

거리 그래프를 이용한 손가락 인식

송지우, 허훈, 오정수

부경대학교 융합디스플레이공학과

Finger Recognition using Distance Graph

Ji-woo Song, Heo Hoon, and Jeong-su Oh

Department of Display Engineering, Pukyong National University

E-mail : sowooji@pukyong.ac.kr, heohoon@pknu.ac.kr, ojs@pknu.ac.kr

요 약

본 논문은 depth image에서 검출된 손가락 경계의 거리 그래프 이용하여 손가락을 인식하는 알고리즘을 제안한다. 거리 그래프는 손바닥 중심과 손 윤곽선 사이의 각도와 유클리디안 거리를 각각 x, y축으로 하여 보여준다. 그래프가 손끝에서 국부 최대점을 갖는다는 사실을 이용하여 손 형태를 얻을 수 있다. 손가락이 손바닥보다 얇은 것을 이용하여 손목의 무게 중심을 찾은 후에, 손목의 무게 중심의 각도를 기준 각도 0° 로 설정하였다. 모의실험 결과는 제안된 알고리즘이 손의 방향에 영향을 받지 않으며 손가락 상태를 보다 잘 검출할 수 있는 것을 보여준다.

ABSTRACT

This paper proposes an algorithm recognizing finger using a distance graph of a detected finger's contour in a depth image. The distance graph shows angles and Euclidean distances between the center of palm and the hand contour as x and y axis respectively. We can obtain hand gestures from the graph using the fact that the graph has local maximum at the positions of finger tips. After we find the center of mass of the wrist using the fingers is thinner than the palm, we make its angle the orienting angle 0° . The simulation results show that the proposed algorithm can detect hand gestures well regardless of the hand direction.

키워드

Kinect, Depth Image, 손가락 인식, 거리 그래프

1. 서 론

최근에 컴퓨터가 사람의 제스처를 인식하고 그에 따른 다양한 동작을 수행하는 컴퓨터-인간 상호작용 알고리즘들에 대해 연구되고 있는 중이다. 특히, 손은 쉽게 다룰 수 있는 신체 부위라는 점에서 손동작을 의사 표현의 중요한 수단으로 사용되고 있다. 일반적으로 손 인식 시스템을 구현할 때 피부색을 이용하는 방법을 주로 사용하는데, 전신이 모두 촬영되는 상황에서 피부색만으로 손을 다루는 것은 신체 부위와 구별하는 건 어려울 뿐만 아니라 조명과 같은 외부 환경에 따라 잡음이 발생하면 이를 처리하기가 힘들다.

Kinect는 Microsoft사에서 개발한 부가적인 컨트롤러 없이 사람의 동작을 읽을 수 있는 주변기기이다. Kinect의 Depth Sensor는 반사된 Laser Beam Pointer로 각 픽셀간 거리를 측정하

여 Kinect 앞의 사용자가 얼마나 떨어져 있는지를 인식할 수 있게 해준다. 또한 Kinect SDK (Software Development Kit)는 인체를 구성하는 주요 골격 20 개에 대한 위치 데이터를 제공해주기 때문에, 해당 관절 포인트를 이용해 신체 부위의 정보를 쉽게 얻을 수 있다. 본 논문에서는 손의 위치를 찾기 위해 Kinect의 기능을 활용한다.

손 제스처 인식을 위한 연구는 다양하게 진행되어왔다. 손의 중심에서 원을 그리는 방법은 손가락의 개수만 셀 수 있으며, 손끝의 좌표 정보나 몇 번째 손가락인지에 대한 정보는 알 수 없다[1]. 또한 손의 크기가 달라지면 오차가 생긴다. Template matching을 이용한 방법은 크기나 각도 변화에 민감하여 조금만 달라도 인식이 매우 저조해진다[2]. convexity defects를 이용한 방법은 손목의 유무가 손가락을 인식하는데 많은 영향을 끼친다[3]. 즉 손목을 제거하지 않으면 정확도가 떨어진다. 본 논문에서는 손의 중

심에서 윤곽선까지의 각도와 거리를 각각 x , y 축으로 거리 그래프를 그려 손의 방향에 영향을 받지 않고 손가락을 검출하는 방법을 제시한다. 또한 그래프의 기준 각도를 손목의 각도로 지정하여 손목의 유무와 손의 방향에 상관없이 손가락을 검출할 수 있음을 보인다.

