

# 스마트 디바이스 사용 여부에 따른 사용자의 이동 상황 판별 기법<sup>†</sup>

(A method of determining the user's state of  
movement based on the smart device usage)

홍민성\*, 목남희\*\*

(Min-Sung Hong and Nam-Hee Mok)

**요약** 스마트 디바이스는 개인 장비이고 내장된 센서들을 사용할 수 있는 능력을 갖고 있다. 따라서 센서 정보를 통해 사용자의 상황을 인지하여 맞춤형 서비스를 제공하는 것이 가능하다. 이러한 맞춤형 서비스를 위해 사용자의 이동 상황은 많은 영역에서 활용이 될 수 있는 중요한 정보이다. 기존의 사용자의 이동 상황을 결정에 관한 연구들은 스마트 디바이스가 가방이나 주머니에 있다고 가정하였기 때문에 사용자가 스마트 디바이스를 사용하고 있는 경우에는 사용자의 이동 상황을 정확하게 결정하는데 적합하지 않다. 본 논문에서는 이러한 문제를 해결하기 위한 사용자의 스마트 디바이스 사용과 이동 상황을 판별하는 알고리즘을 제안한다. 제안한 알고리즘은 스마트 디바이스의 이벤트를 활용하여 사용자의 스마트 디바이스 사용 여부를 판단하고 가속도 센서 데이터를 통해 사용자의 이동 상황을 판별한다. 실제 실험을 통해 분석한 결과 평균 약 75% 정도의 정확도를 보였다.

**핵심주제어** : 스마트 디바이스, 센서, 위치 기반 서비스, 상황 인식

**Abstract** The smart device is a personal device and has the ability to use built-in sensors. Therefore, it is possible to provide individually customized services through recognizing the user's context from the sensor information. For these customized services, the user's moving status is the important information that can be utilized in many areas. Because the existing researches on determining the user's moving status assume that the user has a smart device in his/her pocket or bag, if the user is using his/her smart device it is not suitable for exactly distinguishing the user's moving status. In order to solve this problem, this paper proposes an algorithm to determine the user's using and moving status. Our proposed algorithm utilizes smart device's events to distinguish the smart device usage and the accelerometer sensor data to determine the user's movement. As an analyzing result through the real experiment, the accuracy of our algorithm is about 75 percent on average.

**Key words** : Smart Device, Sensor, Location Based Service, Context Recognition

## 1. 서론

최근 스마트폰을 비롯한 다양한 스마트 디바이스들의 보급이 활발해지고 있다. 스마트 디바이스에는 다양한 센서 및 장치들이 내장되어 있고, 이를 이용하여 사용자의 상황을 인지하는 연구가 활발히 진행되고 있다. 초창기 스마트 디바이스의 센서를 활용하여 상

<sup>†</sup> 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 콘텐츠산업기술포지원사업으로 수행되었음.

\* 단국대학교 컴퓨터과학과, 제1저자

\*\* 단국대학교 경영학부, 교신저자(e-mail: hold\_time@naver.com)

황을 인지하는 연구들은 한 가지의 센서 정보를 활용하는 연구가 대부분이었다. 하지만 최근에는 두 가지 이상의 센서 및 스마트 디바이스에 저장된 사용자의 이력 정보를 활용하여 사용자의 다양한 상황을 더욱 정확히 인지하는 연구들이 진행되고 있다.

이러한 사용자의 상황 정보를 활용하여 사용자가 필요로 하는 서비스를 제공하는 것이 가능하다. 사용자의 상황을 인지하기 위해 스마트 디바이스의 센서 뿐만 아니라 일정, 시간, 어플리케이션 사용내역 등이 사용된다. 사용자의 상황을 결정하는 다양한 요인들 중 하나가 스마트 디바이스 사용자의 이동 상황이다. 예를 들어, 스마트 디바이스 사용자가 동물을 관람 중인 경우를 가정해보자. 내장된 센서 정보를 활용하여 사용자의 이동 상황을 판별한 결과가

정지 상태라면, 사용자가 휴식이나 어떤 동물에 대해 관심을 보이는 중이라고 추론할 수 있다. 나아가 스마트 디바이스의 위치 측정 결과가 매점이나 정자와 같은 휴식 장소가 아니라면, 사용자가 어떤 동물에 관심을 보이고 관찰 중이라고 추론할 수 있다. 더 나아가 위치 측정 결과와 동물원의 지도를 바탕으로 사용자가 관심을 보이는 동물을 추론한 후, 그 동물과 관련된 콘텐츠를 사용자에게 제공할 수 있다. 이러한 서비스를 제공하는데 사용자의 상황 중 이동 상황의 판별은 중요한 역할을 할 수 있다. 본 논문에서는 스마트 디바이스 사용 여부에 따른 사용자의 이동 상황을 판별하는 알고리즘을 제안하고자 한다.

본 논문에서는 타 논문과 비교하기 위해 스마트 디바이스 중 스마트폰을 이용하여 실험 및 검증한다. 본 논문의 구성은 다음과 같다. 2장에서는 스마트폰의 센서를 활용한 상황 인식과 이동 상황 판별의 관련 연구를 소개한다. 3장에서는 스마트폰 사용 여부 및 이동 상황 판별 알고리즘을 제안한다. 4장에서는 제안한 알고리즘과 기존 논문의 알고리즘들의 정확도를 비교 및 분석한다. 마지막으로 5장에서는 결론 및 향후 과제에 대해 소개한다.

