

옥내측위 실시간 보정 알고리즘

임재걸*, 정승환**

*동국대학교 컴퓨터멀티미디어학과

**동국대학교 전자계산학과

e-mail: {yim, honourj}@dongguk.ac.kr

Real Time Correction Algorithm for Indoor Positioning

Jaegol Yim*, Seunghwan Jeong**

*Dept of Computer and Multimedia, Dongguk University

**Dept of Computer Science, Dongguk University

요 약

본 논문은 지도정보를 이용하여 옥내측위 결과를 보정하는 방법을 제안한다. 다양한 옥내측위 관련 연구 결과가 소개되었으나, 지금까지 발표되고 진행되는 연구는 측위의 정확도를 제고하는 방안에 중점을 두고 있다. 그러나 정확도를 아무리 제고하여도 오차를 완전히 제거하기는 불가능하다. 따라서 본 연구는 지도정보를 이용하여 옥내측위 결과를 보정하는 방법을 제안하는 것이다. GPS로 예측한 자동차 궤적을 보정하는 방법으로 map-matching 방법이 널리 연구되었다. 제안하는 방법은 두 선분이 교차하는지 검사하는 함수를 이용하여, 이동단말기가 장애물을 통과하여 움직이는 상황을 나타내는 측위 결과를 실시간으로 즉시 보정한다는 점에서 map-matching 방법과 다르다. 제안하는 실시간 보정 방법은 기존의 map-matching 방법과 함께 측위의 정확도를 제고하기 위하여 사용될 수 있다.

1. 서론

위치기반 서비스(LBS: Location Based Service)는 사용자의 현재 위치를 파악하여, 사용자의 위치를 고려한 유용한 정보를 제공하는 서비스를 일컫는다. 위치기반 서비스에는 첫째, 사용자가 지정한 혹은 사용자의 위치에서 가장 가까운 장소, 생산물, 서비스를 찾아주는 디렉터리 서비스, 둘째, 사용자의 위치를 찾아주는 게이트웨이 서비스, 셋째, 주어진 장소의 이름, 거리와 번지, 혹은 우편번호에 대한 지리적 위치를 결정하여 주는 지오코더(geocoder)와 반대로 지리적 위치를 입력받아 완벽한 주소를 찾아주는 역지오코더(reverse geocoder)를 포괄적으로 지칭하는 위치 유틸리티 서비스, 넷째, 모바일 단말기상에 지리적인 정보를 표현하는 서비스를 일컫는 표현서비스, 다섯째, 사용자에게 적당한 경로를 제공하는 경로 서비스, 등이 있다[1].

위치기반 서비스를 개발하려면 필수적으로 사용자의 위치를 파악해야 한다. 사용자의 위치를 파악하는 측위 방법은 옥외용과 옥내용으로 구분되며, 옥외용은 GPS 방법[2]이 일반적으로 사용되고 있다. 옥내용으로 크리켓(Cricket)[3], 액티브 배지(Active Badge)[4], 등 다양한 시스템이 소개된 바 있는데, 기존의 이러한 방법들은 실내의 일정 영역에 적외선 센서를 부착하거나, 천정에 초음파와 RF 신호 발생기를 부착하는 등 특수한 환경을 꾸며주어야 작동한다.

특수 장비를 이용하는 방법은 범용성과 경제성이 미약하므로, 이미 설치된 무선근거리 통신망을 이용한 옥내측위 방법이 근래에 활발히 소개되었다. 무선근거리통신망을 이용하는 옥내측위 방법은 수신신호의 세기를 이용하는 방법[5-8]과 신호의 도착 시각(TOA: Time of Arrival)을 이용하는 방법[9, 10] 등이 있다. 무선근거리통신망을 이용

하는 방법은 오차가 심하다는 약점이 있다. 그래서 지금까지 이루어진 연구는 정확도를 제고하는 목적에 초점을 맞추었다.

그럼에도 불구하고, 무선네트워크 신호의 굴절, 반사, 간섭 등의 이유로 무선근거리통신망을 이용한 옥내측위 방법은 오차가 아직도 비교적 크다. 따라서 본 논문은 지도 정보를 이용하여 무선근거리통신망을 이용한 측위 결과를 보정하는 방법을 제안한다. 제안하는 방법은 선분의 교차를 검사하는 함수를 이용하여 이동단말기가 장애물을 통과하였다는 측위 결과가 감지되는 즉시 이를 수정한다. 선분 교차 검사 과정의 처리 시간을 단축시키기 위하여 측위 영역을 작은 영역으로 구분함으로써 검사할 장애물의 수를 줄이는 전략을 적용한다.

측위 결과를 보정하는 후처리 방법에 대한 연구가 GPS를 이용한 자동차 항법 시스템에서 이미 널리 연구된 바 있다. 알려진 후처리 방법은 일반적으로 map-matching 방법인데 일련의 측위 결과로 자동차의 궤적을 구하고, 궤적과 거리와 모양이 가장 가까운 도로를 도로망 지도에서 찾아 자동차의 궤적을 그 도로로 정합시킨다. 이 방법은 궤적을 입력 데이터로 사용하는 반면에 제안하는 방법은 측위 결과를 실시간으로 보정하기 때문에 처리 속도가 빠르다. 따라서 실제 응용시스템에서는 일정 기간 동안 본 논문이 제안하는 실시간 보정 방법을 적용하다가 map-matching 방법을 한번씩 적용함으로써 더욱 정확한 궤적을 얻을 수 있을 것으로 사료된다.

