

위치 기반 서비스를 위한 동적 위치 인지 기법

정창훈¹, 김철진^{*}

¹인하공업전문대학 컴퓨터시스템과

A Dynamic Location Recognition Technique for Location-based Service

Chang-Hun Jung¹, Chul-Jin Kim^{*}

¹Dept. of Computer Systems and Engineering, Inha Technical College

요 약 최근 스마트폰의 위치 기반 서비스는 급속도로 성장하는 모바일 기술 중 하나이다. 본 논문에서는 모바일에서 위치 기반 서비스를 사용할 때 특정한 위치에 따라 위치 기반 서비스의 주기를 변경시키기 위한 기법을 제안한다. 이를 통해 위치 기반 서비스의 주기를 위치에 따라 커스터마이제이션 할 수 있으며, 결과적으로 일정한 주기로 위치 기반 서비스를 이용할 때보다 모바일 자원의 낭비를 줄일 수 있는 프로세스를 제안한다.

Abstract The recent location-based services of smart-phones are some of the rapidly growing mobile technology. This paper proposes a technique for modifying the change cycle of location-based services according to the specific location using location based services. This technique shows that the cycle of the location-based services can be customized based on the location. As a result, this technique proposes a process that can reduce the waste of resources compared to the location based services of a constant cycle.

Key Words : Location-based Service, Dynamic Location Recognition, Customization

1. 서론

최근 스마트폰의 사용량이 증가함에 따라 스마트폰은 단순히 전화, 문자의 기능만을 하는 도구가 아닌 일상생활에서 없어서는 안 될 삶의 한 부분으로 자리 잡았다. 그에 따라 스마트폰을 이용한 다양한 기술이 연구·개발되고 있고, 그중에서도 위치 기반 서비스(LBS : Location Based Service)는 GPS, WIFI, 4G를 통해 현재 사용자가 위치한 장소를 기반으로 날씨, 교통, 의료, 지리정보등의 다양한 위치 정보를 얻을 수 있는 서비스로서, 최근 각광 받고 있는 기술 중 하나이다. 또한 이러한 스마트폰의 장점 중 하나는 바로 언제 어디서나 사용할 수 있는 휴대성이며, 그 휴대성을 오랫동안 유지하기 위해서는 모바일 자원의 낭비를 줄여 배터리의 효율성을 높여야 한다.

본 논문에서는 모바일 플랫폼에서 GPS LBS를 사용하는 어플리케이션의 효율성을 높이기 위하여 동적 위치 인지 기법을 제안하고, 이 기법의 가장 큰 장점은 모바일 자원의 낭비를 줄일 수 있다는 것이다. 동적 위치 인지 기법을 사용하면, 사용자의 현재 위치와 임의의 목적지 사이에 n 개의 중간 지점을 얻을 수 있고, 이렇게 얻어진 중간 지점들을 인지함에 따라 기존에 일정했던 GPS LBS의 주기를 목적지까지 남은 거리에 비례하게 커스터마이제이션 할 수 있다. 결과적으로 사용자가 목적지에 가까이 있을 때보다 먼 위치에 있을 때, GPS LBS 주기를 길게하여 GPS와 모바일간의 낭비되는 통신횟수를 줄일 수 있으므로 배터리의 효율성을 높일 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 관련 연구를 주제로 LBS, Geo Code를 설명하고 본 논문에서

^{*}Corresponding Author : Chul-Jin Kim(Inha Technical College)

Tel: +82-32-870-2338 email: cjkim@inhac.ac.kr

Received April 2, 2014

Revised (1st May 21, 2014, 2nd June 9, 2014)

Accepted July 10, 2014

서 제안하는 기술과 관련된 연구 사례를 제시한다. 3장에서는 GPS LBS를 기반으로 위치 기반 서비스를 위한 동적 위치 인지 기법을 제안하고, 4장에서는 그 기법을 버스 알람 어플리케이션 사례에 적용하여 적합성을 검증하며, 그리고 5장에서는 결론을 맺는다.

2. 관련 연구

2.1 LBS

LBS란 Location Based Service의 약자로서 모바일이나 네비게이션, PDA 같은 플랫폼에서 사용가능한 위치 기반 서비스를 말한다. 이 서비스는 사용자의 현재 위치 정보를 얻을 수 있으며 뿐만 아니라 지리, 의료, 상품, 교통등의 생활 전반적인 분야에서 널리 활용되고 있는 서비스이다.

모바일 플랫폼 같은 경우에는 LBS를 사용할 때 GPS를 이용하는 인공위성 신호 방식과 WIFI, 4G등을 이용하는 네트워크 신호 방식으로 구분된다. 인공위성 신호 방식은 오차 범위가 작지만 배터리 소모가 크고 실내에서 사용이 불가하다는 단점이 있으며, 네트워크 신호 방식은 배터리 소모가 작고 실내에서 사용이 가능하지만 오차범위가 크다는 단점이 있다[1].

우선 LBS를 사용하기 위해서는 모바일 어플리케이션의 AndroidManifest.xml 파일에서 퍼미션 설정이 되어 있어야 한다. 이 퍼미션의 의미는 모바일 하드웨어 계층에서 GPS Driver나 Network Driver의 사용여부를 설정하는 것이다. 다음으로는 Table 1을 참고하여 Location API가 제공하는 Location Manager에서 인공위성(GPS) 신호 방식을 이용할 것인지 네트워크(WIFI, 4G) 신호 방식을 이용할 것인지 대한 프로바이더를 선택하여야 한다. 그리고 LBS 데이터의 갱신을 주기적으로 받아올 것인지, 아니면 최초의 한번만 받아올 것인지를 위한 설정을 Table 2를 참고하여 완료한다[2]. 이렇게 퍼미션과 프로바이더, 갱신주기의 대한 설정이 완료되면 그 다음 단계로는 실제로 LBS 데이터를 받아오기 위해 Location Listener 객체를 선언하여 Table 3에 제시된 4개의 메소드를 준비한다[3]. 그 중 현재 사용자의 위치가 갱신될 때마다 호출되는 onLocationChanged()에서 Table 4에 제시된 Location API의 메소드 중 getLongitude()와 getLatitude()를 사용하면 현재 사용자가 위치한 곳의 경

도·위도 데이터를 GPS Driver나 Network Driver를 통해 받아와서 double형 변수에 저장하여 활용할 수 있게 된다[4].