## 2. 관련 연구

스마트폰 기반 상황인식에 대한 연구는 활발하게 진행되고 있다. 상황인식은 사용자의 위치, 시간, 사용자의 행동 및 작업 이력 등과 같은 사용자의 현재 상

황정보를 파악하고 분석하여 사용자 현 상황에 필요한 서비스를 제공하기 위한 기술이다. 이러한 상황정보는 수많은 센서들로부터 수집된 자료들을 분석하여 파악될 수 있다. 실제로 여러 분야에서 현재 이용되고 있는 위치기반서비스(LBS)도 상황인식의 한 분야이다 [1][2]. GPS를 이용한 위치기반서비스와 사용자의 움직임, 선호도, 스케줄 등의 정보를 온톨로지로 구축하여 실제 여행정보안내 스마트폰 앱을 만든 사례가 있다[3].

스마트폰을 이용한 사용자의 행동을 인지하여 서비스를 제공하는 시스템들도 개발되었는데, Bellotti 등은 윈도우 모바일 플랫폼에서 수집된 사용자 정보를 바탕으로 사용자의 행동 (Eating, Shopping 등)을 인식하고 콘텐츠를 추천해주는 Maggiti 시스템을 개발하였다[4]. Carnegie Mellon 대학교의 Context-Aware Mobile Phone 연구는 마이크로폰 센서를 통하여 가속도 및 온도 데이터를 입력 받아 사용자의 상태를 인지하는 SenSay 디바이스를 개발하였다[5]. 원격의료 서비스를 보완 개선하기 위해 스마트폰에 내장된 센서를 이용하여 의료 종사자와 환자를 지원하는 연구가 진행되었으며, 가속도 센서 데이터를 decision tree과 k-nearest neighbour, Sequential Minimal Optimization 분류 알고리즘을 적용하여 사용자의 활동을 인식하였다[6]. 버스 여행을 위해 지정된 목적지에 도달 할 때까지의 남은 시간을 실시간으로 예측하는 서비스를 위한 연구가 진행되었으며, 구축된 도로정보에 사용자의 상황인지 정보를 바탕으로 예측하는데 스마트폰에 내장된 가속도 센서 값을 이용하였다[7]. 돌발적인 상황을 인식하여 사용자에게 알림을 해주기 위한 연구도 진행되었는데, 스마트폰의 가속도 센서를 이용하여 사용자의 추락, 도난과 같은 상황을 인식하여 사용자에게 알려주는 어플리케이션을 개발하였다[8]. 또한 사용자의 허리에 부착하여 일상적인 움직임에 대한 가속도 변화와 기울기의 변화를 측정하여 낙상 감지 시스템을 구현하였다[9].

또한 스마트폰 사용자의 이동 상황을 판별하는 방법에 대해 연구한 사례들이 있다. 스마트폰 사용자가 이동 중인지 아닌지를 판단하기 위해 가속도 센서를 이용하여 사용자의 이동 상황을 판단하는 알고리즘이 연구되었다[10]. GPS와 스마트폰에 자체 내장되어 있는 회전 속도 (orientation) 센서, 네트워크 디바이스를 이용하여 사용자의 이동 상황을 판단하는 알고리즘이 제안되었다[11].

그러나 [10]과 [11]의 연구에서는 사용자의 스마트폰 사용 여부를 반영하지 않고 가방이나 주머니에 넣은 상태를 가정하여 연구를 진행하였다. 사용자가 정지 상황에서 스마트폰을 이용하여 문자를 보내거나 게임을 하는 경우를 가정해보자. 특히 자이로스코프 센서를 사용하는 앱을 이용 중이라면 위 두 논문의 경우 정확도가 현저히 떨어질 것이다.

본 논문에서는 사용자가 스마트폰을 사용 중일 때도 스마트폰 사용자의 이동 상황을 정확하게 판별하는 것을 목표로 한다. 이를 위해 스마트폰의 이벤트 값을 이용하여, 사용자의 스마트폰 사용 여부를 판단한다. 또한 스마트폰에서 제공되는 각 센서들의 데이터를 분석하여, 사용자의 이동 상황을 판단 할 수 있는 의미 있는 센서를 검증하여 사용 여부 판별에 적용한다.

### 3. 제안하는 스마트폰 사용 및 이동 상황 판별 기법

본 장에서는 스마트폰 사용 및 이동 상황을 정의하고, 수집한 데이터 분석을 통하여 스마트폰 사용 및 이동 상황 판별 알고리즘을 제안한다.

#### 3.1 스마트폰 사용 및 이동 상황 정의

본 연구에서는 스마트폰 사용자의 이동 상황을 "기다림", "걷기", "뛰기" 3가지로 구분한다. 또한 스마트폰 사용 여부를 "미사용", "사용" 2가지로 구분한다. 본 논문에서 스마트폰의 사용 여부 및 이동 상황을 <표 1>과 같이 6가지로 정의한다. "사용"은 사용자가 스마트폰을 이용하여 동영상을 보거나 어플리케이션을 이용하는 것이다. 예를 들어, 사용자가 기다리거나 걷는 상태에서 영화를 보거나 문자를 보내거나 게임을 하는 등 스마트폰을 조작하는 상태이다.

