

이동경로 유사도 분석을 통한 매칭 서비스

:최종보고서

컴퓨터공학과/ 2017103950/ 고병후

요 약

현재 스마트폰은 대중들의 일상 대부분을 함께하는 도구가 되었다. 이에 따라 mobile device들은 사용자의 일상 생활을 가리키는 지표가 될 수 있는 데이터들을 다수 저장한다. 본 연구에서는 모바일 디바이스를 이용하여 이러한 데이터를 수집하고 정제하고 가공하여 유의미한 information을 만들어 사용자에게 서비스의 형태로 제공되는 일련의 과정을 소개한다. 이를 바탕으로 스마트폰을 활용해 gps, wifi연결시간, 선형가속벡터 값의 data를 수집하여 일상생활속 이동패턴을 파악하고 이를 통해 유사한 패턴을 보이는 user를 상호 매칭하는 서비스를 기획하고 설계한다.

1. 서론

1.1. 연구배경

현대사회에서 대부분의 사람들에게 스마트폰은 필수적인 도구가 되었다. 대중들은 mobile device와 일상에서 끊임없이 상호작용한다. 이 결과 스마트 폰은 user의 일상 생활과 관련된 많은 data를 생산하고 저장하게 되었다. 해당 data들을 수집하고 가공하여 사용자의 일상의 패턴을 파악하는 유용한 information을 추출할 수 있다면 이를 여러 서비스의 형태로 확장함으로써 사용자 편의를 제공할 수 있다.

본 프로젝트에서는 특히 LBS(Location Based Service)에 초점을 맞추어 연구를 진행한다. 대부분의 사람들은 출퇴근, 등하교 등 일상 속에서 다소 반복적인 이동패턴을 보인다. 이런 이동패턴을 나타낼 수 있는 data(gps, wifi연결시간, 선형가속벡터)를 mobile device를 이용해 수집하고 가공한다. 이를 통해 중앙서버에서 사용자 이동패턴을 분석하여 유사한 이동패턴을 보이는 user를 상호 매칭하는 service를 설계한다. 이를 통해 data를 수집하여 정제하고 가공하여 유의미한 정보를 만들어 사용자에게 서비스의 형태로 제공되는 일련의 과정을 소개하고 제시한다.

기존에 사용자 위치기반 서비스는 실시간으로 방향과 위치좌표를 파악하여 여러 서비스를 제공하는 형식을 취하고는 했다. 하지만 사용자의 위치 및 이동의 특성을 나타내는 데이터는 위치와 방향 말고도 다양하게 존재한다. 예를 들어 와이파이의 연결 여부는 사용자가 해당 위치에 정지하여 시간을 보냈을 가능성을 시사하고 가속도 벡터의 상하방향 반복적인 운동은 사용자가 이동

시에 도보를 이용했는지, 지하철과 버스 등 대중교통을 이용했는지를 알려준다.

본 프로젝트는 이러한 위치, 방향, 가속도 벡터, wi-fi연결 시간 등의 여러 데이터를 지속적으로 수집하여 이를 가공하고 분석하여 유의미한 정보인 '개인의 이동패턴'을 도출하는 것을 목적으로 한다. 다수의 사람들은 일상속에서 출퇴근, 등하교와 같은 반복적이고 일반화할 수 있는 이동 패턴을 갖기에 위에서 언급한 데이터들을 통해 개인별로 대표적인 이동패턴을 도출할 수 있다. 또한 각자의 이동패턴을 도출한다면 이것이 유사한 유저들을 상호 추천하여 동행이 가능한 서로를 인지할 수 있다. 이러한 이동패턴이 유사한 유저들간 연결은 car sharing, 택시비 절감 등의 효과로 이어질 수 있다.

1.2. 연구목표

스마트폰을 이용해 gps, 방향, 흔들림(가속도벡터), wi-fi연결여부 등의 데이터를 지속적으로 수집하고 저장한다. 이를 중앙에서 처리하여 사용자별 대표 이동패턴을 도출하고 해당 패턴이 높은 유사도를 보이는 유저를 상호 추천하는 서비스를 구현한다.

스마트폰이 일상에서 유저의 여러 행동 및 상황을 반영한 데이터를 보유하기에 이를 수집, 가공, 분석하여 사용자의 이동패턴을 파악한다. 이동패턴이 유사한 유저를 매칭하여 동행이 가능하게 해 이동에 있어 car sharing, 택시비절감 등 상호 도움이 되도록 함을 목표로 한다. 구체적인 목표는 아래와 같다.

첫번째, Mobile Device의 back ground service로 필요한 여러 데이터들을 지속적으로 수집하여 중앙 server에 저장한다. 이를 가공, 정제하고 처리하여 유의미한 정보인 이동패턴을 도출한다. 이 패턴을 바탕으로 이동경로, 이동시각, 이동수단의 유사도를 통해 비슷한 특성을 가진 유저를 상호 추천한다.

두번째, 사용자에게 서비스에서 파악한 이동경로 패턴을 GUI를 통해 게시한다.