[Table 1] LBS Provider Type of Location Manager

Artificial Satellite Signal	LocationManager. GPS_PROVIDER
Network Signal	LocationManager. NETWORK_PROVIDER

[Table 2] LBS Data Update Cycle Method of Location Manager

requestLocationUpdates()	Register for location updates using the named provider, and a pending intent
requestSingleUpdate()	Register for a single location update using the named provider and a callback

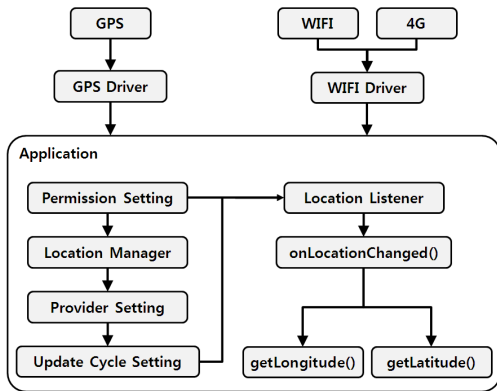
[Table 3] Method of Location Listener

onLocationChanged()	Called when the location has changed
onProviderDisabled()	Called when the provider is disabled by the user
onProviderEnabled()	Called when the provider is enabled by the user
onStatusChanged()	Called when the provider status changes

[Table 4] Method of Location API

getAccuracy()	Get the estimated accuracy of this location, in meters.
getTime()	Return the UTC time of this fix, in milliseconds since January 1, 1970.
getLongitude()	Get the longitude, in degrees.
getLatitude()	Get the latitude, in degrees.

위에 제시된 메소드와 아래의 LBS 데이터의 흐름을 통해 사용자는 현재 위치한 곳의 경도·위도 데이터를 수신할 수 있으며, 이를 통해서 지리, 의료, 상품, 교통등과 관련된 어플리케이션에서 활용하여 위치 기반 서비스를 이용할 수 있다[Fig. 1].



[Fig. 1] Location Based Service Data Flow

2.2 Geo Code

Geo Code는 Google Maps를 기반으로 지리정보에서 주소, 우편번호 또는 경도·위도데이터를 얻을 수 있는 코드이다[5]. 모바일 플랫폼 같은 경우에는 AndroidManifest.xml 파일에서 Http를 사용하겠다는 퍼미션을 설정한 후에 Http Get 메소드를 통하여 Geo Code에 접근할 수 있다[Fig. 2].

```
HttpGet httpGet = new HttpGet("http://maps.google.com/maps/api/geocode/json?address="+destination+"&sensor=false");
```

[Fig. 2] Code of Geo Code Access through Http Get

사용자는 Fig. 2의 코드 중 'destination'이라는 변수 안에 원하는 목적지의 명칭을 넣어 Geo Code에 접근하면 웹 형식으로 되어 있는 지리 정보를 탐색할 수 있다[Fig. 3].

```
{ "results": [ { "address_components": [ { "long_name": "서울역", "short_name": "서울역", "types": "train_station", ..... "geometry": { "location": { "lat": 37.554531, "lng": 126.970663}, ..... "tation", "establishment" } ], "status": "OK" }
```

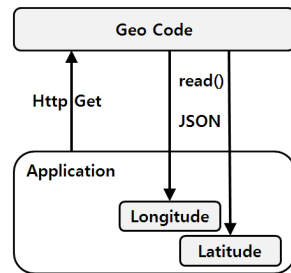
[Fig. 3] Example of Geography Information through Geo Code

다음으로는 read()메소드를 이용하여 Geo Code의 지리정보를 받아온 후에 JSON을 이용하여 파서시키면 원하는 목적지의 경도·위도 데이터 double형 변수에 저장할 수 있게 된다[Fig. 4].

```
while ((geoInfo = stream.read()) != -1) {
    stringBuilder.append((char) geoInfo); }
jsonObject = new JSONObject(stringBuilder.toString());
double desLongitude = ((JSONArray) jsonObj.get("results")).getJSONObject(0).getJSONObject("geometry").getJSONObject("location").getDouble("lng");
double desLatitude = ((JSONArray) jsonObj.get("results")).getJSONObject(0).getJSONObject("geometry").getJSONObject("location").getDouble("lat");
```

[Fig. 4] Code of Geo Code Reception through read() and JSON

이러한 데이터 흐름을 통해 사용자는 Geo Code에 접근하여 원하는 목적지의 경도·위도 데이터를 얻을 수 있고, 결과적으로 이 데이터들을 다양한 분야에 적용하여 활용할 수 있다[Fig. 5].