<표 1> 스마트폰 사용 및 이동 상황 정의

	"미사용"	"사용"
"기다림"	NotUseWait	UseWait
"걷기"	NotUseWalk	UseWalk
"뛰기"	NotUseRun	UseRun

#### 3.2 데이터 수집

스마트폰 사용 여부를 구분하기 위한 이벤트들은 <표 2>와 같다. 스마트폰에서 제공하는 이벤트란 사용자와 어플리케이션 사이 또는 어플리케이션들 간의 일어나는 어떤 일들을 의미하는 것이다. <표 2>에서 OnScreenOnOff 이벤트는 스마트폰의 화면이 켜짐에 대한 상태를 반환하고, onTouchListener는 스마트폰에서 터치 발생 상태를 반환한다. 이 두 이벤트를 이용하여 사용자의 스마트폰 사용 여부를 확인할 수 있다.

예를 들면 OnScreenOnOff가 false를 갖는다면 이는 스마트폰을 사용하지 않고 있다고 판단할 수 있다. OnScreenOnOff가 true이고, onTouchListener가 false를 갖는다면 사용자는 스마트폰을 조작하는 것이 아니라 영화를 보거나 음악을 듣는 등의 행위를 하고 있다고 판단이 가능하다. 만일 OnScreenOnOff가 true이고, onTouchListener가 true를 갖는다면 사용자는 스마트폰을 조작하여 직접 사용하고 있다고 판단할 수 있는 것이다. 본 논문에서는 스마트폰의 "사용"과 "미사용"으로만 구분하기 때문에 OnScreenOnOff 값만을 이용하도록 한다.

<표 2> 본 논문에서 이용하는 이벤트들

Android Event	
이벤트 리스너의 이름	이벤트 발생 시점
onTouchListener	사용자가 화면에 뷰를 터치할 때
onScreenOnOff	화면의 On/Off 상태 체크

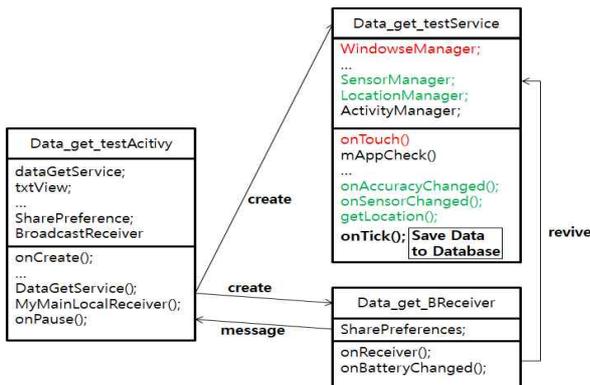
다음 <표 3>은 일반적인 스마트폰에 탑재되어 있는 센서들을 정리한 내용이다.

<표 3> 스마트폰에 내장되어 있는 센서 종류

센서	센서 설명
ACCELEROMETER	3차원 중력 가속도를 측정하는 가속도 센서
GYROSCOPE	회전운동 센서
LIGHT	빛의 밝기를 측정하는 조도 센서
MAGNETIC_FIELD	3차원 자기장 세기를 측정하는 지자기 센서 (나침반)
ORIENTATION	기기의 현재 방향결정
PRESSURE	압력 센서
PROXIMITY	어떤 물체와의 거리를 측정하는 근접 센서
TEMPERATURE	온도를 측정하는 온도 센서

데이터 수집을 위해 <그림 1>에 보이는 바와 같은 안드로이드 프로그램을 SDK 4.0을 이용하여 구현하였으며 이 프로그램은 세 개의 클래스로 구성된다.

Data\_get\_testActivity는 Data\_get\_testService와 Data\_get\_BReceiver를 생성하고 실행시키는 클래스이다. Service는 Sensor 값과 GPS 값이 변할 때마다 스마트폰의 상태에 대해 확인하여 Data를 저장하는 클래스이다. 브로드캐스트 리시버는 서비스가 종료되지 않도록 유지해주는 클래스이다.



<그림 1> 스마트폰의 데이터를 추출하는 프로그램의 구조

### 3.3 데이터 분석

<표 3>에서 언급한 센서의 효용성을 검증하기 위해 <표 4>와 <표 5>에서 <표 1>에서 정의한 6가지

각 상태별로 수집된 데이터의 평균과 분산을 나타낸다. <표 4>와 <표 5>에서 Acc는 3축 가속도센서들의 평균 변화량에 대한 값이며, Acc\_X, Acc\_Y, Acc\_Z는 각 축의 가속도센서 변화량이다. Ori는 3축 회전가속도센서들의 평균 변화량에 대한 값이며, Ori\_X, Ori\_Y, Ori\_Z는 각 축의 회전가속도센서 변화량이다. 이 항목까지는 [10]과 비교하기 위해 0.1초마다 각 상태별로 2분씩 수집한 데이터에 대한 평균과 분산이다. Ori2는 [11]과 비교를 위해 0.25초마다 각 상태별로 2분씩 수집한 데이터로 3축 회전가속도센서들의 평균 변화량에 대한 값이며, Ori2\_X, Ori2\_Y, Ori2\_Z는 이때의 각 축의 회전가속도센서 변화량이다.

센서 데이터 항목 중 온도, 압력, 조도 센서, 자기장 센서, 중력가속도 센서는 움직임에 따른 변화가 없어서 제외하였다.

