모바일센서 데이터 수집 응용과 R을 사용한 분류

임재걸*, 우진석*
*동국대학교 경주캠퍼스 전자계산학과
e-mail:{yim, woo}@dongguk.ac.kr

Sensor Data Collection Mobile Application and Classification Using R

Jaegeol Yim*, Jin-Seok Woo*
*Dept of Computer Engineering, Dongguk University

요 약

본 논문에서는 스마트폰에 장착된 여러 가지 센서들의 센서 값을 수집하는 안드로이드 애플리케이션을 소개하고, 실제로 수집한 데이터를 분석한다. 분석을 위하여 일반적인 통계 분석 도구인 R을 사용한다. 분석 결과 어떤 센서 (Azimuth를 비롯한 orientation 센서) 값은 실용성이 있는 수준으로 정확하다는 것을 알 수 있었으나 또 다른 센서는 (밝기 센서 등) 잡음이 매우 심함을 보인다.

1. 서론

최근 스마트폰을 사용하는 사용자들이 급속히 증가하고 있다[1]. 이들 스마트폰에는 기존의 핸드폰에서는 찾아볼 수 없는 다양한 센서들이 장착되었다. 또한 센서 값을 이용한 다양한 응용이 개발되고 있다. 따라서 본 논문은 센서 데이터를 수집하는 애플리케이션을 소개한다.

센서 데이터가 실용적으로 사용되려면 센서가 정확해야 한다. 본 논문은 센서 수집 응용으로 수집한 데이터를 분석하여 어떤 센서가 실용 수준으로 정확한지 알아본다. 이를 위하여 수집한 데이터를 정보통계학에서 사용되는 프로그램인 R 프로그래밍 언어를 사용하여 분류한다.

2. 관련연구

본 연구는 센서 데이터를 수집하는 안드로이드 애플리케이션을 구현하고, R 프로그래밍 언어를 이용하여 수집된 데이터를 분석함으로 본 절에서는 안드로이드 프로그래밍과 R을 소개한다.

2.1 안드로이드.

안드로이드(Android)는 휴대 전화를 비롯한 휴대용 장치를 위한 운영 체제와 미들웨어, 사용자 인터페이스 그리고 표준 응용 프로그램(웹 브라우저, 이메일 클라이언트, 단문 메시지 서비스(SMS), 멀티미디어 메시지 서비스(MMS)등)을 포함하고 있는 소프트웨어 스택이자 모바일 운영 체제이다. 구글은 안드로이드 응용 프로그램을 사고 팔수 있는 구글 안드로이드 마켓을 제공하고 있다[2]. 안드로이드를 탑재한 스마트폰에서는 안드로이드에서 지원하는 선세를 사용할 수 있는데, 이를 이용하여 음주운전감지[3], 실시간 포트홀 감지[4], 스마트폰을 이용한 이동성과 성 조사[5]등의 애플리케이션들이 개발되고 있다.

안드로이드 내의 센서를 사용하기 위해서는 센서를 사용가능하게 하는 라이브러리를 사용해야 한다. 이때 사용되는 안드로이드OS의 라이브러리는 <표 1>과 같다. <표 1>에서 보이는 라이브러리들 중 android.hardware.Sensor는 센서를 나타내는 클래스로써 사용가능한 센서의 목록을 getSensorList(int)로 얻을 수 있다. 나머지 클래스에 대한 설명은 생략한다[6].

<표 1> 안드로이드에서 사용되는 센서 라이브러리

라이브러리 이름
android.hardware.Sensor
android.hardware.SensorEvent
android.hardware.SensorEventListener
android.hardware.SensorManager
android.location.Location
android.location.LocationListener
android.location.LocationManager
android.net.wifi.ScanResult
android.net.wifi.WifiManager

