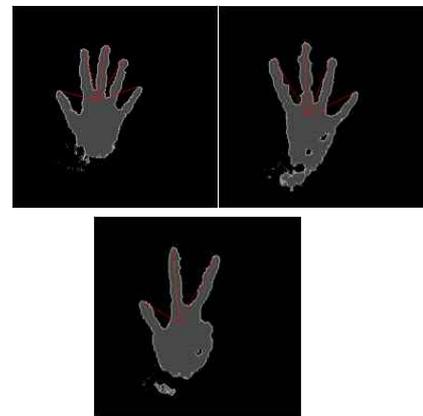


그림 7. 중심점과 손 끝점이 연결된 영상

5. 결론

본 논문에서는 깊이 센서를 이용한 손 재할 시스템으로 손가락의

개수를 확인하는 기법을 제안하였다. 색상과 동작의 깊이를 인식할 수 있는 키넥트의 센서 렌즈(Color and depth-sensing lenses)를 통해 얻은 영상에서 손 영역을 추출한 후 영역의 외곽선을 검출하였고, 중심점과 손 끝점을 이용하여 연결 한 뒤 손가락의 개수를 확인하였다. 실험 결과 손 영역 이외 다른 영역이 조금씩 검출 되었다. 따라서 향후 더 완벽한 손 재할 시스템을 얻을 수 있는 알고리즘의 연구가 필요하다.

감사의 글

본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행되었음

참고문헌

- [1] Wang, R, Popovic,J, “Real-Time Hand-Tracking with a Color Glove,” ACM Trans. Graph., vol. 28, issue 3, no. 63, Aug. 2009.
- [2] R. Kjeldsen, J. Kender, “Toward the use of gesture in traditional user interfaces,” Second IEEE International Conference on Automatic Face and Gesture Recognition, pp.151-156, Oct. 1996.
- [3] Jeffrey L. Wilson, “Microsoft kinect for Xbox360,“ PCMag.com, Retrieved Nov. 2010.
- [4] 최윤철, 임순범 공저, 컴퓨터그래픽스 배우터, 2008년. 생능출판사