<표 4> 스마트폰 “미사용”상태일 때의 이동 상황별 평균과 분산

State TestData	NotUseWait		NotUseWalk		NotUseRun	
	Mean	Variance	Mean	Variance	Mean	Variance
Acc	0.2507	0.8487	5.1329	18.4416	11.0555	27.0699
Acc_X	0.3316	1.8753	3.8180	10.4966	8.7075	46.8877
Acc_Y	0.1805	0.3079	5.7708	44.003	14.0267	94.2394
Acc_Z	0.24	1.275	5.8098	50.4556	10.4622	114.526
Oi	1.2099	102.9402	5.9817	231.4101	6.3778	55.7177
Oi_X	0.2657	0.1471	6.5069	29.0062	2.6729	3.6699
Oi_Y	0.6539	14.2699	1.7607	6.0064	2.8285	12.2859
Oi_Z	1.906	439.3	9.7145	200.7	6.9985	200.9
Ori2	1.4955	96.2659	9.4283	313.0502	8.8033	92.4707
Ori2_X	0.9964	52.7686	12.0672	56.1572	16.1572	175.8492
Ori2_Y	1.0681	28.5741	2.5885	7.7336	2.8563	16.5112
Ori2_Z	2.4221	429.6	13.6291	2715.1	7.3363	328.7

<표 5> 스마트폰 “사용”상태일 때의 이동 상황별 평균과 분산

State TestData	UseWait		UseWalk		UseRun	
	Mean	Variance	Mean	Variance	Mean	Variance
Acc	0.8305	1.2183	1.402	0.6851	3.8471	3.291
Acc_X	0.715	1.1941	1.0305	0.9336	2.7291	4.3204
Acc_Y	0.4256	0.4517	0.8655	0.4146	3.5577	6.3469
Acc_Z	1.3511	4.5306	2.3099	3.7695	5.2544	15.1438
Ori	1.7795	133.257	1.7245	36.8945	2.9028	71.0752
Ori_X	0.5411	1.5325	3.2549	27.1777	0.9378	88.7450
Ori_Y	0.6579	1.1015	1.1702	1.4656	1.6213	1.8054
Ori_Z	4.1129	11844	2.4349	3142	4.3904	6105
Ori2	29	228.657	54.047	2.8136	140.286	4.6284
Ori2_X	0.8878	2.7932	2.4115	4.8647	3.4272	14.2865
Ori2_Y	0.9336	5.0673	1.9186	4.3035	2.5341	7.4892
Ori2_Z	6.8456	1983.9	4.1105	456	7.9240	1162.1

<그림 2>는 3축 가속도 센서의 평균 변화량 분포의 평균을 나타내는 그래프이고, 각 상태별 평균값이 구분되는 것을 볼 수 있다. 데이터의 효율성을 판단하기 이전에 스마트폰의 사용 여부가 이동 상황 판별에 미치는 영향을 알아보려고 한다. <그림 2>에서 “미사용” 시에는 “기다림”, “걷기”, “뛰기”의 평균값이 각각 0.25, 5.13, 11.06인데, “사용” 시에는 “기다림”, “걷기”, “뛰기”의 평균값이 각각 0.83, 1.4, 3.85이다. 또한 “미사용”의 경우 구분이 비교적 명확한데 반해 “사용”의 경우 상태별 평균의 차이가 적은 것은 스마트폰을 사용하는 상황에서 사용자의 눈이 화면에 고정되어 있기 때문에 센서들의 변화량이 적은 것으로 판단된다. 이는 “미사용”의 결정경계를 “사용”의 경우에는 적용하기 어렵다는 사실을 말해준다. 이런 점에서 기존 연구들에는 “사용”의 경우 사용자의 이동 상황 판별이

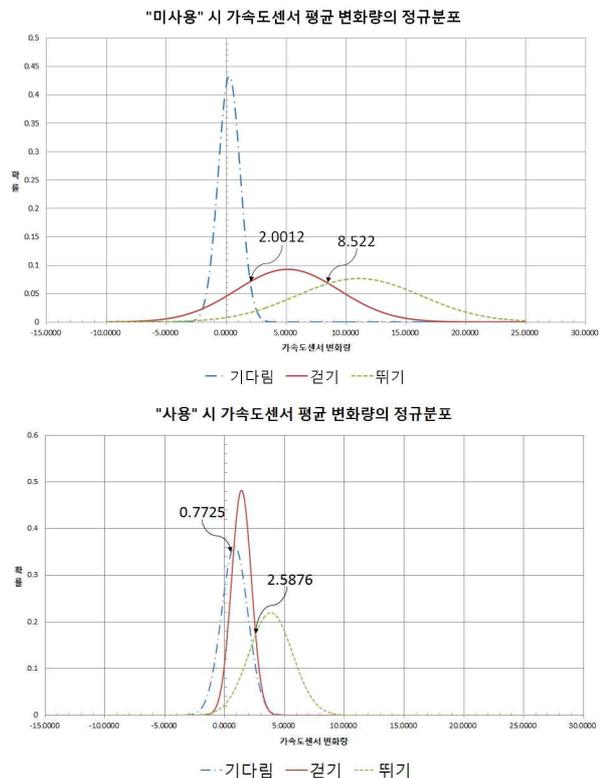


<그림 2> 3축 가속도센서의 평균 변화량

어렵다. 본 논문에서는 이러한 문제를 해결하기 위해 스마트폰에서 제공하는 이벤트로 사용자의 스마트폰 사용 여부를 판단하고 각각에 맞는 결정경계를 적용하여 사용자의 이동 상황을 판단하고자 한다.

