

안드로이드 기반 위치 기록 사진 촬영 시스템

지 현*, 공기석
 한국산업기술대학교 컴퓨터공학
 e-mail : {*ToGGie, kskong}@kpu.ac.kr

Android-based Location Recording Photography System

Hyun Ji, Ki-Sok Kong
 Dept of Computer Engineering, Korea Polytechnic University

요 약

안드로이드 기반의 앱으로, 사진 촬영 시 위치 정보와 메모를 저장할 수 있는 시스템이다. 본 논문에서는 안드로이드에서의 애플리케이션 개발 방법, 카메라에 접근하여 사진 촬영 기능을 구현하는 방법, 위치 정보와 메모를 저장해 둘 수 있는 DB를 구현하고 이용하는 방법, 위치 확인을 위한 맵 호출 방법을 기술하였다.

1. 서론

애플 사의 아이폰 국내 출시를 시작으로 스마트 폰의 공급 속도가 가속화 되었다. 스마트 폰은 GPS, 블루투스, 카메라, 위치 센서, Wi-Fi 등과 같은 다양한 하드웨어 기능을 제공하고, 기존 휴대 전화보다 높은 성능을 바탕으로 다양한 앱을 사용자의 필요에 따라 설치할 수 있게 되었다.

스마트 폰의 열풍 속에서 구글에서는 안드로이드라는 플랫폼을 발표하였다. 오픈된 환경으로 누구나 아무런 제약 없이 쉽게 개발 환경을 갖출 수 있으며, 다양한 스마트 폰 제조사들도 제약이 없는 안드로이드를 채택하여 앞 다투어 다양한 제품들을 출시하였다. 이에 힘입어 최근 안드로이드 폰 중 하나인 삼성의 갤럭시S는 출시된지 70일만에 100만대를 돌파했다[1].

많은 사용자들이 안드로이드 스마트 폰을 사용하고 있고, 이를 이용하여 사진을 찍는 기회가 많아졌다. 사진 촬영 시 GPS를 이용하여 위치 정보를 함께 저장하면 촬영된 사진들의 위치를 언제라도 다시 확인할 수 있다. 이에 추가로 사진과 연관된 메모를 함께 남길 수 있다면, 사진에 좀 더 다양한 정보를 함께 포함시킬 수 있게 된다. 본 논문에서는 사진 촬영, 위치 정보 및 메모 저장, 구글 맵에서 사진 촬영 위치 확인의 기능을 갖춘 안드로이드 앱을 구현하는 방법을 기술하였다.

2. 관련 연구

2.1 기존 유사 앱과의 비교

본 앱 개발을 하면서 관련 유사 앱을 찾아본 결과, 몇 가지 유사한 앱을 확인할 수 있었고, 그 중 현재 아이폰에

서 제공되는 PhotoGeo라는 앱이 위치 정보를 기록, 확인을 할 수 있는 동일한 기능을 제공하고 있었다[2].

<표 1> PhotoGeo와의 비교

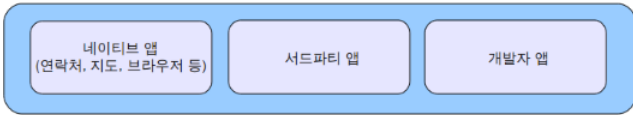
	Photo Geo	연구과제
작동환경	iPhone OS	Android
공통점	- 촬영된 사진의 위치를 구글 맵에 표시한다. - 맵에 촬영된 사진의 이미지가 함께 표시된다.	
차이점	- 촬영된 사진의 날짜와 시간이 맵에 함께 표시된다. - 사진과 사진의 위치를 email로 전송이 가능하다.	- 사진 촬영 시 사진과 관련된 메모를 남길 수 있다. - 프로그램 내의 갤러리에서 메모 확인 가능

위의 <표 1>을 보면 PhotoGeo는 해당 사진과 위치 정보를 함께 e-mail로 전송할 수 있는 기능을 제공하고 있다. 이와는 다르게 본 앱은 사진 촬영 시 사진과 관련된 메모를 함께 저장할 수 있고, 후에 이 메모를 위치 정보와 함께 확인이 가능한 기능을 제공한다는 점이 다르다.

