

가속도 센서 데이터를 이용한 스마트폰 사용자의 제스처 인식

남상하*, 김주희*, 허세경*, 김인철**

*경기대학교 컴퓨터과학과 학부생

**경기대학교 컴퓨터과학과 교수

e-mail:{namsh, chilang, tprudzzang, kic}@kgu.ac.kr

Gesture Recognition from Accelerometer Data on a Smartphone

Sang-Ha Nam*, Joo-Hee Kim*, Se-Kyeong Heo*, In-Cheol Kim**

*Undergraduate Course, Dept of Computer Science, Kyonggi University

**Faculty, Dept of Computer Science, Kyonggi University

요 약

본 논문에서는 스마트 폰에 내장된 3축 가속도 센서를 이용해 제스처 훈련 및 테스트 데이터를 수집하고, DTW(Dynamic Time Warping) 알고리즘을 근간으로 하는 효과적인 제스처 인식 방법을 제안한다. 본 논문에서 제안하는 제스처 인식 방법의 성능을 분석하기 위해 안드로이드 스마트 폰에서 동작하는 제스처 인식 프로그램을 개발하였고, 이것을 이용해 수행한 성능실험 결과를 소개한다.

1. 서론

최근 들어 스마트 폰이 급속히 보급되면서, 국내 스마트 폰 이용자 수 2500만 명의 시대가 열리게 되었다. 이러한 추세는 국내뿐만 아니라 전 세계적으로 일어나고 있다. 스마트 폰은 기존의 휴대폰보다 더욱 다양한 어플리케이션을 제공함에 따라 사용자와 스마트 폰 간의 상호 작용(interaction)이 중요한 기능이자 요소로 떠오르고 있다.

스마트 폰 상의 새로운 인간-기계 상호 작용(HCI) 방식으로는 틸트(Tilt), 웨이크(Shake) 그리고 제스처(Gesture) 등이 있다. 틸트를 이용해 게임을 즐기거나 웨이크를 이용해 알람을 끄는 모습은 주변에서 흔히 일어나는 상황이다. 이러한 틸트와 웨이크는 사용자에게 따라 성능이 크게 좌우되지 않지만 제스처는 3차원 공간상에서 역동적인(Dynamic) 행동 패턴을 분석해야 하므로 사용자마다 다양한 모양이 나온다. 또한 제스처는 손이나 팔, 다리, 머리 그리고 얼굴을 이용해서 전달하는 비언어적 의사소통이다. 인간-기계 상호작용에 있어서 음성인식을 이용한 언어적 의사소통뿐만 아니라 제스처를 이용한 비언어적 의사소통에 관한 연구도 활발히 진행되고 있는 추세이다.

본 논문에서는 스마트 폰에 내장된 3축 가속도 센서를 이용해 제스처 훈련 및 테스트 데이터를 수집하고, DTW(Dynamic Time Warping) 알고리즘을 근간으로 하는 효과적인 제스처 인식 방법을 제안한다. 본 논문에서 제안하는 제스처 인식 방법의 성능을 분석하기 위해 안드로이드 스마트 폰에서 동작하는 제스처 인식 프로그램을 개발하였고, 이것을 이용해 수행한 성능실험 결과를 소개한다.

2. 관련 연구

Sung-Jung Cho의 논문[1]에서는 베이저안 네트워크(Bayesian networks)를 바탕으로 계층적 모델을 만들어 제스처를 1차 분류하였다. 그리고 비슷한 제스처에 대해서 SVM(Support Vector Machine)을 사용한 2차 분류를 통해 정확도를 향상시켰다. Gerrit Neizen의 논문[2]은 은닉 마코프 모델(HMM), 인공신경망(ANN) 그리고 DTW의 성능 비교 실험을 했다.

Tea Marasovic의 논문[4]에서는 7개의 제스처에 대해 주성분분석(Principal Component Analysis)을 이용해 특징 선택(Feature Selection)과 차원 축소(Dimensionality Reduction)를 수행하였다. Ahmad Akl의 논문[5]에서 훈련 단계에서는 DTW와 AP(Affinity Propagation)을 이용해 각 군집(Cluster) 마다 대표(Exemplar)를 뽑아내고 이 대표들과 테스트 샘플간의 DTW 비교로 제스처를 분류하였다. KyoJoong Oh의 논문[6]에서는 모수적 은닉 마코프 모델(Parametric Hidden Markov Model)과 모션 벡터(Motion Vector)을 사용하여 9개의 제스처를 구분했다.