세번째, 서버단에서 사용자의 실제 이동경로들과 이를 일반화한 대표이동경로를 지도위에서 point와 이를 연결한 line형식으로 출력이 가능하게 한다. 또한 사용자별 유사도 도출 근거를 수치화해서 그래프로 표기 가능하게 한다.

2. 관련연구

2.1. Android & Android studio

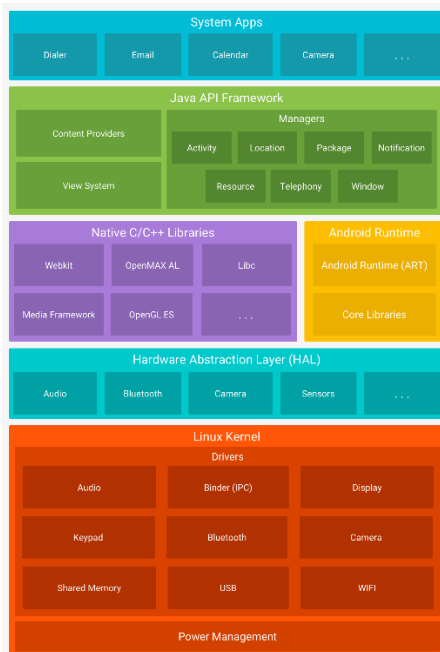


그림 1. Android software stack

해당 서비스를 구현하기 위해 사용자 데이터를 수집할 Mobile Device기기가 필요한데 본 프로젝트에서는 이를 Android 운영체제를 사용중인 스마트 폰으로 한정한다.

Android란 리눅스 커널을 이용한 운영체제이다. 주로 Mobile Device를 대상으로 한다. VM(virtual machine)인 Android RunTime을 통해 application을 별도의 process에서 수행하도록 한다. Android는 현재 다수의 스마트폰에서 사용되고 있다. 운영체제의 아키텍처는 그림1과 같고 각 layer의 역할은 아래와 같다.

1) Linux kernel

Android platform의 기반이다. 스레딩 및 하위 메모리 관리 등 기본기능에 사용된다. Hardware 추상화, 메모리 관리, 드라이버 관리 등을 수행한다.

2) Hardware Abstraction Layer(HAL)

Hardware추상화 계층으로 각 Hardware의 표준 인터페이스를 정의하여 하위 드라이버 구현을 고려하지 않아도 되게 한다.

3) Native Libraries & Android Runtime

저급 수준의 일반적인 library가 위치한다. 컴퓨팅 파워가 약한 Mobile Device특성상 native코드로 구현되어 컴파일 되어있다.

Runtime는 VM으로 컴파일된 application을 os에 맞추어진 형태로 바꾼다.

4) Java API Framework

Android Runtime을 추상화한 계층이다. 개발된 application이 하위 계층과 소통할 수 있도록 지원한다.

5) System Apps (Application)

실제로 개발된 application이다.

이러한 Android os위에서의 클라이언트 앱 구현을 위해 Android Studio를 사용한다.[1]

Android Studio란 Android application개발을 위한 공식IDE(Integrated Development Environment)이다. IntelliJ IDEA를 이용한 코드 편집기를 사용하고 있다. 프로젝트 작성시에 빌드파일은 Gradle Scripts에 표기되며 각 모듈에는 manifest, java, res가 포함된다. 빌드를 위해 Gradle 빌드 시스템을 사용한다. 또한 여러 SDK(Software Development Kit)와 편리한 API를 지원하여 개발이 용이하고 쉽다. 또한 시뮬레이터를 이용해 가상의 스마트폰에서 실험이 가능하다. 이러한 이유로 본 프로젝트에서는 클라이언트 앱 개발을 위해 이를 사용한다.[2]

2.2. GPS

GPS(Global Positioning System)는 위성항법시스템이다. 위성항법시스템은 위성을 통해 위치(Positioning) 및 항법(Navigation), 시각(Timing) 정보(PNT)를 제공해주는 시스템이다.

GPS는 미 국방성 (US Department of Defense) 에서 개발하여 운용중인 인공위성으로부터 수신한 Radio 신호를 이용하여 정지 및 운동하고 있는 물체의 위치를 결정하는 시스템이다. gps에 사용되는 인공위성은 연속적으로 신호를 송신하며 항상 4개의 인공위성이 어디에서도 보일 수 있도록 디자인되어 있다. 즉, gps란 위성에서 발사한 전파를 수신하여 관측점까지 소요 시간을 계산함으로써 관측점까지의 위치를 구하는 time navigation system이다.

위성항법시스템은 군사적 목적은 물론 스마트폰에 탑재될 정도로 작은 수신기만 있으면 사용자의 지리적 위치와 상관없이 신호를 이용할 수 있다. 또한 이동 중에도 실시간 사용이 가능하여 자동차, 선박 등의 길 안내뿐만 아니라 응급구조, 재해 예측 등 활용범위가 매우 넓다.

