

Smartphone Accelerometer-Based Gesture Recognition and its Robotic Application

Sang-Ha Nam[†] · Joo-Hee Kim^{**} · Se-Kyeong Heo^{**} · In-Cheol Kim^{***}

ABSTRACT

We propose an accelerometer-based gesture recognition method for smartphone users. In our method, similarities between a new time series accelerometer data and each gesture exemplar are computed with DTW algorithm, and then the best matching gesture is determined based on k-NN algorithm. In order to investigate the performance of our method, we implemented a gesture recognition program working on an Android smartphone and a gesture-based teleoperating robot system. Through a set of user-mixed and user-independent experiments, we showed that the proposed method and implementation have high performance and scalability.

Keywords : Smartphone, Gesture Recognition, Accelerometer, Robot Teleoperation, DTW, k-NN

스마트폰 가속도 센서 기반의 제스처 인식과 로봇 응용

남 상 하[†] · 김 주 희^{**} · 허 세 경^{**} · 김 인 철^{***}

요 약

본 논문에서는 스마트폰 사용자를 위한 가속도 센서 기반의 제스처 인식 방법을 제안한다. 제안하는 제스처 인식 방법에서는 DTW 알고리즘을 적용하여 새로운 시계열 가속도 데이터와 각 제스처별 대표 훈련 데이터간의 유사도를 측정 한 뒤, k-NN 알고리즘을 적용하여 제스처를 판별한다. 본 논문에서 제안하는 제스처 인식 방법의 성능을 분석해보기 위해, 안드로이드 스마트폰에서 동작하는 제스처 인식 프로그램과 이것을 활용한 제스처 기반 원격 제어 로봇 시스템을 구현하였다. 사용자-혼합 및 사용자-독립 실험들을 통해, 본 논문에서 제안한 제스처 인식 방법과 구현 시스템이 높은 인식 성능과 확장성을 가진다는 것을 보였다.

키워드 : 스마트폰, 제스처 인식, 가속도 센서, 로봇 원격 제어, DTW, k-최근접 이웃

1. 서 론

최근 들어 국내 스마트폰 이용자 수 2500만 명의 시대가 열리는 등 국내외적으로 스마트폰 보급이 급속히 확산되고 있다. 스마트폰은 기존의 휴대폰보다 더욱 강력한 컴퓨팅 파워와 함께 다양한 센서들을 내장함으로써, 이들을 이용한 보다 편리한 사용자 인터페이스 개발에 많은 연구자들의 관심이 집중되고 있다[1].

스마트폰 내장 가속도 센서를 이용한 새로운 사용자 인터페이스(user interface) 방식으로는 틸트(tilt), 셰이크(shake)

그리고 제스처(gesture) 등이 있다[2]. 틸트를 이용해 모바일 게임을 즐기거나, 셰이크를 이용해 스마트폰 알람을 끄는 모습 등은 이제 주변에서 흔히 목격할 수 있게 되었다. 일반적으로 틸트와 셰이크 인식 기능은 스마트폰이나 사용자에 따라 성능이 크게 좌우되지 않지만, 제스처의 경우는 동일한 제스처를 취하더라도 사용자마다 각기 다른 입력 센서 데이터들이 얻어져 높은 인식 성능과 확장성을 얻기 어렵다. 또한, 제스처 인식을 위해서는 3차원 공간상에서 진행되는 역동적인 행동 패턴을 분석해야 하므로 인식을 위한 계산 복잡도도 비교적 높다[2].

제스처는 손이나 팔, 다리, 머리 그리고 얼굴을 이용해서 전달하는 비언어적 의사소통 수단 중의 하나이다. 가속도 센서를 이용한 제스처 인식은 언어적 의사소통 수단인 음성 인식 뿐만 아니라 카메라 영상을 이용한 제스처 인식에 비해서도 계산 복잡도가 낮고, 환경 여건의 영향을 적게 받으며 높은 인식률을 얻을 수 있다는 장점이 있다.

※ 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음.
※ 이 논문은 제38회 추계학술발표대회에서 "가속도 센서 데이터를 이용한 스마트폰 사용자의 제스처 인식"의 제목으로 발표된 논문을 확장한 것임.

† 준 회 원: 경기대학교 컴퓨터과학과 석사과정

** 준 회 원: 경기대학교 컴퓨터과학과 학사과정

*** 종신회원: 경기대학교 컴퓨터과학과 교수

논문접수: 2013년 2월 18일

수정일: 1차 2013년 3월 13일

심사완료: 2013년 3월 13일

* Corresponding Author : In-Cheol Kim(kic@kyonggi.ac.kr)

본 논문에서는 스마트폰 사용자를 위한 가속도 센서 기반의 제스처 인식 방법을 제안한다. 제안하는 제스처 인식 방법에서는 DTW(Dynamic Time Warping) 알고리즘을 적용하여 새로운 시계열 가속도 데이터와 각 제스처별 대표 훈련 데이터간의 유사도를 측정된 뒤, k-NN(Nearest Neighbors) 알고리즘을 적용하여 제스처를 판별한다. 본 논문에서는 새로운 제스처 인식 방법의 성능 평가를 위해 구현한, 안드로이드 스마트폰용 제스처 인식 프로그램과 이것을 활용한 제스처 기반 원격 제어 로봇 시스템에 대해서도 설명한다. 끝으로, 본 논문에서는 성능 평가를 위해 수행한 사용자-혼합 및 사용자-독립 실험 결과들을 소개한다.