[Fig. 5] Geo Code Data Flow

2.3 위치 기반 서비스를 이용한 스마트폰 관광 정보 시스템[6]

연구 [6]에서는 위치 기반 서비스를 이용하여 스마트폰 사용자에게 관광 정보를 제공한다. 이 연구는 LBS를 이용하여 현재 사용자의 경도·위도 데이터가 개발자가 설정한 콘텐츠의 내용을 보여 줄 경도·위도 데이터 범위에 속하면 관광 정보를 보여주는 시스템이다. 이 시스템은 GPS 또는 네트워크 기반의 LBS를 이용하여 사용자에게 관광지에 관한 전문해설사의 음성정보와 이미지 정보를 이용하여 맞춤형, 실감형 콘텐츠를 제공한다. 그리고 관광객은 해당 관광지에서 보고 느낀 점을 사진 또는 동영상 형태로 서버에 전송하여 콘텐츠를 관리할 수 있으며, 다른 사용자와의 공유를 통하여 관광지에 대한 다양한 정보를 얻을 수 있다. 연구[6]은 단순히 LBS를 이용하여 위치 정보 서비스를 제공하는 시스템이지만, 본 논

문에서 제시하는 연구는 이러한 시스템을 조금 더 효율적으로 이용할 수 있도록 기술적인 기법을 제시하는 것이므로 기술적인 차이를 보인다.

2.4 위치기반 서비스의 감성 적용에 관한 연구[7]

연구 [7]에서는 LBS를 통해 사용자에게 위치 정보를 제공하는데 있어, 사용자의 감성을 분석하여 조금 더 현실적으로 원하는 정보를 제공하는 모델을 제시한다. 기존의 위치기반 서비스에서는 사용자의 위치에 기인하여 주변 위치정보를 제공하고 위치 추적 및 길 찾기, 안전 및 보안 목적의 서비스들이 다양한 형태로 제공되어 사용자의 만족을 실현하고 있으나, 사용자 감성에 기인한 정보로는 사용자 만족을 실현해 주지 못하고 있다. 그러나 본 연구에서는 사용자가 입력한 문장을 분석하여 어떠한 감성으로 입력한 것인지를 유추하고 그에 따라 상응하는 위치 카테고리를 추천하는 모델이다. 이 모델에서 위치 정보를 제공하는 핵심적인 요소들은 사용자 감성에 적용되는 정도에 따라 결정된다. 연구 [7]은 LBS와 야날로그적인 기술을 결합하여 위치 정보를 제공해주는 모델을 제시하지만, 본 논문에서는 LBS 시스템 자체를 효율적으로 사용할 수 있는 기술적인 기법을 제시한다. 그러므로 연구 [7]과 본 논문에서 제시하는 연구는 기술적인 차이가 있다.

2.5 푸시 서비스를 이용한 모바일 커스터마이제이션 기법[8]

연구 [8]은 구글에서 제공하는 GCM 푸시서비스를 이용하여 관리자와 사용자간의 데이터 전달을 통한 커스터마이제이션 기법을 제안한다. 관리자는 푸시서비스를 이용하여 사용자 어플리케이션의 변수값을 제어할 수 있고, 그로인해 사용자 어플리케이션은 다양한 형태로 커스터마이제이션 될 수 있다. 이 연구의 큰 장점은 관리자가 시간과 공간의 영향을 받지 않고서도 언제 어디에서나 사용자의 어플리케이션을 커스터마이제이션 할 수 있다는 것이고, 결과적으로 시간과 공간으로 인한 낭비와 불편의 최소화를 기대해 볼 수 있다. 연구 [8]은 관리자가 푸시서비스를 이용하여 사용자의 어플리케이션을 직접 커스터마이제이션 하는 반면, 본 논문에서 제안하는 연구는 LBS를 이용하여 사용자와 목적지간 거리에 따라서 중간지점이 자동으로 커스터마이제이션 된다. 그러므로

연구 [8]은 정적 커스터마이제이션의 개념이고, 본 논문에서 제안하는 연구는 동적 커스터마이제이션의 개념이므로, 두 연구는 커스터마이제이션에 대해서 정적과 동적이라는 차이를 보인다.

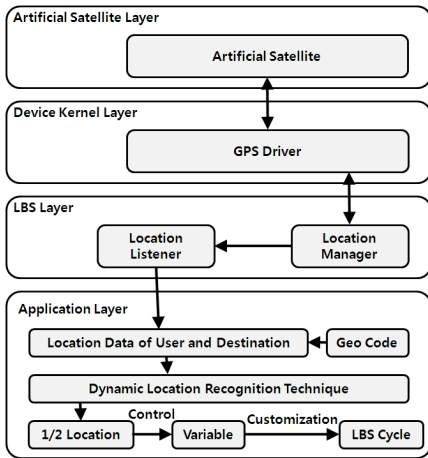
3. 위치 기반 서비스를 위한 동적 위치 인지 기법

본 논문의 위치 기반 서비스를 위한 동적 위치 인지 기법은 모바일 환경의 안드로이드 플랫폼 기반으로 GPS LBS를 이용하여 제안한다.

3.1 GPS LBS를 이용한 동적 위치 인지 구조

본 논문에서 제안하는 동적 위치 인지 기법의 계층도는 Fig. 6와 같다. 인공위성 계층, 디바이스 커널 계층, LBS 계층, 어플리케이션 계층으로 총 4계층으로 구분 된다.

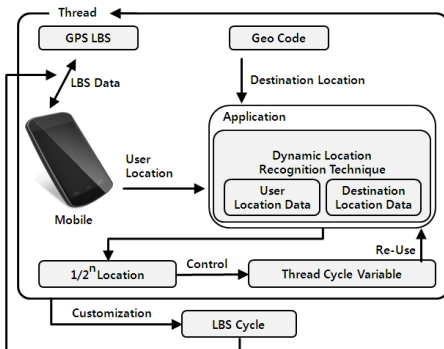
사용자는 어플리케이션에서 GPS LBS를 이용한 Location Manager를 통해 현재 위치 데이터를 요청한다. 요청 데이터는 모바일의 하드웨어 계층인 디바이스 커널 계층에 GPS Driver를 거쳐 인공위성 계층에 전송된다. 그 다음 인공위성 계층에서는 요청 데이터를 수신 후 분석하여 디바이스 커널 계층으로 현재 위치 데이터를 반환한다. 반환받은 GPS Driver는 현재 위치 데이터를 LBS 계층에 Location Manager로 전송한다. LBS 계층에서 현재 위치 데이터를 요청하여 반환받은 Location Manager는 데이터를 Location Listener로 전달하고, Location Listener는 데이터 정보 중 경도·위도 데이터만 얻어내서 어플리케이션 계층으로 전송한다. 그 후 어플리케이션 계층에서는 사용자가 Geo Code를 이용하여 목적지를 설정하면 그 목적지의 경도·위도 데이터와 사용자의 경도·위도 데이터가 동적 위치 인지 기법에 적용된다. 이 과정을 통해 사용자와 목적지간에 1/2 위치 지점을 얻을 수 있고, 얻어진 지점을 동적으로 인지함에 따라 특정한 변수의 값을 변경시켜 결과적으로 LBS 수신 주기를 커스터마이제이션 할 수 있게 된다.



