

Robust User Activity Recognition using Smartphone Accelerometer Sensors

Jeon Myung Joong[†] · Park Young Tack^{**}

ABSTRACT

Recently, with the advent of smart phones, it brought many changes in lives of modern people. Especially, application utilizing the sensor information of smart phone ,which provides the service adapted by user situations, has been emerged. Sensor data of smart phone can be used for recognizing the user situation, Because it is closely related to the behavior and habits of the user. currently, GPS sensor one of mobile sensor has been utilized a lot to recognize basic user activity. But, depending on the user situation, activity recognition system cannot receive GPS signal, and also not collect received data. So utilization is reduced. In this paper, for solving this problem, we suggest a method of user activity recognition that focused on the accelerometer sensor data using smart phone. Accelerometer sensor is stable to collect the data and it's sensitive to user behavior. Finally this paper suggests a noble approach to use state transition diagrams which represent the natural flow of user activity changes for enhancing the accuracy of user activity recognition.

Keywords : Smart Phone, Accelerometer, User Activity Recognition, State Transition Diagram

스마트폰 가속도 센서를 이용한 강건한 사용자 행위 인지 방법

전 명 중[†] · 박 영 택^{**}

요 약

최근 몇 년 동안 스마트폰의 등장으로 현대인들의 생활에 많은 변화를 가져왔다. 특히 스마트폰의 센서 정보를 활용하여 사용자의 상황에 맞는 서비스를 제공하는 응용프로그램들이 많이 등장하고 있다. 스마트폰의 센서 정보는 사용자의 습관이나 행동과 밀접하게 관련되어 있기 때문에 사용자의 상황을 인지하기에 좋은 데이터이다. 현재 모바일 센서 중 GPS 센서는 사용자의 기본적인 행위인지에 많이 활용되고 있다. 하지만 GPS 센서는 사용자의 상황에 따라 수신이 불가능할 수도 있으며 수신된 데이터 역시 부정확할 수 있기 때문에 활용도가 떨어진다. 본 연구에서는 이러한 문제점을 해결하기 위해 모바일 디바이스에 탑재된 가속도 센서 데이터를 중심으로 한 사용자 행위 인지 방법을 제안한다. 가속도 센서는 데이터 수신이 안정적이며, 사용자의 행위에 민감하게 반응하기 때문에 행위인지에 적합하다. 마지막으로 상태 전이도를 활용하여 합리적인 행위변화의 흐름을 적용함으로써 행위인지의 정확도를 높인다.

키워드 : 스마트폰, 가속도, 사용자 행위 인지, 상태 전이도

1. 서 론

최근 스마트폰의 보급으로 현대인들의 생활에 많은 변화를 가져왔다. 그 변화 중 하나는 개인 맞춤형 서비스이다. 기존의 수동적이고 일률적인 서비스 제공방식을 벗어나 사용자의 상황에 맞는 유용한 정보가 선별적으로 제공된다. 이러한 개인 맞춤형 서비스를 제공하기 위해서는 우선 사용자 현재 상황을 정확하게 인지하는 기술이 필요하다.

사용자의 상황(Context)은 물리적 상황, 시간적 상황 등 여러 가지로 나눌 수 있지만 본 논문에서는 물리적 상황으로 규정한다.

스마트폰이 보급되기 전에는 각 신체부위별로 센서 장비를 부착하여 데이터를 수집하고 분석하였다. 하지만 비용 문제와 착용의 불편함으로 인해 대중화 되지 못하였다. 최근에 와서 스마트폰의 대중화를 통해 사용자의 행위를 인지하기에 아주 좋은 환경이 갖추어 졌다. 스마트폰은 사용자가 항상 지니고 다니기 때문에 사용자의 습관이나 행동과 밀접한 관련이 있다. 특히 스마트폰의 센서 데이터를 활용하면 좀 더 정확한 상황을 인지하는데 도움이 된다.

스마트폰에는 다양한 센서 장비가 부착되어 있다. 특히 GPS 센서를 활용하면 사용자의 위치뿐만 아니라 이동속도

※ 이 논문은 2010년도 지식경제부의 재원으로 모바일 플랫폼 기반 계획 및 학습인지 모델 프레임워크 개발사업의 지원을 받아 수행된 연구임.

† 준 회 원: 숭실대학교 컴퓨터공학과 석사과정

** 정 회 원: 숭실대학교 컴퓨터공학과 교수

논문접수: 2013년 7월 16일

심사완료: 2013년 8월 27일

* Corresponding Author : Park Young Tack(park@ssu.ac.kr)

까지 알아낼 수 있다. 이를 통해 사용자의 위치정보, 기본행위를 인지할 수 있고 최종적으로 사용자에게 맞는 서비스를 제공할 수 있다.

하지만 GPS 센서는 기본적으로 위성장비로부터 위치정보를 받기 때문에 실내에서는 신호 방해로 인해 사용이 불가능하다. 또한 흐린 날씨 상황이나 주변에 고층빌딩이 많은 경우 GPS 데이터 수신에 불안정하다. 그리고 유사한 상황에서 사용자의 행위를 구분하기에 어려움이 있다. 예를 들어 사용자가 서 있는 경우와 사용자가 차를 탄 상태에서 신호등으로 인해 정차해 있을 경우, 두 상황 모두 속도가 0으로 같다. 이러한 경우는 사용자의 행위를 구분하기에 어려움이 있다. 마지막으로 GPS 센서는 배터리 소모가 심하다. 이러한 요인들로 인해 실생활에 적용하기에는 문제점이 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 가속도 센서 중심의 상황인지 방식을 제안한다. 가속도 센서는 실외뿐만 아니라 실내에서도 잘 동작하며, 데이터 또한 손실 없이 안정적으로 수집이 가능하다. 그리고 사용자의 행위에 민감하게 반응하기 때문에 유사한 행위를 인지하는데 유리하다. 뿐만 아니라 배터리 소모도 적다. [9]에서 확인할 수 있듯이 가속도 센서를 사용했을 경우와 GPS 센서를 사용했을 경우를 비교해보면 확연한 차이점을 보인다.

2. 관련 연구

A. M. Khan 등의 연구에서는 사용자가 스마트폰을 지니고 있는 위치에 상관없는 특징을 추출하여 AR(Autoregressive) 방법을 사용하여 사용자의 행위를 인지하는 방법을 제안하고 있다[1].

Yu-Chieh Yang 등은 가속도 데이터를 FFT(Fast Fourier Transformation) 한 특징만을 사용하여 사용자의 행위를 인지하였다. 하지만 실험 결과에서 이동수단의 인지 정확도는 낮게 나왔다[2].

2011년 미국 MIT에 SENSEable City Lab에서 이동수단을 인지하는 방법에 대한 연구를 하였다. 사용자 행위의 종류는 Bus, Metro, Walk, Bicycle, Train, Car, Still, Motorcycle 이다. 25 Hz 단위로 샘플링을 하였으며 Car 와 Motorcycle은 한명이 수집하였다. 그리고 나머지 행위는 4명이 수집하였다.

$$\hat{a}_{tot}(t) = \sqrt{\hat{a}_x(t)^2 + \hat{a}_y(t)^2 + \hat{a}_z(t)^2} \quad (1)$$

가속도 데이터는 기본적으로 x, y, z 축의 값으로 이루어져 있으며 방향에 따라 축의 값이 변한다. 즉, 같은 상황일지라도 스마트폰의 방향에 따라 3축의 값이 달라질 수 있다. 그래서 보통 (1)과 같이 각 축의 제곱 값의 합에 루트를 씌워 방향성을 제거한다.(여기서 방향성이라 함은 스마트폰의 tilting을 의미) 즉, 가속도의 크기만을 사용하여 행위를 구

분한다는 것이다. 하지만 이 방법에는 단점이 있다. 사용자의 행위가 전혀 다른 상황에서 가속도의 크기가 비슷한 경우에 문제가 발생한다. 예를 들면, 걷는 경우의 스마트폰 가속도 데이터는 상, 하로 뿐만 아니라 좌, 우, 앞, 뒤의 움직임이 반영되어 나타난다. 하지만 위의 (1)식을 적용하면 이러한 방향성은 제거되기 때문에 가만히 서있는 상태에서 스마트폰을 아래, 위로만 움직여도 가속도의 크기가 비슷하다면 걷는 행위로 인식할 수 있다[3].

또한 Leon Stenneth 등은 가속도 센서데이터와 함께 이동수단 네트워크 정보를 함께 사용하여 bus등의 이동수단을 인지하는 방법을 사용하였다. 이 방법은 높은 정확도를 보여주지만 특정 지역에 국한된 이동수단 네트워크 데이터를 사용하기 때문에 범용적으로 활용하기에는 한계가 있다[4].

Xi Long 연구에서는 의사결정 트리보다는 베이지언 분류기를 사용하여 모델을 생성하였다. 그 이유로 의사결정 트리는 사용자 행위의 종류나 특징 값의 종류가 바뀌었을 경우 새롭게 튜닝을 해야 하기 때문이다. 실험 결과 의사결정 트리보다 정확도는 낮지만 유연성을 갖추게 되었다[13].