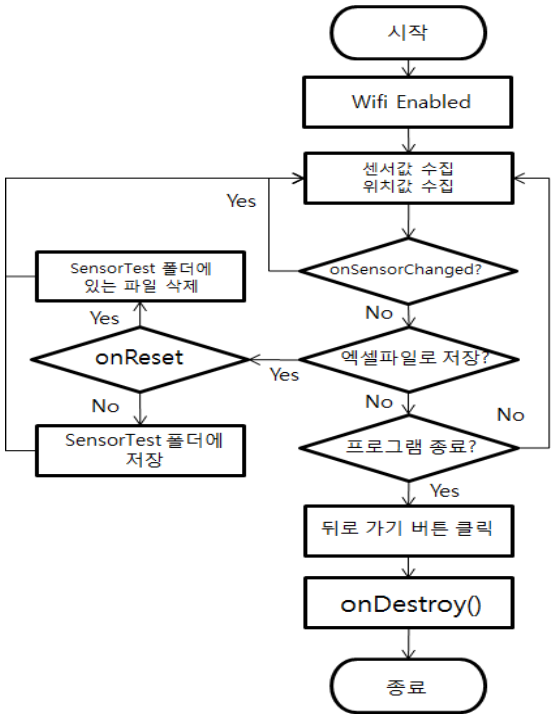
2.2 R 프로그래밍 언어

R 프로그래밍 언어(줄여서 R)는 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어 환경이다. 뉴질랜드 오클랜드 대학의 로스 이하카와 로버트 젠틀맨에 의해 시작되어 현재는 R 코어 팀이 개발하고 있다. R은 GPL 하에 배포되는 S 프로그래밍 언어의 구현으로 GNU S라고도 한다. R은 통계 소프트웨어 개발과 자료 분석에 널리 사용되고 있으며, 패키지 개발이 용이하여 통계학자들 사이에서 통계 소프트웨어 개발에 많이 쓰이고 있다[6]. 이

를 이용하여 고속도로에서의 더블 클러스터 헤드 라우팅 구성의 성능[7], R*트리를 이용한 Microarrays에 대한 유사도 및 클러스터 분석 알고리즘[8] 등의 분야에 사용되고 있다.

3. 센서 데이터 수집 애플리케이션 설계

안드로이드OS가 탑재된 스마트폰에서 센서를 사용하여 현재 위치에서 측정되는 센서들의 값을 각각 화면에 출력하고 출력된 데이터를 저장할 수 있도록 애플리케이션을 구현한다. 또한 리셋 버튼을 생성하여 버튼 클릭 시 저장된 파일을 제거 할 수 있도록 한다.



(그림 1) 센서 데이터 수집 애플리케이션의 설계

이와 같은 응용프로그램의 처리과정은 (그림 1)에 보이는 바와 같다. 우선 Wifi를 Enable시키는데, 그 이유는 일반적으로 Wifi를 끄고 3G또는 4G를 사용하고 있기 때문에 Wifi를 켜는 기능을 우선적으로 수행한다. Wifi를 켠 뒤 센서값 및 위치값 수집을 수행한다. 이때 현재 사용자의 상태에 따라 변화되는 수신된 센서 값들은 변화된 센서 값을 인지하는 onSensorChanged의 기능을 사용하여 사용자의 화면에 변화된 센서 값들을 보여준다. 현재 사용자의 화면에 출력되고 있는 센서값들을 저장 버튼을 클릭하면 엑셀파일로 저장한다. 이를 위하여 public void onSave(View v)클래스를 생성한다. 저장되는 엑셀의 파일 형식은 CSV(Comma Separated Values)로 저장한다. onSave클래스에서는 수신된 결과 값 중 GPS위치 위도, GPS위치 경도 등 총 25개의 데이터들을 저장한다. public void onReset (View v) 클래스를 생성하여 현재 저장된 엑셀파일을 삭제한다. 엑셀파일로 저장하지 않고 프로그램을 종료할 경우 사용자의 스마트폰에 있는 뒤로가기 버튼을 클릭하면 애플리케이션을 종료시킬 수 있다. 애플리케이션을 종료할 시 onDestroy()를 생성하여 애플리케이션 실행시에 실행하였던 모든 센서들을 Destroy함으로써 사용을 중단한다.