[Fig. 6] Structure of Dynamic Location Recognition Technique using GPS LBS

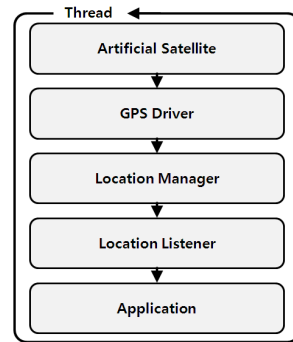
3.2 GPS LBS를 이용한 동적 위치 인지 과정

본 논문에서 제안하는 동적 위치 인지 기법의 과정은 쓰레드를 이용하여 모바일 환경의 사용자가 어플리케이션에서 현재 위치 데이터를 요청하면 GPS LBS를 통해 현재 위치의 경도·위도 데이터를 반환받을 수 있다. 그 후 반환받은 경도·위도데이터는 사용자가 설정한 목적지의 경도·위도데이터와 함께 동적 위치 인지 기법에 사용되어 사용자와 목적지간에 1/2 위치 지점을 추출할 수 있게 된다. 그 후 사용자가 추출된 위치 지점을 인지할 경우에는 쓰레드 주기를 담당하는 변수를 변경시키고, 동적 위치 인지 기법이 재사용되게 된다. 그로인해 사용자와 목적지까지의 1/2, 1/4, 1/8, ..., 1/2ⁿ 위치 지점들을 인지할 수 있고, 이 지점들을 인지할 때마다 쓰레드 주기가 변경되어 결과적으로 LBS 수신 주기를 거리에 비례하게 커스터마이제이션 할 수 있게 된다[Fig. 7].



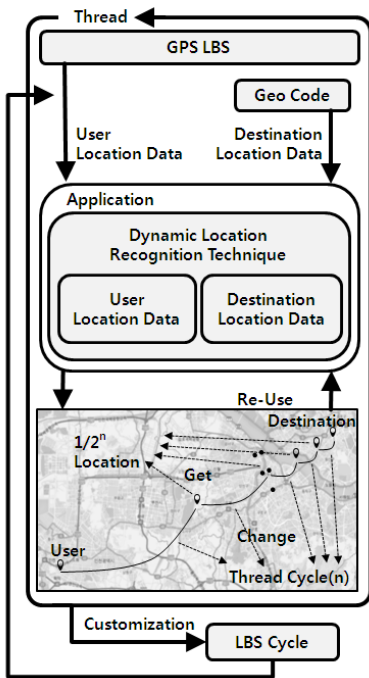
[Fig. 7] Process of Dynamic Location Recognition Technique using GPS LBS

GPS와 모바일 어플리케이션 간에 LBS 데이터 흐름은 사용자가 Location Manager를 통해서 현재 위치 데이터를 요청하면 인공위성으로부터 어플리케이션으로 현재 위치 데이터가 전송된다. 처음 인공위성은 요청데이터를 분석하여 현재 위치 정보를 GPS Driver를 통하여 Location Manager로 전송하고, Location Manager는 Location Listener로 전송한다. 그 다음 Location Listener는 현재 위치 정보 중에 경도·위도 데이터만 추출하여 어플리케이션에 전달하며, 이 과정은 쓰레드를 통하여 반복적으로 발생하게 된다[Fig. 8].



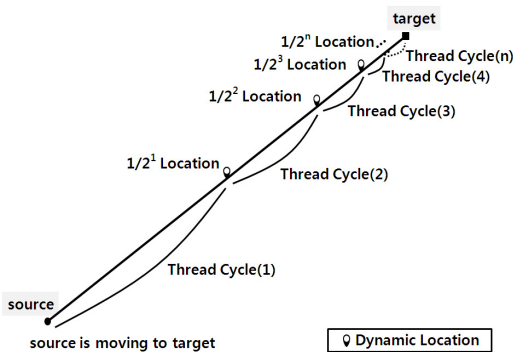
[Fig. 8] User Location Data Flow between GPS and Application

어플리케이션이 경도·위도 데이터를 수신하면 그 데이터는 Geo Code를 통해 설정된 목적지의 경도·위도 데이터와 함께 동적 위치 인지 기법에 사용되어 사용자와 목적지간 1/2 위치 지점을 얻을 수 있게 된다. 그 후, 사용자가 목적지까지 이동하는 중에, 기존에 얻어진 1/2 위치 지점을 인지하게 되면 쓰레드 주기를 담당하는 변수 값이 변경됨과 동시에 동적 위치 인지 기법이 재사용 된다. 이 전체의 과정은 쓰레드를 통해 반복적으로 실행되고, 그에 따라 동적 위치 인지 기법 또한 반복적으로 발생된다. 이로 인해 사용자가 목적지에 도착하기까지 1/2ⁿ 위치 지점들을 인지할 수 있고, 인지된 위치 지점들을 기준으로 쓰레드의 주기를 담당하는 변수값이 변경 된다. 결과적으로 사용자가 목적지까지 이동 중인 상태에서 남은 거리에 비례하게 LBS 수신 주기를 커스터마이제이션 할 수 있게 된다[Fig. 9].