또 다른 연구로는 ARTS Lab.에서 몸에 부착 가능한 형태의 센서를 사용하여 사용자의 행위를 인지하는 연구를 하였다. 7가지의 행위를 인지하며 9가지의 서로 다른 구분자를 사용하여 실험을 하였으며 그 중에서 HMM(Hidden Markov Models)을 사용하여 모델을 생성하였다. 이 연구에서 HMM으로도 높은 정확도를 보이는 모델을 생성할 수 있음을 보였으나 몸에 부착된 형태의 센서 데이터는 실생활의 활용도 측면에서 부족할 수 있다[14].

본 연구에서는 위와 같은 문제점을 해결하기 위해 가속도 데이터의 방향성을 고려한 특징을 추출하여 사용자의 행위를 인지한다. 수직 요소와 수평 요소를 적극적으로 활용하여 유사한 행위간의 인지 정확도를 높인다. 그리고 상태 전이도를 활용하여 학습된 의사결정 트리 모델에 적용함으로써 행위 인지의 정확도를 높인다.

3. 가속도 기반 사용자 행위 인지 시스템

본 장에서는 크게 3가지의 해결방안을 통해 기존 연구의 단점을 극복한다. 먼저 가속도 센서의 수평, 수직 요소 정보를 통해 사용자 행위의 상, 하, 좌, 우 움직임 정보를 이용한다. 이는 사용자의 행위를 좀 더 정확하게 인지하는데 도움이 된다.

두 번째로 의사결정 트리 기반의 구분자를 사용한다. 앞서서 언급했듯이 의사결정 트리는 적은 수의 목표를 구분하기에 성능이 좋으며, 모델이 단순한 구조로 이루어져 있어 모델 분석이 쉽다. 또한 다른 구분자에 비해 성능이 우수하고 인지과정에서 빠른 속도로 인지가 가능하다.

마지막으로 상태 전이도를 사용하여 최종적으로 행위를 결정한다. 행위는 전혀 다르지만 유사한 형태의 가속도 센서 데이터가 나타날 경우, 의사결정 트리 모델만으로 구분

하기에는 한계가 있다. 그래서 합리적인 행위의 흐름을 결정해 놓음으로써 잘못된 행위 인지를 바로잡을 수 있다.

위의 3가지 해결방안을 기반으로 실시간 행위 인지 시스템을 구축한다. 먼저 행위 인지 시스템에서 모델을 학습하는 방법과 인지과정에 관한 구조를 살펴본다. 그리고 모델을 학습하기 위한 가속도 데이터 수집 방법에 대해 살펴보고 수집된 데이터를 통해 수평, 수직 요소를 구하는 방법에 대해 알아본다. 그리고 간단한 전처리를 거친 후 9가지의 특징들을 추출하고 모델을 학습한다. 그리고 상태 전이도를 만들기 위해 만들어진 의사결정 트리 모델을 분석한다. 마지막으로 의사결정 트리 모델과 상태 전이도를 스마트폰에 탑재한다. 지금까지의 과정이 실시간 행위 인지 시스템의 학습 과정이다. 인지 과정에서는 학습과정의 특징 추출 과정과 동일한 절차로 진행된다. 마지막으로 의사결정 트리 모델과 상태 전이도를 활용하여 최종 사용자 행위 인지를 결정한다.

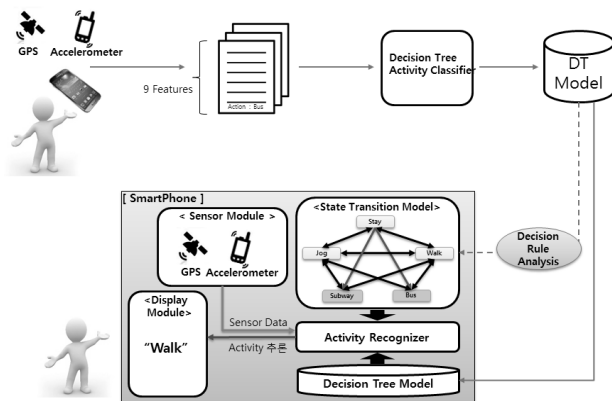


Fig. 1. Real-time user activity recognition system

3.1 실시간 사용자 행위 인지 시스템

위의 Fig. 1은 실시간 사용자 행위 인지 시스템의 구조도이다. 행위 인지 시스템은 크게 행위 인지 모델 학습 과정과 행위 인지 과정으로 나눌 수 있다. 본 연구에서는 Stay, Walk, Jog, Bus, Subway의 총 5가지 사용자 행위를 인지하는 것을 목표로 한다. 먼저 행위 모델 학습 과정을 살펴보면 크게 3가지 단계를 거친다. 안드로이드 스마트폰의 GPS 센서와 가속도 센서로부터 데이터를 수집한다. 수집하는 방식은 50% 윈도우 겹침을 사용하여 수집을 하며, 각각의 행위별로 약 4,000개의 데이터를 수집한다. 수집된 데이터는 수평, 수직 요소를 구하는 과정을 거쳐 이로부터 총 9개의 특징 값을 추출한다. 즉 각각의 행위별로 4000 by 9의 행렬이 생성되는 것이다. 이렇게 생성된 특징 값들은 의사결정 트리를 사용하여 모델을 생성한다. 여러 차례 반복실험을 통한 튜닝을 이용하여 최적의 성능을 나타내는 모델을 생성하면 사용자 행위 인지 모델의 학습 과정이 끝나게 된다.

다음은 행위 인지 과정이다. 행위 인지 과정에서 특징 값 추출하는 단계까지는 사용자 행위 인지 모델 학습과정의 특징 값 추출하는 과정과 동일하다. 다만 상태 전이도를 추가

하여 적용하는 부분이 다르다. 상태 전이도는 일반적인 사용자의 행위 변화 흐름을 반영하여 만들어진다. 또한 앞서 만들어진 의사결정 트리를 분석하여 Subway와 Bus의 인식 정확도를 향상시켜 상태 전이도에 반영한다. 이제 실시간으로 데이터가 들어오면 9개의 특징을 추출하고 스마트폰에 탑재된 모델을 적용하여 사용자 행위를 결정한다. 그리고 상태 전이도를 적용하여 앞에서 결정된 행위의 인식 오류를 판단하여 최종적으로 사용자 행위를 결정한다.

대략적인 행위 인지 모델의 학습 과정과 인지과정에 대해 살펴보았다. 다음은 좀 더 구체적인 행위 인지 모델의 학습 과정과 인지과정에 대해 알아보고 실제 데이터를 어떠한 형태로 표현되고 가공되는지를 설명한다.

3.2 행위 인지 모델 학습 과정

Fig. 2는 행위 인지 모델의 학습과정을 자세히 나타낸 그림이다. 먼저 수집된 가속도 데이터는 Fig. 2와 같이 안드로이드 스마트폰으로부터 3축 가속도 데이터 값을 파일로 기록한다. 그리고 추후 행위 인지의 보정을 위해 GPS 값과 속도 값도 함께 기록한다. 수집된 데이터는 수평, 수직 요소를 구하고 정점에 대한 보정을 한 후 이 값을 통해 9개의 특징 값을 추출한다. 이렇게 만들어진 특징 값은 Weka 툴의 의사결정 트리(J48)를 이용하여 모델을 생성한다. 생성된 모델은 Weka 툴에서 Java의 클래스 형태로 추출할 수 있다. 이렇게 추출된 모델은 안드로이드 스마트폰에 탑재하여 추후 실시간으로 사용자 행위인지를 적용하게 된다.

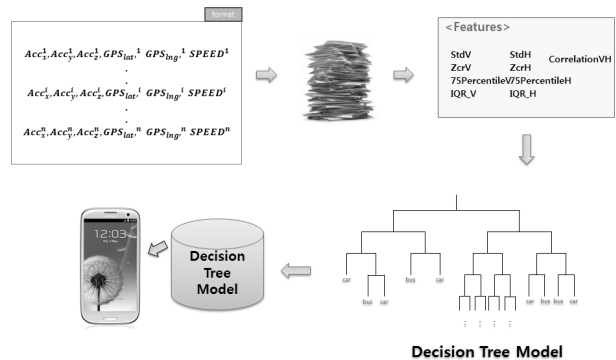


Fig. 2. Learning process of the activity recognition model

제조사에 따른 스마트폰의 종류마다 센서 장비 성능에 차이가 있다. Table 1은 [10]에서 발췌한 내용으로 샘플 주기를 살펴보면 4가지 모델의 가속도 센서에 대한 샘플링 주기를 나타낸다. Nexus One의 모델 경우 가장 빠른 속도로 가속도 데이터 수집이 가능하며 G2의 경우 가장 느린 샘플 주기를 보인다. 1)아래의 참고 링크를 확인해보면 더 많은 디바이스 모델의 샘플링 주기를 확인할 수 있다. 본 연구에서 사용한 Galaxy S3 같은 경우는 약 100 Hz 로서 초당

1) <http://iessendata.blogspot.kr/2012/11/android-accelerometer-sampling-rates.html>

100개의 가속도 데이터를 수집할 수 있다. 이렇게 스마트폰 별로 샘플링 주기가 다르기 때문에 주기를 통일시킬 필요가 있다. 빠른 샘플링 주기는 일부 스마트폰에서 동작하지 않을 수도 있으며 배터리 수명에도 영향이 있다. 본 연구에서는 현재 많이 보급된 스마트폰의 성능을 고려하여 샘플링 주기를 50 Hz 로 설정하였다.

