

## 스마트 폰 기반의 사용자 상황인지 플랫폼에 관한 연구

김기백\*, 장원석\*, 최재완\*, 임철수\*\*, 최종호\*\*\*

### A Study on the User Context Awareness based on the Smart Phone

Kee Baek Kim\*, Won Seok Jang\*, Jae Wan Choi\*, Chul-Su Lim\*\*, Jong Ho Choi \*\*\*

#### 요약

이동통신 및 휴대폰 관련 사업이 급격하게 성장하고, 이를 활용한 다양한 부가서비스가 출현함에 따라 다양한 기능을 제공할 수 있는 스마트 폰은 향후 가장 주목받는 기술로 전망되고 있다. 본 논문에서는 스마트 폰을 활용하여 사용자의 상황에 적합한 인터페이스를 제공하기 위한 상황인지(context-awareness) 인터페이스 플랫폼을 설계하였다. 인터페이스 플랫폼은 스마트 폰에 장착된 센서로부터 측정된 데이터를 분석하여 사용자의 정지 및 이동 상황, 조도, 위치, 장소에 관한 상황을 설정함으로써 인지된 상황에 적절한 이벤트를 발생하여 효율적이고 직관적인 사용자 스마트폰 인터페이스를 제공하는 것을 목표로 하고 있다. 제안한 인터페이스 플랫폼의 효용성을 확인하기 위해 실제 슈팅 게임 시스템에 적용한 결과, 기존의 방법들보다 직관적인 인터페이스를 제공할 수 있음을 확인하였다. 제안한 플랫폼은 게임 환경이나 스마트폰 어플리케이션 환경에서의 인터페이스 구현에 응용이 가능하다.

▶ Keyword : 스마트 폰, 상황인지, 안드로이드, 센서, 인터페이스

#### Abstract

As it is developing and growing up the mobile telecommunication, commercial mobile, and the application on it, the smart phone is attracting public attention techniques, hereafter. In this paper, we can make the context-awareness platform to supply users with the interface for their appropriate contexts using the smart phone. Therefore, we use the sensors in the smart phone, and

• 제1저자 : 김기백 • 교신저자 : 최종호

• 투고일 : 2011. 12. 16, 심사일 : 2012. 01. 12, 게재확정일 : 2012. 01. 17.

\* (주)유토(Uto Co., Ltd.)

\*\* 서경대학교 컴퓨터공학과(Dept. of Computer Engineering, Seokyeong University)

\*\*\* 강남대학교 전자공학과(Dept. of Electronic Engineering, Kangnam University)

※ 이 논문은 전라남도 및 (재)전남문화산업진흥원의 2011년도 문화산업육성지원사업 연구결과임..

the values from them are analyzed. As a result, we can estimate the user's contexts, such as the sitting and standing, movement, the lights, positions, locations, etc. In addition, we happen the suitable events from them for users. Therefore, we can provide the perspective interfaces for them. In addition, to prove the more efficiency of the proposed platform, we implemented the shooting game, and could know that the performance is the more advanced than the others. The proposed algorithm can be used for the mobile environments, which high efficiency is required such as the game, or the implementation of mobile applications which are more efficient to the user interfaces, etc.

▶ Keyword : smart phone, context-awareness, android, sensor, interface

## I. 서론

이동통신 및 휴대폰 사업이 급격하게 성장하고, 이를 활용한 다양한 부가서비스들이 출시됨에 따라 다양한 기능을 제공할 수 있는 스마트 폰은 향후 가장 주목받는 기술로 전망되고 있다. 스마트 폰은 기존의 음성통화 중심의 휴대전화 기능에 인터넷 기능과 PC 유사 기능을 추가하여 다양한 응용 프로그램을 실행할 수 있는 차세대 모바일 단말이다. 스마트 폰을 이용하면 사용자가 언제 어디서나 응용 프로그램을 이용하고 인터넷에 접속하여 정보검색 기능을 수행할 수 있다[1-2].

최근 들어 스마트 폰을 필두로 모바일 유틸리티스 환경을 실현하기 위한 다양한 연구들이 활발히 진행되고 있다. 특히 사용자에게 언제 어디서나 새롭고 편리한 서비스를 제공할 수 있다는 장점에 착안한 핵심기술 중 하나가 상황인지(Context Awareness)기술이다. 스마트 폰은 서비스를 이용하는 사용자의 요청에 가장 알맞고 유용한 정보를 제공하기 위해 주어진 상황(Context)을 적절히 처리하는 과정이 필요하다. 상황을 인지하여 사용자에게 양질의 정보를 제공하는 기술을 상황인지 기술이라 한다[3].

스마트 폰이 상황인지를 위해 필요로 하는 상황정보는 사용자가 컴퓨터와 통신하는 시점의 정보를 의미하는데, 스마트 폰에 장착된 각종의 센서로부터 정보를 수집하여 시간, 위치, 활동 상태 등 모든 정보를 수집할 수 있다. 또한 이 데이터를 적절히 가공하여 상황정보로 변환할 수 있으며, 이 과정에서 해석, 추론, 학습 과정을 거쳐 사용자의 상황에 맞는 적절한 서비스를 제공할 수 있다. 이러한 상황인지 스마트 폰 서비스는 최근 유틸리티스 환경과 결합하여 의료, 교육, 구호, 쇼핑, 개인정보 보호 등 전 분야에 적용될 수 있다[3].

본 논문에서는 범용 스마트 폰 환경에서 상황인지 시스템을 구축하기 위한 플랫폼을 설계하고, 이를 시스템으로 구현하였다. 시스템 구현 시 현재 널리 사용되고 있는 안드로이드, iOS 기반의 모바일 상황인지 플랫폼을 각 OS 환경에 맞게

설계하였다. 설계된 플랫폼은 OS와 기기별 센서로부터 입력되는 값을 이용하여 사용자의 행동 상황을 정의하고, 정의된 사용자의 행동 상황을 추론·해석하여 각각의 어플리케이션을 제어하는 이벤트를 발생한다. 위 과정은 4단계로 정의될 수 있다. 정보를 수집하는 수집단계, 이를 저장하는 저장단계, 각 상황에 맞는 이벤트로 표현하는 재표현 단계, 재표현 후 상황을 해지하는 소멸 단계로 구분된다. 본 논문은 이러한 단계를 통해 효율적인 상황인지 인터페이스를 사용자에게 제공하고 직관적인 방법으로 사용자가 스마트 폰 모바일 기기를 제어하는 방법을 제안하였다. 그림 1에 제안한 플랫폼의 블록도를 나타냈다.

