

# 스마트폰의 가속도계 및 자세계를 활용한 위치예측기 성능 향상 방안

오준환<sup>o</sup>, 김민수, 이영준, 채진석  
인천대학교

{bluecode, minsu, dudwnsbug, jschae}@incheon.ac.kr

## Improvement Performance of Location Predictor Using Accelerometer and Attitude Sensor on Smart Phone

Joonwhan Oh<sup>o</sup>, Min-su Kim, Young-Jun Lee, Jinseok Chae  
University of Incheon

### 요 약

최근의 스마트폰은 UI 또는 엔터테인먼트의 목적으로 가속도계 및 자세계를 제공하고 있으며 이들 정보를 이용한 게임 등의 엔터테인먼트 어플리케이션은 물론 좌표 등의 위치 정보를 통한 위치기반 서비스도 이미 제공중이다. 늘어가는 위치정보 서비스 내에서의 공간정보 질의회수 및 예측회수 증가는 장차 위치 정보 서비스 제공에 있어 서버측 인프라에 부하를 증대시킬 수 있다. 이에 이들 하드웨어가 제공하는 가속도 및 방위각을 활용하여 속도 및 진행방향 정보를 생성하고 이를 위치예측기에 활용하여 연산 속도 및 예측의 정확성 등의 성능을 향상시킬 수 있는 방안을 제시한다.

### 1. 서 론

위치기반 서비스를 효과적으로 제공하기 위한 연구는 여러 방면에서 이루어지고 있다. 위치 정보를 정확하게 추출하기 위한 연구로 GPS, 이동통신망 및 RFID를 이용한 위치 추적 기법 등이 연구되고 있으며, 위치 정보를 효율적으로 가공, 의미있는 정보로 변환하기 위한 기법으로 여러 방식이 제안되고 있다. 특히 노드의 위치 정보를 수집하여 이를 궤적 형태로 만들고, 이로부터 노드의 다음 위치를 예측하는 것은 물류 수·배송 시스템에 있어 차량의 배분의 효율성을 도모하는 등의 그 응용 범위가 크다.

최근 스마트폰이나 PDA폰은 가속도계 및 자세계를 채용. 대부분 스마트폰이나 PDA등에서 동작되는 특성을 고려할 때, 이들 휴대장치에 장착된 가속도계 및 자세계 정보를 이들 위치예측기에 추가 정보로 제공하여 예측의 정확성 및 전체적인 연산 성능을 높일 수 있는가에 대해 논한다.

### 2. 본 론

#### (1) 관련연구

위치 예측에 대해 많은 연구들이 있었다.

대표적인 연구로는 미국에서 개발된 DOMINO[1,2,3]가 있다. 이 시스템은 이동 객체의 현재 위치 정보와 속도, 방향 정보를 이용하여 미래의 위치를 추정하는 기능을 제공한다. 유럽의 CHOROCHRONOS 연구 컨소시엄에서는 시공간 데이터베이스의 특수한 영역으로 이동 객체에 관한 연구가 수행되었으며 특히 이동 객체의 데이터 모델링, 인덱싱, 불확실성 처리에 관한 연구가 주로 진행되었다[4,5,6,7]. 그리고 GPS기반의 수송 관리 시스템과 멀티미디어 시스템을 적용 대상으로 선정하여 응용 시나리오를 제시하였다.

이러한 위치예측기는 질의문에 인자로 위치 정보를 가지고 있으며 추가적으로 속도와 방향 정보를 요구한다.

한편 위치정보만 제공하는 GPS의 특성상 속도와 방향은 필연적으로 수집된 과거 위치정보로부터 연산하여야 하며, 이는 위경도좌표의 투영 변환 등의 복잡한 계산을 통해 얻어지기 때문에 다수의 질의를 동시에 처리할 경우 상당한 CPU자원을 소비할 것을 예상할 수 있다. 따라서 이 연구에서는 스마트폰의 가속도계 및 자세계에서 이들 위치예측기에 유용한 추가 정보를 생성하며, 이 정보가 위치예측기 성능 향상에 기여하는지 알아본다.