Table 1. Sample rate of accelerometer sensor

Model Name	Chipset	Pattern/PIN	Sample Rate
Nexus One	Snapdragon S1	5/5	~ 25 Hz
G2	Snapdragon S2	6/6	~ 62 Hz
Nexus S	Hummingbird	1/0	~ 50 Hz
Droid Incredible	Snapdragon S1	0/1	~ 50 Hz

본 연구와 관련된 연구에서는 하나의 프레임(또는 윈도우) 단위로 가속도 데이터를 분리하여 행위를 인지한다. 보통 3초부터 20초까지를 프레임 단위로 설정한다. 하나의 프레임을 구성하는 시간이 길수록 데이터는 많아져 행위 인지의 성능이 좋아진다. 하지만 행위 인지 시간이 길어진다. 반대로 하나의 프레임을 구성하는 시간이 짧을수록 인지 속도가 빨라져 사용자 입장에서는 좋지만 적은 가속도 데이터를 활용해야 하므로 인지 성능이 저하된다. [3]에서도 이와 같은 상충관계를 고려하여 결정하였다.

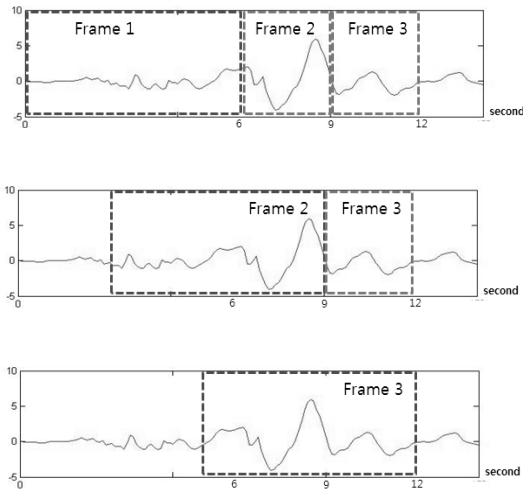


Fig. 3. Window overlap

위의 Fig. 3은 윈도우 겹침을 사용하여 샘플링 하는 과정을 나타내는 그림이다. 가속도 데이터는 신호 데이터의 한 종류로서 연속적인 데이터이며 사용자의 행위에 대한 특징한 패턴을 담고 있다. 그래서 단순한 프레임의 구성은 가속도 데이터의 주기적인 패턴을 담을 수 없다. 그래서 Fig. 3과 같이 가속도 데이터를 수집할 때, 프레임을 겹침 하여 수집하게 되면 사용자의 행위에 대한 패턴을 살릴 수 있다. [3, 7]에서도 같은 방법을 사용하였다. 이 방법의 단점은 프레임이 바뀔 때, 새로운 가속도 데이터로 전부 교체되지 않기 때문에 인지 속도에서 손해를 본다. 하지만 실제로 사용

자가 불편을 느낄 정도로 인지 속도가 느리지 않으며 오히려 모델의 정확도 측면에서 더 큰 장점을 가진다.

본 연구에서는 50% 윈도우 겹침을 사용하였다. Fig. 3에서와 같이 첫 프레임은 300개(50 Hz에서 6초)의 가속도 데이터를 수집하며, 두 번째 프레임부터는 3초 동안 150개의 새로운 데이터로 교체하여 총 300개의 가속도 데이터를 행위 인지에 사용한다. 즉, 3초 간격으로 사용자의 행위를 인지한다.

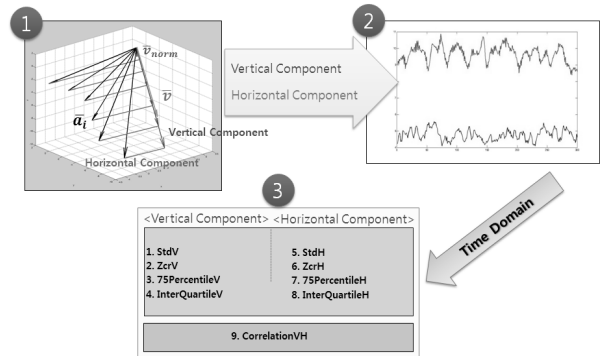


Fig. 4. The process of extracting features

Fig. 4는 특징 추출을 위한 전체적인 과정을 보여준다. 가장 먼저 해야 할 일은 수직 요소와 수평 요소를 구하는 일이다. 하나의 프레임에 300개의 가속도 데이터가 있기 때문에 수직 요소와 수평 요소 역시 300개씩 구할 수 있다. 두 요소의 크기만을 구해서 그래프로 그려보면 Fig. 4의 2번 그래프와 같이 신호 형태의 모양이 나온다. Fig. 4의 2번 그래프는 Jogging의 그래프로서 수직 요소와 수평 요소의 크기 변화가 크다는 것을 알 수 있다. 이 값을 통해서 9개의 특징 값을 추출하는 과정이 Fig. 4의 3번 그림이다. 기존의 관련 연구에서는 2장의 (1)번 식을 통해 구해진 진폭 값을 사용하여 FFT 하여 특징 값으로 많이 사용한다[2-3, 6-7]. 간단하게 설명하면 가속도 데이터에서의 FFT는 시간 영역을 주파수 영역으로 변경한다. 변경한 값을 살펴보면 특정 행위에 나타나는 주파수가 있으며 그 주파수의 값을 구할 수 있다. 이 값들을 통해 특정 행위의 특징 값으로 사용하고자 하는 것이다. 하지만 본 연구에서는 FFT를 적용한 특징 값이 Subway, Stay, Bus의 구분에 큰 영향을 주지 못한다는 것을 실험으로 확인하였다. 그래서 이 부분은 특징 값으로 사용하지 않았다.

관련연구에서 (1)번 식은 스마트폰의 방향성을 고려하지 않는 방법이다. 같은 행위에서도 스마트폰의 방향은 달라질 수 있기 때문에 이와 같은 방법은 좋은 선택이라고 할 수 있다. 하지만 스마트폰의 방향에 관한 정보는 소실된다. 그래서 본 연구에서는 이러한 정보를 반영하기 위해 수직 요소와 수평 요소를 사용한다. 수직 요소는 스마트폰의 방향과 관계없이 중력방향인 아래, 위로의 움직임에 대한 가속도 크기의 변화를 나타낸다. 그리고 수평 요소는 스마트폰의 앞, 뒤로의 움직임에 대한 가속도 크기의 변화를 나타낸다.

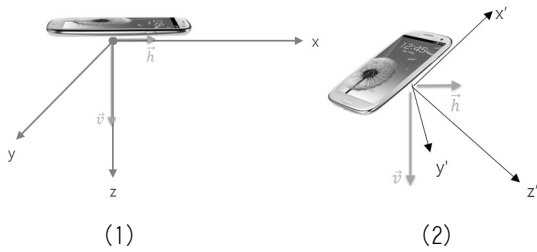


Fig. 5. Changes according to the direction of the axis

Fig. 5의 (1)번 그림은 스마트폰을 평평한 곳에 두었을 경우, 스마트폰의 수직 요소(\vec{v})는 중력 방향과 절대 축 z 와 방향이 같다. 그리고 (2)번의 그림처럼 스마트폰의 방향이 바뀌어도 수직 요소(\vec{v}) 방향은 여전히 중력 방향이다. 즉, 스마트폰의 방향에 관계없이 중력방향의 수직 요소와 앞, 뒤 방향의 수평 요소를 구할 수 있다. 결론적으로 스마트폰의 방향을 고려하지 않지만 전체적인 움직임의 방향은 반영할 수 있게 된다.

수직 요소와 수평 요소를 구하는 방법에 대해 알아보자. 가장 먼저 해야 할 일은 하나의 프레임에 담겨 있는 300개의 가속도 데이터에 대한 평균을 구하는 것이다.

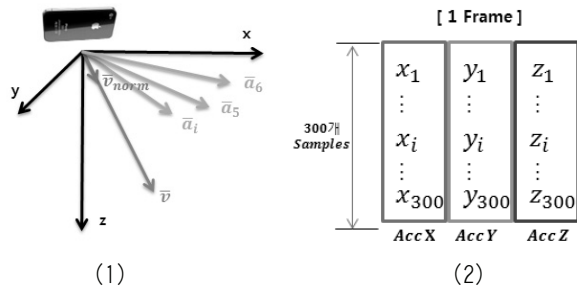


Fig. 6. Unit vector of the accelerometer data

x, y, z 의 3축으로 이루어진 가속도 데이터는 Fig. 6의 (1)과 같이 벡터로 표현이 가능하다. 본 연구에서는 총 300개의 데이터가 있으며 그 중 i 번째 x, y, z 축의 데이터를 \vec{a}_i 라고 정한다. 그리고 Fig. 6의 (2)에서 AccX, AccY, AccZ 각 축의 평균을 구한 \vec{v} 를 구한다. 마지막으로 \vec{v} 의 단위 벡터를 Table 2의 ③ 식을 통해 구한다.