2. 관련 연구

Cho의 연구[3]에서는 휴대폰 내장 가속도 센서를 이용해 제스처를 인식하는 방법을 제안하였다. 이 연구에서는 특징점(feature point)들을 중심으로 가속도 신호를 몇 개의 기본 조각(primitive)들로 나누고, 이 기본 조각들 간의 관계를 학습한 베이저안 네트워크(Bayesian networks) 모델을 이용함으로써 제스처를 1차 분류하였다. 그리고 분류가 어려운 비슷한 제스처들에 대해서는 SVM(Support Vector Machine)을 이용한 2차 분류를 통해 분류 성능을 향상시켰다. 한편, Oh의 연구[4]에서는 연속 가속도 데이터로부터 모션 벡터(motion vector) 시퀀스를 생성하고, 여기에 모수적 은닉 마코프 모델(Parametric Hidden Markov Model)을 적용함으로써 9개의 서로 다른 제스처들을 인식하였다. Akl의 연구[5]에서는 훈련 단계에서 DTW와 AP(Affinity Propagation)을 이용해 유사한 제스처 데이터들을 군집화(clustering)하고, 각 군집별로 대표(exemplar)를 선정할 다음, 테스트 단계에서 테스트 데이터와 이 군집 대표들을 각각 비교함으로써 가장 유사한 제스처 유형을 결정하는 방식을 제시하였다. 하지만 Alk 연구에서 생성하는 각 군집에는 서로 다른 제스처 타입의 데이터들이 함께 포함될 수 있기 때문에, 각 군집 대표가 어떤 제스처 타입을 나타낼지 미리 알 수 없다.

한편, Neizen의 연구[6]에서는 제스처 인식을 위한 대표적인 3 가지 학습 알고리즘들에 대한 성능 비교 실험을 수행하였다. 비교 대상 알고리즘들은 은닉 마코프 모델(HMM), 인공신경망(ANN) 그리고 DTW 등이며, 계산 효율성 및 메모리 효율성 등을 비교하였다. 이 연구에서는 DTW 알고리즘이 다른 두 알고리즘들에 비해 계산 효율성과 메모리 효율성면에서 가장 우수한 결과를 보여주었다. Marasovic의 연구[7]에서는 7개 서로 다른 제스처들에 대한 인식 효율성 향상을 위해, 가속도 데이터로부터 추출한 서로 다른 37개의 특징들에 주성분 분석(Principal Component Analysis) 기술을 적용하여 특징 선택(feature selection)과 차원 축소(dimensionality reduction)를 수행하였다.

지능 로봇, 컴퓨터 게임 분야 등 제스처 인식 기술을 활용한 다양한 응용 연구들이 존재한다. 이 중에서 Neto의 연구[8]에서는 사용자 팔에 부착한 가속도 센서를 이용해 사용자의

실시간 제스처 동작을 인식하고 이것을 로봇 팔에 전달하여 그대로 따라할 수 있도록 함으로써, 편리하게 로봇 작업을 수행하는 연구를 진행하였다. Uribe의 연구[9]에서는 게임용 조이스틱(Joystick)과 Kinect 센서, Wiimote 등을 함께 이용하여 사용자의 제스처를 인식하고, 이것을 기초로 이동 로봇의 움직임을 제어하는 연구를 진행하였다. 앞선 두 연구는 별도의 가속도 입력 장치를 사용한다는 면에서 본 논문의 연구와는 차별화된다. Serafimov의 연구[10]에서는 안드로이드 폰 내장 가속도 센서의 입력 데이터에 은닉 마코프 모델(HMM)을 적용하여 제스처 명령을 인식하고, 이것을 블루투스(bluetooth)를 통해 축구 레고 로봇에 전달하여 수행하는 연구를 수행하였다. 이 연구 역시 스마트폰 내장 가속도 센서를 이용한다는 면에서는 본 연구와 유사하나, DTW 기반의 제스처 인식 방법, 무선 인터넷과 로봇 내장 컴퓨터, 표준 로봇 운영체제 ROS 등을 이용하는 본 연구의 원격 제어 로봇 시스템과는 구성과 운영방식에서 큰 차이가 있다.

3. 가속도 센서 기반의 제스처 인식

3.1 처리 과정

본 논문에서는 스마트 폰 내장 3축 가속도 센서를 이용한 효과적인 제스처 인식 방법과 구현 시스템을 소개한다. 본 논문에서 인식 대상으로 삼는 제스처의 종류는 Table 1과 같다. 스마트폰 사용자가 스마트폰을 한 손에 쥔 채 취할 수 있는 제스처의 종류는 사용 목적에 따라 매우 다양하게 정의할 수 있으나, 본 논문에서는 이동 로봇(mobile robot) 제어, 3D 게임과 같은 가상 환경(virtual environment)에서 캐릭터 제어, 스마트폰 앱(smartphone app) 구동 및 제어 등의 목적을 위해 공통적으로 사용될 수 있는 가장 보편적인 제스처 6 가지를 선정하였다.