단, 실내에서는 오차범위가 커지며 기본적으로 약 10m의 오차범위를 감수해야 한다는 한계점이 있다. 본 연구에서는 사용자의 위치 값을 측정하기 위해 gps좌표를 이용한다. [3]

2.3. 선형가속 센서

Android platform은 기기의 동작을 모니터링 할 수 있는 센서를 다수 지원하는데 본 연구에서는 동작센서를 사용한다. 그 중에서도 흔들림을 감지하기 위해 선형가속 센서를 이용한다.

현대의 일반적인 Mobile device는 여러가지 동작센서를 지원한다. 동작센서란 기기의 동작을 모니터링 할 수 있는 센서이다. 센서의 종류는 센서값 도출의 기반에 따라 분류한다. hardware 기반 센서와 이를 바탕으로 가공한 데이터인 software기반의 센서가 있다. 본 연구에서는 흔들림을 감지하기 위해 중력 값을 제거하고 순수하게 기기에 가해진 가속도만 측정하는 software기반의 선형가속 센서를 이용한다.

해당 프로젝트에서는 선형가속센서 관련 Android studio API인 TYPE_LINEAR_ACCELERATION를 이용하여 기기에 가해진 힘(흔들림)을 측정한다.[4]

동작 센서는 흔들기, 회전, 스윙등 기기에 가해지는 운동을 측정할 수 있다. 기기에 가해지는 동작은 사용자의 의도적 입력(스마트폰을 기우려 화면을 가로로 맞추는 등) 또는 기기가 있는 물리적 환경(기기를 가진 사용자가 자동차를 타고 이동하는 등)에 의해 생성된다.

Android studio에서 동작 센서는 각 SensorEvent에 대해 센서 값의 다차원적 배열을 반환한다. 예를 들어 단일 센서 이벤트에서 가속도계는 세 개의 좌표축에 대한 가속력 데이터를 반환하고 자이로스코프는 세 개의 좌표축에 대한 회전 속도 데이터를 반환한다. 이런 데이터 값은 다른 SensorEvent 매개변수와 함께 float 배열로 반환된다.

해당 프로젝트에서는 관련 Android studio API인 TYPE_ACCELEROMETER를 이용하여 x, y, z축의 가속도를 측정한다. 이를 통해 기기에 가해진 힘(흔들림)을 측정 가능하다.

2.4. Django

Django는 web application framework로 web service를 효율적으로 빠르게 개발하도록 해주는 software다. web application개발에 공통적으로 구현되어야 하는 인증, database의 정보 검색, cookie 관리 등을 쉽게 개발하도록 지원한다. 즉, Django는 다양한 기능을 web application framework라고 하는 재사용 가능한 모듈의 대규모 컬렉션으로 그룹화하여 작업을 더 쉽게 수행할 수 있도록 한다. 개발자는 Django를 사용하여 코드를 보다 효율적으로 구성하고 작성하며 개발 시간을 크게 단축할 수 있다. 특히 양식 처리, 사용자 인증, 관리자 페이지에 대한 편리한 지원이 많다.

MVC(Model-View-Controller)의 디자인 패턴을 지향한다. ORM(Object Relational Mapping)을 지원하여 SQL에 대한 사전지식이 부족하더라도 database를 다룰 수 있다.

위에서 언급한 이유들과 같이 prototype을 단시간 내에 개발하기에 유리하고 Python을 사용하기

에 여러 library를 사용할 수 있다는 장점 때문에 본 프로젝트는 Django를 이용하여 구현한다.

3. 프로젝트 내용

3.1. 시나리오

Mobile Device로 data를 수집하여 이를 바탕으로 이동경로 패턴을 도출한다. 해당 이동경로 패턴을 이용해 타인과 유사도를 검사하여 가장 높은 유사도를 보이는 user를 매칭하는 서비스를 기획하고 구현한다.

이때 수집되는 data는 GPS좌표(위도, 경도), 선형가속벡터 값(x, y, z 값), wifi연결시간(연결시간, 해제시간) 이다. 해당 데이터를 Mobile device에서 background program을 통해 지속적으로 수집하여 server로 전송한다. server에서는 수집된 raw data를 정제, 가공하여 이동패턴이라는 유의미한 information을 생성한다. 이동패턴에는 이동 궤적, 정지장소, 이동수단 등이 포함된다. 해당 이동 패턴을 다른 user와 상호비교하여 유사도를 도출하고 해당 유사도가 가장 높은 유저를 추천하여 상호 매칭한다.

각 단말의 동작 시나리오는 아래와 같다.

3.1.1. Mobile Device

user의 mobile device에서 구현되는 클라이언트 어플리케이션의 시나리오에 관한 시나리오이다.

background에서 지속적으로 data를 수집한다. 해당 data는 server로 전송되고 local storage에서는 삭제된다.

foreground에서 로그인을 비롯한 UI가 출력된다. user의 도출된 이동경로 패턴 결과와 추천받은 유사한 이동경로의 user와 매칭여부를 결정한다.