Fig. 7에서 수직 요소는 \vec{p}_i 벡터이다. 300개의 가속도 데이터 중 \vec{a}_i 데이터가 있다고 가정하면, \vec{a}_i 데이터를 \vec{v}_{norm} 에 투영한 벡터가 수직 요소인 \vec{p}_i 가 되는 것이다. 투영하는 방법은 Table 3에 있다. 여기서 \vec{p}_i 는 수평 요소를 구하기 위해 필요하며 실제로는 p_i^{in} 값을 사용하여 특징 값을 구한다.

수평 요소를 구하는 방법은 다음과 같다. Table 4와 같이 \vec{a}_i 에 앞에서 구한 수직 요소인 \vec{p}_i 을 빼면 수평 요소 \vec{h}_i 값을 구할 수 있다.

Table 2. Expression of unit vector

① i 번째 x, y, z 축의 벡터
$\vec{a}_i = (x_i, y_i, z_i) \quad i = 1, 2, \dots, 300$
② x, y, z 축의 평균
$\vec{v} = (m_x, m_y, m_z)$
③ \vec{v} 의 단위 벡터
$\vec{v}_{norm} = \left(\frac{m_x}{\sqrt{m_x^2 + m_y^2 + m_z^2}}, \frac{m_y}{\sqrt{m_x^2 + m_y^2 + m_z^2}}, \frac{m_z}{\sqrt{m_x^2 + m_y^2 + m_z^2}} \right)$

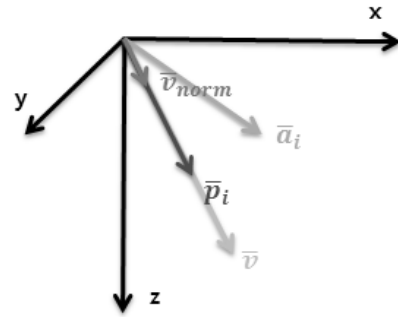


Fig. 7. Vertical component

Table 3. Calculation of vertical component

① Projection
$proj_{\vec{a}_i} \vec{v}_{norm} = \frac{\vec{a}_i \cdot \vec{v}_{norm}}{ \vec{v}_{norm} ^2} \vec{v}_{norm}$
② Inner Product
$p_i^{in} = \langle \vec{a}_i, \vec{v}_{norm} \rangle = \frac{\vec{a}_i \cdot \vec{v}_{norm}}{ \vec{v}_{norm} ^2}$
③ Vertical Component
$\vec{p}_i = p_i^{in} \cdot \vec{v}_{norm}$

Table 4. Calculation of horizontal component

① Horizontal Component
$\vec{h}_i = \vec{a}_i - \vec{p}_i$

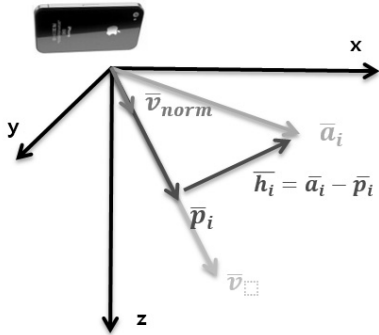


Fig. 8. Vertical, Horizontal component

최종적으로 구한 수직 요소인 \bar{p}_i 와 수평 요소인 \bar{h}_i 값을 Fig. 8에서 확인할 수 있다. 하나의 프레임에 총 300개의 데이터가 있으므로 수직 요소와 수평 요소 역시 300개의 데이터를 생성할 수 있다.

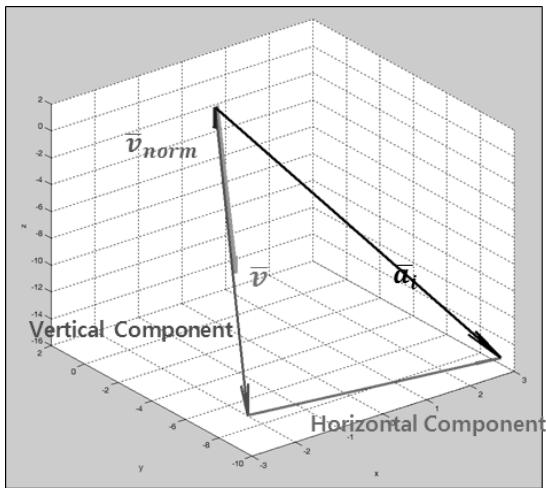


Fig. 9. Vertical, Horizontal component

실제로 수집한 가속도 데이터 중 하나 (\bar{a}_i)를 추출하여 MATLAB을 사용해 그려보면 Fig. 9과 같음을 확인할 수 있다. 여기서 수직, 수평 요소는 크기와 방향을 가진 벡터 형태이다. 하지만 방향의 요소는 이제 필요가 없다. 왜냐하면 300개의 모든 \bar{a}_i 가 \bar{v} 로 투영되어 방향이 동일하기 때문이다. 그래서 본 연구에서는 수직 요소의 크기(p^{in})와 수평 요소의 크기($\|\bar{h}_i\|$)를 사용한다. 앞장의 Fig. 4에서도 ①에서 ②으로 넘어가는 단계에 수직, 수평요소들의 크기만을 사용한다는 것을 확인할 수 있다.

Fig. 10는 가속도 데이터에 대한 두 요소의 크기를 사용하여 그래프로 그린 그림이다. 그래프의 상단에 위치한 선은 수직 요소이며 하단의 선은 수평 요소이다. 5가지의 그래프를 살펴보면 사용자의 행위별로 그래프의 형태가 차이는 것을 알 수 있다. Jog같은 경우는 큰 움직임으로 인해 가속도의 변화가 크며, Car(Bus)의 경우는 상, 하의 움직임

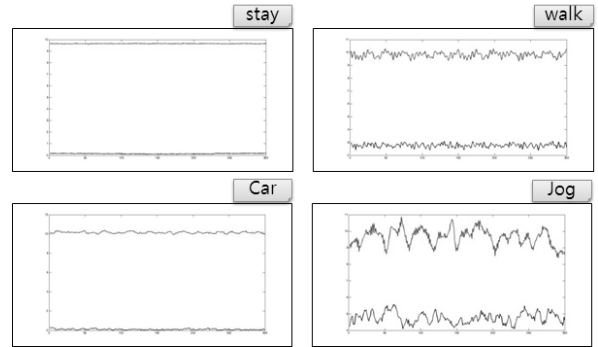


Fig. 10. Vertical, Horizontal component of each activity

보다 좌, 우의 움직임이 더 크다는 것을 알 수 있다. 그리고 Stay의 경우는 거의 움직임이 없으므로 가속도 변화 역시 미비하며, Subway의 경우는 레일 위에서 움직이기 때문에 가속도의 변화가 작다.

Fig. 10에서 Stay의 그래프와 Car의 그래프를 비교해보면 상당히 유사하다. 행위 인지 모델에서도 두 행위의 구분은 쉽지 않다. 이런 문제점을 해결하기 위해 정점 부분을 부각시키는 방법을 사용하였다.

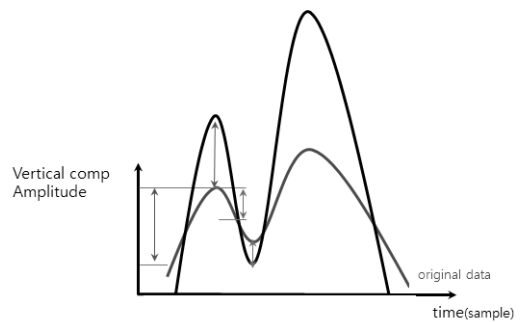


Fig. 11. The method of changing peak point

정점 부분을 부각시키는 방법은 다음과 같다. 기울기가 변하는 지점에 이전의 정점으로부터 현재 정점까지의 크기 값을 더하여 현재 정점의 크기를 변형시킨다. Fig. 11에서 완만한 곡선의 데이터는 기존의 수직 요소의 데이터로써 정점을 부각하여 가파른 곡선의 수직 요소가 만들어졌다.

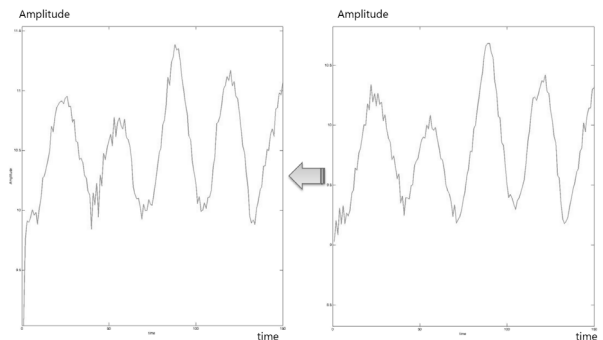


Fig. 12. Comparison of peak point

Fig. 12.은 Jog 의 수직 요소 그래프를 나타낸 그림으로 오른쪽 그래프가 정점을 사용한 그래프이다. 전체적인 형태는 변화되지 않지만 정점의 진폭이 증가하거나 감소하였음을 알 수 있다.