[Fig. 9] Data Flow of User Location Data and Destination Location Data in Application

지금까지 설명한 동적 위치 인지 기법의 핵심 개념은 Fig. 10과 같고, 이것을 크게 둘로 나눈다면 사용자와 목적지간 1/2ⁿ 지점들을 얻는 부분과 얻어진 1/2ⁿ 위치 지점들을 인지하여 쓰레드 주기를 변경시키는 부분으로 구분된다.



[Fig. 10] Core Concept of Dynamic Location Recognition Technique

사용자와 목적지간에 1/2ⁿ 위치 지점들을 얻는 공식은 다음 Fig. 11과 같으며, 사용자의 현재 위치값(source_(x,y))

에 목적지의 위치값(target_(x,y))과 현재 위치값(source_(x,y))의 차이를 반으로 나눈 값을 더하면 된다.

$$\text{between source and target } 1/2^n \text{ Location} = \text{source}_{(x,y)} + \frac{|\text{target}_{(x,y)} - \text{source}_{(x,y)}|}{2}$$

[Fig. 11] Formula for 1/2ⁿ Location

얻어진 1/2ⁿ 위치 지점들을 인지하여 쓰레드 주기를 변경하는 공식은 피타고라스의 정리를 이용하였고, 다음 Fig. 12와 같다. 이것을 설명하자면, 목적지와 현재 위치 사이에서 경도데이터의 차이(target_(x)-source_(x))와 위도데이터의 차이(target_(y)-source_(y))를 각각 제곱하여 서로 더한 값에 루트연산을 하면 된다.

$$\text{between source and target Thread Cycle}(n) = \sqrt{(\text{target}_{(x)} - \text{source}_{(x)})^2 + (\text{target}_{(y)} - \text{source}_{(y)})^2}$$

[Fig. 12] Formula for Thread Cycle

3.3 GPS LBS를 이용한 동적 위치 인지 코드

첫 번째로 사용자는 모바일 환경으로 GPS LBS를 사용하기 위하여 AndroidManifest.xml 파일에서 퍼미션을 설정하여야 한다[Fig. 13].

```
< uses-permission android:name =
"android.permission.ACCESS_FINE_LOCATION" />
```

[Fig. 13] Permission Code for GPS LBS Use at AndroidManifest.xml

그 다음으로는 Location Manager로 LBS를 사용한다고 설정한 뒤, 그 LBS를 requestSingleUpdate 메소드를 통하여 GPS를 이용하기 위한 프로바이더를 설정한다. 그 후 Location Listener의 onLocationChanged라는 메소드와 getLongitude, getLatitude라는 메소드를 통해 현재 위치의 경도·위도 데이터를 double형 변수에 받아들일 수 있다[Fig. 14].

```
public void User_Location_Information() {
    LocationManager locationManager =
        (LocationManager)context.getSystemService
        (Context.LOCATION_SERVICE);
    locationManager.requestSingleUpdate
        (LocationManager.GPS_PROVIDER,
        locationManager, null);
    locationManager = new LocationListener() {
        public void onLocationChanged
        (Location location) {
            double userLon = location.getLongitude();
            double userLat = location.getLatitude(); } } }
```

[Fig. 14] Code for User Longitude Data and Latitude Data Reception

이렇게 사용자의 현재 경도·위도 데이터를 얻은 다음에는 Geo Code를 이용하여 원하는 목적지의 명칭을 'destination'이라는 변수에 넣어서 Http Get 방식으로 목적지의 정보를 탐색할 수 있다[Fig. 15].

```
public void Destination_Location_Information_First() {
    HttpGet httpGet = new HttpGet("http://maps.
    google.com/maps/api/geocode/json?address=
    "+destination+"&ka&sensor=false"); }
```

[Fig. 15] Code of Destination Data Access through Http Get

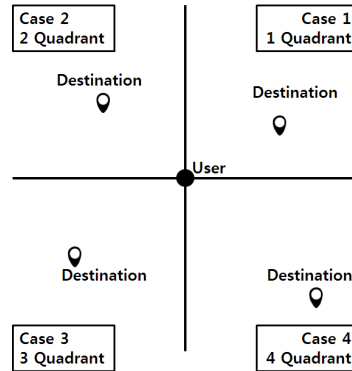
그리고 read()메소드를 통하여 정보를 받아온 후에 경도·위도 데이터를 JSON으로 파서시키면 double형 변수에 넣어 활용할 수 있게 된다[Fig. 16].

```
public void Destination_Location_Information_Second() {
    while((geolInfo = stream.read()) != -1) {
        stringBuilder.append(char geolInfo); }
    jsonObject = new JSONObject(stringBuilder.toString());
    double desLon = ((JSONArray) jsonObj.get("result")).
    getJSONObject(0).getJSONObject("geometry").
    getJSONObject("location").getDouble("lng");
    double desLat = ((JSONArray) jsonObj.get("results")).
    getJSONObject(0).getJSONObject("geometry").
    getJSONObject("location").getDouble("lat"); }
```

[Fig. 16] Code for Destination Longitude Data and Latitude Data Reception through read() and JSON

그 다음 단계로는 사용자의 경도·위도 데이터와 목적

지의 경도·위도 데이터를 동적 위치 인지 알고리즘에 적용시켜 사용자와 목적지간 1/2 위치 지점을 얻어와야 한다. 이때 현재 사용자의 위치를 기준으로 목적지가 위치할 경우는 2차원적으로 우상(1사분면), 좌상(2사분면), 좌하(3사분면), 우하(4사분면)로 총 4가지가 있다[Fig. 17].