1) 회원가입 및 로그인

The image shows two side-by-side mobile app screens for a service named 'Companion'. Both screens have a white background and a black border. The left screen is for login/signup, featuring a blue 'Companion' header, two rounded rectangular input fields for 'ID' and 'PW', and two blue buttons labeled 'sign in' and 'sign up'. The right screen is for registration, featuring the same blue 'Companion' header, three rounded rectangular input fields for 'ID', 'PW', and 'email', and a single blue button labeled 'join us:)'. All text is in a clean, sans-serif font.

그림 2. login and signup UI design

위치정보를 가진 사용자를 분류하고 인지하기 위해 그림2와 같이 회원가입과 로그인을 요구한다. 각 사용자는 고유의 id를 갖고 인증은 jwt(Json web token)을 통해 수행한다.

2) 스마트폰의 센서를 통한 유저 흔적(data) 수집

모바일 단에서 background program으로 지속적이고 규칙적으로 data를 수집한다. 수집되는 data의 종류는 GPS좌표(위도, 경도), 선형가속벡터 값(x,y,z값), wifi연결시간(연결시간, 해제시간) 이다. 해당 기능은 안드로이드 service로 구현되어 유저의 front에서 구현되는 UI와 관련된 프로세스와 독립적으로 구현된다.

3) 일정시간이 지나면 서버로 저장된 데이터 전송

일정 시간이 지나거나 특정 용량에 도달하면 server로 데이터들을 전송한다. HTTP를 이용한 통신을 통해 구현한다.

4) 서버로부터 이동패턴이 유사한 동행인 추천 내용 수신

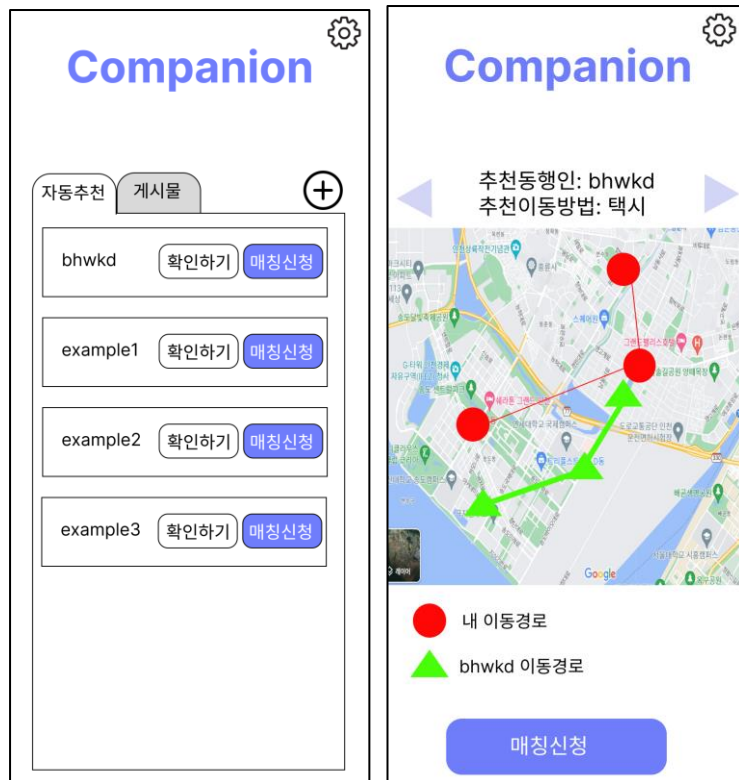


그림 3. Automatic recommendation UI design

server가 자동으로 수집된 data를 바탕으로 해당 사용자의 대표이동패턴과 패턴의 유사도가 높은 유저를 추천하는 데이터를 전송하면 이를 수신하고 출력한다.

5) 유저가 임의의 이동경로 생성

Companion

제목

내용



작성완료

그림 4. route Creation UI Design

서비스가 지속적이고 자동적으로 수집한 데이터를 바탕으로 생성한 이동패턴 외에도 유저가 임의로 이동경로를 게시글 형태로 생성한다. 해당 게시글은 그림4와 같이 제목, 이동경로, 내용으로 구성된다. 이러한 구성으로 인해 생성한 이동경로가 어떠한 이동경로인지 조깅, 출퇴근, 여행 등 내용을 설명 가능하다. 자동으로 생성된 이동패턴이 일상속의 반복적인 이동패턴이라면 해당 이동패턴은 특정한 목적이 분명하고 유저가 단기간에 이루어질 것으로 예상되는 궤도이다.

