

궤적의 방향 변화 분석에 의한 제스처 인식 알고리즘

박장현*, 김민수#

Gesture Recognition Algorithm by Analyzing Direction Change of Trajectory

Jahng-Hyon Park* and Minsoo Kim#

ABSTRACT

There is a necessity for the communication between intelligent robots and human beings because of wide spread use of them. Gesture recognition is currently being studied in regards to better conversing. On the basis of previous research, however, the gesture recognition algorithms appear to require not only complicated algorithms but also separate training process for high recognition rates. This study suggests a gesture recognition algorithm based on computer vision system, which is relatively simple and more efficient in recognizing various human gestures. After tracing the hand gesture using a marker, direction changes of the gesture trajectory were analyzed to determine the simple gesture code that has minimal information to recognize. A map is developed to recognize the gestures that can be expressed with different gesture codes. Through the use of numerical and geometrical trajectory, the advantages and disadvantages of the suggested algorithm was determined.

Key Words : Direction Change(방향 변화), Gesture Code(제스처 코드), Gesture Map(제스처 맵), Gesture Recognition Algorithm(제스처 인식 알고리즘), Trajectory(궤적)

1. 서론

로봇 관련 기술이 눈부시게 발전하면서 지능형 로봇들이 하나 둘씩 우리 생활에 등장하고 있다. 아직까지는 많이 보편화 되지 않았지만 엔터테인먼트용 토이 로봇과 가사용 청소 로봇을 비롯하여 앞으로 다양한 목적을 가진 지능형 로봇의 출현이 전망된다. 이에 따라 사람과 로봇 사이의 자연스러운 의사 전달 방법을 구현하기 위해 많은 연구가 진행되고 있으며 주로 사람과 사람 사이의 가장 대표적인 의사 전달 방식인 음성 인식과 제

스처 인식 분야의 연구가 대부분을 차지한다.

제스처 인식 시스템은 인식을 위해 글러브 센서로부터 얻어진 센서의 출력 값을 이용하는 방법과 시각 기반의 데이터 획득 장치로부터 얻어진 시각 정보를 이용하는 방법으로 나눌 수 있다. 그러나 글러브 센서의 출력 값을 이용할 경우,¹ 반드시 글러브를 착용 하여야 하며 센서와 컴퓨터를 연결하기 위한 선이 필요하여 사용자에게 불편함을 준다. 이러한 문제점은 컴퓨터 시각 정보를 이용하여 해소될 수 있다.² 시각 정보를 이용하기 위하여 Starner 와 Pentland³ 는 색 정보를 이용하여

접수일: 2004년 2월 26일; 게재승인일: 2005년 1월 21일

* 교신저자: 한양대학교 기계공학부

E-mail jpark@hanyang.ac.kr Tel. (02) 2220-0448

한양대학교 대학원 기계설계학과

손을 추적하고, 손의 x, y 좌표, 최소 관성축과의 각도, 경계선 타원의 이심률로 구성된 8 개의 특성 벡터를 추출해 은닉 마르코프 모델(Hidden Markov Model)로 제스처를 인식하였으며, Min⁴ 은 8 방향 코드를 입력으로 하여 은닉 마르코프 모델로 12 개의 동적 손 제스처를 인식하였다. 제스처 인식의 연구와 비슷한 온라인 문자 인식 분야에서는 흘러 쓰기와 단어 단위 입력을 허용하는 등 필기의 자유도를 높이거나 인식률의 향상을 위해 통계적인 모델, 뉴럴 및 퍼지 모델 등의 복잡한 알고리즘으로 제안되었다^{5,6}.

이러한 일련의 연구들은 인식률에서는 어느 정도 만족한 결과를 얻을 수 있으나 알고리즘이 복잡하여 둘 이상의 궤적을 동시에 이용한 제스처의 인식이 어려우며 인식을 위한 트레이닝 과정을 따로 수행해야 하는 문제점이 있다.

본 논문에서는 인식에 필요한 최소한의 정보만을 이용하여 복잡한 제스처를 간단하게 표현할 수 있는 제스처 코드와 제스처 인식을 위한 별도의 훈련과정이 필요 없는 제스처 맵을 제안하여 간단하면서도 다양한 종류의 제스처를 효과적으로 인식할 수 있는 알고리즘을 개발하고자 한다. 그리고 실험을 통하여 본 알고리즘의 우수성과 문제점을 파악하고자 한다.