수직, 수평 요소를 통해 추출할 특징은 총 9개 이다. 각 요소 별로 4개의 특징이 있으며 두 요소의 관계에 대한 특징 1개가 있다.

1) 표준편차(Standard Deviation)

사용자 행위 인지뿐만 아니라 대부분의 구분자 알고리즘의 특징으로 많이 사용한다. 모집단에서 이질적인 성질을 가진 데이터들이 얼마나 분포하느냐를 보기 위한 특징 값이다. 본 연구에서는 가속도 데이터의 분포가 평균값으로부터 얼마나 분포되어 있는지를 볼 수 있다. Jog 나 Walk 같은 경우는 가속도 데이터의 변화가 커 표준편차 값이 크며 Stay 나 Subway 경우는 가속도의 변화가 작기 때문에 표준편차의 값이 작다.

2) 영통과율(Zero-Crossing Rate)

$$zcr = \frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{I}\{s_t s_{t-1} < 0\} \tag{2}$$

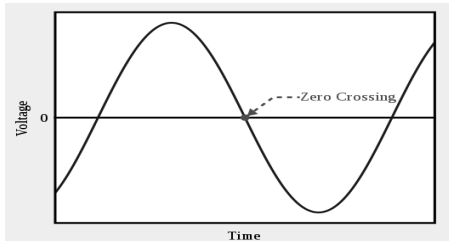


Fig. 13. Zero-crossing rate

영통과율은 기준점인 0을 지나는 신호의 비율을 의미한다. 기본적으로 피치 검출 알고리즘으로도 많이 사용되어 왔다. Fig. 13의 (2)식을 살펴보면 현재 샘플의 값인 s_t 와 이전 샘플의 신호 값인 s_{t-1} 을 곱했을 경우 이 값이 음인지 판단한다. 결과가 음의 값이면 부호가 바뀌었다는 뜻이며 중괄호안의 식은 true 가 되어 1을 반환한다. 반대로 곱한 값이 0보다 크거나 같으면 0을 반환한다. 모든 신호의 길이만큼 그 연산을 하여 합산을 하면 결과적으로 0을 교차한 횟수를 구할 수 있다. 그리고 신호의 길이만큼 교차횟수를 나누어주면 신호가 0을 교차하는 비율이 된다.

가속도 데이터를 적용하였을 경우, 각 행위 별로 영교차율의 비율이 나타날 것이다. 이를 특징 값으로 사용하여 행위 인지에 활용한다.

3) 75분위수(75Percentile)

백분위는 특정 집단의 분포에서 하나의 샘플의 상대적 위

치를 나타내는 방법이다. 75분위수는 전체 데이터에서 75% 위치에 있는 데이터를 말한다.

$$n = \frac{P}{100} \times N + \frac{1}{2} \quad (0 \leq P \leq 100) \tag{3}$$

(3) 에서 P는 퍼센트의 값으로 75분위수의 P값은 75 가 될 것이다. 그리고 N 은 총 데이터의 개수이다. 본 연구에서는 300개의 데이터이므로 N = 300 이다. 위의 데이터를 기반으로 계산을 하면 225.5 가 나온다. 소수점이 나올 경우, 올림을 하면 $n = 226$ 이 된다. 즉, 226번째 값이 75% 위치의 데이터 값이 된다. 300개의 가속도 데이터를 오름차순으로 정렬하여 226번째 값을 찾아내면 그 위치의 값이 75분위수 특징 값이다.

$$\begin{aligned} Q_1 &= CDF^{-1}(0.25) \\ Q_3 &= CDF^{-1}(0.75) \end{aligned} \tag{4}$$

$$IQR = Q_3 - Q_1$$

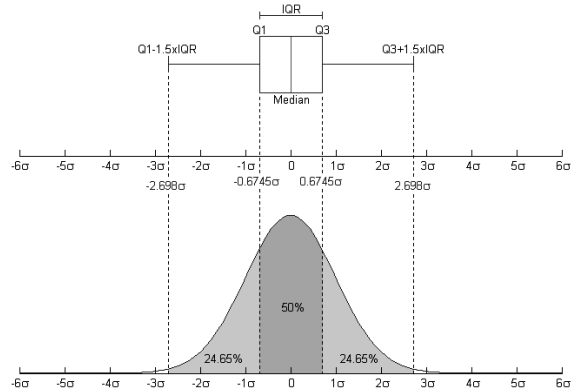


Fig. 14. InterQuartile Range

4) 사분위수 범위(InterQuartile Range)

(4) 에서 Q_1 은 하위 25%의 범위를 의미하여 Q_3 는 75%의 범위를 말한다. 위 (3)의 75분위수에서 P의 값이 25이면 Q_1 이 되며 P의 값이 75이면 Q_3 가 된다. 그래서 Q_3 에서 Q_1 의 값을 빼면 평균으로부터의 50% 범위인 사분위수 범위가 된다. Fig. 13에서 중간의 50% 구간이 사분위수 범위이다.

5) 상호 상관(Cross-Correlation)

$$r_{12} = \frac{1}{N} \sum_{n=0}^{N-1} x_1[n]x_2[n] \tag{5}$$

신호 분석에서 쌍의(non-identical) 파형 형태에 대한 유사도를 측정하는 방법 중 하나가 상호 상관 이다. 계산식은 간단하다. (5) 의 식을 보면 두 신호 x_1 과 x_2 가 있다. 같은 위치의 샘플을 곱하여 모두를 더한 후 총 개수로 나눠주면 상호 상관 값을 구할 수 있다. 값이 비슷한 파형 형태의 두 신호 데이터가 있다고 가정하면, 같은 위치의 신호 곱은 플러스 가 될 것이다. 즉, 값이 클수록 두 신호 데이터가 비슷하다는 이야기가 된다. 본 연구에서는 두 수치, 수평 요소의 상호 상관 값을 구하여 9번째의 특징 값으로 사용하였다.

위의 특징들을 추출했으면 이제 이 특징들을 사용하여 모델을 생성해야 한다. 의사결정 트리 모델은 정보 이득이 가장 낮은 항목부터 노드를 생성하여 트리를 생성한다. 본 연구에서는 Weka 툴킷(v3.6)을 사용하여 행위 인지 모델을 생성하였다. 의사결정 트리 중에서도 J48을 사용하였다. J48은 C4.5 알고리즘을 Java 형태로 구현한 것으로 Weka 데이터 마이닝 툴로 개발 되었다.

의사결정 트리로 생성된 행위 인지 모델의 튜닝은 Weka 툴킷의 옵션에서 조정하였다.

의사결정 트리의 장점 중 하나는 모델 분석이 가능하다는 것이다. 본 연구에서도 이러한 장점을 살려 모델을 분석하였다. Weka에서 생성한 의사결정 트리는 여러 개의 같은 행위에 대한 단말 노드를 가진다. 실제 가속도 데이터의 특징 값들이 도달하는 단말 노드를 분석해보면 같은 행위의 이름을 가진 여러 개의 단말 노드를 확인할 수 있다. Fig. 14을 보면 단말 노드에 7개의 Car 단말 노드가 존재함을 알 수 있다.

여기서 중요한 점은 모든 Car 단말 노드가 유효한 노드가 아니라는 점이다. Fig. 15에서 분석한 내용을 보면 위쪽 박스안의 Car 단말 노드는 91%의 정확도를 보이며 아래쪽 박스안의 Car 단말 노드는 99%의 정확도를 보인다. 또한 훈련 샘플의 대부분이 아래쪽 박스의 Car 단말 노드에 도달함을 알 수 있다. 나머지 Car 단말 노드를 살펴보면 50%대의 정확도를 가지는 단말 노드도 존재한다. 즉, 높은 정확도를 가지는 단말노드와 정확도가 낮은 단말노드가 공존한다는 뜻이다. 그래서 높은 정확도를 보이는 단말 노드를 중점적으로 사용함으로써 의사결정 트리의 정확도를 높일 수 있다.

하지만 Stay, Walk, Jog, Bus, Subway의 5가지 모든 행위에 대해 위의 정책을 적용 하지는 않았다. 위의 정책은 행위 인지의 속도가 느리기 때문이다. 의사결정 트리에서 정확한 특징 값들로 구성이 되지 않으면 높은 정확도를 가지는 단말 노드에 도달하지 못한다. 그래서 본 연구에서는 행위 인지 모델에서 정확도가 떨어지는 Bus 와 Subway의 경우에 위의 정책을 사용하였다.