데이터의 효율성을 2가지 조건으로 판단한다. 첫째로 상태별 평균값 사이의 간격이 큰 것과 두 번째로 상태별 분산이 작은 것을 찾는다. 평균값 사이의 간격이 크다는 것은 상태별로 구분이 명확하다는 것이고, 분산이 작다는 것은 분포가 평균값 주변에 밀집되어 있을 가능성이 높다는 것이다. 결과 0.1초마다 수집한 3축 가속도 센서 평균 변화량의 데이터 이외에는 평균값의 구분이 잘 안되고 상태별 분산이 크다. 그렇기 때문에 본 논문에서는 3축 가속도 센서의 평균 변화량을 이용하여 스마트폰 사용자의 이동 상황을 판별한다.

본 논문에서는 사용자의 이동 상황에 따른 데이터의 변화량이 정규분포를 따를 것이라고 가정하여 확률적으로 모델링한다. 데이터의 분포를 정규분포로 표현하여 제안하는 알고리즘에서 사용자의 이동 상황



<그림 3> 스마트폰 사용 여부에 따른 이동 상황별 가속도센서의 평균 변화량의 분포

판별을 위해 사용하는 “기다림”과 “걷기”, “걷기”와 “뛰기”를 구분할 결정경계를 구하고자 한다. <그림 3>은 스마트폰 사용 여부에 따라 “기다림”, “걷기”, “뛰기” 각각의 이동 상황에서 3축 가속도 센서의 평균 변화량의 정규분포를 그래프로 나타낸 것이며, 그래프에 표시된 수치들은 상태별 확률이 같아지는 지점으로 실험을 통해 얻은 결정경계들이다.

제안하는 알고리즘에서 “기다림”, “걷기”, “뛰기”의 구분을 위한 결정경계로 사용하는 값들은 상태별 정규분포 간 확률이 같아지는 부분으로 정한다. <표 6>은 스마트폰 사용 여부에 따른 이동 상황을 구분하기 위한 결정경계들이다.

<표 6> 사용 여부에 따른 “기다림/걷기/뛰기”의 결정경계

이동 상황 사용 여부	“기다림”&“걷기”	“걷기”&“뛰기”
“미사용”	2.0012	8.522
“사용”	0.7725	2.5876

예를 들어 “미사용”의 경우 2.0012보다 작은 변화량을 보인다면 “기다림”, “걷기”, “뛰기” 중 “기다림”일 확률이 가장 높다는 것이고 2.0012보다 크거나 8.522보다 작은 구간에서는 “걷기”의 확률이 가장 크다는 것이다.

### 3.4 스마트폰 사용 여부 및 이동 상황 판별 알고리즘

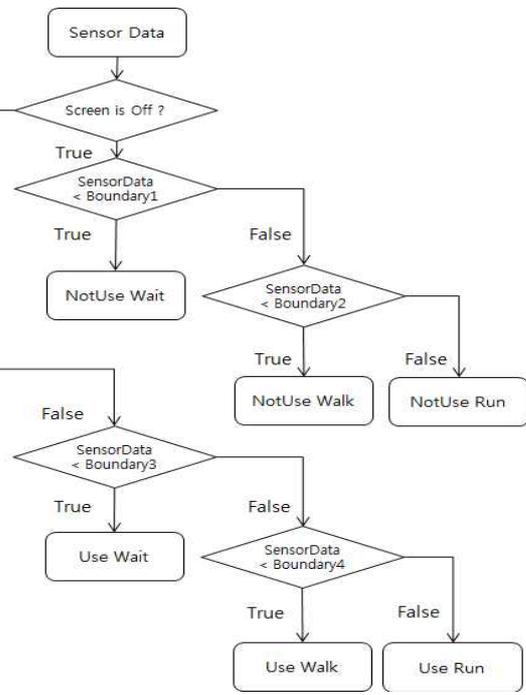
논문 [9]에서는 GPS의 위도, 경도 값에서 사용자의 속도를 구하여 기다림, 차량이동 등을 판별해낸다. 본 논문에서는 GPS가 실외에서는 일반적으로 0~25M 정도의 오차를 갖고, 실내에서는 50~100M 정도의 오차를 갖고, 기기마다 결과값이 차이가 나기 때문에 GPS를 이용하지 않는다. 본 논문에서는 GPS에서 속도를 구하여 차량 이동의 경우를 판단할 수 있다고 가정한다. 현재 스마트폰에서 사용되고 있는 GPS의 오차와 신호가 균일하지 못한 점은 기술적인 문제로써 논외로 한다. 본 논문에서는 스마트폰 사용 여부를 구분하여 사용자의 이동 상황 “기다림”, “걷기”, “뛰기”에 대해 판단하고자 한다. 첫 번째로 화면의 켜/꺼짐 이벤

트를 이용하여 사용자의 스마트폰 사용 여부를 판단한다. “미사용”, “사용” 여부를 확인 후 두 번째로 데이터 분석에서 구한 3축 가속도 센서의 평균 변화량에 대한 결정경계를 이용하여 사용자의 이동 상황인 “기다림”, “걷기”, “뛰기” 상태를 판별한다.