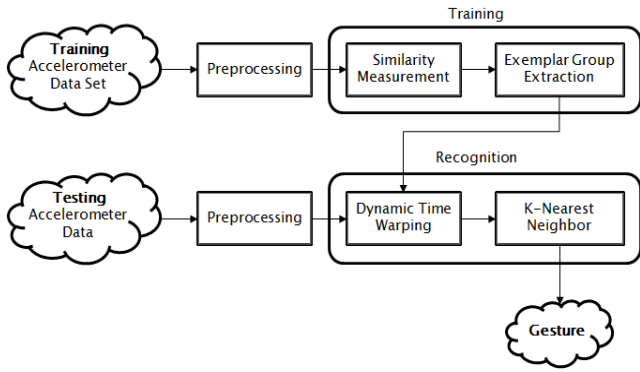
3. 사용자 제스처 인식

3.1 시스템 개요

본 논문에서는 스마트 폰 내장 3축 가속도 센서를 이용하는 효과적인 사용자 제스처 인식 방법과 구현 시스템을 소개한다. <표 1>은 이 시스템에서 구분할 제스처 종류를 나타낸 것이고, <표 2>는 제스처 별 가속도 입력 값의 변화를 나타낸 것이다. <표 2>를 통해 제스처 별로 3축 가속도 입력 값의 변화가 서로 다른 패턴을 보임을 알 수 있다.

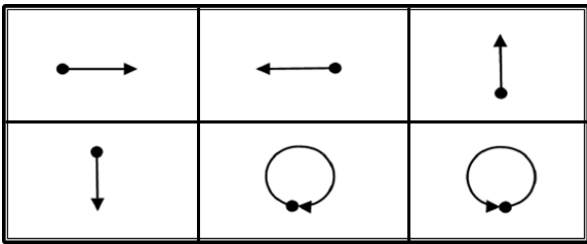
사용자 제스처 인식 시스템의 전체 과정은 (그림 1)과 같다. 데이터는 훈련 데이터와 테스트 데이터로 나뉜다.

※ 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음

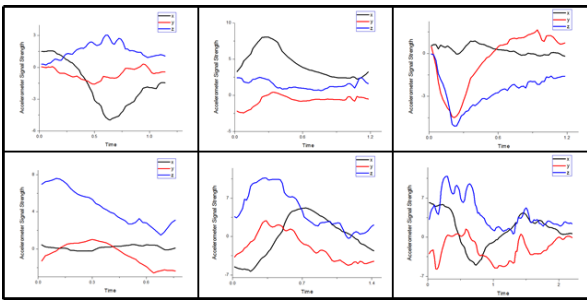


(그림 1) 전체 과정

이 두 데이터는 공통적으로 스마트폰에 내장된 3축 가속도계를 통해 얻어지고, 미리 정한 형식에 맞춰 파일형태로 저장된다. 훈련 데이터와 테스트 데이터는 전 처리(Preprocessing) 단계에서 일부 불필요한 데이터를 삭제하는 과정을 거친다. 그리고 훈련 단계에서는 DTW 알고리즘을 이용한 유사도 측정(Similarity Measurement)을 통해 훈련 데이터들의 유사도 행렬(Similarity Matrix)을 만들고, 이를 바탕으로 대표 집단 추출(Exemplar Group Extraction)을 한다. 인식(Recognition) 단계에서는 실시간으로 입력된 테스트 데이터를 훈련 단계에서 선정된 대표 집단의 훈련 데이터들과 유사도 측정을 하고, K-근접이웃(K-NN) 알고리즘을 이용해 최종적으로 어떤 제스처 인지 판별한다.