6) 임의의 이동경로와 자동 측정된 이동패턴 바탕으로의 추천

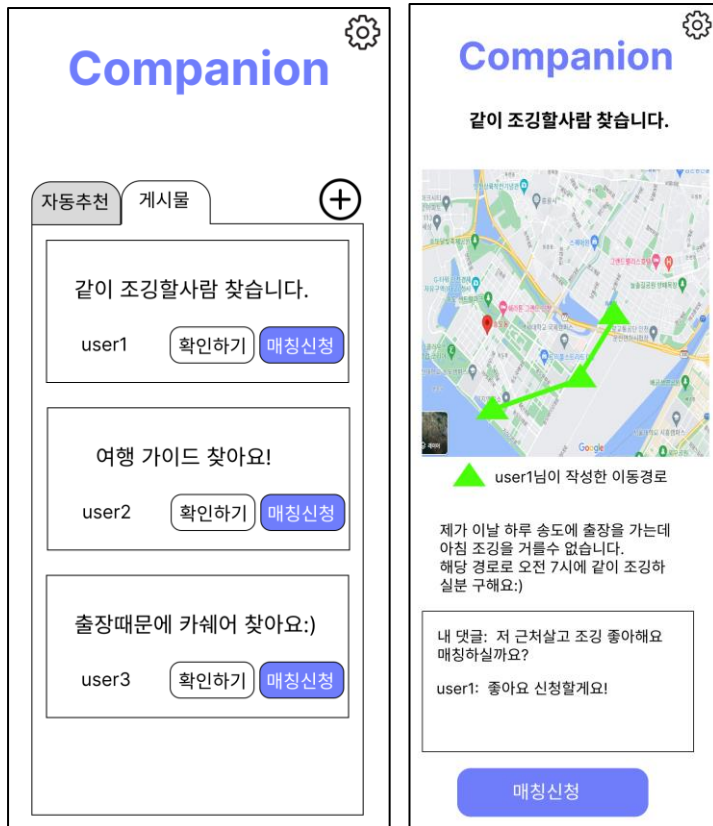


그림 5. Route recommendations generated by users UI

'5)'에서 유저에 의해 임의로 생성된 이동패턴과 서비스가 자동으로 판단한 이동패턴간의 유사도를 검사하여 유사도가 높은 게시글을 리스트 형태로 UI에 출력한다.

3.1.2. Server단 시나리오

Mobile Device로부터 전달받은 data를 정제, 가공한다. 또한 정제된 data를 바탕으로 이동경로 data를 도출하고 해당 data를 바탕으로 user간 유사도를 연산한다. 해당 유사도를 바탕으로 user를 상호 매칭하여 결과를 Mobile단으로 전송한다.

1) 유저로부터 데이터 수집

유저로부터 유저id, 시간, gps좌표, 방향, 가속도벡터, wi-fi연결여부의 데이터를 수신 받는다. 해당 데이터는 raw data로 처리 전까지 server에 쌓아 둔다.

2) 데이터 정제

raw data는 gps 오류, 의미가 적은 data등 많은 noise를 포함한다. 그렇기에 이러한 데이터들을 유의미한 data만 남기도록 정제한다. 해당 과정은 filter를 두어 데이터를 정제시키며 이루어진다.

3) 정보집합 생성

정제된 raw data를 조합하여 의미 있는 data인 combined data를 생성한다. 즉 이동경로 데이터를 생성한다. 이는 출발지, 목적지를 포함한 이동경로, 이동수단(보행, 대중교통)으로 구성된다.

4) 유저의 대표이동경로 파악

이동경로 data를 통해 해당 유저를 대표할 수 있는 이동경로를 도출한다. combined data로 complex data를 생성하는 과정이다. wi-fi연결이 된 시점의 위치, 일정시간동안 gps값의 유지 등을 이용해 정지된 위치를 파악한다. 이 정지된 위치를 바탕으로 사용자의 이동경로가 도출된다.

5) 대표이동패턴을 바탕으로 가장 유사한 유저를 추천

도출한 대표이동경로를 분석해 가장 유사도가 높은 타인을 찾아 매칭한다. 이로써 이동 패턴이 유사한 유저라는 meaning information이 생성된다.

유사도의 기준은 아래의 3가지 기준에 각각 가중치를 두어 계산한다.