[Fig. 17] Case of Destination will be located centered on User Location

이렇기 때문에 사용자와 목적지간 1/2 위치 지점을 얻을 수 있는 코드는 Fig. 18의 4개의 경우를 고려하여 4개의 if문으로 되어있고, 그 if문에 따라서 1/2 위치 지점을 얻어 double형 변수인 'halfLon', 'halfLat'에 저장되는 코드로 이루어져 있다. 또한, 이 코드는 boolean형식의 'DLRTReuseFlag' 변수를 이용하여 기존에 얻어진 1/2 위치 지점을 인지 할 경우에만 재사용되어 1/2ⁿ 위치 지점들을 얻을 수 있게 구현되어 있다.

```
public void DLRT_Half_Location() {
    double halfLon; double halfLat;
    if(DLRTReuseFlag == true) {
        DLRTReuseFlag = false;
        if(userLon < desLon && userLat < desLat) {
            halfLon = userLon + ((desLon - userLon) / 2);
            halfLat = userLat + ((desLat - userLat) / 2); }
        if(userLon > desLon && userLat < desLat) {
            halfLon = userLon - (((desLon - userLon) / 2) * -1);
            halfLat = userLat + ((desLat - userLat) / 2); }
        if(userLon > desLon && userLat > desLat) {
            halfLon = userLon - (((desLon - userLon) / 2) * -1);
            halfLat = userLat - (((desLat - userLat) / 2) * -1); }
        if(userLon < desLon && userLat > desLat) {
            halfLon = userLon + ((desLon - userLon) / 2);
            halfLat = userLat - (((desLat - userLat) / 2) * -1); } }
```

[Fig. 18] Code for 1/2ⁿ Location

```

public void DRLT_Thread_Cycle() {
    int distance = (int)Math.sqrt(Math.abs((desLon-lon)
        *(desLon-db_longitude))+Math.abs((desLat-db_
        latitude)*(desLat-db_latitude)));
    int threadCycle;

    if(userLon < desLon && userLat < desLat) {
        if(userLon > halfLon && userLat > halfLat) {
            DRLTReuseFlag = true; threadCycle = distance; } }
    if(userLon > desLon && userLat < desLat) {
        if(userLon < halfLon && userLat > halfLat) {
            DRLTReuseFlag = true; threadCycle = distance; } }
    if(userLon > desLon && userLat > desLat) {
        if(userLon < halfLon && userLat < halfLat) {
            DRLTReuseFlag = true; threadCycle = distance; } }
    if(userLon > desLon && userLat > desLat) {
        if(userLon > halfLon && userLat < halfLat) {
            DRLTReuseFlag = true; threadCycle = distance; } } } }
    
```

[Fig. 19] Code for Thread Cycle

이러한 코드를 통해 얻어진 1/2 위치 지점은, 사용자가 1/2 위치 지점을 인지함에 따라 쓰레드 주기가 변경되는 코드에 사용된다. 이 코드 또한 사용자의 위치를 기준으로 목적지가 위치할 경우를 고려하여 4개의 if문으로 되어있고, 그에 따라 1/2 위치 지점을 인지할 수 있게 되어있다. 이렇게 1/2 위치 지점이 인지됨에 따라 동적 위치 인지 기법이 재사용되어 1/2ⁿ 위치 지점들이 얻어질 수 있도록 'DRLTReuseFlag' 변수의 값이 'true'로 변경되며, 쓰레드 주기 변수도 피타고라스의 정리를 이용한 'distance' 변수의 값을 통해 거리에 비례하게 값이 변경되어진다[Fig. 19].

지금까지 설명했던 코드들은 하나의 쓰레드에 포함되어 있으며, 그 쓰레드는 위 코드에 의하여 변경되는 쓰레드 주기에 따라 반복적으로 발생하게 된다[Fig. 20].

```

public void run() { super.run();
    while(true) {
        try { Thread.sleep(threadCycle); }
        catch (InterruptedException e) { } }
    handler.post(new Runnable() {
        public void run() {
            User_Location_Information();
            Destination_Location_Information_First();
            Destination_Location_Information_Second();
            DRLT_Half_Location();
            DRLT_Thread_Cycle(); } } } }
    
```

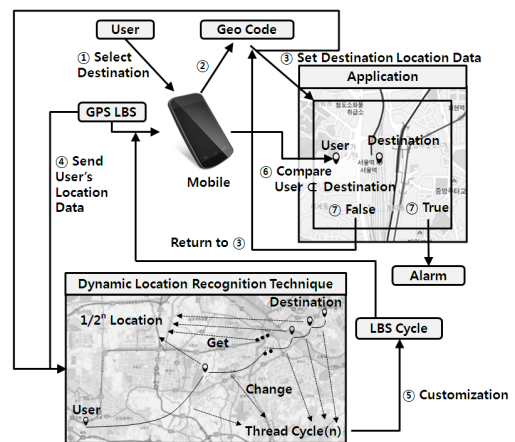
[Fig. 20] Thread Code of Dynamic Location Recognition Technique

그 결과, 사용자는 자신의 위치와 목적지 사이에서 1/2ⁿ 위치 지점들을 인지할 수 있고, 그 위치들을 인지함에 따라 쓰레드의 주기를 변경 시킬 수 있다. 그로인해 결과적으로 거리에 비례하게 LBS 수신 주기를 커스터마이제이션 할 수 있게 되고, 더 나아가서는 디바이스 자원의 낭비를 줄이거나 중간 지점 알람 서비스 등의 다양한 분야에서 활용이 가능할 것이며, 그 결과로 조금 더 효율적인 위치 기반 서비스 사용이 가능하게 된다.

4. 실험 및 평가

본 논문에서 제안한, 위치 서비스를 위한 동적 위치 인지 기법은 사용자가 버스에 탑승하여 어플리케이션으로 하차할 목적지를 설정하면, 목적지 근처에 도착하였을 때 알람이 울리는 어플리케이션을 통하여 적합성을 검증한다.