본 논문의 구성은 2 장에서 제안된 제스처 인식 알고리즘에 대하여 자세히 살펴보고, 3 장에서는 실험 결과를 보여주고 알고리즘에 대한 고찰을 해본다. 결론 및 향후 과제는 4 장에서 제시한다.

2. 제스처 인식 알고리즘

본 논문에서 제안하는 제스처 인식 알고리즘은 손 부분의 인식을 위해 착용한 마커인 원을 인식하고 원 중심의 궤적을 저장하는 부분과 저장된 궤적의 방향 변화를 분석하여 제스처를 인식하는 부분으로 구성된다. 이 과정은 Fig. 1 에서 자세히 보여주고 있다.

원의 궤적을 저장하는 부분에서는 화소당 256 개의 명암도를 갖는 입력된 영상에 임계값(Threshold)을 적용하여 영상을 이치화한 후 노이즈 제거를 위한 필터링을 거쳐 영역별 넓이와 둘레 길이를 계산하여 원형도를 판별함으로써 원을 인식한다⁷. 손 부분의 인식을 위해 원이 표시된

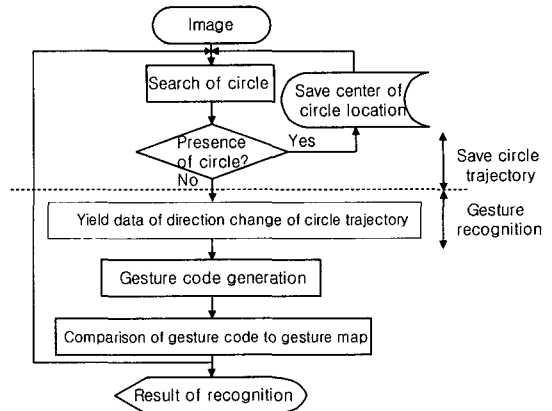


Fig. 1 Gesture recognition algorithm

마커를 이용한 이유는 본 연구에 사용한 CCD 카메라와 비전 보드가 Gray Scale 이므로 손을 인식하는 알고리즘⁸에서 큰 부분을 차지하는 피부의 색깔 정보를 사용할 수 없기 때문이며, 색깔 정보를 이용할 수 있는 CCD 카메라와 비전 보드를 이용한다면 이 부분은 손을 인식하는 다른 알고리즘으로 대체할 수 있다.

제스처를 인식하는 부분은 궤적의 방향 변화를 분석하여 제스처를 본 논문에서 제안한 제스처 코드로 표현하고 이를 본 논문에서 제안한 제스처 맵에서 검색한 후 입력된 제스처를 인식한다.

2.1 제스처 코드

제스처의 인식을 위해서는 생성된 제스처 궤적과 인식의 기준이 되는 제스처 궤적과의 비교과정이 필요하다. 하지만 사용자마다 제스처를 표현하는 속도가 다르며 카메라와 사용자의 거리에 따라서 동일하게 표현한 제스처라도 영상에서의 제스처 궤적 크기가 같지 않다. 따라서 궤적의 속도와 크기 정보를 제외한 방향 변화 정보만을 이용하여 제스처를 인식할 수 있다면 위와 같은 문제점을 해결할 수 있다.

Fig. 2 에서처럼 제스처의 궤적은 방향성을 갖는 곡선과 직선의 조합으로 이루어져 있다. 제스처의 속도와 크기 정보를 제외한다면 제스처 궤적의 구성요소인 곡선과 직선을 표현하기 위한 최소의 정보는 곡선은 출발방향, 도착방향, 회전방향, 회전 수이며 직선은 직선방향이다. 따라서 제스처를 구성요소인 곡선과 직선으로 나누고 순서대로 구성요소의 최소 정보들을 나열하면 복잡한

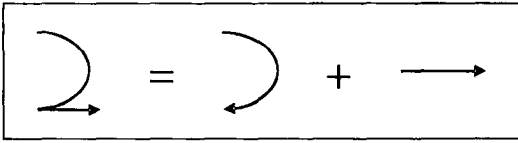


Fig. 2 Elements of gesture trajectory

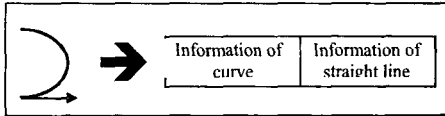


Fig. 3 Concept of gesture code

제스처도 간단한 코드로 나타낼 수 있다. 이를 그림으로 나타내면 Fig. 3 과 같이 표현할 수 있다.

