

안드로이드 환경에서 사용자 소리세기를 이용한 위기대처 시스템

이태경, 김민서*
동국대학교 컴퓨터 공학부
e-mail: pydsk-2002@nate.com

Crisis coping system using the user's voice loudness in android environment

Tae Kyung Lee, Min Seo Kim*
Dept. of Computer Engineering, DongGuk University

요 약

본 논문에서는 긴급통화만 가능하던 기존의 위기대처시스템들과 차별화를 두기 위해, 안드로이드에서 제공하는 시스템 중 하나인 미디어 부분을 이용하여 사용자의 음성을 입력받아 소리세기를 출력시켰다. 또한 GoogleAPI를 활용하여 현재위치를 찾아 긴급 메시지로 전송 가능한 시스템을 제공한다. 본 시스템은 사용자의 편리성과 효율성을 높이기 위하여, 단 1회 실행만으로도 연속적인 위기대처시스템의 인식과 주기적인 실시간 위치를 찾아 메시지 전송이 가능하다. 부가적으로 현재위치를 위도 경도로만 출력 되어 전송 되어지는 것이 아니라, 위도 경도의 값을 주소로 변환하여 출력 하므로서 보다 정확하고 편리하게 서비스를 이용할 수 있다.

1. 서론

2007년 애플(Apple)이 아이폰(iPhone)을 출시한 후 세계는 바야하로 스마트폰(smartphone)폭풍속에 들어있다[4]. 국내에서 스마트폰 가입자가 대중화의 기준점, 500만을 넘어선 것은 2010년 10월이다. 본격적인 스마트폰 대전의 1차전이라 할 수 있는 삼성전자의 '옵니아2'와 애플의 '아이폰'이 2009년 말에 출시되었던 것을 감안하면, 스마트폰은 가히 폭발적으로 성장했다고 할만하다[3].

이에 따라 전국적으로 무선인터넷망이 분포되어 있어 언제 어디서나 스마트폰 하나로 인터넷을 무료로 사용할 수 있게 되었다. 이로 인하여 자신이 원하는 정보를 보다 쉽게 얻을 수 있으며 활용할 수 있기 때문에 스마트폰을 이용하여 어플리케이션을 상용화 또는 개발하고 있는 추세이다[4].

본 논문에서는 사용자의 편리성과 효율성을 높이기 위하여, 단 1회의 실행만으로도 연속적인 위기대처시스템의 인식과 주기적인 실시간 위치를 찾아 메시지 전송이 가능하도록 하였다. 이 뿐만 아니라, 사용자의 음성을 입력받아 소리세기를 출력시켜 기존에 설정한 시스템의 크기보다 크면 위기 대처 시스템이 작동, GoogleAPI를 이용하여 GPS, 3G, Wi-Fi를 이용하여 현재위치를 찾아 위급상황시 전송 가능하게 한다.

본 논문의 제 2장에서는 스마트 시스템 운영체제의 이론적 배경, 시스템에서 사용되는 Open API, GPS, 전화통신과 음성인식에 대하여 설명, 제 3장에서는 사용자 소리세기를 이용한 위기대처 시스템의 구조 및 설계 개념을 설명, 제 4장에서는 실험 결과를 제시하며, 끝으로 5장에서는 결론에 대해 언급한다.

2. 이론적 배경

2.1 안드로이드 운영체제

안드로이드는 사용자의 인터페이스를 제공하며 공식적으로 Java언어를 사용하여 응용 프로그램 개발을 지원한다. 기존의 휴대폰 운영체제인 윈도 모바일, 심비안, 아이폰과 차별화 되는 것은 완전 개방형 플랫폼이라는 점을 가지고 있으며, 안드로이드는 기반 기술인 '소스코드'를 모두 공개함으로써 누구라도 이를 이용하여 소프트웨어와 기기를 만들어 판매할 수 있도록 하였다. 하지만 이러한 개방성은 단점으로도 작용할 수 있는데, 예를 들어 구글에서 운영체제를 새로 업데이트 할 때마다 각 제조사 제품에 즉시 적용되지 못한다는 점이 그러하다.

2.2 Open API

현재 서비스 중인 네이버, 다음, 구글 등의 API를 외부에 공개한 것으로 일반적으로 웹서비스형태로 공개한 것으로, 구글에서 제공하는 Open API를 이용하여 이용할 것이다[1].