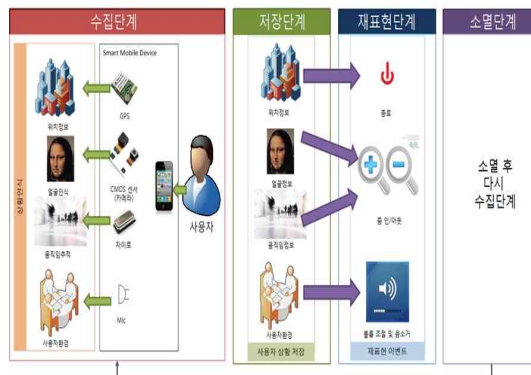


그림 1. 제안한 플랫폼의 블록도  
Fig. 1. Block diagram of proposed platform

## II. 기존 연구 및 사례

모바일 환경에서의 상황인지에 관련된 연구가 지속적으로 수행되고 있다. 그러나 기본적으로 플랫폼은 하드웨어를 제어하는 스마트 폰의 OS에 종속되어 제작되고 있다. 상위 어플리케이션과 모바일 OS를 중간에서 적용하는 컨버전스 형태의 플랫폼으로 제작하는 것이 최근의 동향이다. 그림 2에 상황인지 플랫폼 구조를 나타냈다.

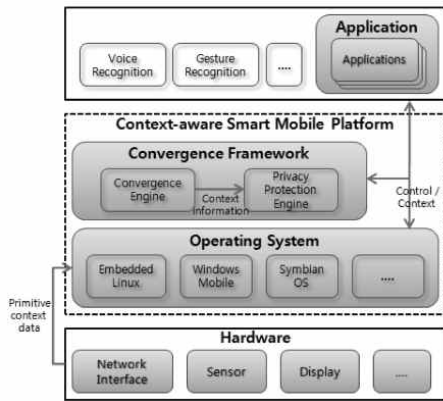


그림 2 상황인지 스마트 모바일 플랫폼 구조  
Fig. 2. Context-awareness mobile platform

모바일 플랫폼에서 상황을 설정하는 연구는 최초 Schilit와 Theimer에 의해 1994년에 논의된 이후, 현재까지 활발하게 진행되고 있다. 초기 연구에서는 주로 LBS(Local Based Service) 중심의 GPS를 사용한 연구가 진행되었으나, 현재는 실제 모바일 센서 값을 이용하여 상황을 정의하는 형태로 변화하고 있다. 특히 2004년 Daniel Siewiorek과 Asim Smailagic의 SenSay를 대표적인 예로 들 수 있다. SenSay에서는 가속도, MIC, 조도 센서를 이용하여 Palm OS 기반의 시스템을 정의하고 있다. 결과적으로는 사용자의 활동과 소음도 및 조도에 따른 상황을 정의한 것이다. 이와 다른 방향의 연구로 K. Rehman은 상황인지 기술에서 시스템과 사용자의 소통문제를 제시하고 이를 위한 소통모델을 개발하였다. 또한 J. Strassner는 상황인지에서 통합적인 상황 정보 처리를 위한 참조 아키텍처인 FOCAL을 제안하였다 [4-9]. 이와 같은 연구를 통해 일반적인 상황정보는 사용자 상황, 물리적 환경 상황, 컴퓨팅 시스템 상황, 사용자-컴퓨터 상호 작용 이력, 기타 미분류 상황 등으로 분류하였으며, 현재는 이를 기반으로 사용자 상황, 신원 상황, 신체 상황, 물리적 환경 상황, 공간 상황, 시간 상황, 환경 상황, 컴퓨팅 시스템 상황, 가용자원 등의 분야로 세분화되었다[10-12].

본 논문에서는 기존 연구들과는 차별화된 것으로 현재 사용되고 있는 스마트 모바일 기기를 대상으로 사용자의 물리적 환경, 공간 상황, 환경 상황 등을 대상으로 연구를 진행하고, 이를 범용 환경용 플랫폼으로 설계하였다.

### III. 스마트 폰 상황인지 플랫폼 설계

본 논문에서 제안한 상황인지 플랫폼은 상용 스마트

폰에 장착된 각 센서들을 이용하여 사용자의 정보를 인지한 후, 이를 다양한 이벤트로 발생시키는 기능을 수행한다. 가속도 센서와 자이로 센서를 이용하여 사용자의 정지 상황 및 이동 상황을 파악하여 어플리케이션 화면의 줌 인/아웃 제어를 제공하고, 잡음 센서를 이용하여 주변 소음도를 측정함으로써 볼륨을 제어하며, 조도 센서를 이용하여 주변 밝기에 따른 화면의 밝기를 제어하고, GPS 센서를 통해 장소에 따른 이벤트 즉 도서관이나 학교에서 어플리케이션을 종료하는 등의 이벤트를 제공한다. 그림 3에 스마트 폰 상황인지 플랫폼을 나타냈다.

#### 1. 센서 기반 플랫폼 설계

본 논문에서는 각종의 센서로부터 계속되는 데이터를 활용하여 이벤트를 생성하는 안드로이드 및 iOS 기반의 플랫폼을 설계하였다. 안드로이드에서는 범용 기기인 삼성전자의 갤럭시 S2에서 제공하는 센서를 기준으로, 그리고 iOS에서는 iPhone 4를 기준으로 플랫폼을 설계하였다. 갤럭시 S2에서 지원하는 센서는 자이로, 가속도, 잡음, 조도, GPS이고, iOS에서 지원하는 센서는 자이로, 가속도, 잡음, GPS 등이다.[1-2]

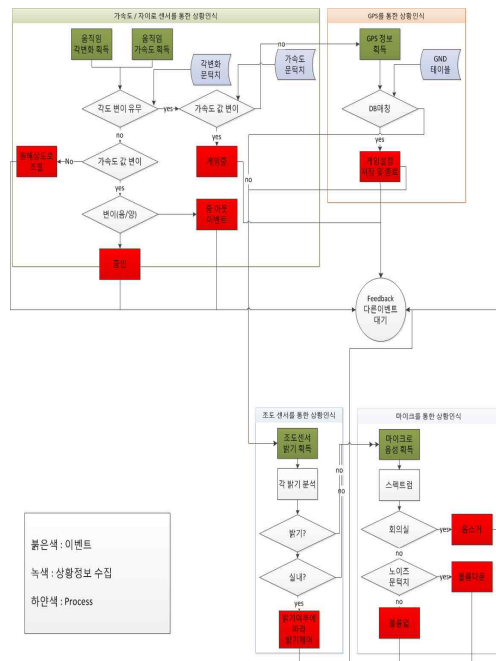


그림 3. 스마트폰 상황인지 플랫폼 흐름도  
Fig. 3. Smart mobile context awareness platform flow chart

2. 자이로 센서를 이용한 이벤트 생성

스마트 폰에 장착된 자이로 센서(gyroscope)는 각속도 센서로 x,y,z 축으로의 변화 값을 표출할 수 있도록 플랫폼을 설계하였다. 이를 통해 기존 가속도 센서의 문제인 이동에 대한 벡터 값이 아닌 절대 값을 추출할 수 있다. 본 논문에서 대상으로 하고 있는 갤럭시 S2와 iPhone 4에서는 동일한 STMicroelectronics사의 L3G4200D를 사용하고 있으므로, OS에서 처리하는 결과 값만 차이가 있다. 그림 4에 안드로이드와 iOS에서 각도별로 처리되는 자이로 센서 결과 값을 나타냈다. 측정값은 x축을 기준으로 90도 회전한 값이며, 90도를 기준으로 다른 회전 값과 대칭되는 것을 알 수 있다. 이를 이용하면 본 논문에서 제안한 사용자의 정지 상태 및 이동 상황 인지가 가능하다.[1-2]