이렇게 생성된 모델을 상태 전이도에 활용된다. Fig. 16은 상태 전이도를 나타낸 그림이다. 상태 전이도의 특징은 3가지로 나눌 수 있다. 먼저 Stay에서 Subway 와 Bus로의 전환은 높은 확률을 고려하여 행위 전환을 한다. 오각형 모양의 Subway 와 Bus 는 행위 인지 모델을 분석하여 만든 정책을 따른다. 두 번째로 Subway 와 Bus에서 Stay로의 전환은 Walk 이나 Jog를 거쳐야 전환할 수 있도록 했다. 운송수

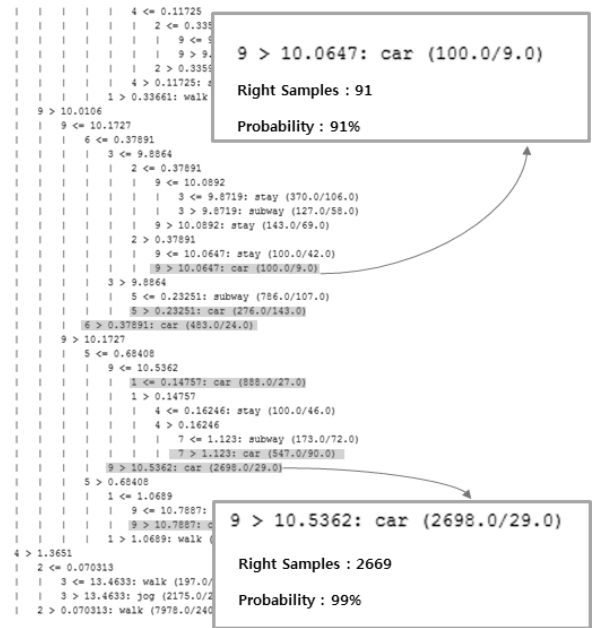


Fig. 15. The analysis of decision tree model

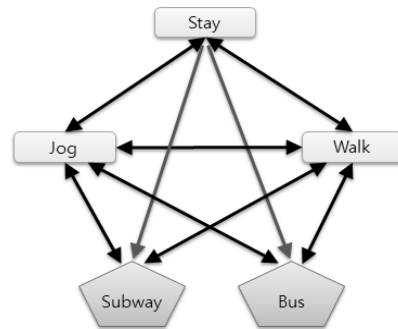


Fig. 16. State transition diagram

단에서 벗어났을 경우, 대부분 Walk 이나 Jog를 거친다. 마지막으로 Subway 와 Bus의 직접적인 상호 전환은 불가능하도록 하였다. 이 특징 역시 Subway 와 Bus 간의 직접적인 전환은 상식적으로 불가능하기 때문이다.

3.3 행위 인지 모델 인지 과정

마지막으로 모델과 상태 전이도를 통하여 실시간으로 사용자의 행위 인지시스템을 구축한다. 실시간 사용자 행위 인지는 Fig. 17의 과정으로 진행된다. 특징 추출 과정까지는 사용자 행위인지 모델 학습과정과 같다. 이식된 모델을 적용하려면 완벽하게 행위인지 모델 학습과정의 특징 값과 같아야 한다. 추출된 특징 값은 학습과정에서 이식된 모델을 통하여 행위를 결정한다. 정해진 행위는 GPS 데이터의 수신여부를 체크하여 수신 될 경우, 인지된 행위와 부합하는지 체크한다. 실내인 경우는 GPS 데이터를 무시한다. 다음 단계로 상태 전이도를 적용한다. 현재 인지된 행위가 이전의 행위에서 바뀌었다면 상태 전이도의 흐름에 부합하는지를 본다. 불가능한 행위 전환이라면 현재 인지된 행위는 무

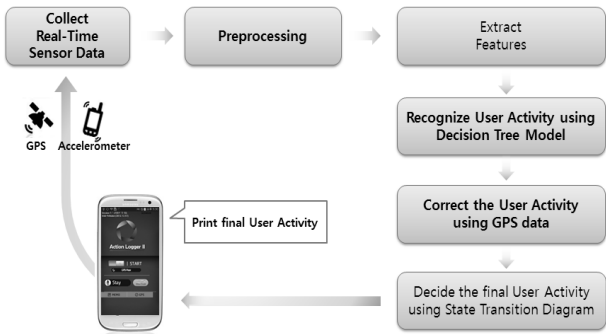


Fig. 17. The process of real-time user activity recognition

시되고 이전의 행위로 계속 이어간다. 최종적인 행위가 결정되었다면 스마트폰 화면에 인지된 행위를 출력해준다. 이와 같은 과정은 3초를 주기로 계속 반복된다.

4. 실험 및 결과

4.1 실험 환경 및 실험 데이터

본 연구에 사용한 스마트폰은 안드로이드가 탑재된 삼성 Galaxy 3(3G)를 사용했다. 실험과정에서 Galaxy 3의 가속도 데이터를 기반으로 만든 의사결정 트리 모델을 다른 제조사 (HTC) 스마트폰에 탑재해서 실험을 해본 결과, 행위의 정확도에서 차이를 보였다. 제조사 마다 측정되는 가속도 센서의 데이터 미세하게 차이이기 때문이다. 그래서 본 연구에서는 하나의 기종으로 실험을 진행하였다.

Table 5. The experimental data

	Galaxy 3		Total	비고
	Pocket	Hand		
Walking	2,132	2,033	4,165	
Jogging	2,214	2,114	4,328	
Bus	2,133	2,124	4,257	Bus, Car
Stay	1,812	2,121	3,933	
Subway	2,056	2,050	4,016	Line 2, 7, 4
Total	10,347	10,442	20,699	

모델을 학습하기 위해 약 2만개 정도의 가속도 데이터 샘플을 수집하였다. Table 5에서 수집한 데이터의 종류를 알 수 있다. 각 행위별로 약 4천개의 데이터를 수집하였으며, 수집 시 주머니 와 손의 두 종류로 나누어 수집하였다. Bus의 가속도 데이터는 Car의 데이터와 병합하여 수집하였고 Subway의 가속도 데이터는 7호선, 2호선, 4호선의 데이터를 병합 수집하였다.

행위 인지 모델의 평가는 10 Folds Cross-validation 방식을 사용하였다. 테스트 집합 과 훈련 집합의 분포가 1:9 로서 약 2만개의 데이터를 사용한다면 2,000개 가 테스트 집합으로 쓰이며 18,000개가 훈련 집합으로 쓰인다.

4.2 실험 결과

본 연구의 실험은 생성된 행위 인지 모델의 정확도를 측정해 본다. 그리고 행위 인지 구분에 기여도가 높은 특징들을 찾아 그 이유에 대해 설명한다. 그리고 같은 환경에서 다른 구분자와 정확도를 비교 평가 해본다. 또한 상태 전이도 사용 전/후의 행위 인지 정확도를 측정해보고 마지막으로 스트레스 테스트를 통하여 실생활에서 제대로 동작하는지를 알아본다.

Table 6. Accuracy of decision tree

Precision	Recall	F-Measure	Class
0.874	0.857	0.865	stay
0.741	0.944	0.942	walk
0.994	0.998	0.996	jog
0.972	0.907	0.917	bus
0.782	0.809	0.795	subway
0.905	0.904	0.905	

Table 7. Confusion matrix of decision tree

	stay	walk	jog	bus	subway	
3370	9	0	70	484	stay	
28	3931	19	24	163	walk	
0	9	4318	0	1	jog	
81	36	2	3862	276	bus	
377	194	3	212	3320	subway	

위의 Table 6와 Table 7은 행위 인지 모델의 정확도를 보여준다. 전체적으로 90% 정도의 정확도를 보인다. 하지만 subway 같은 경우는 정확도가 80% 정도밖에 안 된다. Table 11에서 보면 subway 데이터 인 경우 주로 stay, walk, bus 로 잘못 인지되는 경우가 많다.

Table 8. Contribution of the features

Ranked	attribute Number	Component	feature
1.669	3	Vertical	75Percentile
1.664	4	Vertical	InterQuartile
1.553	1	Vertical	Standard deviation
1.405	5	Horizontal	Standard deviation
1.392	8	Horizontal	InterQuartile
1.364	2	Vertical	Zero-Crossing Rate
1.344	7	Horizontal	75Percentile
1.301	9		Cross-Correlation
0.65	6	Horizontal	Zero-Crossing Rate

위의 Table 8는 특징 값들의 기여도를 볼 수 있는 표이다. 위의 결과는 Weka에서 특징을 평가할 때 많이 쓰이는 방법을 사용하였다. 속성 평가에 "InfoGainAttributeEval"을 사용하였으며 탐색 함수는 "Ranker"를 사용하였다. 본 연구

에서 사용하고 있는 9개의 특징은 모델의 정확도가 가장 높게 나오는 조합으로 구성되어 있다. Table 9를 분석해 보면 수직 요소가 모델의 정확도에 많은 기여를 하고 있음을 알 수 있다. 또한 표준편차와 사분위수가 사용자의 행위를 구분하는데 유용한 특징임을 알 수 있다.