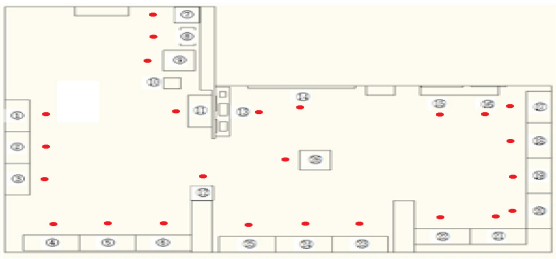
4. 구현

Wifi를 On시켜 Wifi를 자동으로 사용가능 하게 한다. 그러기 위해서 mWifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE)와 mWifiManager.setWifiEnabled(true)를 사용한다. WifiManager는 NETWORK_PROVIDER와 GPS_PROVIDER를 생성하여 각각 네트워크와 GPS로부터 위치정보를 업데이트하여 결과 값을 수신한다. public void onLocationChanged(Location location)를 사용하여 네트워크와 GPS의 값이 바뀔 때마다 새롭게 호출하여 새롭게 수신된 값을 화면에 출력한다. WIFI_SERVIDCE를 사용하여 wifi ap를 검색하하고, public void apSortingWifiSignalLevel()를 사용하여 ap를 신호 세기 순으로 정렬한다. Wifi작동 후에는 SensorEventListener와 LocationListener가 실행되어 SensorManager와 WifiManager의 센서들이 실행된다. 이를 위하여 public class AllSensorTestActivity extends Activity implements SensorEventListener, LocationListener를 사용하여 SensorManager의 Sensor Accelerometer(가속센서), Sensor Gyroscope(자이로 센서), Sensor Light(밝기 센서), Sensor Magnetic_field(자기 센서), Sensor Orientation(방향 센서), Sensor Pressure(압력 센서), Sensor Proximity(근접 센서), Sensor Temperature(온도 센서), Sensor Rotation_vector(회전 센서), Sensor Gravity(중력센서), Sensor Linear_Accelerometer(선형 가속센서) 센서를 선언한다. 또한 선언된 각 센서들의 Listener들을 생성하여 각각의 센서들에게서 결과 값을 수신하여 화면에 출력한다.

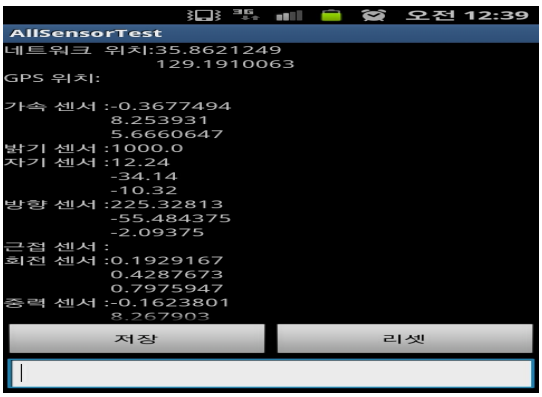
5. 실험

다음의 (그림 2)에서 보이는 박물관에서 센서 데이터 수집 애플리케이션을 실험 하였다. 지도에서 보이는 붉은 점은 각 전시대의 전방 1m지점으로써 실험위치를 의미한다. 27개의 전시대에 대하여 실험하였다.

(그림 3)은 센서데이터 수집 애플리케이션의 실행화면이다. (그림 3)에서 GPS 위치의 데이터가 수집되지 않는 이유는 애플리케이션의 실험위치가 건물 내에 있는 박물관에서 실험하였기 때문이다.



(그림 2) 박물관 도면 및 실험위치



(그림 3) 안드로이드 구현 결과 화면

6. 분석

센서의 정확도를 분석하기 위하여 3절에서 구현한 센서 값 수집 애플리케이션으로 실제 센서 값을 수집하고 분석한 사례를 소개한다.

6.1 데이터 수집

본 논문에서 사용한 실험 장소는 캠퍼스의 도서관내의 박물관을 이용하였다. 박물관을 건물내부에 있으며 총 27개의 전시대가 존재한다. 3절에서 구현한 센서 데이터 수집 애플리케이션을 사용하여 (그림 3)에서 보이는 박물관 내의 위치에서 센서 데이터를 각각의 전시대에서 수집하였다. 실험위치는 각각의 전시대에서 1m 떨어진 지점에서 전시대를 바라보고 수행하였다. 이때, 박물관은 건물 내에 있으므로 GPS 센서의 데이터는 수집되지 않는다.