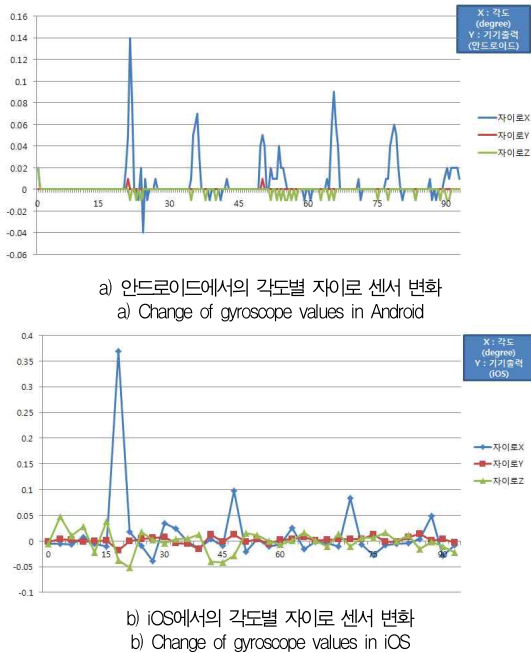


그림 4. 각도별 자이로 센서 변화(x축 90회전)  
Fig. 4. Change of gyroscope values

3. 가속도 센서를 이용한 이벤트 생성

스마트 폰에 장착된 센서 중 가속도 센서(accelerometer)는 외부 충격이나 이동에 대한 벡터 값(roll, pitch, yaw)을 x,y,z 축으로 표현한다. 초창기 스마트 모바일에 장착되었으나 벡터 값을 추출하기 때문에 절대적인 회전각 변화 크기를 추출하지 못하는 특징이 있다. 본 논문에서 대상으로 하고 있는 갤럭시 S2와 iPhone 4는 두 시스템 모두 가속도 센서를

사용하며, 이를 OS에서 처리하는 결과 값에만 차이가 있다. 그림 5는 안드로이드와 iOS에서 움직임별로 처리되는 가속도 센서의 결과 값으로, 이를 이용하면 사용자의 행동상황 정보를 인지할 수 있다. 측정값은 y축을 기준으로 90도 방향으로 이동한 벡터 값이다. 그림 5를 통해 변화하는 방향의 변위는 고정되지만 나머지 두 축의 변위 벡터를 측정하는 것이 가능함을 알 수 있다. 본 논문에서는 제안하는 사용자의 정지 상태 및 이동 상황인지를 위한 기준 값으로 자이로센서와 함께 가속도 센서가 이용된다.

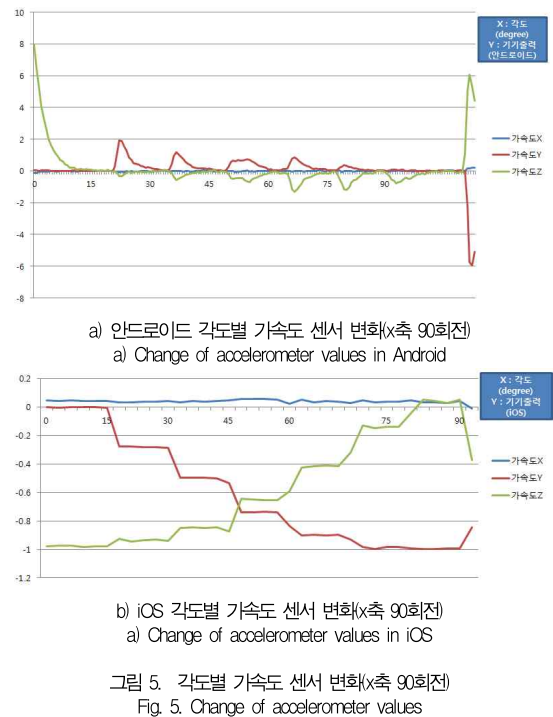


그림 5. 각도별 가속도 센서 변화(x축 90회전)  
Fig. 5. Change of accelerometer values

4. 잠음, 조도, 근접, GPS 센서를 이용한 이벤트 생성

잠음 센서로 널리 사용되고 있는 MIC로 잠음을 감지하고, 이를 주파수 값으로 변환함으로써 플랫폼의 볼륨 조절 이벤트를 생성한다. 안드로이드 기기에만 적용되고 있는 조도 센서에서는 5개의 레벨로 밝기를 계측하고, 조도 레벨을 이용하여 플랫폼의 화면밝기 조절 이벤트를 생성한다. 근접센서는 휴대폰과 인체 부분의 근접 여부를 계측하는 기능을 수행하는 것으로, 본 논문에서 설계한 플랫폼에는 이를 적용하지 않았으나, 실제 제작한 게임에서는 인터페이스로 설계하여 사용자가 빠르게 어플리케이션을 조작하게 하는 이벤트를 생성하도록 하였다. 모든 스마트폰에 장착되어 있는 GPS 센서는 플랫폼

에서 어플리케이션 종료 이벤트를 생성하는 경우에 사용된다.

## IV. 실험 및 결과

### 1. 실험 환경

상용의 스마트 폰을 대상으로 상황인지 결과를 실험하기 위해 현재 널리 사용되고 있는 iOS와 안드로이드 환경에 적합하게 시스템을 설계하였다. 실험에 사용된 OS 버전은 iOS 4.3과 안드로이드 2.3 진저브레이크이다. iOS의 경우 Apple사의 iPhone 4를(A4 1GHz / 512MB DRAM), 그리고 안드로이드의 경우에는 삼성사의 갤럭시 S2(1.2GHz Dual-core ARM / 1GB RAM)을 대상으로 하였다.

### 2. 실험 방법

스마트 폰 기기별로 센서가 다르게 적용되고 있으므로 실험방식을 차별화 하였다. 안드로이드 기기인 갤럭시 S2에서는 사용할 수 있는 센서는 조도, 근접, GPS, 가속도, 자이로, 잠음 센서이고, iOS 기기인 iPhone 4에서는 가속도, 자이로, GPS, 잠음 센서이다. 이는 모바일 플랫폼의 OS 환경에 따라 플랫폼이 제한되기 때문이다.

실제의 사용자 상황은 사용자가 스마트 폰을 휴대한다는 점에서 착안하여 일반적으로 걷고 있는 상황, 달리는 상황, 외부 소음에 따른 환경 상황, 외부 조명에 따른 환경 상황에 대해 정의하였다. 표 1에 본 실험에서 채택한 상황을 요약하여 나타냈다.

각 상황은 사용자가 스마트 폰을 휴대하고 다니는 점을 고려하여 사용자의 정지 및 이동 상태에 대한 행동 정의와, 외부환경에 따른 장소별, 밝기별, 지역별로 이를 분류하고, 각각의 상황에 따라 정보를 수집하는 센서들을 구분하였다. 각각의 센서에서 수집되는 값들을 통해 사용자의 상태를 정의하는 것이 가능하고, 이를 게임 인터페이스에 적용하는 것이 가능하였다.