II. 본 론

Microsoft사의 개발 툴인 Kinect SDK에서는 사람의 신체를 추상화해서 총 20 개의 골격 위치를 제공한다. 이 중 HAND point를 이용하면 깊이 영상에서 손에 해당하는 영역을 쉽게 얻을 수 있다. 피부색을 이용해서 손을 인식하는 방법에 비해 부가적인 연산이 필요하지 않아 빠르고, 적외선 Sensor를 이용하기 때문에 조명의 변화와는 상관이 없어 잡음에 영향을 받지 않는다.

검출된 손 영역에서 손의 중심을 찾기 위해 거리 변환(distance transform)을 이용한다. 거리 변환이란 영상의 한 화소에서 가장 가까이 위치한 물체의 경계선까지 거리를 구하는 연산이다. 손바닥 중심이 손의 경계선에서 가장 멀리 떨어져 있기 때문에 거리 변환 행렬 상에서 가장 큰 값을 가진다.

그림 1(a)는 거리가 큰 값일수록 밝게 보이도록 변환 행렬 범위를 [0, 255]로 정규화 하였다. 그림 1(b)는 구해진 손바닥 좌표와 길이를 표시한 것이다.

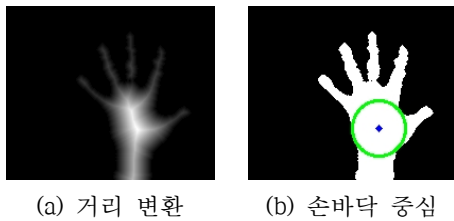


그림 1. 손의 중심 검출

검출된 손 영역의 윤곽선과 손바닥 좌표를 각각 (contour.x, contour.y), (center.x, center.y)라 하면 윤곽선과 손 중심과의 각도는 식 (1), (2)를 이용해서 식 (3)과 같이 구할 수 있다.

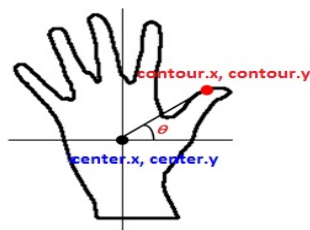


그림 2. 손 중심과 윤곽선의 각도 및 길이

$$X = \text{contour.x} - \text{center.x} \quad (1)$$

$$Y = \text{contour.y} - \text{center.y} \quad (2)$$

$$\theta = \tan^{-1} \frac{Y}{X} \quad (3)$$

마찬가지로 윤곽선에서 손 중심까지의 거리 를 식 (4)에서처럼 구한다.

$$\text{dist} = \left| \sqrt{X^2 + Y^2} \right| \quad (4)$$

그림 3(b)는 그림 3(a)에 대응하는 거리 그래프 프로파일 x 축은 $0 \sim 360^\circ$ 이다. 하지만 이 그래프를 그대로 이용하면 두 가지 문제가 발생한다. 첫째로 이 알고리즘에서는 그래프 상에서 손끝이 국부 최대점을 갖는다는 특징을 이용해서 손가락을 인식하는데, 손목이 존재하면 그림 3(b)에서 보이듯이 손가락뿐만 아니라 손목에서도 국부 최대값을 갖기 때문에 손가락과 구별하기가 어렵다.

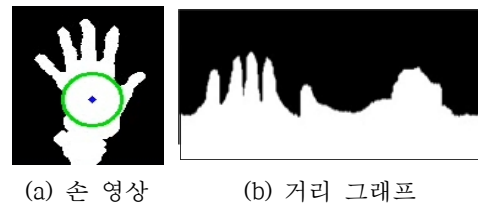


그림 3. 손 영상에 대한 거리 그래프

두 번째로 그림 4에서처럼 손이 회전하면 손가락의 분포 위치가 달라지는데, 만약 하나의 손가락이 그래프의 처음과 끝부분으로 나뉘기면 손가락 개수보다 더 많은 수의 국부 최대값이 검출된다. 이러한 문제점을 해결하기 위해서 손목의 위치를 탐색하는데, Morphology와 convex hull 알고리즘을 이용한다.

