옥내측위 안드로이드 앱 프로토타입 설계

임재걸*, 우진석*, 원정완**
*동국대학교 경주캠퍼스 컴퓨터 공학과
**위로드(주)
e-mail:{yim, woo}@dongguk.ac.kr, one@weroad.co.kr

Design of an Indoor Positioning Android App Prototype

Jeageol Yim*, Jinseok Woo*, Jungwan One**
*Dept of Computer Science, Dongguk University
**Weroad(Ltd.)

요 약

마천루를 비롯한 지하상가와 같은 거대한 건축물 내에서 일을 보는 경우가 흔해짐에 따라, 옥내위치기반 서비스에 대한 요구가 크게 증가하고 있는 실정이다. 근래에 판매되고 있는 스마트폰은 기존의 컴퓨터에 버금가는 계산 능력과 기억 용량을 가지고 있으면서 다양한 센서까지 장착되어 있기 때문에, 그리고 스마트폰은 사용자가 항상 휴대하고 다니는 소지품이기 때문에, 위치기반 서비스를 제공하는 장치로는 스마트폰이 제일 적당하다. 위치기반 서비스 구현에 필수불가결한 요소는 당연히 측위 컴포넌트이다. 본 논문은 스마트폰 센서를 이용하여 사용자의 걸음 수와 진행방향을 판정하고, 이를 바탕으로 사용자의 위치를 판정하는 안드로이드 앱을 설계한다.

1. 서론

최근 스마트폰의 이용인구가 급격히 증가하고 이에 따른 스마트폰의 기능들을 이용한 애플리케이션들이 개발되고 있다. 이러한 애플리케이션들 중 스마트폰에 탑재되어 있는 GPS(Global Posotining System) 또는 센서(Sensor)를 이용한 위치기반애플리케이션들의 개발이 증가하고 있다. GPS를 이용한 옥외측위 애플리케이션들은 스마트폰내의 탑재된 GPS의 성능이 발전함에 따라 식별률 및 정확도가 매우 우수하다. 하지만 GPS를 사용할 수 없는 지하철, 아파트, 상가 등의 옥내에서의 위치기반애플리케이션은 연구가 오랜 기간 되었음에도 불구하고 현재로서는 보편적으로 사용할 수 있는 기술이 개발되고 있지 않다.

본 논문은 보편적으로 사용자들이 사용하고 있는 스마트폰에 내장되어있는 센서들을 이용하여 사용자의 걸음수를 카운트하고 이를 바탕으로 사용자의 현재위치를 측위하는 안드로이드 앱을 설계한다. 또한 본 앱은 사용자의 족적과 걸음수를 스마트폰 화면에 출력한다.

2. 관련연구

실내측위에 관한 연구는 다양한 분야에서 다양한 방법으로 연구되어 지고 있다. [1]에서는 무선 네트워크 기반 실내 측위를 위하여 실내에 설치된 AP(Access Point)의 신호세기인 RSSI (Received Signal Strength Indication) 값을 Fingerprinting 방식을 이용하여 현재위치를 측위 하였다. 하지만 [1]에서 사용한 Fingerprinting 방식은 신호를 사전에 수집하는 시간이 측정 대상 지역의 규모와 분할된 셀에 비례하기 때문에 실제 위치 측위 단계 전에 많은 사전작업이 요구되며 수집된 신호는 주변 환경에 대한 특성을 반영하고 있기 때문에 주변 환경이 바뀌면 새로 데이터 수집을 해야 하는 단점이 존재한다.

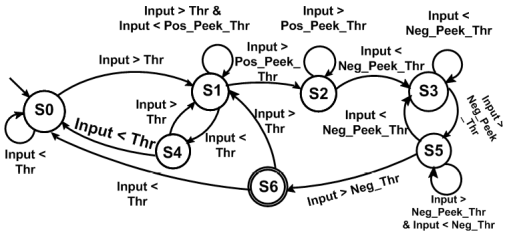
[2]에서는 이동로봇의 목적지에 대한 경로계획을 세우고 자율주행을 하기 위하여 Landmark를 설치하여 이를 이용하여 로봇의 위치를 인식하는 방법이 제안되었다. [2]에서 제안된 방법은 로봇을 위한 방법이므로 사람이 사용할 수 있는 방법과는 다소 거리가 멀다. 또한 측위 하고자 하는 실내에 Landmark를 설치해야 하므로 시간과 비용이 측위공간 크기에 비례하는 단점이 있다.