### 3. 상황인지 실험 및 결과

상황인지 실험은 동일 관점에서 연구된 과거 논문과의 비교를 목표로 과거 Palm OS 기반의 SenSay 방식과 동일한 방법으로 수행하였다. SenSay 방식의 실험을 위해 센서 단말기(sensor box)를 제작하였으나, 본 실험에서는 범용적 환경을 고려하여 범용 안드로이드 기기와 iOS 기기에 적용된

센서만을 대상으로 실험을 진행하였다. 실험의 신뢰성을 높이기 위해 대상 인원을 고정하여 전체 실험을 수행하였고, 각 인원별로 출력된 데이터의 평균으로 실험값을 설정하였다[5].

표 1. 상황 정의  
Table 1. Context definition

	상황 정의	사용 센서
정지 상태	서있는 상황	자이로, 가속도 (안드로이드/iOS)
	앉아있는 상황	자이로, 가속도 (안드로이드/iOS)
이동 상태	걷는 상황	자이로, 가속도 (안드로이드/iOS)
	뛰는 상황	자이로, 가속도 (안드로이드/iOS)
	속보 상황	자이로, 가속도 (안드로이드/iOS)
외부환경 (장소별)	지하철 안	노이즈 (안드로이드/iOS)
	전화벨 소리	노이즈 (안드로이드/iOS)
	일반적 대화	노이즈 (안드로이드/iOS)
	조용한 사무실	노이즈 (안드로이드/iOS)
외부환경 (밝기별)	어두운 상황	조도 (안드로이드)
	일반적 조명	조도 (안드로이드)
	밝을 때	조도 (안드로이드)
외부환경 (지역별)	지역정보	GPS (안드로이드/iOS)

#### 3.1 정지상태 실험

사용자가 정지 상태에 있을 경우, 상황은 크게 2개로 정의할 수 있다. 그 하나는 사용자가 서있는 상황이고, 다른 하나는 앉아있는 상황이다. SenSay의 경우, 이에 대한 정보를 수집하기 위해 가속도 센서만을 이용하였다. 그러나 가속도 센서는 단순 진동 및 충격에 대한 계측만이 가능하기 때문에 정확한 상태를 파악하는 것이 어렵다. 따라서 본 논문에서는 자이로 센서를 사용하여 사용자 상황을 정확하게 수집할 수 있도록 하였다. 그림 6에 OS 별 사용자가 서있을 때와 앉아있을 때의 기기별 센서(자이로/가속도)의 계측 값을 나타냈다. 그림 6에 물리적 변화를 인지하기 위한 진폭 값을 나타냈다.

그림 6과 7에 나타낸 바와 같이 사용자가 서 있는 상태에서 서 있을 경우, 초기 자이로 센서의 값이 크게 변동하고, 가속도 센서의 변위는 크게 차이가 없다. 이는 외부 움직임의 각 변화에 의존하는 자이로센서의 특징에 의해 Z축 값이 크게 변

화하게 되고, 외부 충격 및 크기 변화에 의존하는 가속도센서의 특성에 의해 변위가 고정되기 때문이다. 사용자가 서있었다 않을 때에는 z축으로의 변화량이 급격하게 변화하기 때문에 가속도의 변화량이 적다. 자이로의 Z축이 급격하게 변할 경우에 사용자의 상황은 서 있다가 앉는 상태로 변화했다고 판단할 수 있다. 반대로 사용자가 앉아있는 상황에서 일어나는 것으로 변화할 때에는 역방향으로 Y축 변화량이 급증하기 때문에 이 값을 이용하여 서있을 경우에서 앉는 상황으로 변화하는 것을 구분할 수 있다. 실제의 수치 값을 표 2에 나타냈다.

표 2. 정지 상태에서 센서의 변화량  
Table 2. the change of sensor values in standing and sitting

상태	시간 (s)	OS	가속도 X축	가속도 Y축	가속도 Z축	가속도 진폭 ( $\sqrt{x^2+y^2+z^2}$ )	자이로 X축	자이로 Y축	자이로 Z축	가속도 진폭 ( $\sqrt{x^2+y^2+z^2}$ )	
앉은 경우	1	And	0.15	284	7.65	8.161532	0.03	0.03	0.01	0.043889	
		iOS	-0.119	-0.542	-0.771	0.949829	0.476	0.057	-0.136	0.498318	
	5	And	0.05	0.062	-1.71	1.788016	0.17	0.1	0	0.192231	
		iOS	-0.724	-0.247	0.649	1.003188	-0.003	0.011	0.012	0.016553	
	10	And	1.95	0.7	-0.95	2.28944	-0.01	-0.01	0.01	0.017321	
		iOS	-0.299	-0.038	0.971	1.019797	-0.003	-0.004	0.016	0.016763	
	20	And	...	...	...	...	...	...	...	...	
		iOS	...	...	...	...	...	...	...	...	
	일어난 경우	1	And	-0.04	0.09	0.01	0.098955	0.01	-0.01	-0.01	0.017321
			iOS	-0.681	-0.234	0.736	1.023665	-0.013	-0.008	-0.017	0.071288
		5	And	-1.6	0.04	1.08	1.903833	-0.01	0.01	-0.01	0.017321
			iOS	-0.659	-0.166	0.748	1.014391	0.043	-0.035	0.038	0.080237
10		And	-1.59	-0.6	1.89	2.616307	-0.08	-0.01	0.08	0.113578	
		iOS	-0.779	0.005	0.276	0.836164	-0.116	-1.457	-0.857	1.69433	
20		And	0.78	-0.46	0.75	1.175758	-0.13	-0.17	0.04	0.217715	
		iOS	-1.025	0.056	0.152	1.03721	2.158	0.451	-0.148	2.222247	
60		And	0.01	0.2	-0.23	0.304959	0.09	-0.15	-0.04	0.17944	
		iOS	-0.965	-0.177	0.231	1.074926	-0.027	0.046	0.008	0.053935	
속보		1	And	-0.03	-0.03	0.07	0.081854	0.13	-0.59	0.02	0.604483
			iOS	-0.968	-0.134	0.232	1.009345	-0.006	0.001	-0.005	0.008544
	5	And	-0.03	0.02	-0.07	0.076811	0.11	-0.1	-0.04	0.153948	
		iOS	-0.961	-0.146	0.225	0.997728	-0.026	-0.012	-0.01	0.030332	

3.2 이동상태 실험

사용자가 이동 상태에 있을 경우, 상황은 다양하게 변할 수 있다. 그러나 대부분의 이동의 경우 상황을 정의하기 어렵기 때문에 본 논문에서는 3개의 경우에 대해 고려하였다. 첫 번째는 사용자가 걷고 있는 상황으로 초속 약 0.3~0.5m/s(meter/ seconds)으로 이동 중일 때이고, 두 번째는 속보로 걷는 상황 즉 약 2~2.7m/s로 이동 중인 상황이고, 마지막 세 번째는 전력질주 상황으로 3.75~4m/s로 이동하는 상황이다. 또한 정지상태 실험과 동일하게 가속도 센서는 단순 진동과 충격에 대한 측정만이 가능하기 때문에 정확한 상태를 파악하기 어려운 점을 고려하여 자이로 센서를 함께 사용하였다. 그림 8과 9에 OS 별 사용자 상황 변화 시점의 기기별 출력 값과 물리적 변화를 알기 위한 진폭을 측정할 것이다.