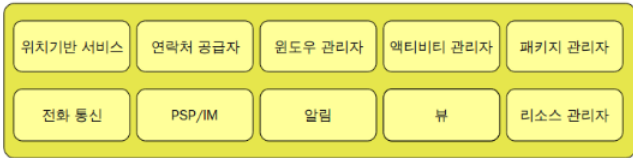
2.2 안드로이드 플랫폼[3]

안드로이드 플랫폼은 구글에서 개발한 모바일 용 운영 체제로서, 리눅스 커널 기반의 오픈 플랫폼이다. 사용 언어는 JAVA이지만, 전통적인 JAVA VM(Virtual Machine)을 사용하지 않고, Dalvik VM을 사용한다. Dalvik VM은 단일 장치에서 복수의 인스턴스가 효율적으로 실행되게끔 보장하도록 설계된 안드로이드 고유의 커스텀 VM이다. 하드웨어와 시스템 서비스에 대한 모든 접근은 Dalvik을 중간 단으로 사용해 관리됨으로, 개발자는 특정 하드웨어 구현에 대해 걱정할 필요가 없게 되었다.

애플리케이션 계층



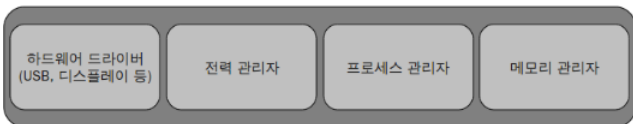
애플리케이션 프레임워크



라이브러리



리눅스 커널



(그림 1) 안드로이드 스택 구조

안드로이드는 (그림 1)과 같은 스택 구조를 갖는다. 리눅스 커널은 하드웨어 드라이버, 프로세스와 메모리 관리, 보안, 네트워크, 전력 관리 등의 핵심 서비스를 담당한다. 라이브러리는 커널 바로 위에서 동작하며, libc와 SSL 같은 다양한 C/C++ 코어 라이브러리뿐만 아니라 다양한 라이브러리를 포함한다. 애플리케이션 프레임워크는 애플리케이션을 만들는데 사용되는 클래스를 제공하며, 애플리케이션 계층은 애플리케이션 프레임워크를 통해 사용할 수 있는 클래스와 서비스를 사용함으로써 안드로이드 런타임 내에서 실행된다.

3. 안드로이드를 이용한 앱 개발 방법[3]

안드로이드 개발이 가능한 OS는 마이크로소프트 윈도우 XP 이상, Apple 맥OS 10.4.8 이상, 리눅스가 가능하다. 개발에 필요한 준비물은 안드로이드 SDK(Software Development Kit)[4], JDK(Java Development Kit)[5], IDE(Integrated Development)가 필요하다. 안드로이드 SDK는 안드로이드 개발자 홈페이지에서 받을 수 있고, JDK는 썬 홈페이지에서 받을 수 있다.

안드로이드 개발에 쓰는 IDE는 ADT(Android Development Tools)를 지원하는 오픈 소스 IDE인 Eclipse[6]를 사용하며, Eclipse 재단 홈페이지를 통해 받을 수 있다. Eclipse에 ADT를 설치하면 안드로이드 SDK를 이용하여 안드로이드 앱 개발이 가능한 환경이 갖춰진다.