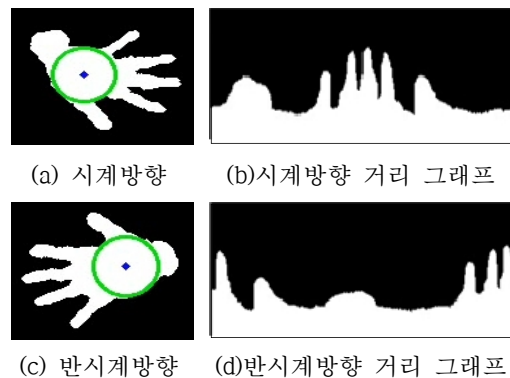


그림 4. 손의 회전에 따른 거리 그래프

손가락은 손바닥에 비해 얇기 때문에 Morphology open 연산을 반복 수행하면 그림 5의 (b)처럼 손바닥 부분만 남는다. 원본 영상에서 손바닥만 남은 영상을 빼면 손가락만을 검출할 수 있다. Morphology 과정 중에 생긴 잡음은 Labeling을 이용하여 제거한다. 마지막으로 그림 5의 (c)와 같이 거리 변환을 통해 구해놓았던 손바닥 부분과 손가락 영상을 더하여 새로운 손을 생성한다.

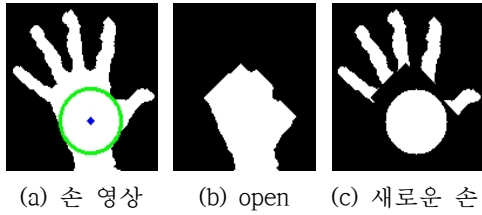


그림 5. 새로운 손을 그리는 과정

최종적으로 손목이 제거된 새로운 손을 생성하기 위해 convex hull 알고리즘을 이용한다. convex hull 알고리즘은 외곽선을 찾는 알고리즘으로 2차원 평면에 N개의 점이 주어졌을 때, 이들 중 몇 개의 점을 골라 이어서 나머지 모든 점을 내부에 포함하는 다각형을 만드는 방법이다. 그림 6(a)는 그림 5(c)에 convex hull 알고리즘을 수행한 결과로 손가락 영역과 손바닥 영역을 포함하는 불록 겹칠 영상이다. 그림 6(a)의 불록 겹칠 영상과 그림 6(b)의 원 영상을 AND 연산 시키면 그림 6(c)와 같은 손목이 제거된 영상을 얻을 수 있다. 손가락 두께가 달라서 open 연산 과정 중 완전히 손가락이 제거되지 않더라도 불록 겹칠은 최외각을 연결하기 때문에 완전한 손의 모양을 얻을 수 있다.

같은 원리로 원래 손 영상인 그림 6(b)에서 손목 부분을 제거한 그림 6(c)를 빼면 손목 부분만 남는다. 추출된 손목 영상의 무게 중심을 계산하면 쉽게 손목 좌표를 얻을 수 있고 이를 기준 각도 0°로 설정하여 거리 그래프를 그리면 손의 방향이 바뀌더라도 손목 위치에 따라 재설정되기 때문에 거리 그래프 상에는 큰 변화가 없다.



그림 6. convex hull을 통한 손목 제거

그림 7은 손의 방향에 따른 거리 그래프를 보여주고 있다. 정 방향 손에 대한 그림 7(b) 거리 그래프와 거의 180° 회전된 손에 대한 그림 7(d)는 유사한 모습을 보여주고 있다. 손목 꺾인 정도와 손가락 펼침 정도에 따라 그래프의 모습은 조금씩 달라지지만 손가락의 순서와 최대점이 분명히 유지되고 있어 손가락을 인식하는 것에는 영향을 주지 않는다.

