

테이블탑 디스플레이에서 활용 가능한 범용적인 터치 제스처 정의 및 인식

박재완, 김종구, 이철우
cyanlip@naver.com, funky20@nate.com leecw@image.chonnam.ac.kr
전남대학교 전자컴퓨터공학과

General Touch Gesture Definition and Recognition for Tabletop display

Jae-Wan Park, Jong-Gu Kim Chil-Woo Lee
Department of Electronics and Computer Engineering
Chonnam National University

요 약

본 논문에서는 터치 제스처의 인식을 위해 시도된 여러 방법 중 테이블탑 디스플레이상에서 HMM을 이용한 제스처의 학습과 사용에 대해 제안한다. 터치 제스처는 제스처의 획(stroke)에 따라 single stroke와 multi stroke로 분류할 수 있다. 그러므로 제스처의 입력은 영상프레임에서 터치 궤적에 따라 변하는 방향 벡터를 이용하여 방향코드로 분석될 수 있다. 그리고 분석된 방향코드를 기계학습을 통하여 학습시킨 후, 인식실험에 사용한다. 제스처 인식 학습에는 총 10개의 제스처에 대하여 100개 방향코드 데이터를 이용하였다. 형태를 갖추고 있는 제스처는 미리 정의되어 있는 제스처와 비교를 통하여 인식할 수 있다. (4 방향 드래그, 원, 삼각형, γ , \sphericalangle 모양 $>$, $<$) 미리 정의되어 있는 제스처가 아닌 경우에는 기계학습을 통하여 사용자가 의미를 부여한 후 제스처를 정의하여 원하는 제스처를 선택적으로 사용할 수 있다. 본 논문에서는 테이블탑 디스플레이 환경에서 사용자의 터치제스처를 인식하는 시스템을 구현하였다. 앞으로 테이블탑 디스플레이 환경에서 터치 제스처 인식에 적합한 알고리즘을 찾고 멀티터치 제스처를 인식하는 연구도 이루어져야 할 것이다.

1. 서 론

인간은 표현의 도구로 제스처를 사용한다. 인간이 비언어적 의사소통 수단인 제스처를 사용하면 좀 더 직관적인 표현이 가능하다. 즉, 전달하고자 하는 목적을 더 명확히 설명하기 위한 보조적인 수단으로써 제스처를 사용한다.

응용프로그램에서 제스처를 활용하면 사용자에게 편리함을 준다[1]. 사용자가 제스처를 사용하면서 편리하게 된 터치기기(device)는 레오파드 OS를 사용하는 애플의 맥북이나 아이팟 터치등이 있다. 이러한 기기는 터치패드에서 제스처를 사용하여 다양한 피드백을 얻을 수 있다. 그러므로 터치제스처를 정의하면 사용자에게 편의를 줄 수 있다.

테이블탑 디스플레이의 입력은 맨손 인터페이스이므로 테이블탑 디스플레이에서는 직관적인 손의 터치제스처를 바로 적용하고 사용할 수 있다. 하지만 기존의 테이블탑 디스플레이에서는 터치 제스처를 정의하지 않고 대부분 OS Value를 사용하거나 응용프로그램에서 요구하는 영

역에 대한 입력으로 제스처를 대체하고 있다.

그리고 테이블탑 디스플레이에서 제스처의 활용은 단지 터치점의 개수를 이용하여 제스처를 표현하고 콘텐츠에서는 제스처의 기능을 대신하는 것이 대부분이다.

본 논문에서는 터치제스처를 입력으로 사용하여 테이블탑 디스플레이에서 활용이 가능한 범용적인 제스처를 정의하고 인식하는 과정에 대해 기술한다.