Table 1. The gesture set

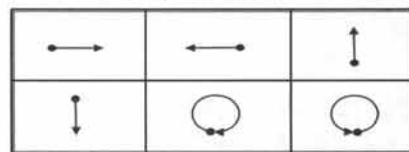


Table 2. Acceleration patterns for gestures

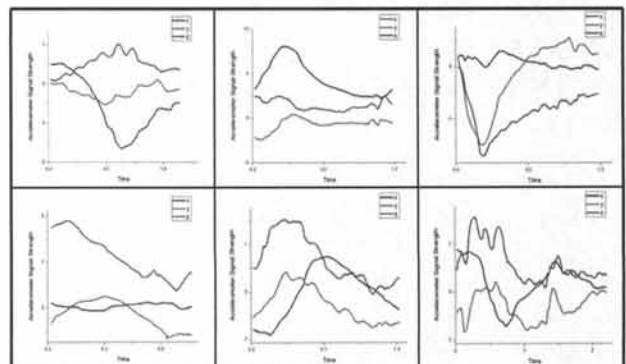


Table 2는 임의의 사용자가 각각의 제스처를 취했을 때, 스마트폰 내장 3축 가속도 센서로 감지되는 전형적인 x, y, z축 가속도 값의 변화를 나타낸 것이다. Table 2를 통해, 우리는 각 제스처별로 비교적 고유한 3축 가속도 입력 값의 변화 패턴을 보인다는 것을 확인할 수 있다. 하지만 다른 센서 데이터를 이용한 제스처 인식과 마찬가지로, 스마트폰 내장 가속도 센서를 이용한 제스처 인식도 동일한 제스처라도 제스처를 취하는 사용자마다, 또 같은 사용자라도 제스처를 취할 때마다 입력 가속도 데이터에 조금씩 차이가 있을 뿐 아니라, 무시하기 어려울 정도의 잡음(noise) 또한 포함하고 있어 안정적인 인식 성능을 얻기 어렵게 한다.

본 연구의 제스처 인식을 위한 전체 처리 과정은 Fig. 1과 같다. 스마트폰 내장 3축 가속도 센서를 통한 입력 데이터는 크게 제스처 인식기 학습을 위한 훈련 데이터(training data)와 실시간 제스처 인식을 위한 테스트 데이터(testing data)로 나뉜다. 이 두 입력 데이터는 공통적으로 스마트폰에 내장된 3축 가속도 센서를 통해 얻어지고, 미리 정해 놓은 형식에 맞춰 파일형태로 스마트폰에 저장된다.

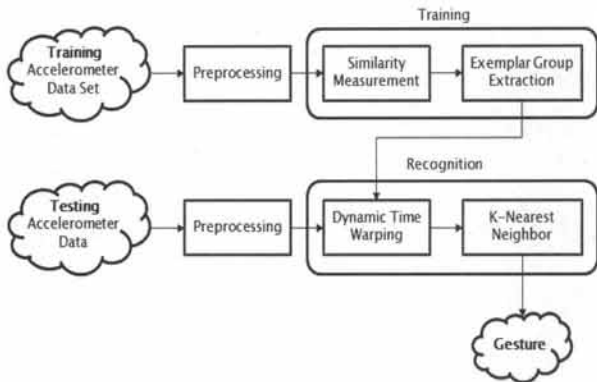


Fig. 1. The overall process

훈련 데이터와 테스트 데이터는 전 처리(preprocessing) 단계에서 일부 불필요한 데이터를 삭제하는 과정을 거친다. 그리고 훈련 단계(training phase)에서는 DTW 알고리즘을 이용해 훈련 데이터들 간의 유사도 측정하여 유사도 행렬(similarity matrix)을 생성하고, 이를 바탕으로 제스처별 대표 집단 추출(exemplar group extraction)을 수행한다. 인식 단계(recognition phase)에서는 실시간으로 입력된 테스트 데이터를 제스처별 대표 집단과 비교하여 유사도를 측정한 뒤, k-근접이웃(k-Nearest Neighbors, k-NN) 알고리즘을 적용해 최종적으로 어떤 제스처인지 판별한다.

3.2 데이터 수집 및 전 처리 단계

데이터 수집(data collection)은 3축 가속도 센서가 내장된 안드로이드(Android) 기반 스마트폰에서 이루어진다.

$$\begin{aligned}
 S_i &= \langle ID_i, User_i, G_i, AccSeq_i \rangle \\
 G_i &\in \{G_1, G_2, \dots, G_6\} \\
 AccSeq_i &= [\langle Acc_{x_1}, Acc_{y_1}, Acc_{z_1} \rangle, \dots, \langle Acc_{x_m}, Acc_{y_m}, Acc_{z_m} \rangle] \quad (1)
 \end{aligned}$$

위 식(1)은 파일에 저장되는 데이터 형식을 나타낸 것이다. 제스처 데이터(S_i)는 일련번호(ID_i), 사용자 이름($User_i$), 제스처 종류(G_i) 그리고 3축 가속도 벡터 $\langle Acc_{x_i}, Acc_{y_i}, Acc_{z_i} \rangle$ 들의 시퀀스($AccSeq_i$) 등으로 구성된다.