[3]에서는 물류 창고 내에서의 사용자의 위치를 기존의 노드간의 신호를 이용하여 측정되는 방법의 단점을 보완하기 위하여 RFID(Radio-Frequency Identification)를 사용하는 방법을 제시하였다. 하지만 [4]에서 나타난 사실과 같이 RFID는 태그의 종류에 따라 인식거리의 차이가 심하며 국제규격이 정해져있지 않아 유럽 및 아시아의 주파수 범위가 달라 상호 호환성이 없다. RFID 도입을 반대하는 사람들의 우려도 만만치 않다. 시민 단체들은 RFID를 통해 개인정보가 노출될 것을 걱정 한다. 신분증에 RFID 태그가 붙어 있을 경우, 같은 주파수를 읽는 판독기만 있으면 누구라도 내 정보를 얻을 수 있는 가능성이 있기 때문이다. 또한, 물품 관리에 바코드 대신 RFID를 이용할 경우, 각 물품마다 일련 번호가 붙기 때문에 소비자가 결재 시 태그를 제거하지 않는다면 소비자의 이동 경로를 다른 사람이 추적할 수도 있다.

현재의 RFID기술은 보안기능이 매우 취약하여, 태그 정보 및 센서 노드의 위변조, 위장리더, DoS 공격, 네트워

크에서 개인 추적 정보 유출 등의 위협에 노출되어 있다. 또한 RFID는 관리 대상이 되는 태그 및 노드의 수가 기존의 네트워크보다 월등히 많고 이들에 의해 구성되는 네트워크가 중앙 집중 구조가 아닌 자율분산 구조이므로, 기존의 네트워크보다 훨씬 더 많은 취약점을 가지고 있다.

[5]는 (그림 1)에 보이는 입력 값이 가속기 센서 값인 유한상태자동기계(Finite State Machine)로 걸음을 탐지한다. 그리고 계층 Support Vector Machine (SVM) 분류기를 이용하여 걸음을 세 클래스로 분류한다, 걷기, 조깅, 달리기. 분류 결과에 따라 더욱 정확한 보폭을 결정한다.



(그림 1) 걸음 탐지 유한상태자동기계

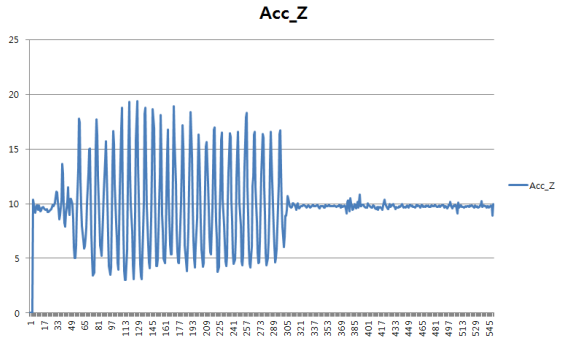
본 논문은 다양하데 보급되어있는 스마트폰을 이용하여 옥내에서의 사용자의 위치를 측정함으로써 초기의 기반시설에 대한 투자비용과 사전정보입력 등의 일련의 과정이 없는 장점을 가지고 있다.

3. 설계

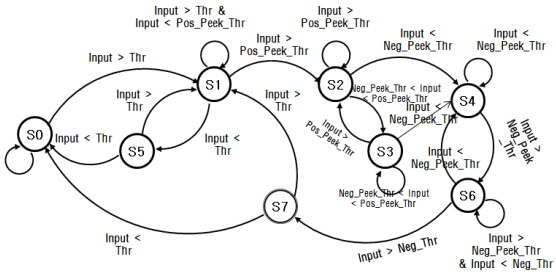
본 논문에서 구현하고자하는 앱은 사용자가 앱을 실행시키고 스마트폰을 들고, 화면을 바라보며 이동하는 것을 전제로 한다. 사용자가 이동시 스마트폰을 바라보며 이동할 경우 스마트폰은 수직으로 흔들리는 움직임을 나타낸다. 이를 이용한 걸음 판단을 위하여 스마트폰의 센서들 중 Z축의 가속도를 나타내는 Accel_Z를 사용한다. (그림 2)는 수직으로 흔들리는 Accel_Z센서 데이터를 보인다. (그림 2)에서 입력되는 센서의 값을 이용한 걸음 판정을 위하여, Thr, Pos_Peek_Thr, Neg_Thr, Neg_Peek_Thr와 같은 Threshold를 사용하며 이는 <표 1>에서 보인다. 걸음을 판정하기 위하여 입력되는 Accel_Z 센서 데이터가 Thr값인 10.0부터 Pos_Peek_Thr값인 11.0이상 상승한 뒤, Neg_Thr값인 9.0와 Neg_Peek_Thr값인 8.0이하까지 하강한다. 그 후 다시 Thr값인 10.0이상으로 값이 상승하면 이를 한걸음으로 판단한다. 이러한 걸음판단을 위하여 (그림 3)과 같은 걸음 탐지 유한상태자동기계를 사용한다.