2. 관련연구

본 논문에서는 터치 제스처 인식을 위해 HMM을 사용하고 있다. HMM은 제스처 인식 분야뿐만 아니라 음성 인식, 필기인식 등에서 뛰어난 인식률을 보이고 있다. 연구[1]은 손의 제스처모델을 모수적 HMM을 이용하여 인식하고 있다. 기본적인 HMM을 이용하면 제스처의 수에 따라 상태(state)의 개수가 늘어나므로 연산 속도가 오래 걸리지만, PHMM은 상태수를 늘리지 않아도 인식 대상을 늘릴 수 있다.

하지만 본 논문에서는 단일 터치 제스처에 대한 인식을 수행하고 있으므로, 터치 surface의 영역구분에 의한 labeling을 사용하지 않는다. 차후 멀티터치 제스처를 인

† 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2009년도 문화콘텐츠산업기술지원사업의 연구결과로 수행되었습니다.

식하기 위해서는 HMM에서 모수적인 상태를 설정할 수 없다.

연구[2]는 MTS(Multi-touch surface)에서 구동되는 GI(Gestural Interface)를 구현하였다. 하지만 응용프로그램을 위한 제스처만을 분류하므로, 다중출력을 기대할 수 없다.

연구[5]는 single stroke 제스처를 HMM이외에도 som (self-organizing map)을 이용하여 인식하고 있다. 그리고 제스처 심볼(symbol) 좌표의 움직임을 추적한다.

본 논문에서는 10개의 제스처에 대한 10개의 상태를 정의하고 학습한다. 학습 과정 이후의 인식과정에서는 터치제스처는 HMM에서 얻어진 확률의 결과만을 이용하여 분류하므로 HMM만을 이용하여 인식이 가능하다.

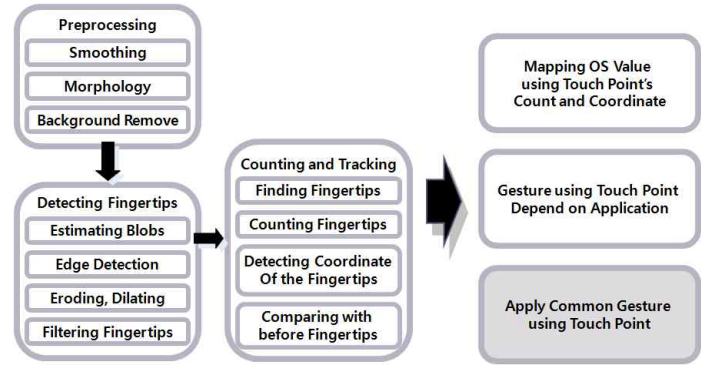


그림 2 터치 인식 과정

3. 실험 내용

3.1 터치점 추출

테이블탑 디스플레이에서 적외선 카메라를 통해 얻은 적외선 영상은 그림[1]과 같다. (c),(d)는 (a),(b)에서 터치 인식 알고리즘을 이용하여 손끝의 터치점을 추출한 영상이다.

그림[2]에서 보는 것처럼 손가락의 터치점을 테이블탑 디스플레이에서 터치좌표로 추출할 수 있다. 적외선 영상에서 터치 점의 좌표를 추출하는 과정은 그림[2]과 같다. 터치점의 좌표는 영상의 프레임의 순서에 따라 추출할 수 있고, 이러한 특징정보를 이용하여 시간에 따라 변하는 터치점에 대한 정보를 추출한다. 추출된 터치점의 좌표를 이용하여 다음 터치점과의 거리이동 및 기울기를 측정하여 8방향코드로 분석한다.

테이블탑 디스플레이에서 터치 제스처는 순차적인 특징을 가진다. 터치제스처는 body, head, hand 제스처와는 다르게 손가락에 의한 터치가 시작되는 순간부터 끝나는 순간까지의 시퀀스를 이용하고 있기 때문이다. 그러므로 흔히 온라인 필기 인식에서 주로 사용하는 HMM을 이용하여 터치제스처를 인식할 수 있다.