Table 1 은 이상적인 제스처 궤적의 몇 가지 구성요소와 각각의 방향 변화를 보여준다. 방향 변화의 그래프는 $i-1$ 번째 프레임에서 i 번째 프레임으로 제스처 궤적이 이동한 방향 벡터를 삼각함수를 이용하여 각도로 나타낸 그래프이다. 반 시계방향의 곡선은 방향 변화 그래프가 지속적으로 증가하며 시계방향의 곡선은 지속적으로 감소한다. 그리고 직선은 방향이 일정하므로 방향의 변화가 없다. 따라서 방향 변화 그래프의 미분치를 이용하면 제스처 궤적이 어떤 방향의 곡선들과 직선들로 구성되어 있는지 판단할 수 있고 이를 알고리즘으로 구현하는 것이 용이해 진다.

하지만 실제 제스처 궤적은 사용자의 입력 여건상 떨림이 발생한다. 이로 인하여 제스처 궤적 방향 변화의 미분치 분석이 용이하지 않다. Fig. 4 와 Fig. 5 는 원점이 출발점인 실제 제스처 궤적의 예와 각도로 표시한 방향 변화 그래프이다. 이러한 문제점을 없애기 위해 각도로 표현된 제스처 궤적의 방향 변화 그래프를 Fig. 6 의 16 방향의 라벨로 정규화 시키면 Fig. 7 과 같이 개선된 방향 변화 그래프를 얻을 수 있다. Fig. 7 에서 원으로 표시된 부분은 궤적의 방향이 각 라벨의 경계일 때 발생하며 i 번째 프레임의 방향 라벨과 $i-1$ 번째 프레임의 방향 라벨의 차인 방향 라벨 변화의 미분치가 연속해서 -1 에서 1 로, 또는 그 반대로 변하는 부분을 0 으로 수정함으로써 평탄화 시킬 수 있다.

Fig. 8 은 Fig. 4 의 제스처 궤적으로부터 앞에서 설명한 라벨링과 평탄화의 전처리 과정을 거친 제스처 궤적 방향 변화 그래프와 방향 변화의 미분

Table 1 Example of elements and direction changes for the ideal gesture trajectory

Type	Trajectory	Direction changes
Curve		
Straight line		

치 그래프이다. 이들 데이터로부터 제스처를 제스처 코드로 나타내기 위한 정보를 구한다.

먼저 방향 라벨 변화의 미분치가 3 이상이 되는 부분을 제스처의 구성요소의 영역을 나타내는 Stage의 경계로 정한다. 그 이유는 곡선의 경우 제스처 입력속도가 빠를 때 방향 라벨이 두 단계까지 변하기 때문이다. 그리고 사용자가 제스처를 입력할 때 방향이 바뀌는 부분에서는 제스처의 속도가 떨어져 의도하지 않은 방향 라벨이 생성될 수 있기 때문에 Stage 경계에서 좌우로 2 개의 방향 데이터는 제외한다. 그다음은 나누어진 Stage 마다 미분치 합 of 절대값을 계산한 후 라벨 경계부분의 궤적을 고려하여 1 이하이면 직선, 2 이상이면 곡선으로 판단한다.