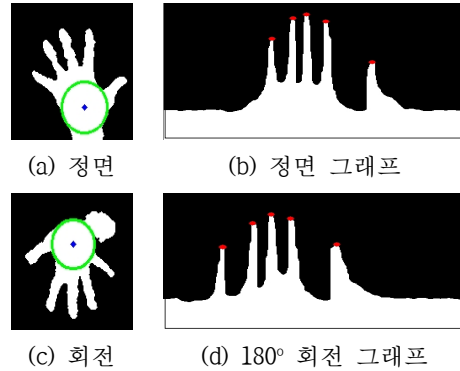


그림 7. 손의 방향에 따른 거리 그래프

III. 실험 결과

제안된 거리 그래프를 이용한 손가락 인식 알고리즘을 평가하기 위해 펼친 손가락 수에 따른 손가락 인식과 손의 방향에 따른 손가락 인식을 표 1과 2에 보여주고 있다.

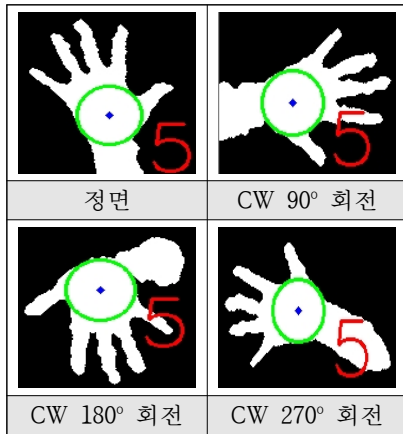
표 1은 거리 그래프를 이용하여 펼친 손가락의 개수를 인식한 결과를 보여주고 있다. 손목이 있음에도 불구하고 개선된 알고리즘을 통해 정확한 손가락의 개수를 인식하고 있음을 확인할 수 있다.

표 2는 거리 그래프를 이용하여 손이 회전함에 따른 손가락 인식을 보여준다. 마찬가지로 손이 회전하여도 정확하게 손가락의 개수를 인식하고 있음을 확인할 수 있다.

표 1. 손목이 존재할 때 손가락 인식

5개	4개	3개
2개	1개	0개

표 2. 회전에 따른 손가락 인식



IV. 결 론

본 논문에서는 거리 그래프를 통해 손가락을 인식하는 방법을 제시하였다. convex hull 알고리즘을 이용하여 손의 크기가 달라져도 어떤 사용자든지 정확하게 손을 검출할 수 있다. 손목의 위치를 탐색하여 그 위치를 기준으로 거리 그래프를 그림으로써 손의 방향이 바뀌어도 손가락의 상태를 잘 인식할 수 있다. 또한 손목이 있어도 손을 인식하는 데 영향을 주지 않는다.

제안된 알고리즘을 사용하면 손가락 개수뿐만 아니라 손끝의 좌표 또한 얻을 수 있기 때문에 향후에는 이 점을 이용하여 더 나은 컴퓨터-인간 상호작용 시스템을 만들 수 있을 것이다.

참고문헌

- [1] S. J. Hoon, "Finger Counting Using Computer Vision," *The Korean Institute of Communications and Information Sciences*, pp. 657-658 Jan. 2013.
- [2] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*, Wiley, 2009.
- [3] In-kyu Choi, Jisang Yoo, "Hand Shape recognition based on geometric feature using the convex-hull", *한국정보통신학회 논문지*, Vol. 18, pp. 1931-1940, Aug. 2014.
- [4] Hanhoon Park, Junyeong Choi, Jong-Il Park, and Kwang-Seok Moon, "A study on Hand Region detection for kinect-based hand shape recognition", *한국방송공학회 논문지*, pp. 393-400, May. 2013.
- [5] <https://developer.microsoft.com/en-us/windows/kinect>
- [6] Minkyu Lee and Jaebong Jeon, "Personal Computer control using kinect", *한국정보과학회 학술발표논문집*, pp. 343-345, 2012.