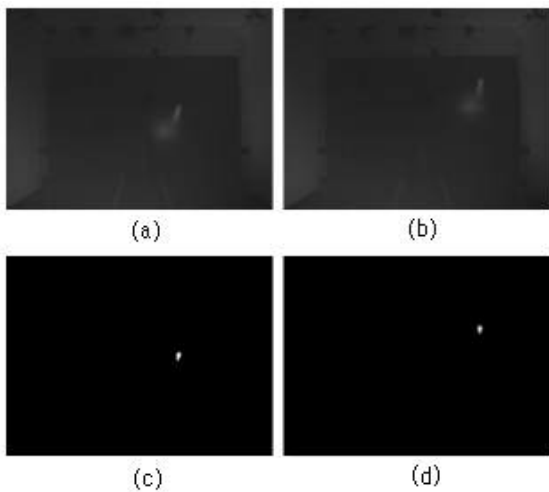
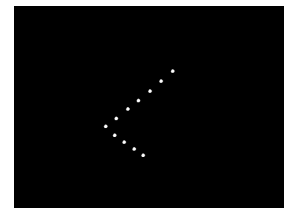
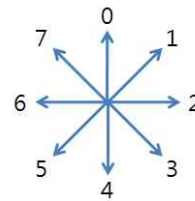


그림 1 (a),(b) 적외선 영상, (c),(d) 터치점 추출 영상



chain code
(5555533333)

그림 3 8방향 코드에 따른 체인코드

3.2 HMM을 이용한 제스처 인식 방법

HMM은 시간적으로 제약을 받는 정보의 구조를 모델링 하는데 뛰어난 모델이다. 음성인식에 가장 많이 사용되는 인식 모델이고, Simple Markov Model만으로 모델링하기 힘든 실제계의 문제를 통계적 매개변수로 접근할 수 있게 해준다. 상태 전이 매개 변수는 순차적인 일련의 사건 발생을 모델링한다. 그리고 관측 심볼 확률 분포는 각 사건의 특징을 유한개의 심볼로 대응시킨다.

HMM은 이러한 두 가지 확률 과정의 결합으로 이루어져 있고, 이 기준에 따라 생성된다. 생성된 HMM은 학습 데이터를 이용한 학습을 통해 적절한 제스처 모델을 구성한다.

인식과정에서는 인식하고자 하는 제스처와 학습이 끝난 후 생성된 모든 HMM의 제스처 모델을 비교한다. 그리하여, 가장 유사하다고 판단되는 제스처 모델을 선택하고 결과를 확률로 나타낸다.

상태전이확률을 이용하여 각 샘플들을 이용한 전처리 단계는 잡음제거와 대표점 추출 과정을 거쳐 손가락 궤적을 체인코드로 변환하는 과정으로 8방향 체인코드를 사용한다.

학습은 각 제스처별로 이루어지며, 해당 제스처의 HMM 모델에 학습결과를 적용한다. HMM의 학습 과정은 각 숫자별로 테이블탑 디스플레이를 통해 수집한 체인코드를 이용하여 숫자별 은닉 마르코프 모델을 구성하는 과정으로, EM알고리즘의 하나인 Baum-Welch 알고리즘을 이용한다. HMM의 인식 과정은 전처리 모듈을 통해 변환된 체인코드가 어느 숫자의 은닉마르코프 모델

에서 나타날 확률이 높은가를 판단하는 과정으로, 각각의 숫자 모델에 전향(Forward) 알고리즘을 적용하여 가장 높은 확률을 보이는 숫자 모델을 최종 인식 결과로 출력한다.

HMM은 아래와 같은 요소로 구성된다.