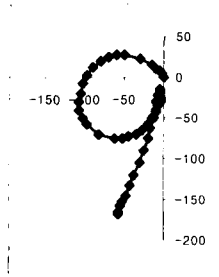


Fig. 4 Trajectory of real gesture

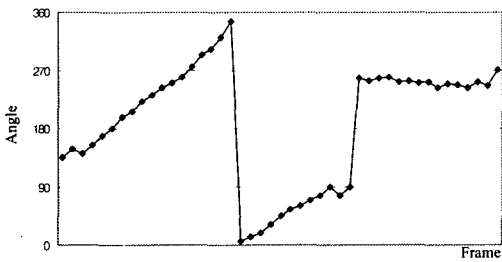


Fig. 5 Direction change of real gesture

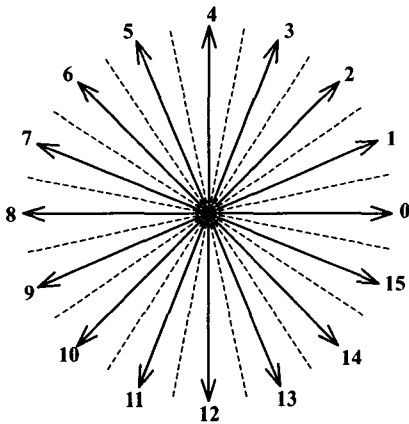


Fig. 6 Label of 16 direction

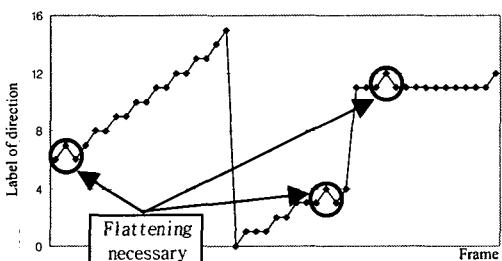


Fig. 7 Direction change graph after labeling

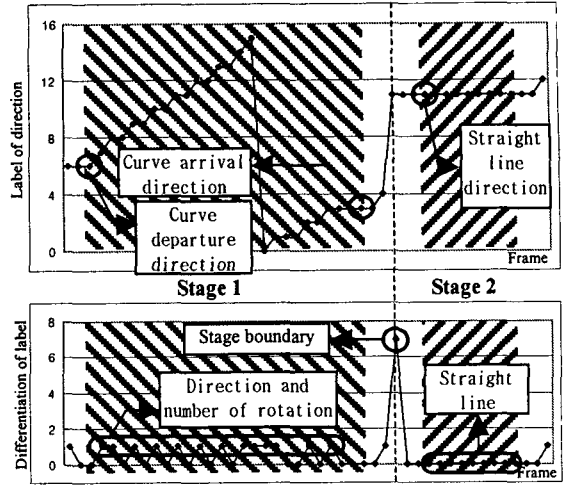


Fig. 8 Direction change graph after flattening (above) and differentiation of the direction change (below)

곡선으로 판단된 Stage의 경우 제스처가 숫자 '8'과 같이 제스처의 회전방향이 바뀌는 궤적을 가져 서로 다른 방향의 곡선이 같은 Stage로 판별되므로 양수 패턴에서 음수 패턴으로 바뀌거나 혹은 그 반대의 경우를 판별하여 추가로 Stage구분을 한다.

곡선 Stage의 경우 출발방향은 첫 번째 라벨 값을, 도착방향은 마지막 라벨 값을 선택하며 제스처의 회전방향은 미분치가 양수패턴이면 반 시계방향으로, 음수패턴이면 시계방향으로 판단한다. 회전수는 미분치의 합의 절대값을 구하여 16으로 나눈 값의 정수 부분을 취한다. 직선 Stage의 경우 직선 방향은 가장 많은 빈도수를 가지는 라벨로 정한다.

이러한 제스처 구성요소들의 정보들을 Stage마다 4개의 Cell을 할당하여 Fig. 3의 제스처 코드화 개념과 같이 나열하면 Fig. 9와 같이 제스처를 간단한 제스처 코드로 나타낼 수 있다. 제스처 코드의 첫 번째 Cell은 제스처가 몇 개의 Stage로 구성되어 있는지를 나타내는 Index이며 다음 절에서 설명할 제스처 맵에서의 빠른 검색을 위해 사용된다. 제스처 코드에서 곡선 Stage의 경우 Cell마다 순서대로 출발방향 라벨, 도착방향 라벨, 회전방향, 회전 수의 정보가 위치하며 회전방향의 경우 반 시계방향은 '1'로, 시계방향은 '-1'로 표시한다. 직선 Stage의 경우 첫 번째 Cell에는 직선 방향 라벨을, 세 번째 Cell에는 직선 Stage임을 나타내는 '0'을 표시하며 두 번째와 네 번째 Cell은 사용하지 않는다.