본 논문에서 Open API 중 Google Map API를 사용하면 자바스크립트를 통해 구글 Map을 웹페이지에 포함 할 수 있기 때문에 Google Map API를 선택하여 현재위치를 찾아 위치를 표시하고 현재의 위치 주소까지 출력할 수 있도록 한다.

2.3 GPS

현재 출시되어 있는 GPS에 관한 서비스를 보면 자동으로 내비게이션의 지도를 업데이트를 해주며, 수도권 및 고속도로의 상황을 실시간을 알수 있게 해준다. 이 기능들

중 현재의 위치를 잡아 지도를 업데이트 해주는 기능을 이용하여, 본 논문에서 OpenAPI와 GPS를 활용하여 자신의 현재위치를 지도상의 표시점으로 표시할 수 있도록 한다.

2.4 전화통신과 음성인식

전화통신을 사용하기 위해서는 전화통신을 위한 안드로이드 API인 TelephonyManager가 필요하다.

TelephonyManager란 통화중 상태, 단말기 전화번호, 기기 식별자, 일련 번호 등을 얻기 위한 설정이다. 이 부분을 사용함으로써 기기의 현재 위치 검색이 가능하다.

<표 1> TelephonyManager Class에서 제공하는 기능

Method	Description
getCallState	휴대폰의 통화상태를 알려줌
getDataActivity	데이터 송신 상태를 알려줌
getDataState	데이터 통신 접속 상태를 알려줌
getSimState	SIM의 상태를 알려줌
getLineNumber	제 1회선의 전화번호를 알려줌

음성 인식 기술은 상당히 이론적 배경을 요구하기 때문에 단순한 음파 분석으로는 구현하기 어렵다. 하지만 안드로이드가 탑재된 기기에서는 음성 인식 기술을 사용할 수 있도록 액티비티가 제공되기 때문에 음성 인식에 대한 특별한 지식이 없더라도 자신의 어플리케이션에서 음성 인식을 사용할 수 있다[6].

본 논문에서는 안드로이드에서 제공하는 음성을 받아들이는 부분을 이용하여 음성의 소리 크기를 출력하여 설정한 음성의 크기 보다 클시, 위기대처 시스템이 작동 하도록 한다.

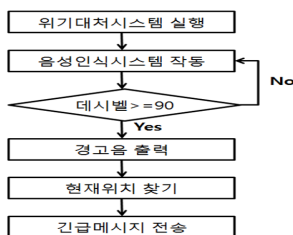
3. 소리세기와 위치정보를 이용한 시스템

3.1 시스템 설계 흐름

기본적인 사용자 소리세기를 이용한 위기대처 시스템은 4가지 시스템으로 이루어진다. 위기대처시스템의 흐름과 비상시스템의 흐름에 대해서 설명한다.

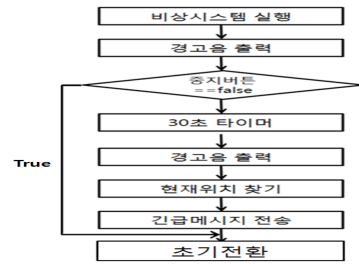
3.1.1 위기대처시스템 흐름

아래의 [그림 1]의 위기대처시스템의 설계 흐름을 보면 위기대처시스템이 실행된 후 순차적으로 음성인식시스템도 작동한다. 음성인식 시스템이 작동되면 자동으로 데시벨을 메시지로 출력시키며 90dB 미만일때, 음성인식시스템이 계속 반복된다. 90dB이상일 경우 조건을 만족하므로 순차적으로 수행되어 경고음을 출력, 현재위치 찾기, 긴급메시지 전송으로 시스템이 종료된다.



[그림 1] 위기대처시스템 설계 흐름

3.1.2 비상시스템 흐름



[그림 2] 비상시스템 설계 흐름

[그림 2]의 비상시스템의 설계 흐름을 보면 비상시스템이 실행된 즉시 경고음이 출력된다. 중지 버튼 또한 표시 되는데 30초 안에 중지버튼을 누르면 비상시스템이 종료된다. 30초안에 중지 되지 않으면 자동으로 현재위치를 찾아 긴급메시지로 현재의 위치를 전송하게 된다.