그림 8과 9에 나타낸 바와 같이 사용자의 상황변화에 따라 두 센서의 값에는 큰 변화가 유발된다. 특히 전력 질주의 경

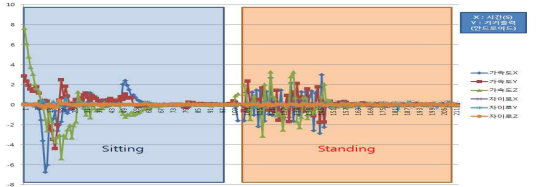
우에는 큰 변화를 보이고 있다. 전력 질주 시에는 가속도 센서의 경우 각 축별로 변위의 차이가 약 2~3 정도의 차이가 생긴다. 이는 평소 1이하의 차이를 유지하는 가속도 센서의 변위를 감안하면 매우 큰 차이를 보이는 것을 알 수 있다.

표 3. 이동 상태에서 센서의 변화량  
Table 3. the change of sensor values in the movement

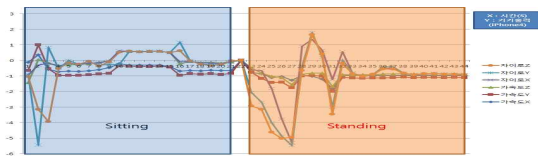
상태	시간 (s)	OS	가속도 X축	가속도 Y축	가속도 Z축	가속도 진폭 ( $\sqrt{x^2+y^2+z^2}$ )	자이로 X축	자이로 Y축	자이로 Z축	가속도 진폭 ( $\sqrt{x^2+y^2+z^2}$ )	
걷는 경우 (0.3~0.5m/s)	1	And	-1.85	-0.28	1.43	2.35495222	0	0.01	-0.01	0.0173210	
		iOS	-0.653	-0.197	0.744	1.00833344	0.833	0.637	0.574	1.1856392	
	10	And	-2.6	-5.6	3.51	7.10211940	0.01	-0.01	-0.01	0.0173210	
		iOS	-0.759	-0.05	0.024	0.76102265	-0.666	-1.768	-0.823	1.9702303	
	20	And	-2.43	-0.51	-0.98	2.69394448	0	0.02	-0.04	0.0447236	
		iOS	-0.864	-0.081	0.09	0.87244312	-1.457	-1.239	0.484	1.9734658	
	60	And	...	...	...	...	...	...	...	...	
		iOS	...	...	...	...	...	...	...	...	
	속보 경우 (2.0~2.7m/s)	1	And	1.82	-0.29	-1.64	2.32383115	-0.13	0.23	0.01	0.2693603
			iOS	-1.03	0.032	0.145	1.04421693	-0.25	2.837	-0.88	2.2017715
		70	And	-1.57	0.9	-2.5	3.08624367	-0.07	-0.08	-0.05	0.1174730
			iOS	-1.111	-0.316	0.409	1.22533995	-0.794	-2.198	-0.611	2.4153339
뛰는 경우 (3.75~4.0m/s)		1	And	-0.98	2.01	-3.55	4.19559292	0.21	0.5	-0.02	0.4086633
			iOS	-0.976	-0.141	0.222	1.01081205	-5.77	0.238	-0.514	5.7916655
		10	And	-1.31	-2.85	0.85	3.24978460	0.09	-0.59	0.05	0.5931558
			iOS	-1.518	0.236	0.237	1.56464341	0.744	-3.008	-0.95	3.3153594
		20	And	-1.67	-1.62	3.76	4.42163996	0.03	-0.01	0	0.032782
			iOS	-1.017	0.234	-0.017	1.04371164	-3.008	0.117	0.301	3.0849459
		60	And	...	...	...	...	...	...	...	...
			iOS	...	...	...	...	...	...	...	...
	뛰는 경우 (3.75~4.0m/s)	1	And	-3.2	2.33	-4.72	6.16013798	-0.05	0.02	-0.01	0.0640324
			iOS	-1.401	-0.403	0.617	1.58300315	2.438	1.335	0.855	2.9480222
		70	And	-1.24	-3.92	0.12	4.11319826	0.08	-0.12	-0.03	0.1473019
			iOS	-1.392	-0.464	0.464	1.52282333	0.005	-0.107	0.046	0.1168232
1		And	-6.38	-1.19	5.56	8.54599906	-0.34	-1.27	0.03	0.4352010	
		iOS	0.185	-0.083	1.003	1.02600341	1.885	3.31	-0.035	3.8153228	
10		And	5.03	-0.96	3.16	6.01731657	-0.1	0.28	0.07	0.3354508	
		iOS	-2.039	0.112	0.727	2.16762404	1.25	-3.84	0.345	4.0566882	
20		And	-6.77	8.09	5.2	11.6825254	-0.41	-0.14	0.05	0.4651925	
		iOS	-0.225	0.501	1.328	1.43708385	-3.765	1.589	2.047	4.5703678	
60		And	...	...	...	...	...	...	...	...	
		iOS	...	...	...	...	...	...	...	...	
70	And	-3.27	-2.27	8.86	9.71315602	0.13	-0.39	-0.02	0.4115821		
	iOS	-1.155	-0.397	1.673	2.07136742	1.019	-3.01	4.37	5.40227317		
70	And	-4.76	2.49	-9.37	4.11068337	0.24	-0.68	-0.09	0.7250489		
	iOS	-1.218	-0.016	-0.373	1.27393445	-0.617	4.216	0.526	4.2834331		

자이로 센서의 경우, 최대 18~21 정도의 차이가 발생하며 이는 일반적으로 1 미만의 차이를 보이는 자이로 센서 값을 감안하면 큰 변위 차가 발생하는 것을 알 수 있다. 그러나 걷는 상황과 속보의 차는 상황 정의가 용이하지 않다. 자이로센서의 변화량은 거의 차이가 없으나, 가속도 센서의 위상 변화의 크기 차를 기준으로 시스템을 설계하면 상황인지가 가능하다. iOS의 경우에는 센서의 물리적 변화의 절대 값을 제공하는

특징을 가지고 있어 실제 변화량에서 자이로 값이 크게 변동하는 특징을 가지고 있고, 안드로이드의 경우에는 변위에 대한 크기를 제공하기 때문에 가속도의 변화가 크다는 것을 알 수 있다. 프레임워크 설계 시 각 OS의 특징을 고려하여야 한다. 이를 반영하여 iOS는 자이로 변화를 중심으로, 그리고 안드로이드는 가속도 센서의 변화를 중심으로 설계하였다. 표 3에 다음 상황에 대한 정의를 수치로 표현한 결과를 나타냈다.