- ①대표 이동패턴의 각 정지위치(출발지, 목적지)의 거리
- ②이동수단(보행, 자동차, 전철)의 일치,
- ③이동 시각의 유사함

3.2. 아키텍처

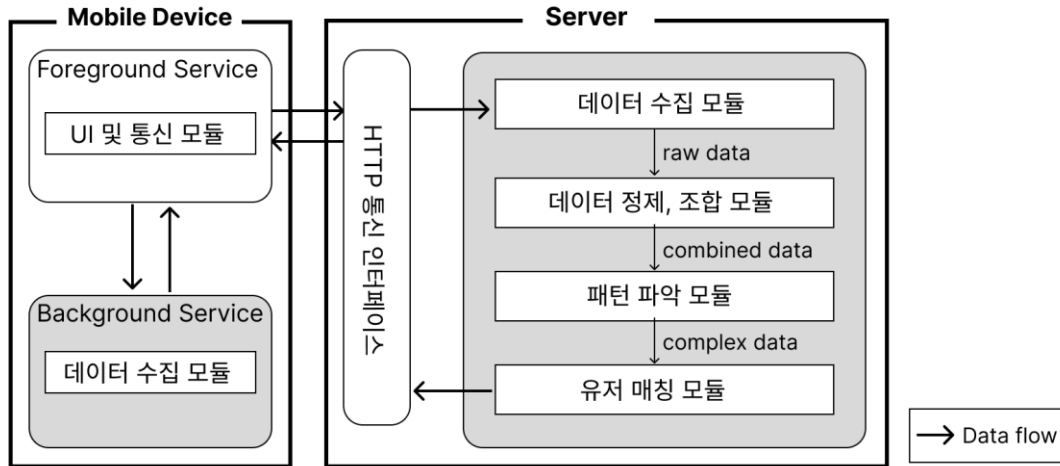


그림 6. Architecture overview

본 프로젝트의 전체적인 Architecture는 그림2와 같다. 유저단은 데이터를 수집하는 background service와 UI가 위치한 foreground service로 구성되며 server는 각 역할에 따른 모듈들로 구성된다.

3.3. 시스템 설계

시스템은 그림1과 같이 구성된다. 해당 절에서는 data flow를 중심으로 data가 처리되는 과정을 기술하며 시스템의 전체적인 구조와 흐름을 제시한다.