의사결정 트리의 모델을 분석함으로써 모델 생성에 사용된 특징 값들의 중요도를 알아볼 수 있다. 아래 Fig. 18과 Fig. 19은 의사결정 트리 기반의 행위 인지 모델을 나타낸 그림이다. 각 행위별로 대표적인 단말 노드를 찾아 가속도 데이터가 어떠한 특징 값을 거쳐 단말 노드에 도달하는지를 분석하였다.

다음 Fig. 18과 Fig. 19.에서 의사결정 트리의 상위 노드를 구성하는 특징 값은 사분위수 범위(IQR) 과 영통과율(ZCR) 이다. 그리고 대표적인 단말 노드는 대부분 수직 요

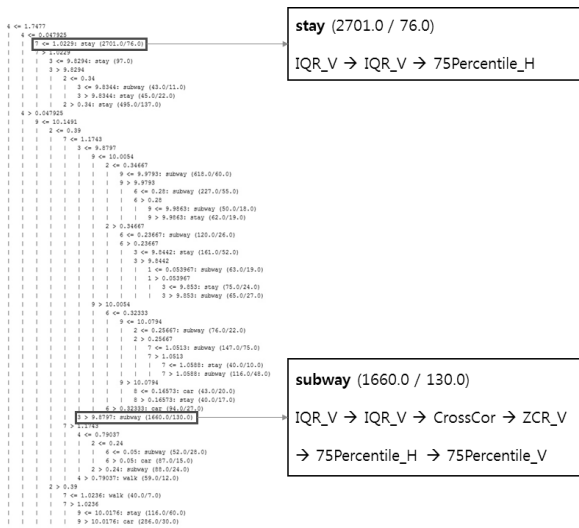


Fig. 18. Analysis of activity recognition model #1

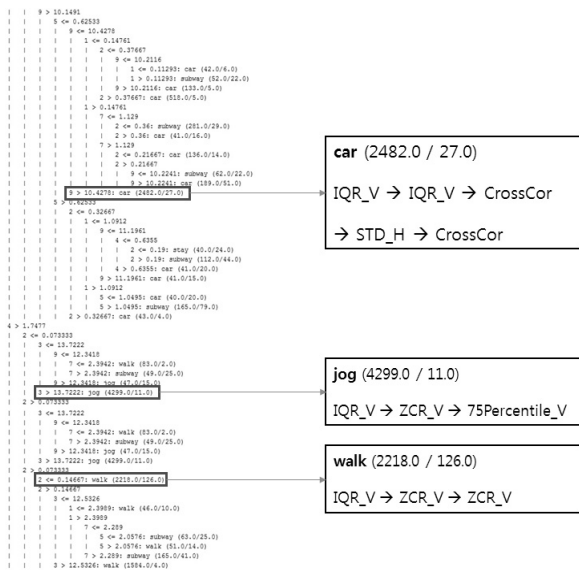


Fig. 19. Analysis of activity recognition model #2

소가 비중 있게 사용되었음을 확인할 수 있다. 결국 상, 하의 움직임이 사용자의 행위를 구분하는데 유용함을 알 수 있다. Table 9는 각 행위에 대한 대표노드를 구성하고 있는 트리의 구체적인 특징 값 조건을 나타낸다.

Table 9. Representative node of activity recognition model
Structure of Decision Tree Model

1. Stay Activity

IF IQR_V <= 1.7477 THEN
IF IQR_V <= 0.047925 THEN
IF 75Percentile_H <= 1.0229 THEN
return Stay

2. Subway Activity

IF IQR_V > 0.047925 THEN
IF IQR_V <= 0.047925 THEN
IF CrossCor <= 10.1491 THEN
IF ZCR_V <= 0.39 THEN
IF 75Percentile_H <= 1.1743 THEN
IF 75Percentile_V <= 9.8797 THEN
return Subway

3. Car Activity

IF IQR_V > 0.047925 THEN
IF IQR_V <= 0.047925 THEN
IF CrossCor > 10.1491 THEN
IF STD_H > 0.62533 THEN
IF CrossCor > 10.4278 THEN
return Car

4. Jog Activity

IF IQR_V > 1.7477 THEN
IF ZCR_V <= 0.073333 THEN
IF 75Percentile > 13.7222 THEN
return Jog

5. Walk Activity

IF IQR_V > 1.7477 THEN
IF ZCR_V > 0.07333 THEN
IF ZCR_V <= 0.14667 THEN
return Walk

Table 10. Accuracy of Support Vector Machine

Precision	Recall	F-Measure	Class
0.818	0.864	0.841	stay
0.948	0.94	0.944	walk
0.999	1	0.999	jog
0.935	0.893	0.913	bus
0.773	0.773	0.773	subway
0.897	0.896	0.896	

Table 11. Accuracy of BayesNet

Precision	Recall	F-Measure	Class
0.858	0.81	0.833	stay
0.891	0.957	0.923	walk
0.999	0.995	0.997	jog
0.853	0.827	0.84	bus
0.7	0.71	0.705	subway
0.862	0.862	0.861	

Table 12. Accuracy of Neural Network

Precision	Recall	F-Measure	Class
0.813	0.834	0.824	stay
0.945	0.929	0.937	walk
0.997	0.999	0.998	jog
0.925	0.871	0.897	bus
0.753	0.791	0.771	subway
0.889	0.887	0.887	

Table 13. Accuracy of Logistic Regression

Precision	Recall	F-Measure	Class
0.749	0.888	0.813	stay
0.914	0.926	0.92	walk
0.998	0.998	0.998	jog
0.877	0.725	0.793	bus
0.653	0.646	0.649	subway
0.841	0.837	0.837	

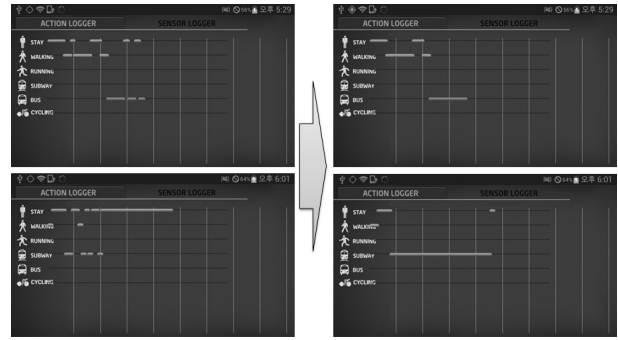


Fig. 20. The screen of using state transition diagram

Table 10 부터 Table 13 까지 5개의 서로 다른 구분자를 사용하여 정확도를 평가해보았다. 입력되는 특징 값은 본 연구에서 의사결정 트리 모델을 생성하기 위해 사용하는 특징 값과 동일한 값이다.

모든 구분자에서 대체로 80% 후반의 정확도를 보였다. 그리고 Subway 의 경우, 대부분의 구분자에서 상대적으로 정확도가 낮았다. 반대로 Walk, Jog 의 경우는 데이터의 특징이 뚜렷하기 때문에 높은 정확도를 보였다.

구분자 마다 적합한 특징의 종류가 존재하겠지만 기본적으로 모델의 정확도에 영향을 미치는 것은 충분한 양의 데이터와 노이즈가 적은 데이터를 사용하는 것이다. 본 연구에서도 충분한 데이터를 사용하여 대부분의 구분자를 통한 모델에서 높은 정확도를 보였다.

상태 전이도의 사용여부에 따른 정확도는 정량적으로 분석하기가 쉽지 않다. 다만 동일한 행위에 대해 오류의 빈도를 관찰함으로써 정확도를 가늠할 수 있다. Fig. 20은 상태

Table 14. Stress Test #1

No.	list of test	the number of test	result
1	When standing in the subway	13	There is some difference in speed, but Properly recognized subway
2	When sitting in the subway	13	There is some difference in speed, but Properly recognized subway
3	When standing in the bus	15	It is recognized immediately, and no problem recognizing bus
4	When sitting in the bus	15	It is recognized immediately, and no problem recognizing bus
5	When standing in an escalator or elevator	20	Does not recognize exactly (Recognizing subway or stay)
6	When turning on the Action Logger in the subway	13	After starting the subway, recognized the subway exactly.
7	When turning on the Action Logger in the bus	15	After starting the bus, recognized the bus exactly.
8	When operated for more than 4 hours, whether to work or not,	20	There is no if the mobile phone is turned off.

Table 15. Stress Test #2

No.	list of test	the number of test	result
9	When the bus stops due to a signal light or traffic jam	13	No problem to recognize the bus
10	When boarding the bus for more than two hour, is there any problem?	13	No problem to recognize the bus
11	When waiting the bus or subway at the bus stop or subway station	15	Most recognized correctly but sometimes recognized subway
12	Result of activity recognition at the daily life including bus or subway	15	No problem to recognize the bus and subway but there is a difference in the speed of recognition
13	When entering into the building after alighting from the car or bus	20	Order is all right (Bus -> Walking -> Stay)
14	When walking down from the underground parking lot in the car(bus)	13	Order is all right (Bus -> Walking)
15	When going out to the outdoor after riding the car(bus) in the underground parking lot	15	After 1 or 2 minutes, recognizing the car(bus) exactly.
16	Can it recognize the bus at rush hour?	20	No problem to recognize the bus

전이도 의 사용여부에 따른 행위 인지의 오류 빈도를 눈으로 확인할 수 있다. 왼쪽 그래프는 상태 전이도를 사용하지 않았을 경우의 그래프이며 오른쪽 그래프는 사용했을 경우의 그래프이다. 상태 전이도를 사용하지 않았을 때, Bus 나 Subway 같은 행위는 매끄럽지 못한 선을 볼 수 있다. 하지만 상태 전이도를 적용한 그래프를 살펴보면 노이즈가 거의 없는 형태로서 매끄러운 선의 형태로 나타남을 알 수 있다. 이는 상태 전이도에서 운송수단인 경우, 반드시 Walk 이나 Jog를 거쳐야하기 때문에 왼쪽의 그래프처럼 Stay 상태로 전환될 수 없다.