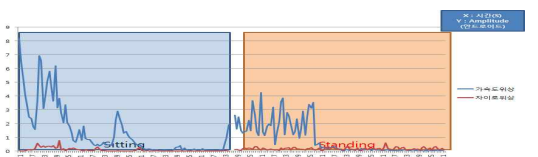


a) 정지 상황의 안드로이드 자이로/가속도 센서 변화  
a) Change of gyroscope and accelerometer values in the standing and sitting via the Android device

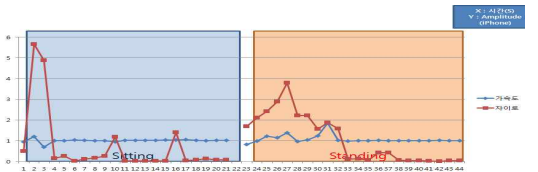


b) 정지 상황의 iOS 자이로/가속도 센서 변화  
b) Change of gyroscope and accelerometer values in the standing and sitting via the iOS device

그림 6. 정지 상황의 자이로와 속도 센서 변화  
Fig. 6. Change of gyroscope and accelerometer values in the standing and sitting via each devices

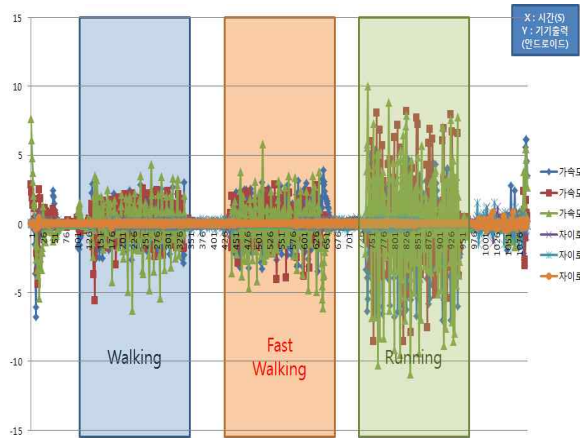


a) 정지 상황의 안드로이드 자이로/가속도 센서 진폭 변화  
a) Change of gyroscope and accelerometer amplitude in the standing and sitting via the Android device



b) 정지 상황의 iOS 자이로/가속도 센서 진폭변화  
b) Change of gyroscope and accelerometer amplitude in the standing and sitting via the iOS device

그림 7. 정지 상황의 자이로/가속도 센서 진폭 변화  
Fig. 7. Change of gyroscope and accelerometer amplitude in the standing and sitting via each devices

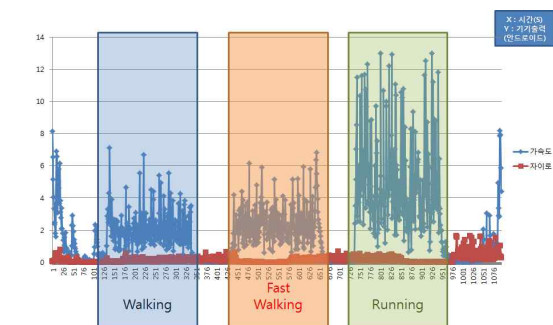


a) 이동 상황의 안드로이드 자이로/가속도 센서 변화  
a) Change of gyroscope and accelerometer values in the movement via the Android device

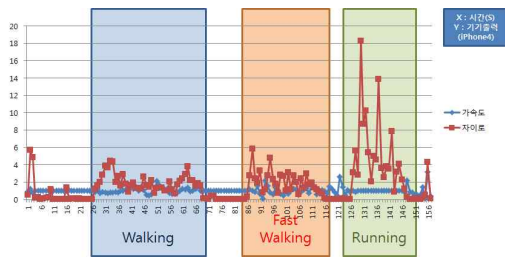


b) 이동 상황 iOS 자이로/가속도 센서 변화  
b) Change of gyroscope and accelerometer values in the movement via the iOS device

그림 8. 이동 상황 기기별 자이로 / 가속도 센서 변화  
Fig. 8. Change of gyroscope and accelerometer values in the movement via each devices

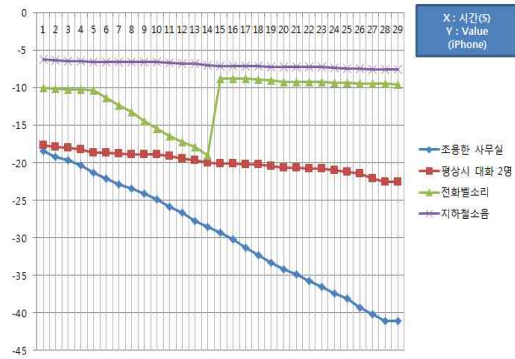


a) 이동 상황의 안드로이드 자이로/가속도 센서 진폭 변화  
a) Change of gyroscope and accelerometer amplitude in the movement via the Android device



b) 이동 상황의 iOS 자이로/가속도 센서 진폭 변화  
 b) Change of gyroscope and accelerometer amplitude in the movement via the iOS device

그림 9. 이동 상황의 자이로/가속도 센서 진폭 변화  
 Fig. 9. Change of gyroscope and accelerometer amplitude in the movement via each devices



b) 장소별 iOS 기기 소음값 측정 결과  
 b) Result of sound values via iOS mobile in each places

그림 10. 장소별 소음값 측정 결과  
 Fig. 10. Result of sound values in each places

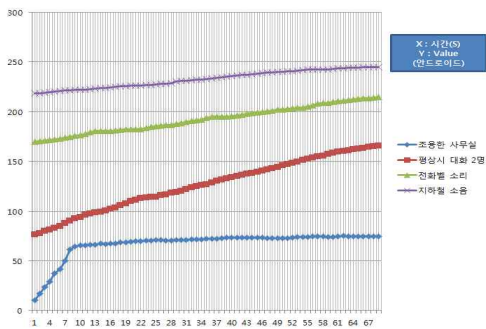
3.3 외부상황 장소변화 실험

사용자가 스마트 폰을 이용하여 주로 이동하는 공간을 중심으로 잡음 센서 값을 판단하는 형태로 실험을 진행하였다. 소음도 측정표에서 정의하는 각각의 레벨에서 실험을 진행하였다. 실험의 정확성을 위해 CEM사의 DT-85A 데시벨 측정기를 사용하여 각 상황의 데시벨 크기를 유지하고, 이를 통해 각 스마트 폰의 잡음 센서가 반응하는 값을 측정하는 형태로 실험을 진행하였다. 상황별 소음도를 표 4에 나타냈다.

표 4. 장소별 소음도  
 Table 4. Sound level of each places

장소	소음도(DB)
조용한 사무실	48~53
평상시 대화 2명	58~63
전화벨 소리	69~72
지하철 소음	80~85

장소별 소음도를 기준으로 각각의 상황별 센서의 변화값을 그림에 나타냈다. 표 5는 실제 측정된 데이터의 값이다.



a) 장소별 안드로이드 기기 소음값 측정 결과  
 a) Result of sound values via Android mobile in each places

표 5. 장소 및 시간별 측정 소음값  
 Table 4. Sound values of each places

장소	기기	1s	5s	10s	...	1min
조용한 사무실	안드로이드	10.42	41.67	70.31	...	74.73
	iOS	-18.42	-20.32	-32.25	...	-42.61
평상시 대화 2명	안드로이드	76.6	125.96	159.80	...	169.74
	iOS	-17.66	-18.16	-20.22	...	-22.53
전화벨 소리	안드로이드	169.84	177.72	183.97	...	217.61
	iOS	-10.02	-10.36	-9.00	...	-9.88
지하철 소음	안드로이드	218.22	222.33	235.84	...	246.71
	iOS	-6.25	-6.49	-7.18	...	-8.50