<표 1> 제스처 종류



<표 2> 제스처 별 가속도 입력 값의 변화

3.2 데이터 수집 및 전 처리 단계

데이터 수집은 3축 가속도계가 내장된 안드로이드 기반 스마트폰에서 이루어진다. 데이터 측정 주기로는 센서 값이 변할 때 마다 기록되는 방식을 사용한다. 따라서 안드로이드에서 제공하는 가장 빠른 간격을 이용한다.

$$S_n = \langle ID_n, User_n, G_n, AccSeq_n \rangle$$

$$G_n \in \{G_1, G_2, \dots, G_6\}$$

$$AccSeq_n = [\langle Acc_{x_1}, Acc_{y_1}, Acc_{z_1} \rangle, \dots, \langle Acc_{x_m}, Acc_{y_m}, Acc_{z_m} \rangle] \quad (1)$$

위 식(1)은 파일에 저장되는 데이터 형태를 나타낸 것이다. 제스처 데이터(S_i)는 일련번호(ID_i), 사용자 이름($User_i$), 제스처 종류(G_i) 그리고 일련의 가속도 값($AccSeq_i$)으로 이루어진다.

앞서 언급했듯이 제스처는 사용자에게 상당히 의존적이다. 이를 분석하기 위해 사용자 이름이 데이터의 한 부분을 차지한다. 제스처 종류는 <표 1>과 같이 이 시스템에서 구분하고자 하는 6가지 종류로 제한하였다. 그리고 일련의 가속도 값은 x축, y축, z축으로 각각 나누어서 저장하는데 이는 두 데이터 간의 유사도 비교를 할 때 각 축끼리 계산해야 하기 때문이다. 전 처리 단계는 훈련 및 테스트 데이터에 포함된 불필요한 데이터를 삭제한다. 데이터 수집 시 수집 버튼을 누르는 순간(Action Down)부터 떼는 순간(Action Up)까지 3축 가속도 센서 값을 기록한다. 이로 인해 데이터의 처음 부분과 끝 부분은 불필요한 센서 값이 저장될 확률이 높으므로 모든 데이터의 앞(Front), 뒤(Rear) 각각 5% 씩 총 10%의 데이터를 삭제한다.

3.2 훈련 단계

훈련 단계에서는 DTW 알고리즘을 사용하여 유사도 행렬을 구하는 과정과 이를 이용한 대표 추출 방법이 진행된다. 먼저 DTW 알고리즘은 시간이나 속도가 다른 두 시퀀스 데이터 간의 유사도를 측정하는 알고리즘으로써 주로 시계열 패턴인식에 적용한다. 이 알고리즘은 두 시퀀스의 시작 성분부터 끝 성분까지 동적 계획법으로 계산한다. 이 알고리즘을 이용하면 비선형 매핑 함수를 최적으로 찾아가면서 동시에 비교가 이루어진다. 알고리즘의 결과 값은 두 시퀀스의 차이를 나타내므로 바꿔 말하면 유사도라도 표현될 수 있고 이 값이 작을수록 두 데이터가 비슷하다는 것을 의미한다. 본 논문에서 다루는 입력 데이터도 제스처의 경로(Trace)에 따른 시계열 데이터이므로 DTW 알고리즘을 적용한다.

$$D_{i,j} = d(p_i, q_j) + \min\{D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\} \quad (2)$$

$$d(p_i, q_j) = (p_i - q_j)^2 \quad (3)$$

임의의 두 시퀀스 데이터 $p = [p_1, \dots, p_k]$ 와 $q = [q_1, \dots, q_l]$ 가 있다고 가정하자. 이 둘 사이의 유사도는 식 (2)을 통해 계산되는데 본 논문에서 사용할 거리 함수 $d(\cdot, \cdot)$ 는 식 (3)과 같다. 이러한 계산 과정을 거쳐 얻은 두 데이터 간의 유사도 값은 식 (4)와 같다.

$$DTW(p, q) = D_{k,l} \quad (4)$$

본 논문에서 다루는 제스처 훈련 및 테스트 데이터는 앞서 데이터 수집 단계에서 x축, y축, z축으로 나누어 놓았다. 따라서 아래 식 (5)을 이용해서 최종적인 두 제스처 데이터 간의 유사도를 계산한다.

$$DTW(S_a, S_b) = \sqrt{D_{k,l}^2(x) + D_{k,l}^2(y) + D_{k,l}^2(z)} \quad (5)$$

이러한 방법을 사용하면 정확성이 높고, 특별한 모델 만들지 않아서 은닉 마코프 모델이나 인공 신경망과 같은 알고리즘에 비해서 인식 속도가 빠르다.[2] 이러한 점은

스마트 폰 상에서 실시간 제스처 인식 시스템을 설계할 때 장점이 된다.