앞서 언급했듯이 제스처 데이터는 사용자에게 상당히 의존적이다. 제스처 인식의 사용자 의존성을 분석해 보기 위해 제스처를 취한 사용자의 이름도 데이터 라벨(label)의 한 부분으로 포함시켰다. 제스처 종류는 Table 1의 6가지 종류로 제한하였다. 전 처리 단계(preprocessing phase)에서는 훈련 및 테스트 데이터에 포함된 불필요한 데이터 부분을 삭제한다. 각 제스처에 대한 데이터 수집은 시작 버튼을 누르는 순간부터 때는 순간까지 3축 가속도 센서 값을 기록함으로써 이루어진다. 이때 각 제스처를 나타내는 가속도 벡터 시퀀스($AccSeq_i$)의 시작 부분과 끝 부분에는 제스처의 한 부분으로 보기 어려운 불필요한 가속도 데이터 값이 포함될 가능성이 높으므로, 모든 제스처 데이터(S_i)에서 가속도 벡터 시퀀스($AccSeq_i$)의 앞부분과 뒷부분에서 각각 5% 씩 총 10%의 데이터를 삭제한다.

3.3 훈련 단계

훈련 단계(training phase)에서는 DTW 알고리즘을 사용하여 유사도 행렬(similarity matrix)을 구하는 과정과 이를 이용한 대표 추출(exemplar extraction) 과정이 진행된다. DTW(Dynamic Time Warping) 알고리즘은 시간이나 속도가 다른 두 시퀀스 데이터 간의 유사도를 측정하는 대표적인 알고리즘으로서, 주로 시계열 패턴 인식 분야에 폭넓게 적용되어 왔다. 이 알고리즘은 두 시퀀스의 시작 성분부터 끝 성분까지 동적 계획법을 적용하면서, 두 시퀀스 간의 최적의 비선형 대응 함수(nonlinear mapping function)를 찾는다. DTW 알고리즘의 결과 값은 두 시퀀스 간의 거리(distance) 혹은 차이(difference)를 나타내므로, 이 값이 작을수록 두 시퀀스 데이터가 유사하다는 것을 의미한다. 본 논문에서 다루는 제스처 입력 데이터도 각 제스처의 경로에 따른 시계열 가속도 데이터이므로, 입력 데이터 간의 유사도 측정을 위해 효과적인 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 데이터 수집 단계에서 각 축(x축, y축, z축)별로 나뉘어 3축 가속도 데이터가 기록된다. 따라서 아래의 식 (5)를 이용해 각 축별 거리를 결합함으로써, 최종적인 두 제스처 데이터 간의 유사도를 계산한다.

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

이러한 DTW 알고리즘은 시계열 데이터 분류(time series data classification)를 위한 은닉 마코프 모델(HMM)이나 인공 신경망(ANN)과 같은 다른 알고리즘들에 비해, 특별히 모델 학습을 위한 단계가 필요하지 않고, 인식 속도도 빠르다는 장점이 있다[6]. 이러한 DTW의 특성은 스마트폰 상에서 수행되는 실시간 제스처 인식 시스템을 구현할 때 큰 장점이 될 수 있다.

훈련 단계의 첫 번째 과정은 각 제스처 별로 식 (6)과 같은 유사도 행렬(SM_i)을 만드는 것이다.

$$SM_i = \begin{bmatrix} sm_{1,1}^i & sm_{1,2}^i & \dots & sm_{1,z}^i \\ sm_{2,1}^i & sm_{2,2}^i & \dots & sm_{2,z}^i \\ \vdots & \vdots & \ddots & \vdots \\ sm_{z,1}^i & sm_{z,2}^i & \dots & sm_{z,z}^i \end{bmatrix} \quad (6)$$

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

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 번째 과정은 식 (8)과 같이 합 행렬(TM_i)을 구한다.

$$TM_i = [tm_1^i, tm_2^i, \dots, tm_z^i] \quad (8)$$

$$tm_r^i = \sum_{s=1}^z sm_{r,s}^i$$

합 행렬의 요소(tm_r^i)는 해당 훈련 데이터가 같은 제스처를 나타내는 다른 훈련 데이터들로부터 어느 정도의 거리에 놓여있는지를 나타낸다. 즉, 합 행렬(TM_i)에서 가장 작은 값 tm_r^i 을 가지는 훈련 데이터가 다른 훈련 데이터들과 거리의 합이 가장 작다는 것을 알 수 있고, 이것은 곧 이 훈련 데이터가 같은 제스처를 나타내는 훈련 데이터 집합의 가장 중심에 위치한다는 것을 의미한다. 따라서 합 행렬을 토대로 각 제스처 타입별로 훈련 데이터 집합의 가장 중심에 위치한 대표(exemplar)를 선정할 수 있다. 이렇게 선정된 대표는 추후 제스처 인식 단계에서 제스처 판별을 위해 테스트 데이터와의 비교에 이용된다.