그림 10과 표 5에서 볼 수 있듯이 각 상황별로 데이터의 입력 값에 변화가 있음을 알 수 있다. 잡음 센서의 경우 소음도가 높은 환경일수록 값이 커지는 것을 알 수 있고, 약 1분 후부터 일정한 값으로 변화가 수렴하였다. 그 이유는 초기 소음도가 목표치에 도달하는 시간이 걸리기 때문이다. 이를 기반으로 총 20분 간 입력된 데이터를 통해 각 상황별 기준치를 추정할 수 있다. 안드로이드의 경우 74.05dB를 기준으로 조용한 사무실인 48~53dB 환경에 사용자가 있는 것으로 판단할 수 있고, 평상시 대화 수준인 58~63dB 환경의 경우에는 169.00dB가 기준 값이 되며, 전화벨 소리 및 주변 소음이 69~72dB인 환경에서는 217.00dB가 기준 값이 된다. 또한 지하철 소음이 246.00dB 이상일 경우에는 사용자가 지하철역사나 지하철 내에 있는 것으로 상황을 인지할 수 있다.

iOS에서 조용한 사무실의 경우에는 -42.05dB이고, 평상시 대화 수준의 경우에는 -21.00dB 이상이며, 전화벨 소리 및 주변 소음의 경우에는 -9.50dB 이상이다, 지하철 소음의 경우에는 -8.50dB이다. 표 6에 실험을 통해 실측한 데이터를 나타냈다.

표 6. 기기별 소음도의 임계치값  
Table 6. the threshold of sound level via each mobiles

장소	소음도(dB)	안드로이드	iOS
조용한 사무실	48~53	74.05	-42.05
평상시 대화 2명	58~63	169.00	-21.00
전화벨 소리	69~72	217.00	-9.50
지하철 소음	80~85	246.00	-8.50

3.4 외부상황 조도변화 실험

iOS 조도 센서의 경우에는 Apple의 정책상 사용이 불가하나, 안드로이드의 경우에는 센서를 통해 조도의 밝기 유무를 확인 할 수 있고, 이를 통해 사용자의 환경을 인지할 수 있다. 그러나 표현의 범주가 3가지로 국한되어 있어 사용자가 어떤 환경에 있는지의 여부를 조도 값에 따라 확인하여야 한다. 표 7에 상황을 인지하기 위한 조도 값을 나타냈다. 이 데이터는 Australian Greenhouse Office에서 발간한 Strategic Study of Household Energy and Greenhouse Issues에서 발취한 조도표를 기준으로 하였다. 표 8은 안드로이드 기기에서 측정이 가능한 조도 레벨이다 [13].

표 7. 조도 기준  
Table 7. Light level based on LUX

장소	조도(lux)
밤하늘	0.002
보름달	0.01
맑은 하늘의 어두운 황혼	3.4
거실	50
복도/화장실	80
매우 어두운 낮	100
관장 오피스 조명(Australia)	320
맑은 날의 해돋이 혹은 해넘이	400
인공조명	1000
낮	10,000~25,000
직사광선	32,000~130,000

표 8. 안드로이드 조도 기준  
Table 8. the light level in the Android mobile

레벨	조도
1	0
2	10
3	100
4	1000
5	10000

조도 기준으로 어떤 상황일 때 모바일 레벨이 변하는지를 측정하기 위해 조도계 LX-1010B를 이용하여 레벨에 따른 조도(lux) 값을 측정하였다. 그림 11에 나타난 바와 같이 조

도 값에 따라 레벨이 변화한다. 약 40 lux 이하까지는 레벨 2로 거실이나 방 조도가 어두운 환경으로 인지되고, 밤이나 0lux 이하의 경우에는 레벨 1로 인지된다. 109lux 정도의 흐린 날씨에서 낮 정도의 밝기는 레벨 3으로, 일반적인 실내 형광등부터 인공조명 정도의 밝기 즉 직사광선을 제외한 밝기는 레벨 4로 인지된다. 또한 직사광선과 밝은 날씨의 태양 밝기 정도는 레벨 5로 인지된다. 이를 근거로 표현한 각 레벨별 조도 값(lux) 과 이에 따른 사용자 상황을 표 9에 나타냈다.

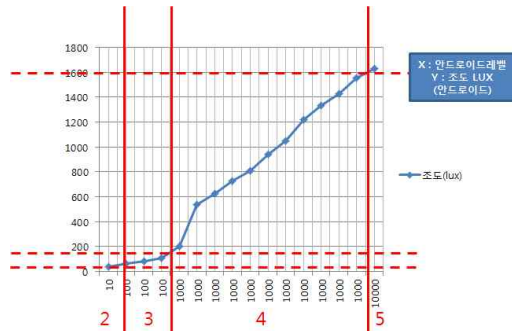


그림 11. 안드로이드 조도 레벨  
Fig. 11. Android light level via LUX

3.5 외부상황 지역변화 실험

모든 스마트 폰에는 GPS 센서가 장착되어 있으므로 LBS(Local Based Service)를 구현 할 수 있다. 본 실험에서는 스마트 폰 상황인지 인터페이스를 구현하기 위해 각 좌표를 입력하고 이에 대한 이벤트를 설정하였다. 이벤트는 도서관이나 학교와 같이 정숙이 필요한 장소에서 어플리케이션이 종료되는 형태로 제작하였고, 이를 기반으로 각 좌표를 데이터베이스에 저장하였다.

표 9. lux 별 안드로이드 조도 레벨 결과  
Table 9. the result of Android light level via LUX

레벨	값	lux	상황
1	0	0~5	어두운 밤이나 밤
2	10	5~40	빛이 새어들어오는 정도
3	100	63~110	형광등이 켜지지 않은 복도, 화장실 및 매우 흐린 날씨의 낮
4	1000	200~1558	인공조명 즉 형광등 및 조명이 존재하는 상황
5	10000	1630 ~	직사광선 및 낮

3.6 상황인지 프레임워크를 사용한 모바일 게임 제작

본 실험을 통해 도출된 결과를 이용하여 각 OS 환경에 맞는 상황인지 플랫폼을 제작하였다. 검증 방법으로 본 논문에서 설계한 플랫폼으로 환경을 제어할 수 있는 슈팅 게임 형태

의 콘텐츠를 제작하였다. 적용 환경은 소리에 따라 게임의 난이도 조절(스나이퍼의 흔들림 증가), 밝기에 따라 게임 밝기 조절, 움직임에 따른 줌인/아웃 기능을 적용하였다. 그림 12에 게임에 적용한 콘텐츠 결과를 나타냈다.



그림 12 상황인지 게임 콘텐츠(Street Sniper)  
Fig. 12. Context-awareness game contents(Street Sniper)

스마트 폰 기기별로 센서가 다르게 적용되고 있으므로 실험의 방식은 두 환경에서 차이가 있다. 안드로이드 기기인 갤럭시 S2에서는 사용할 수 있는 센서는 조도, 근접, GPS, 가속도, 자이로, 잡음 센서인 것에 반해, iOS 기기인 iPhone 4에서는 가속도, 자이로, GPS, 잡음 센서이다. 이는 모바일 플랫폼의 OS 환경에 따라 제한되기 때문이다.