S0은 사용자가 정지하여 있는 상태를 뜻한다. S1은 사용자가 한발을 떼었을지도 모르는 상황이며, S2는 사용자가 한발을 뗀 상태이다. S3와 S4는 뗀 한발을 땅에 디딘 상태를 의미하며, S6은 사용자가 발을 디딘 후 다시 몸을 일으키는 상태이며, S7은 최종적으로 한걸음을 걸었다는 것을 의미한다.

또한 옥내측위 앱의 순서도는 (그림 3)와 같이 구성된다. 스마트폰에서 앱을 실행하였을 때 먼저 스마트폰에서 사용할 수 있는 센서들을 활성화시킨다. 센서가 활성화된 상태에서 입력된 값을 .CVS파일 형식으로 저장한다.



(그림 2) 스마트폰센서의 Accel_Z의 값



(그림 3) 걸음 탐지 유한상태자동기계

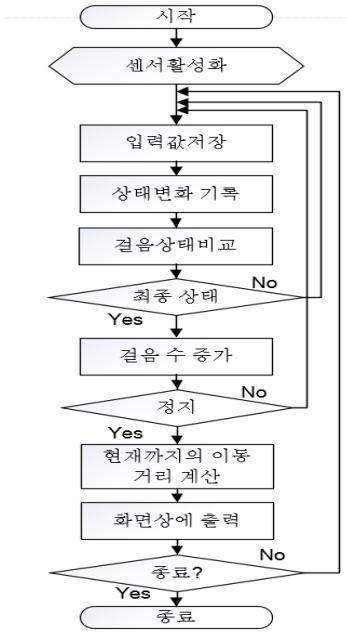


그림 4 옥내측위 앱의 순서도

입력된 센서값을 정해진 Threshhold값을 이용하여 상태변화를 기록한다. 이 상태변화를 기준으로 사용자의 걸음상태를 판단하며, 최종적으로 걸음을 걸은 것이 아니라며 다시 센서값을 받아 입력하는 프로세서로 돌아가고, 걸

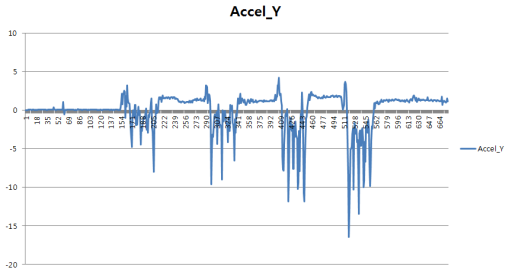
음을 걸은 것으로 판단이 된다면 걸음수를 추가한다. 한걸음을 걸었다고 판단될 경우 사용자의 다음 상태가 정지인지 아닌지를 판단하게 된다. 정지가 아니라면 입력값을 저장하는 프로세서로 돌아가며, 정지했다면 현재까지의 이동거리를 계산하여 화면상에 출력한다. 최종적으로 앱을 종료한다면 모든 프로세서를 종료하여 앱을 종료하고, 종료하지 않고 계속 이용할 경우 입력 값을 저장하는 프로세서로 돌아간다.

4. 구현

본 앱은 모바일 OS 중 하나인 안드로이드에서 작동하는 앱으로 Android SDK(Software Development Kit)을 사용하여 구현한다. 안드로이드 OS를 탑재한 모바일기기인 스마트폰에서 센서값을 구하기 위하여 센서 매니저를 생성하고, 각각의 센서를 객체로 생성한 다음 이들을 센서 리스너에 등록한다. 이 센서 리스너들은 센서값들이 변하는 이벤트가 발생할 때 마다 변화되는 센서값을 50mm/s 마다 문자열버퍼에 저장하였다가 엑셀문서에 저장한다. 한걸음을 측정하기 위하여 유한상태자동기계를 사용하며, 상태별 Threshold값을 <표 1> 같이 사용한다.