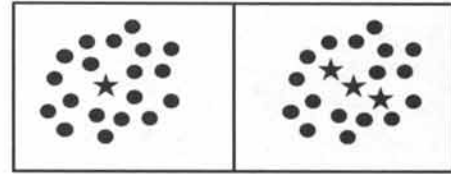


Fig. 2. Two different ways of exemplar extraction

Fig. 2는 두 가지 대표 추출 방법을 비교해서 나타낸 그림이다. 왼쪽은 각 제스처 별로 한 개의 대표(single exemplar)를 선정하는 방법이고, 오른쪽은 다수의 대표들(multiple exemplars)을 선정하는 방법이다. 많은 기존 연구들에서는 각 제스처를 나타내는 훈련 데이터 집합에서 그 제스처를 가장 잘 표현할 수 있는 대표 훈련 데이터를 하나씩만 선정 한 뒤, 테스트 데이터와의 비교를 통해 제스처를 판별하였다. 다수의 훈련 데이터들에서 최선의 방법으로 단일 대표를 선정한다고 하더라도, 결국 다른 관점에서 보면 각 제스처별로 훈련 데이터를 하나만 사용하는 것이다. 예컨대, 각 제스처별 훈련 데이터들 중 가장 중앙에 위치한 데이터를 그 제스처의 단일 대표로 선정한다고 하더라도, 각 제스처를 나타내는 훈련 데이터 수가 많고 훈련 데이터들 간의 변이가 심해 데이터 분포가 넓게 퍼질 경우, 단일 대표로 선정된 하나의 훈련 데이터만으로는 제스처 데이터들의 다양성을 충분히 표현하기에는 부족하다. 이러한 점을 보완하기 위해, 본 논문에서는 Fig. 3의 오른쪽과 같이 각 제스처별로 훈련 데이터 집합에서 그 제스처를 가장 잘 나타낼 수 있는 다수의 대표들(multiple exemplars)을 선정하는 방법을 제안한다. 이 방법에 따르면, 각 제스처 별 훈련 데이터들 중 가장 중앙에 위치한 약 25%의 데이터들을 해당 제스처를 나타내는 대표들로 선정한다. 이를 위해 앞서 설명한 합 행렬 계산을 통해 다른 훈련 데이터들과의 거리가 가장 작은 것부터 오름차순으로 정렬한 다음, 상위 25%의 훈련 데이터들을 대표들로 선정한다. 이 방법은 대표들의 수가 증가함에 따라 인식을 위한 비교 시간이 다소 증가하는 한계점은 있지만, 제스처 인식 적용 대상자인 사용자들 간의 변이, 스마트폰 종류에 따른 가속도 센서의 차이 등을 극복하고 제스처 인식의 정확도를 향상시키는데 도움을 줄 수 있다.

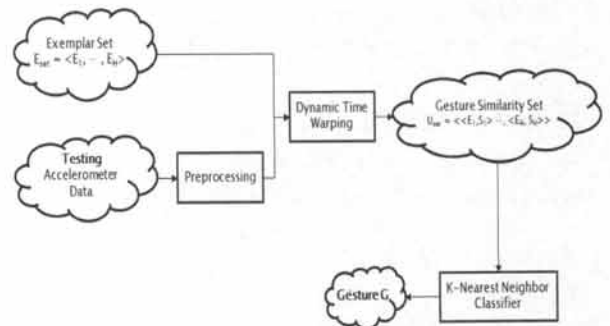


Fig. 3. The recognition phase

3.4 제스처 인식 단계

위와 같은 훈련 단계를 거쳐 선정된 제스처별 대표 집합(exemplar set)을 바탕으로, Fig. 3과 같은 실시간 제스처 인식 단계(recognition phase)가 수행된다. 제스처 인식 단계는 세부적으로 2단계로 구분할 수 있다. 첫 번째 단계는 전 처리가 완료된 테스트 데이터와 대표 집합(E_{set})의 각 멤버들 간의 유사도를 DTW 알고리즘을 이용하여 측정한다. 그 결과로 제스처 유사도 집합(U_{set})이 얻어진다. 이 집합은 대표 집합의 각 멤버(E_i)들이 테스트 데이터와 어느 정도 유사한 지 유사도(S_i)를 함께 표현한 것이다. 그 다음 단계는 k-최근접 이웃 알고리즘(k-Nearest Neighbors)을 이용해서 테스트 데이터와 가까운 k개의 후보 제스처 집합(G_{set})을 골라내고, 이 후보 제스처 집합으로부터 다수결 원칙에 의해 하나의 제스처로 최종 판별해낸다.



Fig. 4. Screenshots of the implemented gesture recognition system

4. 구현과 응용

앞서 제시한 스마트폰 가속도 센서 기반의 제스처 인식 방법을 토대로, 실제로 안드로이드 스마트폰용 제스처 인식 프로그램을 구현하였다. Fig. 4는 각각 데이터 수집 단계와 실시간 제스처 인식 단계를 위한 안드로이드 어플리케이션의 실행 화면들을 보여주고 있다. 왼쪽의 수집 탭을 통해서 제스처별 데이터 수집, 파일 삭제, 훈련과 초기화 기능 등이 제공되고, 오른쪽의 인식 탭을 통해서는 훈련된 인식기를 새로운 입력 데이터에 적용하여 제스처를 판별하는 실시간 제스처 인식 기능이 제공된다.