#### (2) 안드로이드

##### 1) 소개

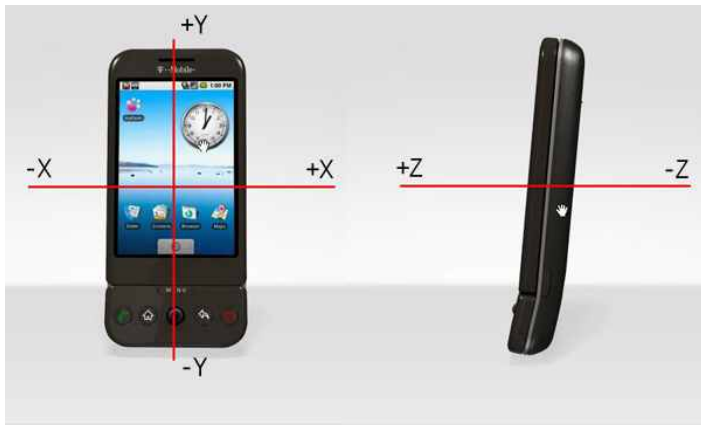
안드로이드는 Google이 공개한 Linux 기반의 개방형

모바일플랫폼으로서, Google이 주도로 2007년 11월에 설립된 다양한 통신사 및 제조사, S/W 벤더가 참여하고 있는 OHA(Open Handset Alliance)에서 개발되었다. 단순히 운영체제(OS)만으로 구성된 것이 아니라, Middleware, UI(User Interface), 인터넷 Browser 및 기타 어플리케이션으로 구성되어 있다.

특히 이 논문에서 주목하는 것은 안드로이드의 하드웨어로, 가속도계 및 자기 센서를 권장하고 있다. 즉, 안드로이드 기반 이동 단말은 이들 센서를 탑재하고 있을 가능성이 높다.

2) 가속도계

그림 1 및 그림 2는 안드로이드의 가속도계가 사용하는 좌표축을 설명하고 있다. 좌표축은 전형적인 왼손 좌표계로서, 각 축의 양수 방향은 각각 단말기 우측, 상방, 디스플레이가 사용자와 가까워지는 방향을 나타낸다.



<그림 1> 가속도계의 좌표축

노드가 피벗(Pivot)되었을 경우, 좌표축도 함께 회전하여 결과적으로 Y축 양수가 하늘을 향하게 된다.



<그림 2> 피벗(Pivot)시 가속도계의 좌표축

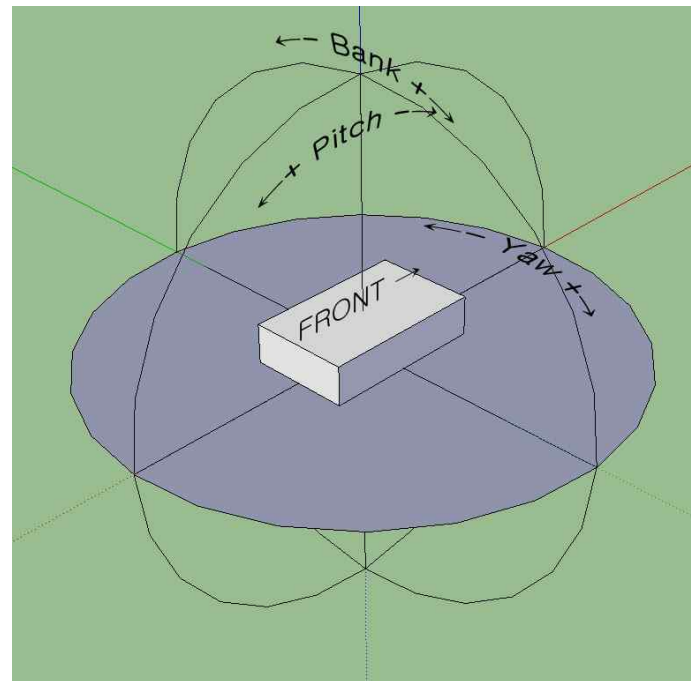
3) 나침반

기본적으로 자세계도 가속도계와 같은 좌표계를 가지고 있으며, 각 축에 대한 회전을 표현한다.