3.2 음성인식에 따른 시스템 개발

3.2.1 소리세기를 이용한 시스템

이 부분은 안드로이드에서 제공하는 음성인식 API를 이용하여 소리세기를 측정하는 부분이다. 음성을 받는 부분, 음성의 소리크기를 측정하는 부분, 소리의 크기를 알려주는 부분으로 구분되어 수행된다. 음성인식 API Package중 android.media.AudioFormat을 사용한다[6].

<표 2> 소리세기 시스템 부분

```

public int calculatePowerDb(short[] sdata, int off, int samples)
{
    double sum = 0;
    double sqsum = 0;
    for (int i = 0; i < samples; i++)
    {
        final long v = sdata[off + i];
        sum += v;
        sqsum += v * v;
    }
    double power = (sqsum - sum * sum / samples) / samples;
    power /= MAX_16_BIT * MAX_16_BIT;

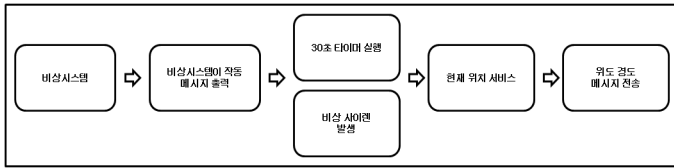
    double result = Math.log10(power) * 10f + FUDGE;
    return (int)result;
}
  
```

<표 2>의 코드는 소리세기를 측정하는 소스의 일부분으로서 소리의 크기를 계산하는 부분이다. 0을 최대 전력, sqsum을 들어오는 신호의 세기를 계속 더해서 더한 최종 값 (sqsum-sum*sum/samples)/samples의 계산식을 이용하여 결과 값을 구한후 power에 임의로 저장한다. 그 후 MAX_16_BIT의 값을 32768, FUDGE의 값 0.6f으로 지정해 주었기 때문에 이 값을 가지고 위의 식처럼 계산 한 후 result로 dB값을 변환하여 TextView를 이용하여 출력 한다.

3.2.2 비상 시스템

이 부분은 비상 시스템으로서 음성인식 API를 사용하지 못할 시 사용 가능 하도록 하였다. 비상 시스템이 수행 될 시 비상시스템이 작동되었다는 메시지, 사이렌 소리와 함께 30초 타이머가 수행되어 30초 타이머 안에 중지 하지

않으면 현재 위치 서비스가 수행되어 비상연락망을 이용한 현재위치의 위도, 경도 메시지가 전송 수행된다.



[그림 3] 비상 시스템

<표 3>의 코드는 비상시스템 소스의 일부분으로서 비상벨을 울리는 부분과 30초 타이머 부분을 설명한다. Intent를 이용하여 MediaPlayer의 비상벨 부분을 울려와 실행 시킨후, 30초 타이머를 작동한다. 위의 소스를 보면 second == 0이 되면, 이 또한 Intent를 이용하여 현재 위치가 전송되는 부분으로 연결된다.

<표 3> 비상시스템 부분

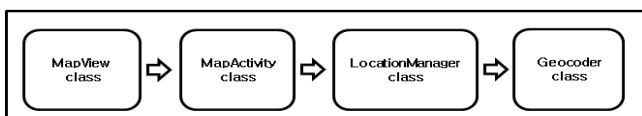
```
fileOfMusic = MediaPlayer.create(DecibelMain.this,R.raw.decibel);
fileOfMusic.setLooping(true);
fileOfMusic.start();

private Runnable doUpdateTimer = new Runnable(){
    public void run(){
        if(second == 0){
            Intent intentmain2_2 = newIntent(Menu2.this,Menu3_sms
startActivity(intentmain2_2);
            Toast.makeText(Menu2.this, "현재위치가 전송됩니다",
            Toast.LENGTH_SHORT).show();
        }
        second--;
        text = (TextView)findViewById(R.id.main2_txt);
        String strFormat = String.format("%02d:%02d:%02d",
            hour,minute,second);
        text.setText(strFormat);
        handler.postDelayed(doUpdateTimer, 1000);
    }
};
```

3.2.3 현재위치 서비스 시스템

안드로이드 SDK가 제공하는 위치 기반 서비스 API를 활용하여 특정한 하드웨어 장치의 제공자를 이용해서 단말기의 현재 위치를 알아내어 수치로 된 위치 좌표를 사용자에게 친숙한 주소나 지도 이미지로 변환하여 출력하여 수행한다.