본 연구에서 개발된 스마트폰 가속도 센서 기반의 제스처 인식 시스템은 인간-컴퓨터 인터페이스(HCI), 인간-로봇 인터페이스(HRI) 분야 등 다양한 응용분야에 활용될 수 있다. 본 논문에서는 앞서 제안한 제스처 인식 방법을 활용해 원격 제어 이동 로봇 시스템을 개발한 응용 사례를 소개한다. 스마트폰 가속도 센서 기반의 제스처 인식 기능을 이용한 원격 로봇 제어는 무선 리모콘과 같은 별도의 부가 장비 없이도 근거리망(LAN)이나 인터넷을 통해 비교적 원거리의

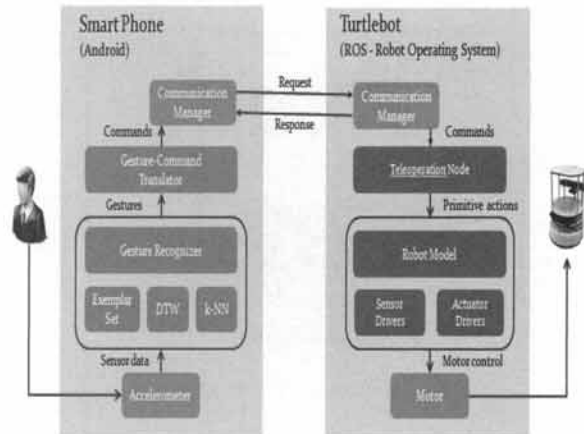


Fig. 5. A gesture-based teleoperating robot system

로봇 제어도 가능하다는 잇점이 있을 뿐만 아니라, 음성이나 영상 제스처 인식 기술을 이용한 방법에 비해 소음이나 조명 변화 등 주변 환경의 여건에도 매우 강건한 신뢰성을 얻을 수 있고, 사용자 편의성도 높다는 장점이 있다.

본 연구에서 개발한 제스처 기반의 원격 제어 로봇 시스템의 전체 구성은 Fig. 5와 같다. 전체 시스템은 스마트폰 사용자의 2차원 제스처 명령을 실시간으로 인식해 서버로 전송하는 클라이언트 부분과 스마트폰으로부터 받은 제스처 명령에 따라 이동 동작들을 실제로 수행하는 로봇 서버로 구성된다. 스마트폰 클라이언트와 로봇 서버 간에는 무선 인터넷인 Wi-Fi를 통해 메시지가 전달된다.

Turn Right	Turn Left	Go Forward	Stop	Rotate Right	Rotate Left

Fig. 6. Gestures vs robot action commands

스마트폰 사용자의 제스처와 이에 대응하는 로봇 동작 명령의 종류는 Fig. 6과 같다. 로봇은 스마트폰 사용자의 제스처에 따라 직진(Go Forward), 정지(Stop), 좌/우 회전(Turn/Rotate Left/Right)의 기본적인 동작들을 수행한다. 회전 동작 중 Turn은 미리 정해진 각도만큼 좌 또는 우로 1회 회전하는 동작을 의미하고, Rotate는 정지 또는 다른 명령이 전달될 때까지 좌 또는 우로 회전을 계속하는 동작을 의미한다.

이 시스템 구현에 사용한 이동 로봇은 Fig. 7과 같은 Willow Garage사의 Turtlebot이다. iRobot사의 Create 로봇을 하부 몸체로 채택하고 있는 Turtlebot은 2개의 모터 구동바퀴와 무게 중심을 바로잡기 위한 2개의 지지바퀴 등 총 4개의 바퀴로 이동한다. Turtlebot에는 센서 데이터 처리와 의사 결정 연산, 통신 및 데이터 저장을 위해 넷북(Netbook)을 주 두뇌로 사용하며, 시각 인식을 위한 영상 입력 장치로 Microsoft사의 Kinect를 장착하고 있다.



Fig. 7. Turtlebot: a mobile robot

Turtlebot의 넷북에는 기본 운영체제로 Ubuntu 11.10과 로봇 운영체제인 ROS(Robot Operating System)를 설치하며, Wi-Fi 모듈을 통해 인터넷에 연결될 수 있다.

스마트폰 클라이언트 측에서는 사용자가 취하는 제스처에 따라 3축 가속도 측정치를 입력하여, 앞서 설명한 방식대로 대표 집합(exemplar set)과 DTW 기반의 유사도 계산을 한 다음, k-최근접 이웃 선정 방식으로 제스처를 판별하는 제스처 인식기(Gesture Recognizer)가 있고, 인식된 제스처를 로봇 동작 명령(robot action command)으로 변환해주는 제스처-명령 변환기(Gesture-Command Translator)가 있다. 이렇게 변환된 로봇 동작 명령은 통신 관리자(Communication Manager)를 통해 로봇 서버 측에 전송된다.

한편, 로봇 서버 측에서는 통신 관리자(Communication Manager)가 수신한 로봇 동작 명령을 원격 제어 모듈(Teleoperation Node)을 통해 로봇의 기본 동작들(primitive actions)로 변환한 다음, 센서 드라이브들(Sensor Drives)과 모터 드라이브들(Motor Drives)들을 포함한 가상 머신인 로봇 모델(Robot Model)을 거쳐 실제로 이동 동작을 실행하는 모터 제어 신호(motor control)로 변환한다.



(a) Stop and Go Forward actions



(b) Turn Right and Turn Left actions

Fig. 8. Robot teleoperations with smartphone gestures

Fig. 8은 구현된 원격 제어 로봇 시스템을 스마트폰 사용자의 제스처로 제어하는 모습들을 보여주고 있다. Fig. 8의 (a)는 제스처에 의해 정지(Stop) 동작과 직진(Go Forward) 동작을 수행하는 모습을 왼쪽과 오른쪽에 각각 보여주고, (b)는 우 회전(Turn Right) 동작과 좌 회전(Turn Left) 동작을 수행하는 모습을 보여주고 있다. 각 그림 속 스마트폰 화면에는 인식된 제스처들이 표시되어 있다.