3.3.1. 데이터 수집

```
mysql> DESC sensor_data;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
user_id	varchar(30)	NO	MUL	NULL	
time	timestamp	NO		NULL	
latitude	double	NO		NULL	
longitude	double	NO		NULL	
acceleration_x	float	NO		NULL	
acceleration_y	float	NO		NULL	
acceleration_z	float	NO		NULL	
onwifi	tinyint(1)	NO		NULL	

9 rows in set (0.00 sec)

그림 7. Database schema

Mobile Device의 background service에서 일정주기로 data를 수집한다. 기본적으로 정해진시간t

마다 주기적으로 data를 수집하며 아래 기술한 특정 data가 수집되는 조건발생시에도 수집을 진행한다.

- ① GPS좌표(위도,경도) : 시간t 경과마다 수집.
- ② 선형가속벡터(x,y,z) : 시간t 경과마다 수집.
- ③ wifi접속 여부 : 시간t 경과시마다 수집. 혹은 wifi 연결, 해제 이벤트 발생시에 수집.

하나의 data가 수집되는 조건이 발생하면 다른 data들도 동시에 수집되어 저장된다.

해당 data는 sever로 전달되어 그림 7과 같이 설계된 DataBase에 저장된다. 이렇게 센서로부터 수집된 Raw Data는 data 정제 단계로 넘어간다.

3.3.2. 데이터 정제

Mobile Device의 sensor로부터 수집된 raw data는 noise를 내포하기에 filtering 과정을 거쳐야 한다. 아래는 각 data별 수행되는 필터의 종류와 역할이다.

◦GPS좌표(위도,경도)

- ① location filter : gps는 일정정도의 오류를 내포하며 실내에서는 오류의 정도가 커짐으로 이를 wifi 연결로 인해 확인되는 위치 정보 등으로 보정한다. sensor detect와 동시에 이루어져야 한다.
- ② vector filter : 수집된 data를 시간을 기준으로 정렬하면 인접한 index사이에 gps좌표간 euclidean distance를 구할 수 있고 이를 해당 index사이의 시간 차로 나누어 속도를 구한다. 해당 속도가 지나치게 큰 값이라면 해당 data를 error로 간주하여 삭제한다.[5]

◦선형가속벡터(x,y,z)

- ① absolute filter : 가속도의 값이 지나치게 큰 값으로 측정된 data는 오류로 간주하여 인접한 시간의 vector의 산술평균으로 값을 바꾼다.

이렇게 정제된 data는 이동경로라는 정보로 도출되고자 다음 단계로 넘어간다.

3.3.3. 데이터 조합

오류가 보정된 raw data를 바탕으로 이동경로라는 data를 도출한다. 이동경로 data는 일종의 graph로 구성된다. vertex는 상태(정지 혹은 이동)와 머문시간(도달시간, 출발시간)으로 구성된다. edge는 이동수단이 저장된다. 해당 data를 도출하는 방법은 아래와 같다.

① 시간으로 정렬했을 때 인접한 gps간의 (거리/시간)으로 도출한 속도가 일정 값 STOP_DECISION 이하인 연속한 data들을 합쳐서 하나의 vertex를 구성한다. 상태는 정지로 판단된다. 합쳐진 data들의 시간 값의 배열을 T라고 했을 때 도달시간은 $\min(T)$ 로 출발시간은 $\max(T)$ 로 구성된다.[6] wifi연결이 True인 data도 정지로 판단한다.

② 인접한 data간 속도가 STOP_DECISION이상인 data는 이전의 data와 합쳐지지 않고 독립적으로 구성되어 상태는 이동중으로 저장된다.

③ 위의 ①, ②과정이 끝난 vertex간의 가속벡터(흔들림), 속도에 따라 decision value를 두어 도보인지 자동차인지 지하철인지 판단하여 edge에 이동수단을 저장한다.

위의 방식으로 도출된 이동경로 data는 전체 데이터가 연결되어 있다. 그러므로 유사한 위치 값을 갖는 정지상태 vertex를 기준으로 회귀주기가 반복됨을 판단한다면 해당 주기 별로 데이터를 자르거나 혹은 24시간을 기준으로 자를 수 있다.

이렇게 구성된 combined data는 패턴파악 모듈로 전달된다.[7]

3.3.4. 패턴파악

user마다 이동경로 data가 복수 개 저장 되어있다. 일반적으로 요일마다 이동패턴이 반복됨으로 요일 별로 이동경로 data를 분류한다. 가장 빠른 시간의 정지vertex를 출발지로 가장 느린시간의 정지vertex를 도착지로 판단한다. 요일별 이동경로 데이터들에 대해 정지부터 도착까지 vertex간의 좌표의 평균을 구해 요일마다 하나의 대표 이동경로 패턴을 도출한다.

3.3.5. 의미 있는 정보 추출

user마다 갖는 대표이동경로(패턴)을 바탕으로 상호간에 유사도를 측정한다. 유사도 측정 방식은 rank를 계산하는 것이다. 출발, 도착지의 거리가 가까운지, 이동시간이 유사한지, 이동수단이 같은지에 대해 각각 점수를 배정해 해당 점수에 따라 가장 큰 유사도를 갖는 user를 찾는다.

3.3.6. 이동패턴 출력

모바일 단으로부터 유사한 이동경로를 갖는 user에 대한 request를 받으면 해당 리스트를 response한다. 응답에는 이동경로 data가 포함되어 있다. 해당 이동경로를 모바일 단에서 구글 map에 vertex를 원으로 표현하여 출력해 유저에게 제공한다.

3.4. 요구사항

3.4.1. Mobile app(User) 요구사항

- background에서 지속적으로 data를 수집하는 기능의 application이 필요하다.
- Mobile단에서 user와 그 data들을 관리할 RDBMS가 필요하다. 이곳에 데이터를 축적해 두었다가 특정 시간이 지나면 server로 이를 전송하고 모바일 단말기에서는 삭제한다.