현재 위치의 정보를 알아내기 위해서는 LOCATION_SERVICE로 getSystemService()를 호출해서 locationManager 인스턴스를 얻는다. 여기서 특별한 권한이 필요하지 않는다. 다음으로 응용프로그램에 필요한 위치 정보의 수준에 따라 적절한 권한을 요청하는 요소를 AndroidManifest.xml 파일에 추가한다. 이 작업은 세밀한 위치정보를 얻을 수 있다. 마지막으로 Geocoder 객체를 이용해 주어진 Location 객체의 수치 좌표들을 주소나 지명으로 변환하여 출력을 수행하여, MapView를 이용한 현재 위치 표시와 현재 위치 주소를 메시지로 전송 수행한다.



[그림 4] 현재위치 서비스 시스템

아래의 <표 4>의 코드를 보면 현재 위치를 찾는 부분의 일부분의 코드이다. geoLat는 경도, geoLng는 위도,

GeoPoint는 현재의 경도와 위도를 찾아 그 위치 부분을 어립잡아 지도를 이용해 보여주는 부분이다. GeoPoint에서 보여주는 부분에서 mapMarkerParams를 이용하여 맵마커로 현재의 상세한 위치를 표시해준다. 다음으로 Geocoder를 위도,경도의 값을 가지고 상세히 주소 표시해주는 부분이다.

<표 4> 현재위치 서비스 시스템

```
public void onLocationChanged(Location location) {
    final double geoLat = location.getLatitude();
    final double geoLng = location.getLongitude();
    GeoPoint newPoint = new GeoPoint((int)(geoLat * 1E6),
        (int)(geoLng*1E6));
    mapControl.animateTo(newPoint);
    mapControl.setZoom(18);
    MapView.LayoutParams mapMarkerParams = new
MapView.LayoutParams(LayoutParams.WRAP_CONTENT,LayoutParams.WRAP
_CONTENT, newPointMapView.LayoutParams.TOP_LEFT);
    ImageView mapMarker = new ImageView(getApplicationContext());
    mapMarker.setImageResource(R.drawable.green_dot);
    map.addView(mapMarker, mapMarkerParams);

    Geocoder geoCoder = new Geocoder(getBaseContext(), Locale.getDefault());
    String Result = "";
    try{
        Iterator<Address>addresses = geoCoder.getFromLocation(geoLat, geoLng,
1).iterator();
        if (addresses != null) {
            while (addresses.hasNext()) {
                Address namedLoc = addresses.next();
                String placeName = namedLoc.getLocality(); //도시
                String featureName = namedLoc.getFeatureName(); // 번지수
                String country = namedLoc.getCountryName(); //나라
                String road = namedLoc.getThoroughfare(); // 동네
                int addIdx = namedLoc.getMaxAddressLineIndex();
                for (int idx = 0; idx <= addIdx; idx++) {
                    String addLine = namedLoc.getAddressLine(idx);
                    Result += String.format("%s",addLine);
                }
            }
            Toast.makeText(getBaseContext(), Result, 2000).show();
            LocText.setText("위도 : "+ geoLat+ "경도 : "+ geoLng+ Result);
        } catch (IOException e) {
        }
    }
}
```

3.2.4 비상연락망을 이용한 메시지 전송

이 부분은 위기대처 시스템이 수행되면 위의 설명과 같이 위치 서비스 시스템에서 수행된 결과 값을 가지고 SmsManager객체의 정적 메서드 getDefault()로 SmsManager 객체를 얻는다. sendMessage() 메시지를 이용하여 위치 서비스 시스템에서 수행된 결과 값을 메시지로 전송 수행한다.

<표 5> 비상연락망을 이용한 메시지 전송

```

public void onLocationChanged(Location location) {
    final double geoLat = location.getLatitude();
    final double geoLng = location.getLongitude();
    GeoPoint newPoint = new GeoPoint((int)(geoLat * 1E6),
        (int)(geoLng*1E6));

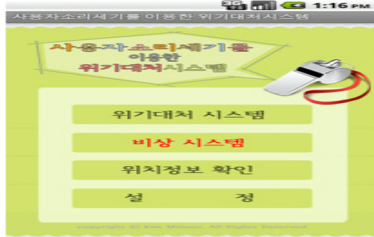
    Intent intent_main3_sms = new Intent(Intent.ACTION_VIEW);
    String smsBody ="위도:"+ geoLat+ "경도:"+ geoLng+ Wn"+
        "http://maps.google.co.kr/?q="+ geoLat+ ", "+ geoLng;
    intent_main3_sms.putExtra("sms_body", smsBody);
    intent_main3_sms.putExtra("address", "");

    intent_main3_sms.setType("vnd.android-dir/mms-sms");
    startActivity(intent_main3_sms)
}
    
```