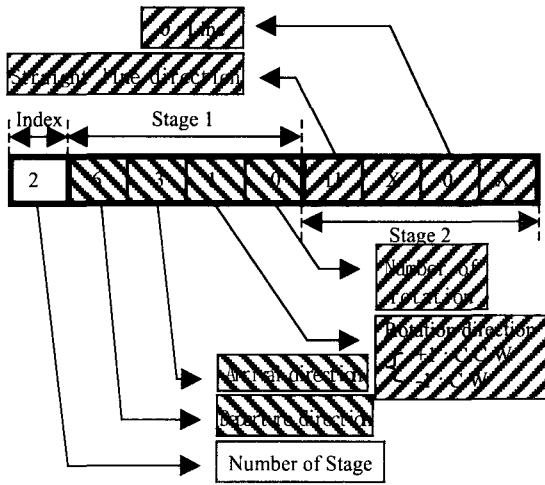


Fig. 9 Structure of gesture code

2.2 제스처 맵

제스처 입력의 여건상 사용자가 제스처를 항상 동일한 궤적으로 표현하지 않기 때문에 같은 제스처라도 조금씩 다르게 표현될 수 있다. 궤적의 작은 차이는 제스처 코드화의 전처리 과정인 라벨링에 의해 교정될 수 있지만 여기에는 한계가 있다. 더욱이 사용자가 다를 경우에는 같은 제스처라도 궤적의 차이가 확연하며 이로 인하여 다른 제스처 코드가 생성된다. Fig. 10 은 동일한 제스처라도 사용자에게 따라 제스처 코드가 다른 경우를 보여준다.

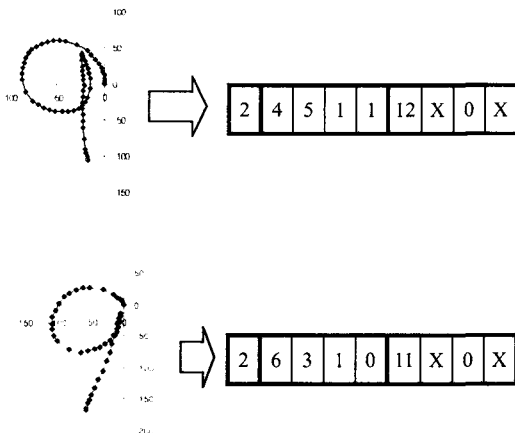


Fig. 10 Different gesture codes from same gesture

따라서 제스처 인식의 기준이 되는 제스처 맵은 이런 경우를 고려하여 다르게 표현될 수 있는 동일한 제스처의 여러 제스처 코드를 포함할 수 있는 정보를 가져야 하며 빠른 검색을 위해 간단한 구조로 구성되어야 한다. 이러한 조건들을 만족시키기 위해 본 논문에서 제안한 제스처 맵의 구조는 Fig. 11 과 같다. 제스처 맵의 제일 처음 Cell 은 제스처 코드와 마찬가지로 제스처가 몇 개의 Stage 로 구성되어 있는지를 나타내는 Index 이며 그 다음 Cell 부터 Stage 당 6 개의 Cell 이 차례대로 할당되어 있다. 곡선의 경우 순서대로 출발 방향 라벨의 최소값, 최대값, 도착방향 라벨의 최소값, 최대값, 회전방향, 회전 수가 나열되어 있으며 직선의 경우 직선방향 라벨의 최소값, 최대값이 첫 번째와 두 번째 Cell 에, 직선임을 나타내는 '0' 값이 다섯 번째 Cell 에 나열되어 있고 나머지 Cell 은 사용하지 않는다.

최종적인 제스처의 인식은 입력된 제스처 궤적으로부터 제스처 코드를 생성한 후, 같은 Index 를 가진 제스처 맵을 검색하여 조건이 일치하면 해당하는 제스처 맵의 인식 결과를 보여주는 과정으로 이루어진다.

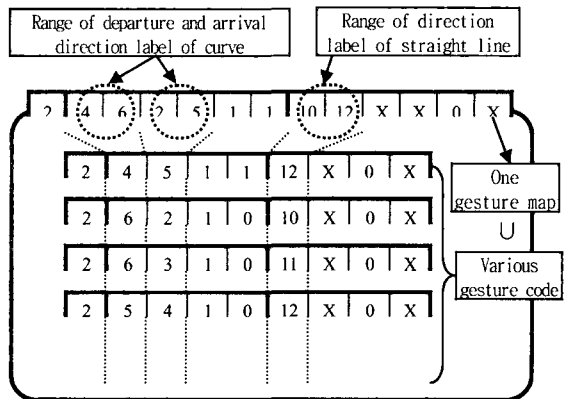


Fig. 11 Structure of gesture map

3. 실험 및 결과

3.1 실험 장치의 구성

본 논문에서 제시한 알고리즘은 Pentium 4 2.4GHz CPU 와 512MB Memory 사양의 IBM 호환 PC 에서 구현되었으며 영상입력을 위한 하드웨어 환경으로는 Sentech 사의 STC-1100 CCD 카메라와 Coreco 사의 PC-Vision 보드를 사용하였다. 알고리

들 구현을 위한 소프트웨어 환경으로는 Microsoft Visual C++ 6.0 을 사용하였다. Fig. 12 는 데이터 흐름을 자세히 보여준다.