훈련 단계의 첫 번째는 식 (6)과 같이 각 제스처 별로 대표 추출을 위한 유사도 행렬(SM_i)을 만드는 것이다.

$$SM_i = \begin{bmatrix} sm_{1,1}^i & sm_{1,2}^i & \cdots & sm_{1,z}^i \\ sm_{2,1}^i & sm_{2,2}^i & \cdots & sm_{2,z}^i \\ \vdots & \vdots & \ddots & \vdots \\ sm_{z,1}^i & sm_{z,2}^i & \cdots & sm_{z,z}^i \end{bmatrix}$$

$$SM_i \in \{SM_1, SM_2, \dots, SM_6\} \quad (6)$$

$$sm_{r,s}^i = DTW(S_a, S_b) \quad (7)$$

이 행렬의 요소($sm_{r,s}^i$)는 식 (7)과 같이 DTW 알고리즘을 이용해서 동일한 제스처를 나타내는 서로 다른 훈련 데이터들 간의 유사도를 측정된 결과이다. 이와 같이 각 제스처 별로 행렬을 만드는 이유는 각 제스처 별로 대표 집단을 추출하기 위함이다.

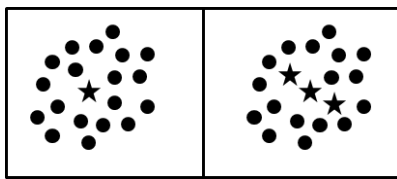
훈련 단계의 두 번째는 식 (8)과 같이 합 행렬(TM_i)을 구한다.

$$TM_i = [tm_1^i, tm_2^i, \dots, tm_z^i]$$

$$TM_i \in \{TM_1, TM_2, \dots, TM_6\}$$

$$tm_r^i = \sum_{s=1}^z sm_{r,s}^i \quad (8)$$

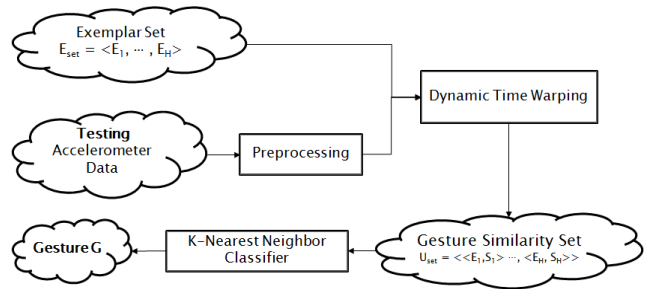
합 행렬의 요소(tm_r^i)는 해당 훈련 데이터가 자신을 제외한 다른 모든 훈련 데이터들로부터 어느 정도의 유사도 값을 가지는지 나타낸다. 즉 합 행렬(TM_i)에서 가장 작은 값을 가지는 훈련 데이터가 다른 모든 훈련 데이터들과 거리 차의 합이 가장 작다는 것을 알 수 있고 이는 가장 중심에 위치한다는 것을 의미한다. 이를 바탕으로 각 제스처 마다 가장 중심에 위치한 대표를 선정한다. 이렇게 선정된 대표는 차후에 제스처 인식 단계에서 테스트 데이터와의 비교에 사용된다.



(그림 2) 대표 추출 방법

(그림 2)는 두 가지 대표 추출 방법을 비교해서 나타낸 그림이다. 왼쪽은 각 제스처 별로 한 개의 대표를 선정하는 방법이고 오른쪽은 대표 집단을 선정하는 방법이다. 기존의 방법은 각 데이터간의 유사도 합이 가장 작은 데이터 즉 가장 중앙에 위치한 훈련 데이터를 대표 시퀀스로 선정한다. 그러나 다른 관점에서 보면 훈련 데이터를 하나만 사용한 것과 차이가 없을 수도 있다. 그리고 훈련 데이터 수가 많을수록 훈련 데이터의 분포가 넓게 퍼질 가능성이 높기 때문에 하나의 훈련 데이터로는 이 모두를 대표하기에 부족하다. 그리고 사용자 간의 편차가 심한 제스처