6.2 분석 과정

R 프로그래밍 언어에서 지원하는 Data Mining 중 본 논문에서는 Clustering을 사용한다[10]. 또한 여러 Clustering방법 중 K-Means를 사용한다. R 프로그래밍 언어를 사용하기위해서는 R프로그램을 필요로 한다. R프로그램에는 Package를 설치하여 Package내의 함수를 사용하는 방식을 사용하고 있다. R 프로그램을 설치하면 핵심적인 패키지는 R과 함께 설치되며, CRAN(The Comprehensive R Archive Network)을 통해 700개 이상의 Package를 내려 받을 수 있다. Clustering의 K-Means는 기본적으로 제공되는 Package이다. K-Means은 1x1이상의 행렬을 요구한다. 이때 가로는 센서들의 이름으로, 세로는 실험한위치의 데이터를 사용한다. 아래의 <표 2>는 K-Means를 위하여 행렬을 제작하는 명령어이다. 이때 사용되는 데이터는 Azimuth를 사용한다. 첫 번째 줄은 cells변수에SensorData.csv내의 데이터를 입력하는 명령어이다. 이때, sep=“,”의 의미는 ,를 이용하여 각각의 데이터 간에 “,”가 나타나면 다음 데이터로 인식하여 각각의 데이터를 나누는 의미이다. 두 번째 줄을 rnames에 DataName.csv내의 데이터를 입력하는 명령어이다. 세 번째 줄은 cnames에 SensorName.csv내의 데이터를 입력하는 명령어이다. 네 번째 줄은 x에 행렬을 만들어 입력하는 명령어이다. 이때, cells의 데이터들을 row의수를 1로, column의 수를 31개로 하여 만들겠다는 의미이며, dimnames는 앞에서 정의한 row와 column의 수가 맞을 경우 배열 또는 행렬을 생성하는 명령어이다. byrow는 row로 입력되는 데이터들을 변환하는 것을 의미한다. 이러한 실험 시 SensorData.csv, DataName.csv, SensorName.csv들은 원하는 센서와 그 데이터들로만 새로 생성하여 사용가능하다. 위의 데이터들을 바탕으로 표2와 같이 R 프로그램 내에서 프로그램을 작성하면 result라는 행렬이 생성된다.

<표 2> K-Means를 위하여 행렬을 제작하는 명령어

```
database <- scan(“SensorData.csv”, sep=“,”)
rnames <- scan(”DataName.csv”, sep=“,”)
cnames <- scan(”SensorName.csv”, sep=“,”)
result <- matrix(cells, nrow=1, ncol=31, byrow=TRUE,
dimnames=list(rnames, cnames))
```

<표 2>에서 입력된 database, rnames, cnames, result의 값을 확인하기 위하여 아래의 <표 3>과 같이 R프로그램에서 작성하면 값을 확인 할 수 있다.

<표 3> 입력된 데이터의 출력

```
print(database)
print(rnames)
print(cnames)
print(result)
```

생성된 행렬 result를 사용하여 아래의 <표 4>와 같이 입력하면, x를 10개의 그룹으로 100번 kmeans하여 그 결과를 km에 입력하게 된다. 이후 print(km)명령을 입력하면 km의 결과를 볼 수 있으며 결과는 (그림 4)와 같다. 또한 kmeans의 결과를 그래프로 보고 싶다면, plot(result, col = km$cluster)명령과 points(km$center, col = 1:2,

pch = 8)을 이용하면 결과를 그래프로 확인 할 수 있다. 이때 col =1:2의 pch = 8 의미는 clustering한 결과로 나뉜 그룹들의 평균 값들에 여덟 개의 점을 가지는 포인트를 그리는 명령어이다. 이 결과는 (그림 4)와 같다.