- N : 상태의 수,
 $S = \{S_1, S_2, \dots, S_N\}$: 상태의 집합,
 q_t : 시간 t 의 상태
- M : 관측 심볼의 수,
 $V = \{v_1, v_2, \dots, v_M\}$: 관측 심볼의 집합
- A : $\{a_{ij}\}$: 상태 전이 확률 분포
 $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N$
 : 상태 i 에서 상태 j 로 전이할 확률
- $B = \{b_t(k)\}$: 관측 심볼 확률 분포,
 $b_j(k) = P(v_k | q_t = S_j), 1 \leq j \leq N, 1 \leq k \leq M$
 : j 에서 심볼 v_k 를 관측할 확률
- $\pi = \{\pi_i\}$: 초기 상태 확률 분포,
 $\pi_i = P(q_1 = S_i), 1 \leq i \leq N$: 초기 상태가 i 일 확률

일반적으로 하나의 HMM은 $\lambda = (A, B, \pi)$ 로 표시된다. 주어진 모델과 관측열 $O = O_1, O_2, \dots, O_\tau$ 에 대해 생성 확률은 아래와 같다.

$$P(O|\lambda) = \sum_{\text{for all } q} \left[\pi_{q_1} b_{q_1}(O_1) \prod_{i=2}^{\tau} a_{q_{i-1}q_i} b_{q_i}(O_i) \right] \quad (\text{수식 1})$$

HMM의 응용에는 세 가지의 해결해야 할 문제가 있다. 즉, 평가, 해석, 그리고 학습에 있으며 이들은 각기 Forward-Backward 알고리즘, Viterbi 알고리즘 그리고 Baum-Welch 알고리즘으로 해결된다. 다음 그림은 본 논문에서 사용된 일반적인 Left-Right HMM의 형태를 보여준다.

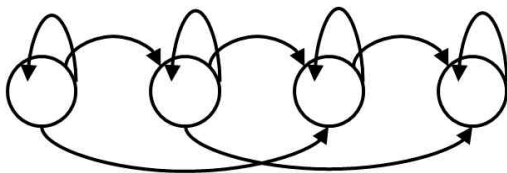


그림 4 Left-Right HMM 모델

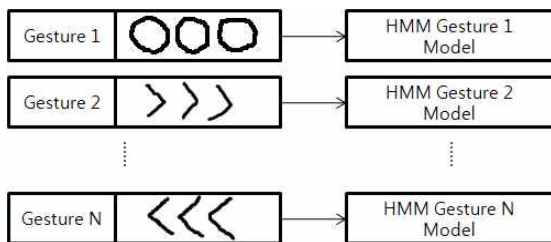


그림 5 제스처 학습 과정

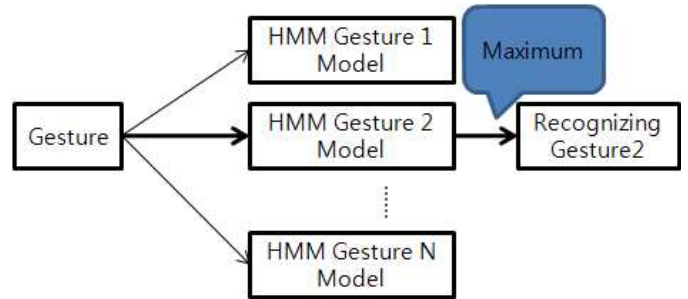


그림 6 제스처 인식 과정

그림 [5]은 제스처 학습 과정을 나타낸 것이고, 그림 [6]는 HMM을 사용한 제스처 인식 과정을 보여준다.

4. 실험 결과

실험에서는 10개의 제스처마다 각각 10개의 샘플을 이용하여 각각의 체인코드를 학습하였다. 각 제스처의 특징점이 되는 터치점은 테이블탑 디스플레이를 이용하여 추출하였다.

HMM의 학습 데이터는 테이블탑 디스플레이에서 10개의 제스처를 10번 행동하여 총 100개의 체인코드를 만들어 데이터베이스화 하였다. HMM에서의 상태수는 제스처에 따라 10개로 하였고, 8방향 코드를 사용하였다.

본 실험에서는 범용적인 제스처를 사용하여 제스처의 사용이 응용프로그램에 제한받지 않도록 하였다. 응용프로그램에 제한되는 제스처는 제스처를 인식하지 않더라도 응용프로그램의 요구를 피드백으로 보여준다. 하지만 범용적인 제스처는 인식 결과에 따라 수행결과를 보여주므로 제스처의 인식에 대한 장점을 나타낼 수 있다.