<그림 4>는 제안하는 알고리즘의 의사코드(Pseudo Code)와 순서도이다. i) OnScreenOnOff 리스너의 값을 통해 사용자의 스마트폰의 “미사용”과 “사용” 여부를 확인한다. ii) OnScreenOnOff 리스너의 값이 true인 경우 “미사용”으로 판단한다. 다음으로 스마트폰의 가속도센서에서 얻은 3축 가속도센서들의 평균 변화량을 구하여 결정경계 Boundary1과 2를 이용하여 “기다림”, “걷기”, “뛰기”를 판단하게 된다. iii) OnScreenOnOff 리스너의 값이 false인 경우 “사용”으로 판단한다. 다음으로 스마트폰의 가속도센서에서 얻은 3축 가속도센서들의 평균 변화량을 구하여 결정경계 Boundary3과 4를 이용하여 “기다림”, “걷기”, “뛰기”를 판단하게 된다. Boundary들은 데이터 분석의 <표 6>에 있는 결정경계 값들이다. Boundary1은 “미사용”시 “기다림”과 “걷기”를 구분하기 위한 결정경계이고 Boundary2는 “미사용”시 “걷기”와 “뛰기”를 구분하기 위한 결정경계이다. Boundary3은 “사용”시 “기다림”과 “걷기”를 구분하기 위한 결정경계이고 Boundary4는 “사용”시 “걷기”와 “뛰기”를 구분하기 위한 결정경계이다.

```

Input : 3축 가속도의 평균 값(from Sensor Data)
Output : 사용자의 상태
IF (Screen is off==true) THEN //“미사용”
  IF Data<Boundary1 THEN PRINT(NotUse Wait)
  ELSEIF Data >= Boundary1 && Data < Boundary2 THEN
    PRINT(NotUse Walk)
  ELSEIF Data >= Boundary2 THEN PRINT(NotUse Run)
ENDIF
ELSEIF (Screen is off==false) THEN //“사용”
  IF Data < Boundary3 THEN PRINT(Use Wait)
  ELSEIF Data >= Boundary3 && Data < Boundary4
    THEN PRINT(Use Walk)
  ELSEIF Data >= Boundary4 THEN PRINT(Use Run)
ENDIF
ENDIF
  
```



<그림 4> 스마트폰 사용자의 이동 상황을 판단하는 알고리즘

#### 4. 제안한 알고리즘의 정확도 비교 분석

<표 7>은 [10]과 [11]의 알고리즘과 제안하는 알고리즘에 대한 스마트폰 사용 여부에 따른 사용자의 "기다림", "걸기", "뛰기" 상태 판별에 대한 정확도를 비교한 것이다.

<표 7> 제안하는 알고리즘의 정확도 비교

		논문 [10]	논문 [11]	제안하는 알고리즘
"미사용"	"기다림"	59.3	99.6	97.3
	"걸기"	10.6	0.4	75.5
	"뛰기"	판단안함	0	65.1
"사용"	"기다림"	34.7	97.1	69.2
	"걸기"	2.06	0.03	67.7
	"뛰기"	판단안함	0	71.7

<그림 5>는 제안한 알고리즘을 테스트하기 위해

수집한 데이터의 스크린 샷이다. Tran\_Acc는 3축 가속도센서들의 변화량의 평균이며, Tran\_Ori는 3축 회전가속도 센서들의 변화량의 평균 값이다. [10]과 비교하기 위해 0.1초마다 각 상태별로 10분씩 상태별로 6000개의 데이터를 수집하였다. Ori2는 [11]과 비교를 위해 0.25초마다 10분씩 상태별로 2400개의 데이터를 수집하였다.