실험에 사용된 제스처 영상의 크기는 480x360 화소이며, 각 화소는 256 개의 명암도를 가진다. PC-Vision 보드는 CCD 카메라로부터 입력되는 영상을 초당 30 프레임의 속도로 처리하며 본 논문에서 제안한 알고리즘을 이용하여 실시간으로 제스처 인식을 수행했을 경우 초당 26 프레임의 속도로 저하되었다. 하지만 이는 원 인식 과정에서의 부하에 의한 것이며 제스처 인식 부분에서의 부하로 인한 속도 저하는 없었다.

3.2 제스처 인식 실험 결과

먼저 숫자 '2'의 궤적을 갖는 제스처 인식 실험을 행하였다. 인식을 위해 사용한 제스처 맵은 Fig. 13 과 같으며 곡선과 직선의 순서로 이루어진 숫자 '2'의 여러 가지 패턴을 고려하여 작성하였다. 여러 명의 사용자들에 의해 입력된 제스처 인식 결과는 Table 2 와 같다.

인식에 실패한 제스처 궤적은 숫자 '2'의 제스처가 예상과는 다르게 곡선과 곡선으로 이루어진 궤적이었다. 인식에 실패한 궤적의 제스처 코드들은 Fig. 14 와 같으며 이들의 추가적인 인식을 위해 Fig. 15 와 같은 제스처 맵을 추가로 사용하였다. 즉, 하나의 제스처에 둘 이상의 제스처 맵 사

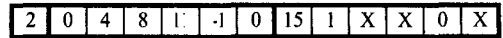


Fig.13 Gesture map for the experiment

Table 2 Result of recognition

Result	Trajectory of gesture
Success	
Failure	

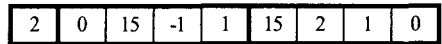
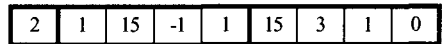
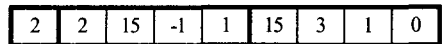


Fig. 14 Gesture codes failed in recognition

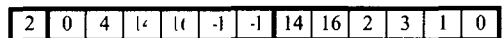


Fig. 15 Gesture map added to recognize the failed codes

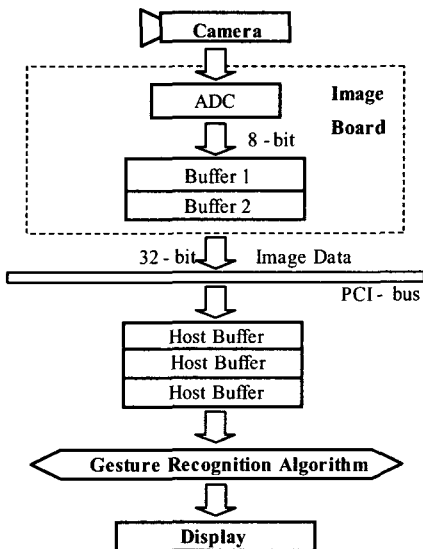


Fig. 12 Data flow chart

용이 가능하며 이러한 제스처 맵의 자유로운 확장성에 의하여 사용자마다 다양하게 표현한 제스처의 효과적인 인식이 가능해진다.

위와 마찬가지로 방법으로 몇 가지 도형과 나머지 9 개의 숫자들에 대한 제스처 인식 실험을 행하였다. 실험한 제스처의 종류와 개수는 Table 3 과 같다. 도형은 각각 3~5 개의 Stage 로 이루어진 삼각형, 사각형, 별의 제스처 인식을 실험하였으며 도형을 표현할 수 있는 다양한 획의 순서와 방향은 고려하지 않고 한 가지 경우만 고려하였다. 숫자의 경우 '8'은 사용자에게 따라 시작방향이 시계방향인 궤적과 반 시계방향인 궤적이 있으므로 두 가지 패턴의 인식을 위해 두 개의 제스처 맵을 사용하였다.