단말기가 책상 위에 바로 놓여졌을 때, 즉 디스플레이가 화면을 향하고 있다고 할 때 다음과 같이 표현된다.

<표 2> 안드로이드의 좌표계 설명

| 축  | 표현    | 범위                          |
|----|-------|-----------------------------|
| X축 | Pitch | -90: 상방이 하향<br>90: 후방이 하향   |
| Y축 | Bank  | -180: 좌측이 하향<br>180: 우측이 하향 |
| Z축 | Yaw   | 0:동 90:서 180:남 360:북        |



<그림 3> 안드로이드의 좌표계 설명

(1) 정보 수집 방안

안드로이드는 SensorManager라는 객체를 통하여 하드웨어 이하 계층을 투명하게 한다. 이 객체는 가속도계, 자세계 및 자기장 센서 등의 정보를 관리하는 객체이며 이 객체를 통해 각 센서의 값을 읽어오게 된다.

1) 가속도계로 속도 구하기

이동 단말의 이동 속도는 가속도의 적분으로 계산한다. 계산에 이용할 가속도는 안드로이드의 가속도계를 이용하여 측정한다.

<표 3> 안드로이드의 가속도계 측정 코드

```
public void onSensorChanged(int sensor,
float[] values) {
    synchronized (this) {
        Log.d(tag, "onSensorChanged: " +
sensor + ", x: " + values[0] + ", y: " + values[1]
+ ", z: " + values[2]);
        if (sensor ==
SensorManager.SENSOR_ORIENTATION) {
            xViewO.setText("Orientation X: "
+ values[0]);
            yViewO.setText("Orientation Y: "
+ values[1]);
            zViewO.setText("Orientation Z: "
+ values[2]);
        }
        if (sensor ==
SensorManager.SENSOR_ACCELEROMETER) {
            xViewA.setText("Accel X: " +
values[0]);
            yViewA.setText("Accel Y: " +
values[1]);
            zViewA.setText("Accel Z: " +
values[2]);
        }
    }
}
```

2) 자세계 정보로 진행방향 구하기

진행 방향은 자세계의 Yaw 출력을 이용하여 단말기가 향한 방위각을 구한다.

<표 4> 안드로이드의 자세계 측정 코드

```
public void onSensorChanged(SensorEvent
event) {
    SensorManager.getRotationMatrix(mR, mI,
mGData, mMData);
    SensorManager.remapCoordinateSystem(mR,
SensorManager.AXIS_MINUS_X, mR);
    SensorManager.getOrientation(mR,
```

```
mOrientation);
    float incl = SensorManager.getInclination(mI);
    if (mCount++ > 50) {
        final float rad2deg =
(float)(180.0f/Math.PI);
        mCount = 0;
        Log.d("Compass", "yaw: " +
(int)(mOrientation[0]*rad2deg) +
" pitch: " +
(int)(mOrientation[1]*rad2deg) +
" roll: " +
(int)(mOrientation[2]*rad2deg) +
" incl: " +
(int)(incl*rad2deg)
);
    }
}
```

3. 시 험

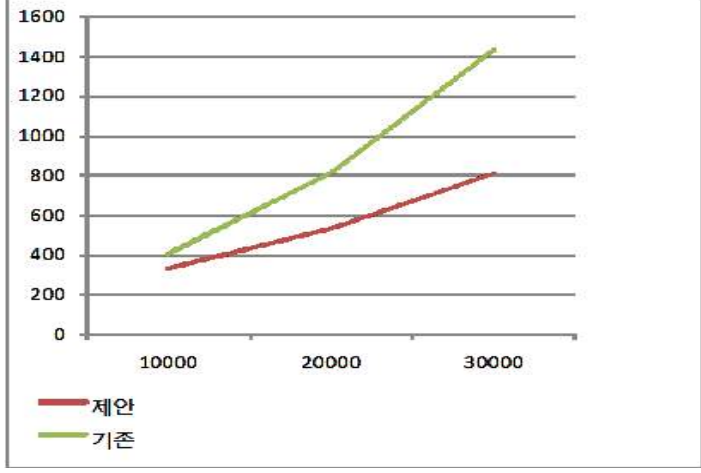
(1) 시험 설계

10,000건, 20,000건, 30,000건의 임의 좌표를 생성하며, 이 때 생성하는 좌표의 범위는 위도 36~38도, 경도 126도~127도로 제한한다.