#	U	V	W	X	AC	AD	AE	AF
1	Tran_Acc	Tran_Acc	Tran_Acc	Tran_Acc	Tran_Ori	Tran_Ori	Tran_Ori	Tran_Ori
2	18.137299	3.9151155	5.748405	7.267273187	5.64896	2.767273187	30.573185	5.64896
3	11.898981	1.7656802	4.7595107	6.341357723	12.386095	5.82176	9.05549	9.08716687
4	10.056474	4.2989502	2.30300903	5.52811077	11.099535	12.40893	16.46906	13.32584167
5	7.9837643	3.4545135	7.9089977	6.848425167	65.3048	12.67438	87.423	35.13406
6	1.9959413	1.0747375	5.2969208	2.789199867	15.730095	4.52958	3.8851	8.048258333
7	1.19328	3.1474457	1.282715	3.8838484	30.789645	2.33727	34.77507	22.633995
8	17.5796355	3.5312806	6.0645904	9.038502187	4.99499	20.085156	2.81477	11.964912
9	9.2888031	0.4606018	2.072708	3.9407043	0.80287	6.072262	4.55008	3.808404
10	4.9130859	1.3818054	7.8302308	4.708374033	7.63527	22.740827	8.51227	12.962789
11	18.1170945	26.17735	0.076767	14.700435	0.0959	10.117203	7.7966	6.009001
12	19.345276	30.937087	0.8444366	17.0426653	20.82231	12.3594164	9.68397	14.28856547
13	6.985794	2.303009	5.6807556	4.989852867	5.163596	0.2636647	8.42934	4.6188669
14	8.5211333	13.4341	4.889856	8.98174067	1.366454	1.3007267	1.45303	1.37340367
15	3.5312805	10.056472	13.434219	9.00732833	6.255445	0.8718207	7.92456	5.017275233
16	1.6888733	0.614136	2.7636108	1.688873367	6.1427	3.4234577	4.82255	4.7962359
17	3.1474457	0.690903	6.9857942	3.608047633	9.684815	2.1658972	9.76929	7.2066674
18	0.5373688	7.753464	2.1494752	3.480102667	11.41845	0.1100289	8.72374	6.750739633
19	1.6121063	0.460602	4.3454163	3.20727082	4.89042	1.0514897	1.316	2.41963233
20	2.9939117	2.226242	0.2303009	1.8168182	5.593164	0.5757264	6.09664	4.088510133
21	8.4443664	7.216095	6.1413574	7.267272933	10.279386	2.2785891	1.31505	4.6243417
22	6.3718063	15.9675293	0.9212036	7.753463723	14.00039	0.8334038	6.10216	6.97851267
23	1.9959412	1.3818054	2.5338099	1.970352167	0.185076	0.7368898	4.09028	1.6707486
24	9.9797055	28.4805447	41.3006285	26.58695957	5.068674	6.8080637	7.42721	6.434649233
25	3.608047	4.913085	46.520782	18.34730467	5.41207	1.4325	0.54911	2.46456
26	4.6827853	24.71896426	7.7534635	12.38507102	2.712375	1.836131	0.55728	1.701928667
27	5.4504847	6.37180626	4.606018	5.476048633	9.166145	0.877989	3.29414	4.446091333
28	5.6807556	0.076767	3.83834834	3.198623847	5.78323	1.6029074	1.1196	2.82887133
29	3.6848144	6.2181245	6.14135744	5.34809878	9.20884	2.1761834	2.7396	6.2163278
30	0.76769687	0.6141355	2.7636109	1.813805357	1.01138	0.423982	2.26495	1.25254
31	4.60601813	2.226242	5.9110565	4.24777221	7.2895	0.77272	2.37057	3.47596667
32	3.1474457	9.5958707	0.9979706	4.880429	17.39706	1.601698	6.03861	8.545789333
33	5.2989208	0.9212036	2.7636109	2.993911767	1.73044	0.677896	0.02311	1.810472
34	6.4484253	5.9878235	2.2262421	4.887496967	3.34559	7.527743	11.07812	7.317151
35	6.4484252	38.4602508	11.4382796	18.7823182	3.86896	0.9285	3.70381	2.83756667
36	10.0564727	28.86498	11.2079777	16.70961013	3.41623	1.284337	0.93098	1.870535667

<그림 5> 알고리즘 테스트를 위해 수집한 데이터

논문 [10]은 3축 가속도 센서의 평균값의 변화량을 이용하여 이동 상황을 판별하였다. 이 논문에서는 하나의 레코드(평균값)로 판단하지 않고 10개의 레코드 간 변화량을 구하여 0.2의 결정경계를 넘는 구간의 수가 과반수이상이면 "걸기"로 판단하고, 과반수미만이면 "기다림"으로 판단한다. 이 논문에서 사용한 결정 경계인 0.2는 데이터 셋에 대해 임의의 결정경계를 적용하여 알고리즘의 정확도를 구해 이 중 가장 높은 정확도를 보인 결정경계의 수치이다. [10]의 정확도 결과를 보면 "미사용"의 경우 논문에서 주장했던 정확도에 비해 현저히 낮게 나온다. 이는 데이터 셋의 차이 때문이라고 판단할 수 있다. 하지만 "사용"의 경우에는 "기다림"의 경우 34.7%의 정확도를 보이고 "걸기"의 경우 2%의 정확도를 보인다. "기다림"과 "걸기"를 하나의 결정경계로 구별하기 때문에 "사용"시 "기다림"의 정확도는 신뢰할 수 있는 수치라 아니게 된다.

논문 [11]는 y축 각속도 센서의 변화량을 이용하여 이동 상황을 판별하였다. 이 논문에서는 수집한 데이터 셋을 분석하여 상태별 최소/최대 결정경계를 구하였다. 하지만 실제 실험을 통해 알고리즘의 정확도를 평가하지는 않았다. 위 논문에서 제시한 결정경계를

바탕으로 수집된 데이터 셋에 적용하여 정확도를 구해본 결과 “미사용”, “사용”에서 “기다림”의 정확도는 매우 높은 수치를 보인다. 하지만 “걷기”의 정확도가 0에 가깝다는 것은 결론적으로 판별이 어렵다는 것을 나타낸다.

<표 7>의 결과에서 본 논문에서 제안하는 알고리즘이 논문 [10]과 [11]보다 “미사용”의 경우 월등한 정확도와 판별력을 보인다. “사용”의 경우에는 “기다림” 상태에서 타 논문 [10]보다 높고 [11]보다 낮은 정확도를 보인다. 하지만 “기다림”과 “걷기”를 구분하는데 필요한 결정경계가 하나라는 점을 고려한다면 [11]의 “사용”시 “기다림”의 매우 높은 정확도는 0에 가까운 “걷기”의 정확도로 인해 판별력이 매우 떨어지는 것으로 판단할 수 있다. 따라서 본 논문의 알고리즘은 타 논문의 것들보다 판별력이 높은 것을 알 수 있다.