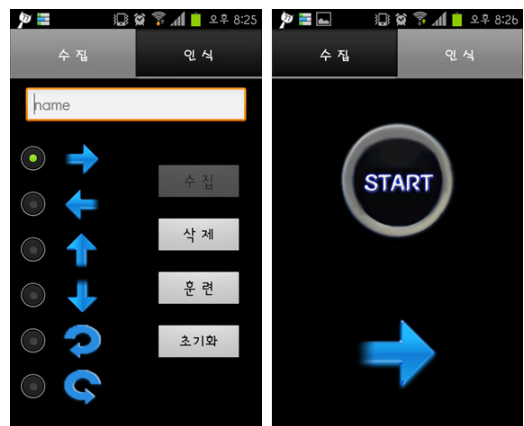
데이터를 하나의 훈련 데이터로 대표하기에는 무리가 따른다. 이러한 점을 보완하기 위한 방법 즉 제스처 인식의 범용성을 넓히고 정확도를 높이기 위한 방법으로 오른쪽과 같은 대표 집단 추출 방법을 제안한다. 제스처 별로 훈련 데이터 하나만 대표로 뽑지 않고 훈련 데이터 수의 25%에 해당하는 중앙에 모인 군집을 대표로 뽑는다. 이를 통해 여러 사용자의 훈련 데이터가 대표 집단에 속할 수 있게 하고 넓은 분포를 가진 훈련 데이터 집단을 효과적으로 대표하여 정확도를 향상시킬 수 있다. 이 방법은 AP 알고리즘을 사용한 방법[5]과 비교해서 여러 개의 대표를 뽑는다는 점은 같지만 군집화를 하지 않는다는 점이 다르다.



(그림 3) 제스처 인식 단계

3.3 제스처 인식 단계

위와 같은 훈련 단계를 거쳐 만들어진 대표 군집을 바탕으로 (그림 3)과 같이 제스처 인식 단계가 수행된다. 제스처 인식 단계는 세부적으로 2단계로 구분할 수 있다. 첫 번째 단계는 전 처리가 완료된 테스트 데이터와 대표 집단(E_{set}) 간의 유사도를 DTW알고리즘을 이용하여 측정한다. 그 결과로 제스처 유사도 집단(U_{set})이 얻어진다. 이 집단은 대표 집단의 각 요소(E_i)와 테스트 데이터간의 유사도(S_i)가 쌍을 이뤄 테스트 데이터가 대표 집단의 각 요소로부터 어느 정도의 유사도 값을 가지는지 나타낸다. 그 다음 단계는 K-최근접 이웃 알고리즘(K-Nearest Neighbor)을 이용해서 테스트 데이터와 가까운 후보 제스처 집단(G_{set})을 추출하고, 후보 제스처 집단에서 다수결 원칙에 의해 하나의 제스처 결과로 판별한다. 이때 K는 제스처 유사도 집단 요소 개수의 10%이며 홀수개로 제한한다.



(그림 4) 프로그램 실행 화면

4. 구현 및 평가

앞서 제시한 사용자 제스처 인식 방법을 이용하여 실제로 안드로이드 스마트 폰용 제스처 인식 프로그램을 구현하였다. (그림 4)는 실험 데이터 수집과 실시간 분류를 위한 안드로이드 어플리케이션의 실행 화면이다. 왼쪽의 수집 탭에서는 제스처별 데이터를 수집하고, 파일 삭제, 훈련 그리고 훈련 초기화 기능이 있다. 그리고 분류 탭은 수집 탭에서 훈련된 내용을 바탕으로 어떤 제스처인지 분류하는 기능을 제공한다.

사용자	제스처 종류	훈련 데이터	테스트 데이터	총 데이터
10명	6개	10번씩	10번씩	1200개

<표 3> 수집한 데이터 개수

본 논문에서 성능 평가를 위해 수집한 데이터 개수는 <표 3>과 같다. 10명의 사용자에게 6개의 제스처를 20번씩 총 1200개의 데이터를 수집하였다. 먼저 대표 시퀀스를 한 개만 선택하는 방식과 대표 집단을 선택하는 방식의 성능 분석 비교 실험을 한 결과는 <표 4>와 같다. 10명의 사용자가 수집한 600개의 데이터로 훈련하고, 600개의 데이터로 테스트하였다. 괄호로 나타낸 숫자는 본 논문에서 취한 대표 집단 추출 방법으로 성능 평가를 한 결과이다. 대표를 하나만 추출하는 방법은 평균 86%의 인식률을 보였고, 대표 집단 추출 방법은 평균 91%로 약 5%가량 향상된 인식 결과를 보였다.