4. 세부 설계 및 구현

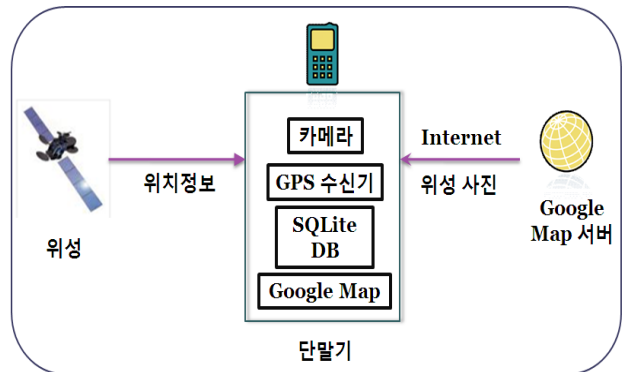
4.1 개발 환경

- 개발 OS -Windows XP SP3
- 개발 PC 사양 -Intel CoreDuo U2500, 2G ram
- 개발 폰 -Nexus One 2.2
- 안드로이드 SDK 2.1 기반
- Eclipse에서 Google Map 2.1 API 사용
- 사용 언어는 JAVA

(그림 2) 개발에 사용된 하드웨어와 소프트웨어 본 앱을 개발하기 위한 개발 환경은 위 (그림 2)과 같이 구축하였다.

4.2 시스템 구성

시스템 구성은 아래의 (그림 3)와 같이 크게 위치 정보를 보내주는 위성, 단말기인 안드로이드 용 스마트 폰, 구글 맵 서버로 분류 된다.



(그림 3) 시스템 구성도

이 앱을 사용하여 스마트 폰에 있는 카메라로 촬영을 하고 사진을 저장하게 되면, 동시에 내장된 GPS 수신기를 이용하여 위성으로부터 위치 정보를 받아와 메모와 함께 이 내용을 DB(DataBase)에 저장한다. 이 저장된 사진, 메모, 위치 정보는 본 앱의 갤러리에서 확인 가능하며, 사진이 촬영된 위치를 확인할 때 구글 맵을 사용하게 되는데 구글 맵 서버에 해당 위치의 위성 사진과 맵을 요청하여 보여주게 된다.

4.3 사진 촬영

사진을 촬영하려면 우선 카메라 변수를 선언하여 카메라를 제어하고, 카메라의 프리뷰를 띄우기 위하여 SurfaceView를 이용한다. 그리고 위치 정보를 함께 받아와야 하기 때문에 GPS에 접근하여 아래 (그림 4)처럼 GPS의 정보를 수시로 갱신해야 사진을 찍는 순간의 위치 정보를 기록할 수 있다.

```
public class cameragps extends Activity implements SurfaceHolder.Callback
{
    private void Updatelocation(Location location)
    //GPS의 위치 정보를 갱신하는 역할

    public void surfaceCreated(SurfaceHolder holder)
    //카메라 화면이 비춰질 surfaceview를 생성

    public void surfaceDestroyed(SurfaceHolder holder)
    //surfaceview를 소멸

    Button inbutton = (Button)findViewById(R.id.inbutton);
    inbutton.setOnClickListener(new Button.OnClickListener()
    public void onClick(View v)
    {
        //사진 촬영 시 위치 정보, 이미지 저장, DB 저장 처리
    }
}
```

(그림 4) 카메라와 GPS를 사용하기 위한 클래스
 사진이 촬영되는 이벤트가 발생하면 이미지 파일을 기본 저장소인 SD Card에 JPEG형식으로 저장을 하게 된다. 이 때 저장되는 파일명은 사진이 촬영되는 시간으로 저장을 한다. 동시에 DB에 위치 정보, 메모가 입력된다.

4.4 갤러리

갤러리는 촬영된 사진들을 볼 수 있는 기능을 제공한다. 아래 (그림 5)를 보면 저장소에서 저장된 이미지들의 목록을 가져오고, 그 이미지 목록들을 갤러리에 보내주게 되고 저장소에 있는 이미지 파일들이 미리보기 식으로 보여지게 된다. 이 목록 중 하나의 이미지를 선택하게 되면 해당 이미지를 화면에 표시를 해주게 된다. 이와 동시에 선택된 이미지와 연결되어 있는 위치 정보와 메모를 파일명으로 DB에서 검색하여 읽어오게 되고 메모의 내용은 이미지와 같이 출력해준다.

```
public class gallery extends Activity
{
    private List<String> ReadSDCard()
    //저장소에 있는 파일들의 목록

    gridview.setOnItemClickListener(new OnItemClickListener()
    //선택된 사진 확대 출력

    public class ImageAdapter extends BaseAdapter
    //갤러리를 위한 Adapter 선언
    }

    public class imagezoom extends Activity
    {
        //확대된 이미지와 메모 출력
    }
}
```