실제 사용자의 상황은 사용자가 스마트 폰을 휴대한다는 점에서 착안하여 일반적으로 걷고 있는 상황, 달리는 상황, 외부 소음에 따른 환경 상황, 외부 조명에 따른 환경에 대해 설정하였다. 표1은 실험에서 설정한 상황에 대한 요약이다.

사용자의 정지 및 이동 상태에 대한 행동을 설정한 후 외부 환경에 따른 장소/밝기/지역 별로 이를 분류하였고, 각 상황에 따라 정보를 계속하는 센서를 구분하였다. 계속 값을 통해 사용자의 상태를 정의할 수 있었으며, 이를 이용하여 게임 인터페이스에 적용함으로써 그 유효성을 확인하였다.

## V. 결 론

본 논문에서는 상용 모바일에 장착된 센서들을 이용하는 사용자 상황인지 플랫폼을 설계하였다. 상황인지 플랫폼을 설계하기 위해서는 각 센서들로부터 사용자의 상황 정보를 습득하는 정보의 수집 단계, 이를 데이터화하여 저장하는 저장단계, 저장된 데이터를 이벤트로 표현하는 재표현 단계, 재표현이 끝나면 다시 정보를 수집하기 위해 이벤트를 소멸하는 소멸단계로 구성된다. 이러한 각각의 단계에서 가속도, 자이로, 잡음, 조도, GPS 등 상용 스마트 폰 모바일 환경의 센서를 활용하여 사용자의 상황을 인지하였다.

가속도 자이로 센서를 이용하여 사용자의 앉고 서있는 정지 상황을 인지하였고, 사용자가 뛰고, 걷고, 빠른 속도로 걷는 상황 또한 인지하였다. 또한 잡음 센서를 이용하여 상황별 사용자의 외부환경과 대화 여부를 인지하였다. 외부 환경 중 조도에 따른 상황을 인지하기 위해 외부 상황의 밝기를 판별하였다. 그리고 GPS를 사용하여 각 위치 정보에 맞는 이벤트를 발생하였고, 범용적인 플랫폼 개발을 위해 현재 스마트폰 시장을 양분하고 있는 안드로이드와 iOS의 환경에 맞는 플랫폼을 설계하였다. 플랫폼의 유효성을 확인하기 위해 슈팅 게임 콘텐츠를 제작하여 플랫폼으로 제어하였다. 결과적으로 사용자에게 더욱 직관적이고 손쉬운 인터페이스를 제공할 수 있다는 것을 확인하였다.

상용의 플랫폼을 설계하기 위해서는 다양한 범용 단말기를 대상으로 실험이 진행되어야 하며, 각 상황에 적절한 콘텐츠 생성에 관한 연구가 향후의 연구방향이다.

## Acknowledgment

본 연구는 전라남도 및 (재)전남문화산업진흥원의 2011년도 문화산업육성지원사업의 연구결과로 수행되었음.

This research is supported by JenllaNamdo and Jeonnam Culture Industry Promotion Agency(JCIA) in the Culture Industry Promotion & Support Program 2011.

## 참 고 문 헌

- [1] <http://developer.apple.com/>
- [2] <http://developer.android.com/index.html>
- [3] J.M. Kang, T. G. Ko, S. S. Seo, B. J. Sung, John Strassner, J. Kim, C. I. Park, W. K. Hong, "Smart mobile platform for context-awareness service," KIISE, 2010.05, pp57~67.
- [4] A.K. Dey and G.D. Abowd, "Towards an Understanding of Context and Context-Awareness," Aug, 1999.
- [5] Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Neema Moraveji, Kathryn Reiger, Jeremy Shaffer, "SenSay:A Context-Aware Mobile Phone", Wearable

Computers, 2003. Proceedings. Seventh IEEE International Symposium on, 1530-0811, Oct. 2005, pp. 248-249.

[6] Robert Schmohl, Uwe Baumgarten, "Mobile services based on client-server or P2P architectures facing issues of context-awareness and heterogeneous environments", Wireless and Mobile Communications, 2008. ICWMC '08. The Fourth International Conference on, Vol 10, Aug 2008, pp118-124.

[7] S. Y. Lim, J. D. Heo, "Technology Trends of Context Aware Computing Applications", ETRI Electronics and Telecommunications trends, Vol 19., NO 5., Oct. 2004, pp31~40.

[8] D. Salber, A.K. Dey, and G.D. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," in Proceedings of CHI'99, May, 1999, pp.434-441.

[9] D. Salber, A.K. Dey, and G.D. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," in Proceedings of CHI'99, May, 1999, pp.434-441.

[10] M. Lamming and M. Flynn, "Forget-Me-Not: Intimate Computing in Support of Human Memory," in Proceedings of FRIEND21: International Symposium on Next Generation Human Interfaces, Feb, 1994, pp.125-128.

[11] Y. Nishibe et al., "Mobile Digital Assistants for Community Support," AAAI Magazine, Vol.19, o.2, Summer, May, 1998, pp.31-49.

[12] Y. Nishibe et al., "Mobile Digital Assistants for Community Support," AAAI Magazine, Vol.19, o.2, Summer, May, 1998, pp.31-49.

[13] Strategic Study of Household Energy and Greenhouse Issue, Australian Greenhouse Office, June. 1998.

[14] Dong-Wook Jang, Surgwon Sohn, Kwang-Rok Han, Bok-Keun Sun, "A Design of Transducer Interface Protocol for Context-aware Middleware", The Korea Society of Computer and Information, Vol 16, No9, Sep, 2011, pp45~55

[15] Jong-Hyun Park, Ji-Hoon Kang, "Personalized

Recommendation based on Context-Aware for Resource Sharing in Ubiquitous Environments", Vol 16, No9, Sep, 2011, pp19~26

저 자 소 개



김기백

현재 : (주) 유토 연구원

관심분야: 3D 재구성, 증강현실, 콘텐츠, 멀티미디어

Email : k2b@utoweb.co.kr



장원석

현재 : (주) 유토 이사

관심분야: 콘텐츠, 멀티미디어, 증강현실

Email : stone@utoweb.co.kr



최재완

1988 단국대학교 전자공학과 공학사

2004 남서울대학교 디지털정보공학과 공학석사

현재 : (주) 유토 대표이사

관심분야: 멀티미디어 콘텐츠, 증강현실

Email : ceo@utoweb.co.kr



임철수

1985 서울대학교 컴퓨터공학과 공학사

1988 Indiana University 전산공학과 공학석사

1995 서강대학교 자계산학과 공학박사

1988 ~ 1994 (주)아시아나항공 시스템 센터

1994 ~ 1997: (주)SK 텔레콤 차장

현재 : 서경대학교 컴퓨터공학과 교수

관심분야: 콘텐츠, 멀티미디어, 증강현실

Email : cslim@skuniv.ac.kr



최종호

1982 중앙대학교 전자공학과 공학사

1984 중앙대학교 전자공학과 공학석사

1987 중앙대학교 전자공학과 공학박사

현재 : 강남대학교 전자공학과 교수

관심분야: 영상정보통신, 제스처인식

Email : jhchoi@kangnam.ac.kr