Table 3. Data collection

Users	Gestures	Training	Testing	Total
10	6	10	10	1200

5. 실험 및 평가

본 논문에서 제안한 제스처 인식 방법의 성능을 평가하기 위한 실험을 전개하였다. 실험을 위해 Table 3과 같이, 총 10명의 실험자로부터 6가지 종류의 제스처 각각에 대해 훈련용과 테스트용 데이터를 각각 10회씩 수집하여 총 1200개의 제스처 데이터를 수집하였다. 제스처 데이터의 사용자 의존성 문제를 고려하여, 인식 성능을 2 가지 다른 방식으로 분석해보았다. 먼저, 사용자-혼합 실험들(user-mixed experiments)에서는 10명의 사용자에게서 수집한 600개의 훈련 데이터를 사용자 구분 없이 혼합하여 인식기를 학습한 뒤, 동일한 10명 사용자 집단으로부터 수집한 별도의 600개 테스트용 데이터로 테스트하여 인식률을 측정하였다. 또한, 사용자-독립 실험들(user-independent experiments)에서는 10명 중 임의로 선택한 5명의 사용자들로부터 수집한 300개 훈련 데이터로 인식기를 학습한 뒤, 나머지 5명의 사용자에게서 수집한 별도의 300개 데이터로 테스트하여 인식률을 측정하였다. 또한, 본 연구에서는 제스처별로 단일 대표(single exemplar)를 선정하는 방식에 비해, 본 논문에서 제안하는 방식대로 다수의 대표들(multiple exemplars)을 선정하는 방식이 어느 정도 효과가 있는지도 비교 실험해 보았다.

Table 4와 Table 5는 각각 사용자-혼합 실험과 사용자-독립 실험들의 결과를 혼동 행렬(confusion matrix)형태로 보여주고 있다. 행렬의 괄호 속 숫자는 다수 대표 선정 방식의 인식 결과를, 괄호 밖의 숫자는 단일 대표 선정 방식의 인식 결과를 각각 나타낸다. Table 4의 사용자-혼합 실험들에서는 단일 대표 선정방식이 평균 86%의 인식률을 보인 반면, 본 논문의 다수 대표 선정방식은 이보다 약 5% 향상된 평균 91%의 인식률을 보였다. 한편, 상하, 좌우 직선 동작을 이용한 비교적 단순한 4 가지 동적 제스처들은 대부분 평균 95%이상의 높은 인식률을 보인 반면, 좌 회전과 우 회전 제스처들은 이보다는 조금 낮은 평균 78~80% 정도의 인식률을 보였다. 이 결과를 통해, 본 연구에서 제안한 제스처 인식 방법의 높은 인식 성능과 사용자 독립성을 확인할 수 있다.

Table 5의 사용자-독립 실험들에서는 단일 대표 선정방식이 평균 81%의 인식률을 보였고, 본 논문의 다수 대표 선

정방식은 이보다 약 8% 향상된 평균 89%의 인식률을 보였다. 이 결과는 사용자-혼합 실험의 경우보다는 인식률이 조금 낮지만, 사용자-독립 실험의 경우에도 여전히 본 논문에서 제안한 제스처 인식 방법의 높은 인식 성능을 확인할 수 있다.

Table 4. Confusion matrix for user-mixed experiments

	→	←	↑	↓	○	○
→	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)

Table 5. Confusion matrix for user-independent experiments

	→	←	↑	↓	○	○
→	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)

Table 6. Result for robot teleoperation experiments

Total	Successes	Failures	Success Rate
100	98	2	98%

또한, 본 논문에서는 제스처를 이용한 원격 제어 로봇 실험을 수행하고 그 결과를 분석해 보았다. 먼저, 사용자의 제스처에 따라 로봇이 올바른 동작을 수행하는지 실험하였다. Table 6은 스마트폰 사용자가 임의로 선택한 서로 다른 제스처 명령을 100회 수행하고, 이 중 몇 번을 성공적으로 제스처 명령을 인식하고, 해당 동작을 로봇이 정상 수행하였는지 실험한 결과를 보인다. 총 100회 실험 중, 제스처 인식 오류나 인식된 제스처와 다른 동작을 수행한 경우는 없었고, 단지 로봇 동작이 정상적으로 수행을 완료하지 못한 경우는 2번 있었다. 따라서 본 실험에서는 약 98%의 제스처 명령 실행 성공률을 보였다.

한편, 제스처 명령을 이용한 원격 로봇제어에서 시간 효율성을 알아보기 위해, 각 단계별 수행 시간을 분석해보았다. Table 7은 단계별 수행 시간을 분석한 결과를 보여준다.

Table 7. Time consumption for robot teleoperation

Step	Time (sec)	Acc. Time (sec)
Gesture recognition	0.25	0.25
Command translation	0.15	0.40
Communication	0.90	1.30
Primitive Action translation	0.13	1.43
Motor control	0.03	1.46

Table 7을 통해, 스마트폰 사용자의 제스처 명령 인식에서 시작하여 로봇이 해당 동작 수행을 완료할 때까지 평균 1.46초 정도의 시간이 소요되었음을 알 수 있다. 이것은 사용자가 제스처로 네트워크를 통해 로봇을 원격 제어 할 때 2초 이내에 로봇이 빠르게 반응할 수 있다는 것을 의미한다. 단계별 수행 시간을 살펴보면, 제스처 인식 단계는 약 0.25초로서 전체 수행 시간의 약 17%만을 차지한 반면, 스마트폰과 로봇 간 통신(communication) 단계는 약 0.9초로서 전체 수행 시간의 약 62%를 차지하였다. 제스처 명령이 전달되는 무선 인터넷 상황에 따라 통신 시간은 달라질 수도 있을 것이다. 이 실험결과를 통해 제스처 명령을 이용한 원격 로봇 제어에서 전체 수행 시간에 비해 제스처 인식을 위한 지연 시간은 크지 않고, 제스처 인식이 매우 효율적으로 이루어졌음을 알 수 있다.