<표 1> 상태별 Threshold의 기준 값

Name	Thr	Pos_Peek_Thr	Neg_Thr	Neg_Peek_Thr
Value	10.0	11.0	9.0	8.0



(그림 5) 스마트폰센서의 Accel_Y의 값

스마트폰 센서에서는 스마트폰의 화면이 위로 향하였을 때 상하의 움직임을 측정하는 자이로센서의 값은 Y이다. 이 Y값을 걸음으로 측정하려면 Y의 움직임을 센서로 읽고 저장하여 이동 중인지 또는 정지인지를 판단한다. 이때, 이동 중인 경우는 핸드폰이 상하로 크게 흔들리며, 정지일 경우 상하의 움직임이 작을 것이다. 이러한 값의 변화는 (그림 5)에서 보이는 바와 같다.

스마트폰내의 센서값이 상승하는 경향과 하강하는 경향을 인지하여 (그림 3)에 보이는 걸음 탐지 유한상태자동기계와 <표 1>에 보이는 상태별 Threshold의 기준값의 이용하여 각각의 Threshold값과 센서의 입력값인 Input값을 비교하여 S0부터 S7까지 수행되었을 경우 한걸음으로 판단하는 알고리즘은 <표 2>에 기술되었다.

사용자가 정지하였을 때 출발지점에서 현재 정지지점까지의 거리와 위치를 측정하기 위해서 ori_x와 초기출발지점의 좌표인 sX와 sY를 사용한다. 처음과정으로 현재의 방위각cur_ori와 기준방위각인 stdOri를 비교하여 사분면 중 어느 사분면인지를 판단한다.

<표 2> 걸음 수 측정 알고리즘

```
if (input > Thr && (status == 0 || status == 4 || status
== 6)) {
        status = S0;
    }
else if (input > Thr && (status == 0 || status == 4 ||
status == 6)) {
        status = S1;
    }
                                .
                                .
                                .
else if (input > Neg_Thr && status == 5 ){
        status = S6;
    }
```

판단된 사분면과 현재의 방위각인 cur_ori의 차이를 계산하여 그 값을 difOri에 대입한다. 기준좌표에서 한걸음 움직인 위치까지의 좌표를 계산하기 위하여 수식1을 사용한다. 수식1에서 550은 한걸음을 걸었을 때의 평균보폭을 뜻하고, 50을 나눈 것은 1cm당 Pixel의 수를 나타내기 때문이다.

$$\cos Val1 = (\cos(radians(difOri)) \times 550)/50$$
$$\sin Val1 = (\sin(radians(difOri)) \times 550)/50$$

수식 1 좌표의 거리계산을 위한 수식

기존의 수학적 사분면의 좌표계산과 안드로이드 핸드폰화면에서의 좌표계산에는 차이가 있다. 기존의 사분면에서는 (0,0)의 좌표가 왼쪽하단으로 설정되어 사용되고 있다. 하지만 안드로이드 핸드폰화면내의 (0,0)의 좌표는 왼쪽상단을 기준으로 설정되어 사용되고 있기 때문에 <표 3>과 같은 차이를 가지게 된다. sX와 sY는 각각 스마트폰 화면상의 좌표를 뜻하며, sX와 sY를 각각의 사분면에서 sX는 cosVal1을 sY는 sinVal1을 증가일 경우는 +를 감소일 경우는 -를 계산하여 계산값을 각각 sX와 sY에 대입한다. 이렇게 계산된 sX와 sY는 가각 listdisX배열과 listdisY배열에 저장한다.

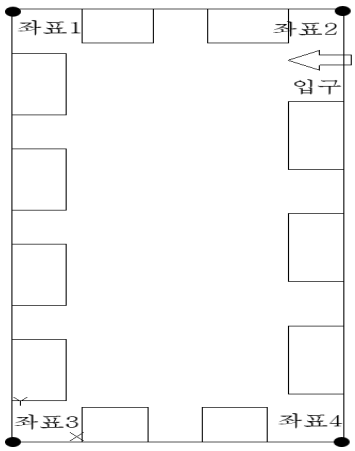
또한 Android의 Canvas기능을 사용하여 핸드폰화면에 정사각형의 Canvas를 그린 뒤 그 listdisX와 listdisY을 각각 읽어 drawcircle을 이용하여 사용자가 정지할때마다 정지하기까지의 걸음수와 족적을 Canvas위에 출력한다.