본 연구내용을 기반으로 구현된 사용자 행위 인지 어플리케이션을 사용하여 Table 14, Table 15의 총 16가지 항목을 테스트하였다. 에스컬레이터나 엘리베이터의 탑승과 같은

상황은 학습되지 않은 상황이기 때문에 사용자의 행위인지에 오류가 있었다. 또한 사용자가 처한 상황의 진동정보가 제대로 전해지지 않는 경우 행위 인지의 속도가 느려졌다.

Fig. 21은 사용자 행위 인지 응용프로그램의 전체적인 사용자 인터페이스이다. 왼쪽의 화면에는 행위 인지를 시작, 정지 할 수 있는 화면으로 시작 시 실시간으로 사용자의 행위를 인지하여 화면에 출력해 준다. 그리고 오른쪽 위의 그림은 과거의 사용자 행위를 기록해둔 파일을 리스트 형태로 볼 수 있다. 오른쪽 아래의 그림은 과거나 실시간으로 사용자의 행위를 가로막대 형태로 보여준다. 그리고 하단의 붉은색 선은 GPS의 데이터를 수신할 없다는 뜻으로 수신이 가능하면 녹색으로 나타난다.

이와 같은 구성을 통해 사용자의 행위 전환 과정을 눈으로 확인할 수 있으며 그리고 수집된 사용자 행위 데이터는 또 다른 상위 상황을 추론할 수 있는 기반이 된다.

5. 결론 및 향후 연구

본 연구는 가속도 센서 데이터 중심의 의사결정 트리를 생성하고 이를 분석하여 사용자의 행위를 인지하는 방법을 제안하였다. 기존의 가속도 센서를 이용한 사용자 행위 인지는 유사한 가속도 데이터를 보이는 행위에 대하여 정확도가 떨어졌다. 그 이유는 스마트폰 이동 방향에 대한 속성을 사용하지 않고 오로지 크기만을 사용하였기 때문이다. 하지만 본 연구에서는 수직 요소와 수평 요소의 크기를 사용하여 스마트폰의 방향성을 고려한 특징 값을 추출하였다. 또한 의사결정 트리의 장점을 살려 모델을 분석하였고 이를 상태 전이도에 적용하였다. 그리고 상태 전이도에서는 일반적인 사용자의 행위 흐름을 정의하여 인지 오류를 줄였다.

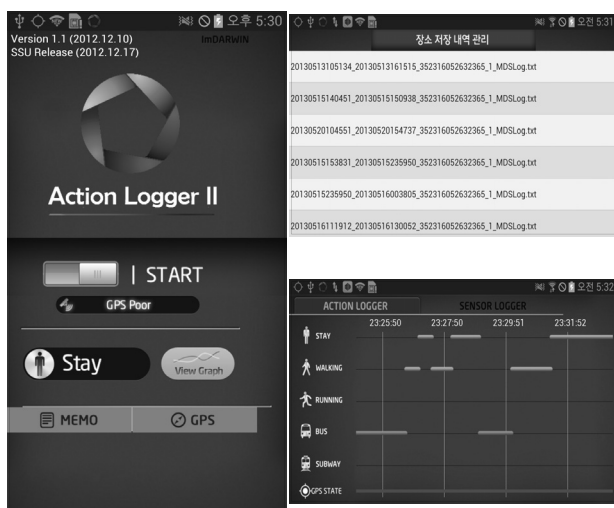


Fig. 21. The application UI of user activity recognition

이를 행위 인지과정에 적용함으로써 행위 인지 정확도를 향상시켰다.

Subway의 행위를 인지하는 경우는 가속도 데이터가 다른 행위의 가속도 데이터와 유사하여 상대적으로 인지 오류가 많았다. 이러한 부분은 조금 더 개선할 여지가 있다. 그리고 스마트폰의 가속도 데이터를 수집할 때 손과 주머니의 경우만을 수집하였다. 향후에는 좀 더 다양한 위치에서 데이터를 수집하여 행위 인지의 성능을 올릴 필요가 있다. 그리고 다양한 행위를 추가하여 실생활에 활용할 수 있도록 해야 할 것이다. 마지막으로 배터리 성능을 고려한 데이터 수집을 고려하여 실제로 사용했을 때 불편함이 없도록 해야 할 것이다.

참 고 문 헌

- [1] A. M. Khan, Y.-K. Lee, S. Y. Lee, T.-S. Kim, "Human Activity Recognition via An Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis", Future Information Technology (FutureTech), 2010 5th International Conference on, 2010.
- [2] Yu-Chieh Yang, Tatsuo Toida, Chin-Ming Hong, "Transportations Prediction Using Build-in Triaxial Accelerometer in Cell Phone", Department of Industrial Education of National Taiwan Normal University, 2010.
- [3] Vincenzo Manzoni, Diego Maniloff, Kristian Kloeckl, Carlo Ratti, "Transportation mode identification and real-time co2 emission estimation using smartphones", Massachusetts Institute of Technology (MIT), 2011.
- [4] Leon Stenneth, Ouri Wolfson, Philip S. Yu, Bo Xu, "Transportation Mode Detection using Mobile Phones and GIS Information", GIS '11 Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp.54-63, 2011.
- [5] Pekka Siirtola, Juha Roning, "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data", International Journal of Interactive Multimedia and Artificial Intelligence, pp.38-45, 2012.
- [6] Arvind Thiagarajan, James Biagioni, Tomas Gerlich, Jakob Eriksson, "Cooperative Transit Tracking using Smart-phones", SenSys '10 Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, pp.85-98, 2010.
- [7] Ben Nham, Kanya Siangliulue, Serena Yeung, "Predicting Mode of Transport from iPhone Accelerometer Data", Stanford University CS229 Machine Learning, 2008.
- [8] Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore, "Activity recognition using cell phone accelerometers", ACM SIGKDD Explorations Newslette Vol.12, pp.74-82, 2010.
- [9] Fehmi Ben Abdesslem, Andrew Philips, Tristan Henderson, "Less is More: Energy-Efficient Mobile Sensing with SenseLess", MobiHeld '09 Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds, pp.61-62, 2009.
- [10] Adam J. Aviv, Benjamin Sapp, Matt Blaze, Jonathan M. Smith, "Practicality of Accelerometer Side Channels on Smartphones", ACSAC '12 Proceedings of the 28th Annual Computer Security Applications Conference, pp.41-50, 2012.
- [11] Yang, Jun, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones", Proceeding of the 1st international workshop on Interactive multimedia for consumer electronics, pp.1-10, 2009.
- [12] David Mizell, "Using Gravity to Estimate Accelerometer Orientation", Proceedings of the Seventh IEEE International Symposium on Wearable Computers(ISWC'03), 2005.
- [13] Xi Long, Student Member, IEEE, Bin Yin, Ronald M Aarts, Fellow, IEEE, "Single-Accelerometer-Based Daily Physical Activity Classification", 31st Annual International Conference of the IEEE EMBS Minneapolis, Minnesota, USA, September 2-6, 2009.
- [14] Scuola Superiore Sant'Anna, Piazza Martiri della Liberta, "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers", Computational Intelligence and Neuroscience, Vol.10, 2010.
- [15] Mehmet Sonercan, Sinan Dincer, "User State Tracking using Smartphones", the degree of Bachelor of Science in Boğaziçi University Computer Networks Research Laboratory, 2011.
- [16] Sian Lun Lau, Klaus DAVID, "Movement Recognition using the Accelerometer in Smartphones", Future Network & MobileSummit 2010 Conference Proceedings Paul Cunningham and Miriam Cunningham(Eds) IIMC International Information Management Corporation, 2010.



전 명 중

e-mail : jjmmjj@gmail.com

2010년 숭실대학교 컴퓨터공학과(학사)

2013년 숭실대학교 컴퓨터공학과

석사과정

관심분야: 기계 학습, 상황 인지,

에이전트 시스템, 빅 데이터



박 영 택

e-mail : park@ssu.ac.kr

1978년 서울대학교 전자공학과 (학사)

1980년 한국대학교 컴퓨터공학과 전산학
(석사)

1992년 Univ. of Illinois at Urbana
Champaign(박사)

1981년~현재 숭실대학교 컴퓨터학과 교수

관심분야: 인공지능, 에이전트 전문가 시스템, 시맨틱 웹