(그림 5) 이미지 목록을 받아와 보여주기 위한 클래스
 갤러리에서 이미지를 선택했을 경우, 폰의 메뉴 버튼을 누르게 되면 이 이미지에 관련된 위치 정보를 확인할 수 메뉴와 현재 이미지와 위치 정보, 메모를 삭제하는 삭제 메뉴를 사용할 수 있다.

4.5 위치 확인

갤러리에서 사진을 촬영한 위치 확인 요청이 들어오면 안드로이드에서 기본적으로 제공하는 구글 맵을 이용하게 된다.

```
public class mapactivity extends MapActivity
{
    mymap = (MapView)findViewById(R.id.mapview1);
    mapcon = mymap.getController();
    //맵, 컨트롤러 선언

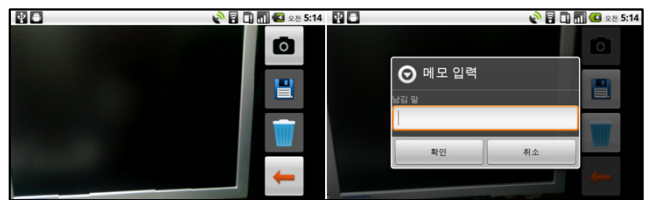
    poverlay = new MyPosition();
    //맵에 오버레이로 현 위치를 표시
}

public class MyPosition extends Overlay
{
    //표시될 오버레이를 구현
}

public class dbAdapter
{
    //DB를 생성, 제어하기 위한 기능 구현
}
```

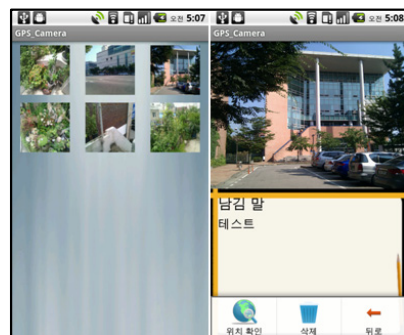
(그림 6) 맵 호출과 오버레이 표시하기 위한 클래스
 맵을 호출하기 위해 MapView를 이용하여 맵을 호출하게 된다. 호출된 맵을 DB에 입력되어 있는 위치로 이동시키고 맵을 컨트롤하기 위해 (그림 6)과 같이 맵 컨트롤러를 선언한다. 호출된 맵 위에 현재 위치 마킹과 미리 보기용 이미지를 그려줘야 하므로, Overlay를 이용하여 맵에 표시되는 미리보기 이미지와 현재 위치를 마킹을 맵 위에 레이어로 그려주게 된다.

4.6 개발 결과 및 분석



(그림 7) 사진 촬영 화면

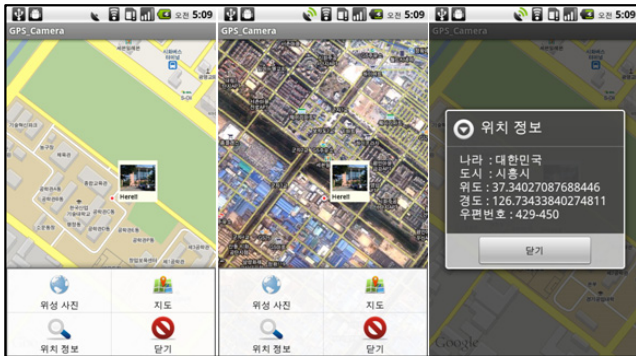
사진 촬영을 하게 되면 (그림 7)과 같이 사진을 저장함과 동시에 메모를 기록하게 된다. 메모 내용을 기록하게 되면 DB에 파일명과 위치 정보, 메모가 전송되어 기록하게 된다. 이 때, 저장되는 파일명은 사진이 촬영되는 시간으로 기록된다. DB에 기록된 정보들은 사진들을 다시 확인할 때 위치 확인과 메모 확인을 위해 쓰이게 된다.