생성된 3개 영역에 대해 각각의 알고리즘을 5회 반복하여 수행하고, 각 회차의 수행속도를 평균한다.

대조군으로 사용할 위경도 기반 위치 추정 기법에서 두 좌표로부터 방위각 및 거리를 산출하는 수식은 Vincenty's Formula를 사용하였다. 측정 단위는 ms이다.

(2) 알고리즘 시험



<그림 4> 수행속도 비교

10,000건부터 30,000건까지의 3개 영역에 대해 각 알고리즘을 5회 수행한 결과에 대한 평균을 그래프로 나타내었다. 그 결과 가속도와 방향으로부터 다음 위치를 추정하는 것이 좌표쌍을 사용할 때보다 약 50%정도의 성능 향상이 있음을 알 수 있었다.

#### 4. 결 론

본 논문에서는 하드웨어를 적극 활용하여 가속도와 방위각 정보를 추가로 수집하고 이들 데이터를 활용하여 각 단말의 다음 위치를 예측하는 방법을 제안하였다. 제한된 정보만을 활용하여 다음 위치를 예측하는 방법과 하드웨어를 적극적으로 활용하여 가속도 및 방위각 정보를 예측에 활용하는 방법을 비교하였을 때 제안한 방법이 나은 연산 속도를 보이는 것을 확인하였다.

한편 이번 시험은 위치예측기에 입력하기 전 단계인 입력값 생성 단계에서 수행한 시험으로 위치예측기의 성능 향상에 실질적으로 기여하는지에 대한 문제는 해결되지 않았다. 이는 추후 연구에서 실제로 위치예측기에 적용해 볼 필요가 있다고 생각된다.

#### 5. 참고문헌

- [1] P.Sistla, O.Wolfson, S.Chamberlain, and S.Dao, "Modeling and Querying Moving Objects," Proceedings of the Thirteenth International Conference on Data Engineering(ICDE13), Birmingham, UK, Apr. 1997.
- [2] O.Wolfson, S.Chamberlain, S.Dao, and L.Jiang, "Location Management in Moving Objects Databases," Proceedings of the Second International Workshop on Satellite-Based Information Services (WOSBIS'97), Budapest, Hungary, Oct. 1997.
- [3] O.Wolfson, B.Xu, S.Chamberlain, and L.Jiang, "Moving Objects Databases: Issues and Solutions," Proceedings of the 10th International Conference on Scientific and Statistical Database Management (SSDBM98), Capri, Italy, Jul. 1998, pp. 111-122.
- [4] M.Erwig, R.H.Güting, M.Schneider, and M. Vazirgiannis, "Abstract and Discrete Modeling of Spatio-Temporal Data Types," Chorochronos Technical Report, CH-98-14, 1998.
- [5] M.Erwig, R.H.Güting, M.Schneider, and M. Vazirgiannis, "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Objects in Databases," *GeoInformatica* Vol. 3, No.3, 1999, pp.269-296.
- [6] R.H.Güting, M.Böhlen, M.Erwig, C.S.Jensen, N. Lorentzos, M.Schneider, and M. Vazirgiannis, "A Foundation for Representing and Querying Moving Objects," *ACM Transactions on Database Systems*, Vol. 25, No. 1, Mar. 2000, pp. 1-42.
- [7] L.Forlizzi, R.H.Güting, E.Nardelli, and

M.Schneider, "A Data Model and Data Structures for Moving Objects Databases," Proceedings of the ACM SIGMOD Conference, Dallas, Texas, May 2000, pp. 319-330.