<표 3> 기존의 사분면 좌표계산과
안드로이드핸드폰화면에서의 좌표계산의 차이

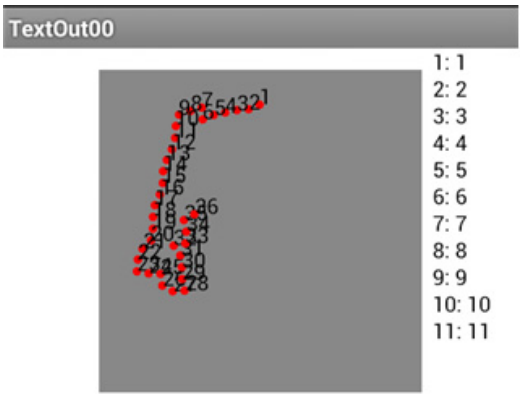
사분면	기존의 사분면		안드로이드에서의 사분면	
	X값	Y값	X값	Y값
1	증가	증가	증가	감소
2	증가	감소	증가	증가
3	감소	감소	감소	증가
4	감소	증가	감소	감소

5. 실험

본 옥내측위 앱은 옥내에서의 사용을 목적으로 하고 있기 때문에 현재 동국대학교 자연과학대학 4층에 위치한 S402강의실을 (그림 6)과 같이 배치하여 박물관으로 가정하여 실험하였다. 본 평면도는 오토캐드로 작성되었으며, 좌표1에서 좌표2까지와 좌표3에서 좌표4까지 의 실제 측정거리는 6m10cm이며, 좌표1에서 좌표3까지, 좌표2에서 좌표4까지의 실제 측정 거리는 12m 72cm이다.



(그림 6) 오토캐드도면



(그림 7) 실험결과화면

실험은 (그림 6)에 보이는 입구에서부터 시계 반대방향으로 사용자가 진행하여 다시 입구로 나가는 경로로 실험하였고, 평면도에서 보이는 사각형들은 박물관내의 전시물이라고 가정한 뒤 각각의 전시물 앞에서 한번 씩 정지하여 실험하였다. 실험의 결과 화면은 (그림 7)에서 보인다. (그림 7)에서 붉은 점들은 사용자의 족적을 나타내며 붉은 점의 옆에 나타난 숫자들은 걸음수를 의미한다. 화면의 오른쪽의 표시된 숫자들은 사용자의 정지 횟수를 나타낸 것이다. 실험 시 실험장소의 전시물이라고 가정한 11곳의 장소에 각각 정지하며 한바퀴 도는데 필요한 걸음수는 29걸음이다. 하지만 방향을 전환하는 부분에서의 움직임이 실제걸음보다 2번 또는 3번 가량 추가적으로 계산되어 실험 시 36걸음이 측정되었다. 50번의 실험결과로 80%의 정확도를 보였다.

6. 결론

본 논문은 스마트폰 센서 값을 이용하여 옥내에서의 걸음수를 측정하고 이를 화면에 보여줌으로써 사용자의 위치를 측정하는 방법을 제안하였고, 실험결과를 보였다. 스마트폰내의 센서들이 전자기기들의 영향을 받았을 경우나 다양한 스마트폰기기간의 센서 정확도 차이에 의하여 사용자가 걸어간 방향 또는 경로 등이 잘못 측정 될 수 있다. 이를 위하여, 측정 방위각의 가중평균을 이용하는 방법, 칼만필터를 이용하는 방법, 그리고 파티클 필터를 이용하는 방법 등을 적용할 예정이다. 또한 정지 시에 정지한 위치의 정보를 제공하거나 동영상을 보이는 등의 앱을 개발할 예정이다. 이러한 다양한 연구로, 궁극적으로 옥내에서 사용자의 위치를 정확히 판정하는 스마트폰 앱을 개발하고, 나아가서 실용적인 옥내위치기반 서비스를 제공하는 스마트폰 앱을 개발할 예정이다.

사사

본 논문은 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(C0033172)의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

[1] 이장재, 권장우, 정민아, 이성로 “실내 측위 결정을 위한 Fingerprinting Bayesian 알고리즘” 한국통신학회논문지 제35권 제6호(네트워크 및 서비스), 2010.6, 888-894 (7 pages)

[2] 이재영, 채의성, 유원필 “이동로봇을 위한 IR랜드마크 기반의 실시간 실내 측위 시스템” 제어로봇시스템학회 논문지 제12권 제9호, 2006.9, 868-875 (8 pages)

[3] 노귀용, 송진국, 정창렬 “u-창고관리를 위한 RFID 테그 기반의 위치 보정 기법” 한국정보학회 논문집, 제14권 제2호, 2009.2. 149-155(7pages)

[4] http://ko.wikipedia.org/wiki/RFID

[5] M. Alzantot, M. Youssef, "UPTIME: Ubiquitous Pedestrian Tracking using Mobile Phones," IEEE Wireless Communications and Networking Conference 2012, pp. 3204-3209