다음은 임의의 사용자에게 대한 제스처 인식 성능을 알아보는 실험이다. 이 실험을 위해 5명의 사용자가 측정한 300개 데이터로 훈련하고 나머지 5명의 사용자가 측정한 300개 데이터로 테스트하였다. 실험 결과는 <표 5>와 같다. 대표 집단 추출 방법을 사용한 제스처 인식 방법이 평균 89%의 인식률을 보였다. 이는 제스처 별 단일 대표 시퀀스를 이용한 방법의 평균 81% 인식률보다 약 8%가량 향상된 결과이다.

	→	←	↑	↓	○	○
→	98(100)			2(0)		
←		94(95)		2(1)	3	1
↑			94(95)	1(0)		5
↓	2(0)	21(4)		76(95)	1	
○	5	5(0)		9(14)	78(77)	3(4)
○	1	14(11)		1(4)	3(2)	81(82)

<표 4> 사용자 혼합 실험 결과

대표 집단 추출 방법은 계산 속도가 비교적 느리지만 훈련 데이터의 양에 비례해서 대표 집단이 선정되게 된다. 이를 통해 여러 사용자의 샘플이 선택 될 확률이 높아지게 되고 결과적으로 사용자 제스처 인식 성능이 향상됨을 알 수 있다.

	→	←	↑	↓	○	○
→	47(45)				1(5)	2(0)
←		45(46)			(1)	5(3)
↑	(5)	3(0)	35(44)		1	11(0)
↓		1(6)		47(44)		2(0)
○				7(0)	23(39)	20(11)
○			1(0)	1(0)	1	47(49)

<표 5> 사용자 독립적 실험 결과

5. 결론

본 논문에서 DTW 알고리즘을 근간으로 하는 효과적인 사용자 제스처 인식 시스템을 제안하였다. 그리고 샘플 수집에 대한 제약을 두지 않고 사용자간의 편차를 최소화하기 위한 노력의 일환으로 각 제스처마다 대표 집단을 선택해서 훈련하는 방법을 제안하였다. 이 시스템은 히든 마코프 모델이나 인공 신경망에 비해 훈련 및 테스트의 계산 속도가 빠르고 간단한 알고리즘을 통해 높은 성능을 기대할 수 있다는 장점이 있지만 데이터들에 대한 표준화 방법을 사용하지 않아서 군집화가 어렵다는 단점이 있다.

향후 시스템의 성능 개선과 제스처 종류를 추가하기 위해 히든 마코프 모델을 이용한 훈련 방안, 길이가 다른 시퀀스(Sequence) 데이터의 효과적인 군집화를 위한 방안 그리고 이를 응용하여 로봇 분야에 접목 시킬 방안 등을 연구할 계획이다.

참고문헌

- [1] Sung-Jung Cho, et al. "Two-stage Recognition of Raw Acceleration Signals for 3-D Gesture-Understanding Cell Phones", Proc. of 10th Int. Workshop on Frontiers in Handwriting Recognition, 2006
- [2] Gerrit Neizen and Gerhard P. Hancke, "Evaluating and Optimizing Accelerometer-based Gesture Recognition Techniques for Mobile Devices", Proc. of IEEE AFRICON-09, 2009.
- [3] Jiayang Liu, et al. "uWave: Accelerometer-based Personalized Gesture Recognition and its Applications", Proc. of IEEE PerCom-09, 2009.
- [4] Tea Marasovic and Vladan Papic, "Accelerometer-Based Gesture Classification Using Principal Component Analysis", Proc. of IEEE SoftCOM-11, 2011.
- [5] Ahmad Akl, et al. "A Novel Accelerometer-Based Gesture Recognition System", IEEE Transaction on Signal Processing, Vol. 59, No. 12, pp. 6197-6205, 2011.
- [6] KyoJoong Oh, et al. "Gesture Recognition Application with Parametric Hidden Markov Model for Activity-Based Personalized Service in APPiME", Proc. of IEEE CogSIMA-11, 2011.