사용자가 광역버스에 탑승 후 원하는 목적지를 설정하면 Geo Code를 통하여 목적지의 위치데이터를 얻게 되고, 그와 동시에 GPS LBS를 이용하여 사용자의 현재 위치 데이터를 반환받는다. 다음으로는 사용자의 위치 데이터와 목적지의 위치 데이터가 동적 위치 인지 기법에 적용되며, 그로인해 사용자와 목적지간 1/2ⁿ 위치 지점들을 얻을 수 있게 된다. 그리고 사용자가 목적지까지 이동하는 중에 얻어진 1/2ⁿ 위치 지점들을 인지함으로써 GPS LBS 주기가 커스터마이제이션 되어 점점 짧아지게 된다. 이렇게 위치에 따라 커스터마이제이션 되는 GPS LBS 주기를 기반으로 계속해서 현재 위치 데이터를 수



[Fig. 21] Data Flow of Bus Alarm Application

신하여, 목적지 위치데이터와 비교하여 목적지 근처에 도착하였을 경우에는 알람이 울리는 어플리케이션이다. 결과적으로 사용자와 목적지간 거리에 비례하여 GPS LBS 주기가 커스터마이제이션 되며, 그로인해 모바일 자원의 낭비를 줄일 수 있게 된다[Fig. 21].

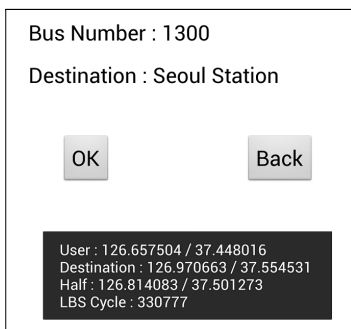
본 논문에서 실험할 버스 어플리케이션의 시나리오는 다음과 같다. 사용자는 인하공업전문대학에서 서울역행 1300번 버스를 탑승하여, 버스 알람 어플리케이션의 목적지를 서울역으로 설정한다. 그렇게 되면, 사용자가 목적지까지 이동 중에 1/2ⁿ 위치 지점들을 얻을 수 있게 되고 얻어진 1/2ⁿ 위치 지점을 지나게 될 때마다 GPS LBS 수신 주기가 커스터마이제이션 된다. 결과적으로 목적지에 가까워질수록 GPS LBS 수신 주기가 짧아지고, 사용자가 목적지에 도착 하였을 경우에는 알람이 울리게 되는 시나리오이다.

첫 번째로 사용자는 버스를 탑승한 후, 어플리케이션의 목적지 선택 액티비티에서 탑승한 버스의 번호와 하차할 목적지를 선택할 수 있다[Fig. 22].



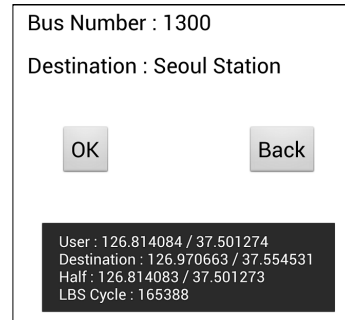
[Fig. 22] Destination Selection Activity

목적지 선택이 완료되면 GPS LBS를 통해 현재 위치 데이터를 수신하고, 수신한 현재 위치 데이터와 설정한 목적지 데이터를 기반으로 동적 위치 인지 기법이 실행된다. 따라서 1/2ⁿ 위치 지점과 GPS LBS 주기가 셋팅되고, 이 데이터들은 어플리케이션의 Toast Message 기능을 통하여 확인할 수 있다[Fig. 23].



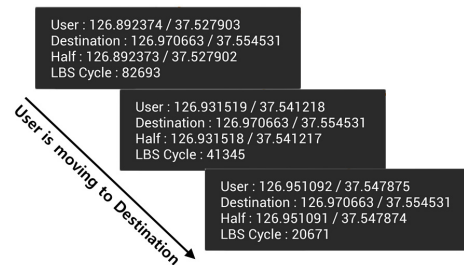
[Fig. 23] Use of Dynamic Location Recognition Technique

이로서 사용자는 1/2ⁿ 위치 지점을 얻게 되었고, 330777(약 5분 30초)의 GPS LBS 주기로 현재 위치 데이터를 계속해서 수신하게 된다. 그 후 1/2ⁿ 위치 지점을 지나게 되면 동적 위치 인지 기법이 재사용되어 1/2² 위치 지점을 얻게 되고 GPS LBS의 주기는 165388(약 2분 40초)로 더 짧아지게 된다[Fig. 24].

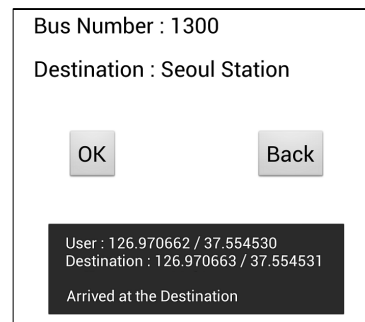


[Fig. 24] Reuse of Dynamic Location Recognition Technique

이런 과정을 통하여 사용자가 목적지에 가까워질수록 GPS LBS 주기는 점점 더 짧아지게 된다[Fig. 25].



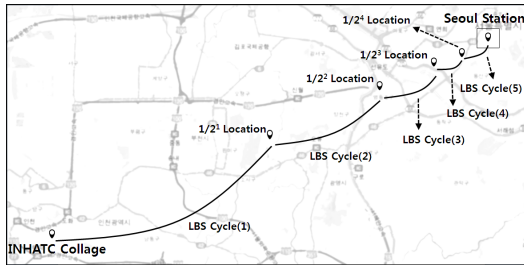
[Fig. 25] LBS Cycle Change according to User Location



[Fig. 26] Toast Message when User arrived at the Destination

그리고 사용자가 목적지 근처에 도착하게 되면 알람이 발생되며, 어플리케이션이 종료된다[Fig. 26].