- Server로부터 전달받은 User의 대표이동 경로를 지도에 표기할 수 있어야 한다. 이때 정지 위치 및 주요 동선지점을 pin으로 표시, 각 pin은 line으로 이어지는 구조로 표기되도록 한다.
- data수집시에 이상이 있거나 서버와의 접속이 원활히 진행되지 않는다면 유저, 서버단에 모두 이를 알릴 수 있는 push알림이 있어야 한다.

3.4.2. Server 요구사항

- float값으로 전송될 gps좌표계산을 컴퓨팅 파워에 따라 int로 변환해 연산한다.
- 간단한 data는 json으로 송수신 하지만 데이터의 크기에 따라 FTP를 고려해야 한다.
- 분석에 필요한 임계값 (gps값의 차이로 정지여부를 판단할 때 기준 등)들을 설정하기 위해 수집된 데이터를 matplotlib등을 통해 시각화해 확인해야 한다.
- 한 유저에 대해 여러 개의 이동 동선 데이터를 지도에 중첩하여 구분되게 표기해야 한다. 이를 위해 이동 동선당 다른 색을 부여한다.

4. 프로젝트 결과

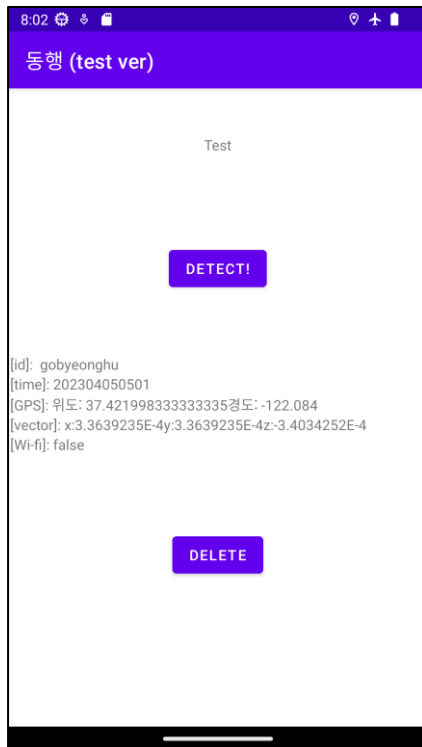


그림 8. Mobile Device의 sensor detection 예시

user_id	time	latitude	longitude	acceleration_x	acceleration_y	acceleration_z	onwifi
test_user_01	2023-04-19 07:04:17	37.2459902	127.076736	3.37854	0.00340271	-3.41475	0
test_user_01	2023-04-19 07:09:17	37.2482411	127.0781278	8.37893	0.0044108	-1.41443	0
test_user_01	2023-04-19 07:14:17	37.2509739	127.0792972	0.287939	1.47741	0.41453	0
test_user_01	2023-04-19 07:19:17	37.2538553	127.0753253	0.107279	0.164454	-3.41453	0

그림 9. Server단의 Data 저장 sample

그림8은 모바일 어플리케이션에서 수집한 sensor data를 UI로 출력한 모습이다. 이와 같이 raw data에 해당하는 값들을 유저단에서 수집한다. 해당 데이터는 그림9와 같이 서버의 data base에 저장된다. raw data는 서버단의 데이터 정제 모듈과 데이터 조합 모듈을 거쳐 이동경로라는 그래

프 형태의 데이터로 생성된다. 해당 이동경로 데이터는 유저간 유사도를 분석하여 매칭된다.

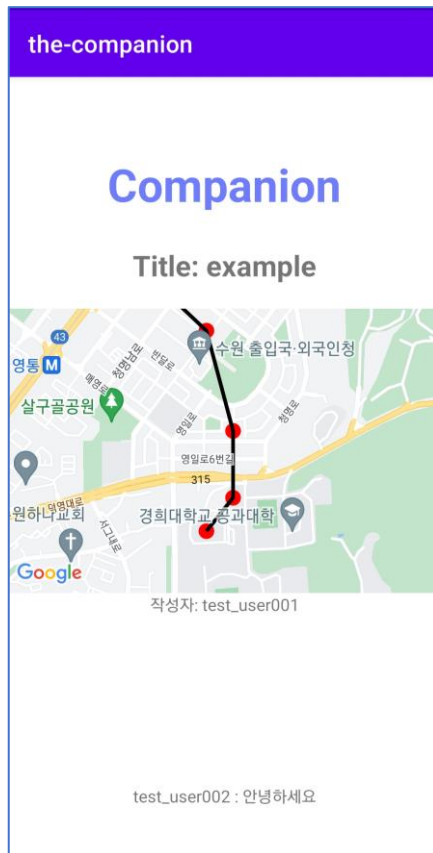


그림 10. 이동경로 UI출력 sample

그림10은 이동경로데이터를 모바일 단에서 출력한 결과이다. 이동경로 데이터를 구성하는 정지, 이동궤적 두가지로 나뉘는 vertex는 붉은색 원으로 표시되고 해당 vertex들을 선으로 연결한 상태로 데이터를 시각화 한다. 해당 지도는 google map api로 구현되어 구글맵 위에 표시되고 주변 장소 검색 등 구글맵의 기능들을 그대로 활용 가능하다.

5. 결론

5.1 기대효과

Mobile device를 통해 user의 data를 수집하고 이를 이용하여 이동경로라는 information을 도출해 해당 이동경로가 유사한 user를 매칭해 주는 서비스를 제안하였다.

본 프로젝트는 유저가 일상속에서 갖는 반복적이고 일반적인 이동경로를 Mobile device를 통해 수집한 data를 이용해 파악하고 해당 패턴의 유사성을 보이는 타인을 매칭한다. 이 서비스를 통해 사용자는 본인과 동행이 가능한 사람을 찾을 수 있다. 이러한 매칭이 실제 동행으로 이어진다면 현재 다수 존재하는 car sharing, 렌터카사업 등에 이동경로가 비슷한 유저를 추천하는 본 시스템이 적용하여 더욱 높은 효과를 기대할 수 있을 것이다.

또한 특정 이동 목적에 맞는 이동경로를 유저가 게시글 형태로 생성하여 해당 이동경로와 유사한 이동패턴의 유저를 시스템이 추천하여 게시글을 노출하는 서비스를 제안하였다. 해당 기능은 기존에 존재하지 않던 형태의 서비스로 이동경로 기반의 새로운 유저간 연결을 제공할 수 있다.

5.2 추후 연구 방향

본 프로젝트는 확장성을 고려하여 각 기능별로 분리되어 설계되어 있다. 예를 들면 현재 유저의 이동수단을 파악함에 있어 가속도 벡터의 값을 사전에 정해진 상수로 크기를 비교하여 도보인지 교통수단에 탑승했는지를 파악한다. 또 유사한 이동경로 패턴을 갖는 유저를 탐색할 때 분기문을 이용하여 판단하도록 설계되어 있다. 이러한 군집화, 유사도 분석등의 판단들은 향후 보다 많은 데이터가 축적되면 deep learning 및 machine learning algorithm을 적용하여 정확도를 높일 수 있다. 이러한 점을 고려하여 향후에는 데이터 수집량을 늘리고 알고리즘의 정확도를 높이는 방향으로 연구를 진행한다.

6. 참고문헌

[1]<https://developer.android.com/guide/platform>

[2] <https://developer.android.com/studio/intro>

[3]한진희, "GPS 수신기 설계 기술 및 동향",2007

[4]<https://developer.android.com/guide/topics/sensors/>

[5] 양승국, 백혜정, 김제민, 전명중, 박영택 "개인 경로 학습을 위한 GPS 좌표 기반 이동 궤적 추출", 2011

[6]김선용, 박범준, 정재진 "모바일장치의 GPS를 이용한 사용자 이동경로 분석 및 이동경로 추천 시스템", 2011.2

[7] 양혜진 "모바일의 위치데이터를 활용한 사용자 조사 방법", 2015.5