실험은 예상하지 못한 각각의 제스처 패턴의 인식을 위하여 제스처 맵을 계속 수정해 나간 결

과 한 가지의 경우를 제외하고 모두 인식에 성공하였다. 인식에 실패한 경우는 숫자 '0'과 '6'의 제스처가 종종 구분이 되지 않고 모두 '0'으로 인식이 된 것이며 이는 '0'과 '6'의 제스처 코드가 Fig. 16 에서처럼 비슷한 경우가 발생하기 때문이다. 이런 경우의 제스처 구분은 본 논문에서 제안한 제스처의 방향 변화 정보만을 이용한 인식 방법으로는 가능하지 않으며 더 많은 정보를 이용하는 다른 방법이 요구된다.

Table 3 Gestures used for experiment

Number	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Figure	Triangle, Quadrangle, Star

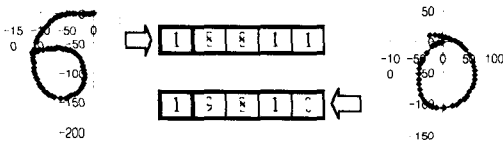


Fig. 16 Similar gesture codes from different gesture

4. 결론

제스처의 방향 변화 정보만을 이용하여 복잡한 제스처를 실시간으로 인식할 수 있는 알고리즘을 개발하였다. 비록 제스처 인식에 필요한 최소의 정보만을 사용하였음에도 불구하고 사용자마다 다르게 표현할 수 있는 제스처를 효과적으로 인식할 수 있는 제스처 코드와 제스처 맵을 제안하였으며 특히, 제스처 인식의 기준이 되는 제스처 맵은 수정과 확장이 용이하여 사용자의 제스처를 학습하여 인식할 수 있는 지능형 로봇의 구현에 본 알고리즘이 효과적으로 사용될 수 있을 것으로 기대된다. 그러나 특수한 경우에는 정보의 부족으로 인하여 다른 의도로 표현한 제스처들을 구분하지 못하는 문제점이 발견되기도 하였다. 하지만 이러한 문제점은 제스처 기반의 명령 인식이 가능한 지능형 로봇이나 가전 제품의 지령 제스처를 적절히 선정함으로써 해결할 수 있다.

본 연구에서 제안한 알고리즘을 효과적으로 이용할 수 있는 향후 과제로는 손의 궤적뿐만 아니라 팔꿈치나 어깨 등의 궤적을 동시에 사용하는

제스처 인식, 스테레오 비전 시스템을 이용하여 얻은 3 차원의 제스처 궤적을 사용한 제스처 인식, 그리고 앞에서 언급한 제스처 맵의 자동 학습 알고리즘 등이 있다.

참고문헌

1. Lee, C. and Xu, Y., "Online, Interactive Learning of Gestures for Human/Robot Interfaces," ICRA, Vol. 4, pp. 2982-2987, 1996.
2. Baudel, T. and Lafon, M.B., "Charade: Remote Control of Object using Free-Hand Gestures," Communications of the ACM, Vol. 36, No. 7, pp. 28-35, July 1993.
3. Starner, T. and Pentland, A., "Visual Recognition of American Sign Language Using Hidden Markov Models," International Workshop on Automatic Face and Gesture-Recognition, pp. 189-194, 1995.
4. Min, B., Yoon, H., Soh, J., Yang, Y. and Ejima, T., "Hand Gesture Recognition Using Hidden Markov Models," IEEE International Conference on Systems, Man, and Cybernetics, pp. 4232-4235, 1997.
5. Shin, B.K. and Kim, J.B., "On-line Hangul Character Recognition based on Letter-Spotting Technique," Journal of Korea Information Science Society, Vol. 23(B), No. 11, pp. 1135-1144, 1996.
6. Kim, C.W., Kim, B.M. and Kang, O.H., "Online Recognition Using Fuzzy Direction Code And HMM," Journal of Korea Information Science Society, Vol. 24(B), No. 5, pp. 551-560, 1997.
7. Kang, D.J. and Ha, J.E., Digital Image Processing, SciTech, pp. 218-263, 2003.
8. Lee, G.M., Moon, D.S., Kim, S.W. and Kim, M.H., "Hand Gesture Recognition and Tracking under Natural Environment," Journal of Korea Multimedia Society, pp. 577-582, Autumn 1999.