또한 “사용”의 경우, “미사용”의 경우보다 정확도가 전체적으로 낮아지는 것을 볼 수 있다. 이는 사용자가 기기를 조작하는 경우 가속도 및 각속도가 발생하기 때문에 얻어지는 센서 값들의 변화량의 폭이 증가한 결과로 판단된다.

## 5. 결 론

지금까지 스마트 디바이스에 내장된 센서를 이용하여 사용자의 이동행태를 판별하는 연구는 다양하게 진행 되었다. 그러나 사용자의 스마트 디바이스 사용 여부를 고려하지 않고 진행한 연구들이라는 점에서 일반적인 상황에는 적합하지 않았다. 본 논문에서는 이러한 문제점을 해결하기 위해 스마트 디바이스에서 제공되는 이벤트를 이용하여 사용자의 스마트 디바이스 사용 여부를 인지한다. 또한 스마트 디바이스 사용 여부에 따른 사용자의 이동 상황을 판별하기 위해 본 논문에서 정의한 6가지 상태별 가속도 센서의 평균 변화량 데이터를 정규분포로 모델링하여 각 상태를 구분하기 위한 결정경계들을 구하였다. 얻어진 결정경계 값들을 이용한 제안한 알고리즘을 평가한 결과 타 논문에 비해 “미사용”의 경우 월등하거나 비슷한 판별의 정확도를 보이며, “사용”의 경우 높은 판별의 정확도를 보였다.

향후 연구로서 스마트 디바이스를 사용하지 않는 경우, 타 연구에 비해 낮은 알고리즘의 정확도를 기

학습이나 패턴인식 기법을 활용하여 보완할 것이다. 더 나아가 HMM (Hidden Markov Model)등을 이용하여 사용자의 이동행태를 예측할 수 있도록 할 것이다.

## 참 고 문 헌

- [1] P. Zheng and L. M. Nio, "The rise of the smart phone," IEEE Distributed System Online, Vol.7, No.3, 2010.
- [2] L. Yu and other, "Applying Context-awareness to Service-oriented Architecture," In proceedings of IEEE International E-Business Engineering Conf. (ICEBE'09), pp. 397-402, 2009.
- [3] 이재길, 주용진, 박수홍, "온톨로지기반 상황인지 추론 시스템 설계 및 구현 - 여행정보안내 스마트폰 앱을 사례로", 한국지형공간정보학회지, 제20권, 제4호, pp. 67-75, 2012.
- [4] V. Bellotti, B. Begole, E. H. Shi, N. Ducheneaut, J. Fang, E. Isaacs, T. King, M. W. Newman, K. Partridge, B. Price, P. Rasmussen, M. Roberts, D. J. Schiano, and A. Walendowski, "Activity-based serendipitous recommendations with the magitti mobile leisure guide," CHI 2008 Proceedings, pp. 1157-1166, 2008.
- [5] Siewiorek, D., A. Samaligic, J. Furukawa, N. Moraveji, K. Reiger, and J. Shaffer, "SenSay: A context-Aware Mobile Phone," In Proc. of IEEE Symposium on Wearable Computers (ISWC), pp. 248-249, 2003.
- [6] Sian Lun Lau, I. Konig, K. David, B. Parandian, C. Carius-Dussel, M. Schultz, "Supporting patient monitoring using activity recognition with a smartphone," Wireless Communication Systems (ISWCS), 2010.
- [7] King-Seng Ang, Chen-Khong Tham, "Smartphone-based vehicular and activity sensing", Networks (ICON), 2012 18th IEEE International Conference on, pp. 1-6, 2012.
- [8] 차경애, 여선동, "스마트폰 가속도 센서 기반의 돌발 상황인식 어플리케이션 개발," 한국산업정보학

회 논문지, 제17권, 제5호, pp. 9-17, 2012.

- [9] 류정탁, “3축 가속도센서와 기울기 센서를 이용한 낙상감지시스템 개발,” 한국산업정보학회 논문지, 제18권, 제4호, pp. 19-24, 2013.
- [10] 임재걸, 이계영, 심규박, 레콩탄, “스마트폰이 이동 중인지 아닌지를 판단하는 방법,” 한국멀티미디어학회 논문지, 제15권, 제5호, pp. 632-638, 2012.
- [11] 하동수, 박성준, “스마트폰 기반의 사용자 이동상태 판별 알고리즘,” 한국컴퓨터정보학회 논문지, 제16권, 제3호, pp. 167-174, 2011.



**홍 민 성** (Min-Sung Hong)

- 학생회원
- 단국대학교 컴퓨터공학과 학사
- 단국대학교 컴퓨터과학과 석사과정
- 관심분야 : 상황인식, 센서, 모바일

네트워킹, 데이터마이닝



**목 남 희** (Nam-Hee Mok)

- 정회원
- 이화여자대학교 정치외교학과 학사
- Kentucky State University 회계과정 이수
- Cleveland State University 회계정보 석사
- 단국대학교 경영학 박사
- 단국대학교 상경대학 경영학과 조교수
- 관심분야 : 경영 정보시스템, 의사결정, 데이터마이닝

논문접수일 : 2013년 09월 26일  
1차수정완료일 : 2013년 10월 26일  
2차수정완료일 : 2013년 11월 20일  
게재확정일 : 2013년 11월 25일