<표 4> kmeans를 수행하는 명령어

```
km <- kmeans(x, 10, 100)
```

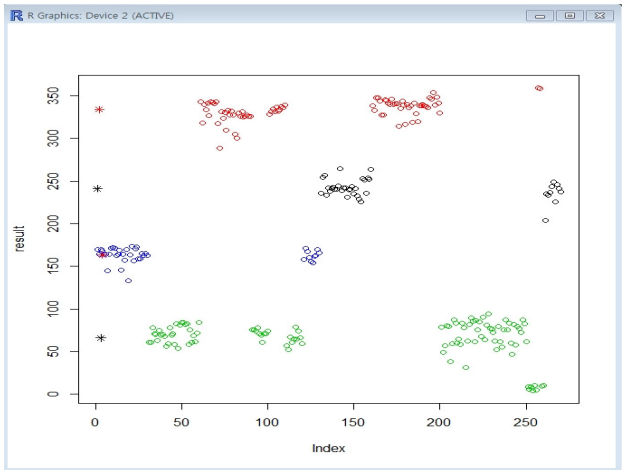


그림 4 kmeans의 결과를 그래프로 나타낸 화면

위 실험에서는 4개의 그룹으로 나누어지는 실험결과를 얻었기 때문에 다음 실험에서는 나누어진 4개의 그룹중 하나의 그룹을 지정하여 해당하는 그룹의 데이터들을 사용하여 실험 하도록 한다.

두 번째 실험으로는 4개의 그룹 중 South로 분류된 1번, 2번 , 3번, 13번 전시대를 대상으로 pitch데이터를 이용하여 clustering한다. 실험 시 대상이 되는 전시대가 4개이므로 clustering그룹의 수를 4개에서부터 시작하도록 한다. 3개의 그룹으로 실험한 결과에서 13번 전시대에서는 2번과 3번의 그룹의 결과만을 가지고 1번 그룹의 결과를 가지지 않으므로 13번 전시대는 pitch데이터로 분류가 가능하다.

세 번째 실험으로는 두 번째 실험에 분류가 되지 않은 1번, 2번, 3번 전시대의 센서 데이터 중 Brightness를 이용하여 clustering한다. Brightness센서의 데이터는 동일한 위치에서 센서 데이터를 수집했음에도 불구하고, 10 또는 100의 데이터수치가 균일하게 수집되지 않아 clustering결과 분류하지 못하는 것을 보였다.

7. 결론 및 향후 과제

본 논문은 안드로이드를 이용하여 스마트폰에서 사용가능한 데이터수집 애플리케이션을 구현하였고, 구현한 애플리케이션을 박물관내에서 실험하여 구현한 애플리케이션이 올바르게 작동함을 보였다. 또한 수집된 데이터를 R 프로그램을 이용하여 clustering하였으며, 데이터 중 Azimuth를 이용하여 clustering한 결과 4개의 그룹으로 나뉘었으며, 이는 4방위로 나누고자한 개발 초기의 목적에 부합하였다. 이중 South로 다시 정의한 그룹에서 pitch로 clustering한 결과 13번 전시대를 분류할 수 있음을 보였다. 또한 Brightness센서의 데이터를 사용하여 clustering하였으나 센서 데이터의 값의 잡음이 심하여 올바른 clustering의 결과를 얻을 수 없었다. 이와 같은 방법으로 수집된 데이터를 이용하여 clustering하면 사용자가 원하는 수준의 분류를 할 수 있음을 보였다.

참고문헌

[1] http://www.mbap.co.kr/Report
[2] http://ko.wikipedia.org/wiki/안드로이드_(운영_체제)
[3] Jiangpeng Dai, Jin Teng, Xiaole Bai, Zhaohui Shen and Dong Xuan “Mobile Phone Based Drunk Driving Detection” Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS
[4] Artis Mednisy, Girts Strazdinsy, Reinholds Zviedrisy, Georgijs Kanonirs, Leo Selavoy “Real Time Pothole Detection using Android Smartphones with Accelerometers” Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on
[5] Carlo Tacconi, Sabato Mellone, Lorenzo Chiari “ Smartphone-Based Applications for Investigating Falls and Mobility” 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops
[6] http://developer.android.com/
[7] http://ko.wikipedia.org/wiki/R_
[8] Bilal R. Qazi, Adnan Muhtar, Wanod Kumar and Jaafar M. H. Elmirghani “Performance of a Double Cluster Head Routing Scheme in a Motorway Environment” IEEE Globecom 2010 Workshop on Broadbad Wireless Access
[9] Jiaxiong Pi, Yong Shi and Zhengxin Chen “Similarity and cluster analysis algorithms for Microarrays using R* trees” Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference Workshops (CSBW'05)
[10] Feng Qian, Guang-min Hu, and Xing-miao Yao “Unicast Network Loss Tomography using #R-cast Probes Scheme” 2009 International Conference on Communications and Mobile Computing
[11]http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R
[12] http://terms.naver.com/entry.nhn?docId=858059