6. 결론

본 논문에서는 DTW 알고리즘과 k-NN 알고리즘을 근간으로 하는 효과적인 스마트폰 가속도 센서 기반의 제스처 인식 방법을 제안하였다. 그리고 사용자간 변이를 감안하여 훈련 단계에서 각 제스처마다 그 제스처를 대표할 다양한 다수의 대표들(multiple exemplars), 즉 대표 집단(exemplar group)을 선정하는 방식을 제안하였다. 새로 제안한 제스처 인식 방법을 기초로 안드로이드 스마트폰에서 동작하는 제스처 인식 프로그램을 구현하였고, 이것을 활용하여 제스처 기반 원격 제어 로봇 시스템도 구현하였다. 사용자-혼합 실험 및 사용자-독립 실험들을 통해, 본 논문에서 제안한 제스처 인식 방법과 구현 시스템이 높은 인식 성능과 확장성을 가진다는 것을 입증하였다.

향후 연구에서는 제스처별 대표 집단 선정 작업을 위해 가변 길이의 시계열 데이터들에 대한 효과적인 군집화(clustering) 방법을 적용하는 연구와 기존의 제스처 인식 연구들에서 많이 이용된 은닉 마코프 모델(Hidden Markov Model, HMM)[10]과의 실험적 비교 등을 진행할 계획이다.

참고 문헌

- [1] J. Kim, S. Nam, S. Heo, I. Kim, "Design of an Activity Recognition System using Smartphone Accelerometer", KIPS

Transaction on Software and Data Engineering, Vol.2, No.1, pp.1-6, 2013.

- [2] J. Liu, L. Zhong, J. Wickramasuriya, V. Vasudevan, "uWave: Accelerometer-based Personalized Gesture Recognition and its Applications", Pervasive and Mobile Computing, Vol.5, pp.657-675, 2009.
- [3] S. Cho, E. Choi, W. Bang, J. Yang, "Two-stage Recognition of Raw Acceleration Signals for 3-D Gesture-Understanding Cell Phones", Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, 2006.
- [4] K. Oh, Y. Jeong, S. Kim, H. Choi. "Gesture Recognition Application with Parametric Hidden Markov Model for Activity-Based Personalized Service in APPiME", Proceedings of the First IEEE International Conference on Situation Awareness and Decision Support, 2011.
- [5] A. Akl, C. Feng, S. Valaee, "A Novel Accelerometer-Based Gesture Recognition System", IEEE Transaction on Signal Processing, Vol.59, No.12, pp.6197-6205, 2011.
- [6] G. Neizen, G. P. Hancke, "Evaluating and Optimizing Accelerometer-based Gesture Recognition Techniques for Mobile Devices", Proceedings of IEEE AFRICON-09, 2009.
- [7] T. Marasovic, V. Papic, "Accelerometer-Based Gesture Classification Using Principal Component Analysis", Proceedings of the 19th IEEE International Conference on Software, Telecommunications and Computer Networks, 2011.
- [8] P. Neto, J. N. Pires, A. P. Moreira, "Accelerometer-Based Control of an Industrial Robotic Arm", Proc. of the 18th IEEE International Symposium on Robot and Human Interactive Communication, 2009.
- [9] A. Uribe, et al., "Mobile Robotic Teleoperation using Gesture-Based Human Interfaces", Proceedings of IEEE Latin American and Colombian Conference on Automatic Control and Industry Applications(LARC), 2011.
- [10] K. Serafimov, et al., "Using Mobile-phone Accelerometer for Gestural Control of Soccer Robots", Proceedings of Mediterranean Conference on Embedded Computing(MECO), 2012.



남 상 하

e-mail : namsh@kyonggi.ac.kr

2013년 경기대학교 컴퓨터과학과(학사)
2013년~현 재 경기대학교 컴퓨터과학과
석사과정

관심분야: 인공지능, 기계학습, 지능로봇



김 주 희

e-mail : chilang@kyonggi.ac.kr

2003년~현 재 경기대학교 컴퓨터과학과
학사과정

관심분야: 인공지능, 기계학습,
모바일컴퓨팅



허 세 경

e-mail : tprudzzang@kyonggi.ac.kr

20008년~현 재 경기대학교 컴퓨터과학과
학사과정

관심분야: 인공지능, 기계학습,
모바일컴퓨팅



김 인 철

e-mail : kic@kyonggi.ac.kr

1985년 서울대학교 수학과 (학사)
1987년 서울대학교 전산과학과(이학석사)
1995년 서울대학교 전산과학과(이학박사)
1996년~현 재 경기대학교 컴퓨터과학과
교수

관심분야: 인공지능, 기계학습, 지능형시스템