다음은 위 실험의 시나리오를 도식화 한 것이다[Fig. 27].

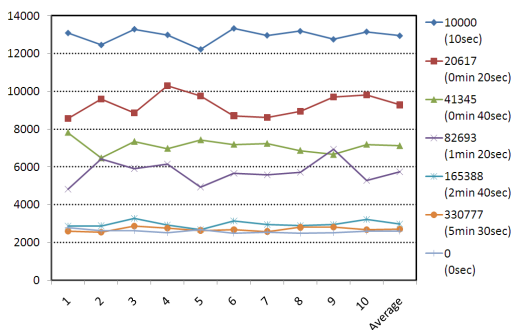


Location	Longitude	Latitude
Inhac College	126.657604	37.448016
1/2° Location	126.814083	37.501273
1/2° Location	126.892373	37.527902
1/2° Location	126.931518	37.541217
1/2° Location	126.951091	37.547874
Seoul Station	126.970663	37.554531

	Cycle	(Time)
LBS Cycle(1)	330777	(5min 30sec)
LBS Cycle(2)	165388	(2min 40sec)
LBS Cycle(3)	82693	(1min 20sec)
LBS Cycle(4)	41435	(0min 40sec)
LBS Cycle(5)	20671	(0min 20sec)

[Fig. 27] Scenario of Bus Alarm Application

다음은 이클립스의 Android Trace Check 기능을 이용하여, 위 실험에서 추출된 LBS 주기에 따른 CPU 사용량 그래프이다. LBS 주기를 기준으로 20초간의 CPU 상태를 10번씩 조사하였고, 주기가 짧을수록 CPU 사용량이 많다는 것을 알 수 있다[Fig. 28].



[Fig. 28] CPU Trace Check according to LBS Cycle

위의 사례를 통해 위치 서비스를 위한 동적 위치 인지

기법의 실현 가능성을 검증하였으며, 제안된 연구와 Fig. 28을 통해 자원의 낭비를 줄일 수 있는 것 또한 검증하였다.

5. 결론 및 향후 연구과제

본 논문은 위치 기반 서비스를 위한 동적 위치 인지 기법을 제안하였다. GPS LBS를 이용하여 수신한 사용자의 현재 위치와 Geo Code를 이용하여 얻은 목적지의 위치사이에 1/2° 위치 지점을 얻은 뒤, 사용자가 이 지점을 인지하게 되면 특정한 변수의 값을 변경 시킬 수 있다. 이 과정을 쓰레드를 통하여 반복적으로 발생시키고, 그에 따라 얻어진 1/2° 위치 지점들을 기준으로 쓰레드 주기를 담당하는 변수값을 변경 시키면, 결과적으로 1/2° 위치 지점들에 따라 LBS 주기를 커스터마이제이션 할 수 있게 된다. 본 논문에서는 동적 위치 인지 기법을 버스 알람 어플리케이션에 적용하여 적합성을 검증하였고 또한 이를 통해 자원의 낭비를 줄일 수 있는 것을 검증하였다. 향후 모바일 디바이스 자원의 효율성을 고려하여 개발되는 어플리케이션에 이용될 수 있도록 본 동적 위치 인지 기법을 정형화한다.

References

- [1] All-round programmer - Unikys, "[Android App. Development Application] Import and Optimize of Location GPS Location", <http://unikys.tistory.com/283>, 2012.
- [2] Android Developers, "LocationManager", <http://developer.android.com/reference/android/location/LocationManager.html>, 2013.
- [3] Android Developers, "LocationListener", <http://developer.android.com/reference/android/location/LocationListener.html>, 2013.
- [4] Android Developers, "Location", <http://developer.android.com/reference/android/location/Location.html>, 2013.
- [5] Wikipedia, "Geocoding", <http://en.wikipedia.org/wiki/Geocoding>, 2013.
- [6] Kim S.H., Kim J.U., Kim H.J., Park D.G., "A Tour Information System on Smart Phone using Location Based Service", Journal of Korea Multimedia Society Vol.

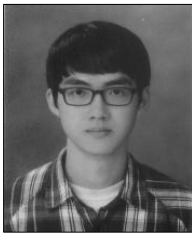
15, No. 5, p:677-691, May 2012.

DOI: <http://dx.doi.org/10.9717/kmms.2012.15.5.677>

- [7] Lee H.S., "A Thesis on Apply Emotion to Location-BasedService", Konkuk University Master Thesis Paper, Dec 2012.
- [8] Jung C.H., Ye J.H., Kim C.J., "A Mobile Customization Technique using Push Service", Journal of the Korea Academia-Industrial Cooperation Society Vol. 14, No. 9, p:4498-4506, Sep 2013.
- DOI: <http://dx.doi.org/10.5762/KAIS.2013.14.9.4498>
-

정 창 훈(Chang-Hun Jung)

[준회원]



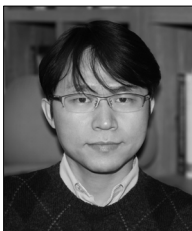
- 2008년 3월 ~ 2014년 2월 : 인하 공전 컴퓨터시스템과

<관심분야>

위치 기반 서비스, 커스터마이제이션, 소프트웨어 공학
데이터베이스, 모바일 서비스, 3D게임

김 철 진(Chul-Jin Kim)

[종신회원]



- 2004년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2004년 3월 ~ 2009년 2월 : 삼성 전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 조교수

<관심분야>

컴포넌트 기반 개발 방법론, 컴포넌트 커스터마이제이션,
모바일 서비스, 클라우드 컴퓨팅