학습에 사용한 제스처는 그림[7]과 같다.

테이블탑 디스플레이에서 그림[7]과 같은 제스처를 인식시킨 결과는 표[1]과 같다. 본 논문에서 사용한 제스처는 방향벡터가 중복되는 부분이 적고, 그 수가 많지 않으므로 표[1]과 같은 인식률을 얻을 수 있었다.

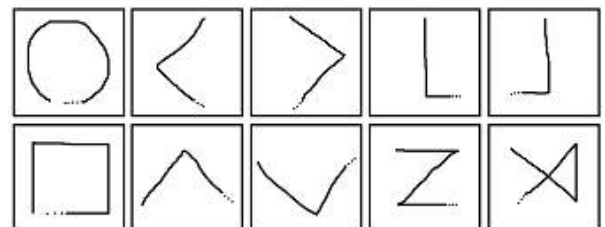


그림 7 학습 제스처

표 1 테이블탑 디스플레이에서의 인식 결과

숫자	실험데이터 개수	정인식 개수	정확율(%)
0	10	9	90%
1	10	8	80%
2	10	10	100%
3	10	10	100%
4	10	10	100%
5	10	10	100%
6	10	10	100%
7	10	10	100%
8	10	8	80%
9	10	8	80%
전체	100	93	93%

5. 결론

본 논문에서는 테이블탑 디스플레이상에서 HMM을 이용한 터치제스처 학습과 인식에 대하여 제안하였다. 실험데이터는 테이블탑 디스플레이 환경에서 수집하였지만, 터치 제스처 학습은 사용자가 어떤 제스처를 요구하고 사용하는가에 대해 따라 변할 수 있다. 그러므로 다양한 사용자의 터치 제스처가 수집되어야 할 것이다.

그리고 테이블탑 디스플레이의 스크린에 터치되는 터치점들을 유클리디언 거리에 의해 터치점의 좌표를 획득하고, 체인코드를 생성하였으므로 프로그램의 성능에 영향을 받는다. (카메라의 프레임 속도에 의존한다)

따라서 테이블탑 디스플레이 환경에 적합한 HMM의 상태를 정의하는 연구도 진행해야 할 것이다. 또한 문자 인식에 사용하는 인공신경망 등의 알고리즘과 비교를 통해 터치제스처 인식에 가장 적합한 알고리즘에 대한 연구도 함께 이루어져야 할 것이다.

참고 문헌

[1] Wilson et al., "Parametric Hidden Markov Models for Gesture Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, No. 9, pp. 884-900, 1999.

[2] Westerman, W. and Elias, J.G. Multi-touch: A New Tactile 2-D Gesture Interface for Human Computer Interaction. Human Factors and Ergonomics Society 45th Annual Meeting. 2001.

[3] J. Ou, X. Chen, J. Yang, "Gesture Recognition for Remote Collaborative Physical Tasks Using Tablet PCs", in: Proc. of IX IEEE Intl. Conf. on Computer

Vision, Work, on Multimedia Technologies in E-Learning and Collaboration (Nice, 2003).

[4] C. Mertz, P. Lecoanet, "GRIGRI: Gesture Recognition on Interactive Graphical Radar Image", Published in Progress In Gestural Interaction Proceedings of Gesture Workshop 96. 1996 Springer.

[5] Caridakis, G., Karpouzis, K., Pateritsas, C., Drosopoulos, A., Stafylopatis, A., Kollias, S., "Hand trajectory-based gesture recognition using self-organizing feature maps and Markov models.", In: IEEE Internat. Conf. on Multimedia and Expo, 2008.

[6] Jie Yang and Yangsheng Xu, "Hidden Markov Model for Gesture Recognition," tech. report CMU-RI-TR-94-10, Robotics Institute, Carnegie Mellon University, May, 1994.