<표 5>의 코드는 비상연락망을 이용한 메시지 전송 코드 부분이다. <표 4>에서 찾은 위도와 경도의 부분을 가지고 다시 한번 호출하여 위도, 경도, 주소의 값을 임의로 저장하여 기본 제공하는 문자시스템에 전송하여 위도의 값과 경도의 값 지도를 띄어 현재위치를 볼 수 있는 값으로 변환하여 전송한다.

4. 실험결과

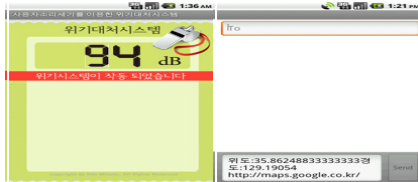
그림 6은 사용자 소리세기를 이용한 위기대처시스템이 실행 될 때 사용자가 시스템을 선택할 수 있도록 하고 있다.



[그림 6] 초기 메뉴 선택화면

4.1 위기대처 시스템 실행

그림 7은 위기대처 시스템을 실험한 결과 이다. 음성인식 API를 이용하여 음성을 받아들인 후 소리의 세기를 바로바로 출력 시켜 보여주는 것을 확인 하였다. 그 이후 설정한 90db이상으로 인해 현재 위치 정보가 전송 되는 것을 확인 할 수 있었다.



[그림 7] 위기대처 시스템 실행화면

4.2 비상 시스템 실행

그림 8은 음성을 이용할 수 없을 경우 사용하는 비상 시스템으로서 비상 시스템을 선택하면 실행 된다는 메시지와 함께 비상 사이렌, 30초 타이머, 중지버튼이 나타나는 것을 확인 했다. 30초 타이머 안에 중지를 시키지 않았더니, '현재 위치를 전송합니다' 라는 메시지와 함께 메시지 전송

창이 출력 되었다. 타이머가 진행 되는 도중에 중지 버튼을 누르면 사이렌이 중지되며 재실행이 가능하도록 대기 상태에 있는 것을 확인 하였다.



[그림 8] 비상 시스템 실행화면

4.3 위치정보 확인

그림 9는 현재의 위치정보를 확인하여 메시지로 전송이 가능하도록 하는 부분이며, 현재 위치의 주소를 출력하는 것을 확인 하였다.



[그림 9] 위치정보 확인 실행화면

5. 결론

본 논문에서는 기존의 위기대처시스템과 차별화를 두기 위해 음성인식API, 현재위치 전송을 사용하여 시스템을 개발 사용자의 편리성과 효율성을 높일 수 있는 실험을 해보았다. 음성API를 이용하여 사용자의 목소리 크기를 측정하여 출력하는 것을 볼 수 있다.

부가적으로 GPS 현재위치 기능을 이용하여 사용자위치를 주소로 변환하여 출력 하므로써 보다 정확하고 편리하게 서비스를 제공한다.

참고문헌

- [1] 손수국 조승호, “안드로이드 프로그래밍의 이해와 실제” 생능출판사 2010
- [2] 우종정, “안드로이드 프로그래밍의 이해”, 생능출판사, 2011
- [3] 김정훈, “안드로이드 프로그래밍”, 성안당, 2009
- [4] 세인 콘더 로런 다시, “시작하세요! 모바일 소프트웨어 개발 안드로이드 프로그래밍”, 위키북스, 2010
- [5] 이준호, “안드로이드 2.2 프로그래밍”, 위키북스 2010
- [6] 안병익 양성봉, “위치기반 정보서비스를 위한 위치조회 방법 및 구조”, 추계학술대회 U-방재국토의 구현, 2006
- [7] 신현빈, “GPS를 이용한 여행 도우미 서비스“, 한국멀티미디어학회 추계학술발표논문집, 2010
- [8] Tricorder, <http://code.google.com/p/moonblink/wiki/Tricorder>
- [9] <http://developer.android.com/index.html>