(그림 8) 갤러리 작동 화면

갤러리에서는 촬영된 사진들을 확인할 수 있다. (그림 8)을 보면 첫 화면은 사진 목록이 보여지게 된다. 목록 중에서 보고 싶은 사진을 선택하면, 선택된 사진의 파일명을 DB에서 검색하여 해당 사진에 남겨진 메모 내용과 사진

을 볼 수 있게 된다. 이 상태에서 안드로이드 폰의 메뉴 버튼을 누르게 되면 현재 사진을 삭제할 수 있는 삭제 버튼과 사진이 찍힌 위치를 맵에서 확인할 수 있는 위치 확인 버튼, 목록으로 되돌아가는 뒤로 버튼을 사용할 수 있다.



(그림 9) 맵을 호출하여 위치를 확인하는 화면

갤러리에서 위치 확인 버튼을 누르게 되면 DB에서 해당 파일명을 검색하여 일치하는 위치 정보를 불러오게 된다. 이렇게 불러온 위치 정보는 맵을 불러올 때, 해당하는 위치로 맵을 이동시켜 (그림 9)와 같이 사진을 찍은 위치를 표시해준다. 맵 화면에서 메뉴 버튼을 누르게 되면 위성 사진과 지도로 맵 레이어를 변경할 수 있고, 위치 정보 버튼을 누르게 되면 지오코더를 이용하여 해당 위치 정보의 주소를 보여주게 된다.

결과물을 테스트를 해본 결과, 갤러리에서 사진들이 많아지게 되면 원하는 사진을 찾기가 힘들어지게 된다. 이를 보완하기 위해서 갤러리에서 사진의 주소나 메모, 촬영된 날짜를 기준으로 검색을 할 수 있도록 하여 찾는 사진을 좀 더 빠르게 찾을 수 있도록 보완이 필요하다고 생각된다.

5. 결론 및 향후 연구 과제

이 연구를 통해 개발된 앱을 이용하게 되면 많은 사람들이 휴대하고 다니는 스마트 폰을 이용한 사진 촬영이 단순히 사진만 촬영하여 저장하는 것이 아닌 위치 정보와 메모를 함께 남기게 됨으로써 사진에 이미지와 함께 추가적인 정보를 기록할 수 있게 된다. 사용 예를 들어보면 여행을 다니면서 이 앱을 이용하게 되면 여행지에서의 찍은 사진의 위치와 메모가 남아, 여행지에 대해 보다 더 다양하고 풍부한 남길 수 있게 된다.

본 앱을 개발하면서 안드로이드 OS가 탑재된 스마트 폰에서 카메라 제어, GPS 접근 제어, DB 생성 및 제어, 파일 입출력, 갤러리를 구현했다.

후에 갤러리에서 촬영된 사진의 양이 늘어났을 경우를 대비해 원하는 이미지를 좀 더 빠르고 쉽게 찾기 위해 사진이 촬영된 날짜나 위치 정보, 혹은 메모의 내용을 바탕으로 검색이 가능하도록 연구가 계속될 것이다.

6. 참고 문헌

[1] 파이낸셜 뉴스, “갤럭시S’ 70일 만에 100만대 판매 신기록”, 2010.09.01.

[2] PhotoGeo, “<http://www.chainlink.com/photoGeo/>”

[3] 리토마이어 (조성만 역), 프로페셔널 안드로이드 애플리케이션 개발, 제이펍, 2009.07

[4] 안드로이드 SDK(Software Development Kit), “<http://developer.android.com/sdk/>”

[5] JDK(Java Development Kit), “<http://java.sun.com/javase/downloads/index.jsp>”

[6] Eclipse, “<http://www.eclipse.org/downloads/>”