2. 관련연구

다양한 옥내 측위 시스템이 발표되었다. 예를 들면, 크리켓[3] 시스템에서는 고정 지점에 부착된 비컨(beacon)들이 고주파 신호와 초음파 신호들을 동시에 송신하면 이동물체에

부착된 수신기가 이 신호들을 수신하고, 신호들의 도착 시각의 차이를 바탕으로 거리들을 구하고, 비컨들의 좌표와 거리를 바탕으로 자신의 좌표를 구한다.

액티브 빔지[4] 시스템에서는 이동물체들에 적외선을 발사하는 송신기가 부착되어 있고, 방마다 여러 개의 수신기가 고정 지점에 설치되어 있다. 적외선은 벽을 통과하지 못하기 때문에 이동물체가 어느 방에 들어오면 그 방에 설치된 고정 수신기들만 그 이동물체의 적외선을 감지할 수 있다. 수신기는 중앙컴퓨터에 연결되어 있어서 수신기에 감지된 적외선을 바탕으로 이동물체가 어느 방 안에 있다는 것을 판정한다. 이상에서 본 바와 같이, 크리켓이나 액티브 빔지 시스템에서는 측위를 위한 특수 장비가 필요함을 알 수 있다. 이외에도 특수 장비를 이용하여 사용자의 옥내 위치를 파악하는 다양한 연구 결과가 발표된 바 있다.

사용자가 어디에 있는지 인터넷을 사용할 수 있도록 하기 위하여 무선근거리통신망이 거의 어느 곳에서도 제공된다. 따라서 무선근거리통신망을 이용하여 사용자의 위치를 판정할 수 있다면 옥내 측위를 위한 특수 장비가 필요 없으므로 경제적이고 구현하기에 편리하다. 무선근거리통신망을 이용한 옥내측위 방법이 [5-10]에 소개된다. [5-8]은 AP(access point)의 신호의 세기를 바탕으로 이동단말기와 AP간의 거리를 구하고, 이 거리를 바탕으로 이동단말기의 위치를 구한다. [5-7]은 정확도를 제고하기 위한 방법을 소개하는데 반하여, [8]은 처리 속도를 개선하는 방법을 소개한다. AP 신호는 간섭과 반사 등의 영향을 받음으로 정확도가 비교적 낮다는 단점이 있다. 그래서 신호세기가 아닌 신호가 전송되는데 걸리는 시간으로 거리를 구하는 방법이 [9-10]에 소개된다.

[9]는 AP 신호가 도착하는데 걸리는 시간(TOA)을 이용하여 거리를 구하는 방법을 소개한다. 이 방법을 이용하려면 정확도가 높은 시계가 장착되어야 하고 물리계층에 가까운 계층의 통신을 해야 한다. 이러한 요구를 만족하도록 laptop에 보조적으로 설치할 수 있는 하드웨어를 소개하고 이동단말기와 AP간의 거리를 구하는 실험결과를 소개한다.

[10]은 TOA를 이용하여 이동단말기와 AP 사이의 거리를 구하고, 이것을 Kalman Filter[11]에 적용하여 이동단말기를 추적하는 방법을 소개한다. TOA 방법이 신호세기(RSSI) 방법보다 더 정확하기 때문에 본 논문이 제안하는 방법은 RSSI를 이용하는 기존의 방법보다 더 정확할 것이라고 예상할 수 있다. 나아가서 Kalman filter의 인수들의 값을 통계치를 적용하여 배정하고, 예상 단계에서 이동단말기가 움직이는 방향과 최근의 속도를 이용하는 방법을 이용하여 평균 오차가 1m 내외인 추적 방법을 본 논문은 제안한다.

추적의 결과를 지도 정보를 이용하여 보정하는 방법이 자동차 항법 시스템 개발을 위하여 이미 널리 연구되었다. [12]는 최근의 연속적인 GPS 포인트 몇 개를 이용하여 직선을 계산하고, 이것을 현재 route라고 한다. 비슷한 방법으로 과거의 route도 구한 다음, 이들의 각도를 비교하여 차이가 적으면 가장 가까운 직선도로로 매핑하고, 아니면 주행각도를 교정하는 알고리즘을 제안한다.

센서 처리장치로부터 읽은 위치벡터들을 합하여 관찰 궤적을 생성한다. 관찰궤적은 일련의 원소들인데 각 원소는 (position, heading, speed)로 구성된다. 관찰궤적에서 직선 세그먼트들을 추출하고, 이것과 가장 가까운 도로세그먼트를 찾아 현재 주행 중인 도로로 결정하는 방법이 [13]에 소개된다.

데이터를 입력받아 자동차의 위치, 방향, 도로상의 시설물 등에 대한 가설들과 확률을 설정하고, 이 가설들을 각각 거나 결합하는 과정을 거쳐 궤적을 구하는 map-matching 방법이 [14]에, 추적오류를 고려한 map-matching 작업의 새로운 모형을 개발하고, 추적 데이터 오류를 증명 가능한 정확한 방법으로 이용하여 계산 결과로 얻은 궤적의 질을 보증하는 빠른 map matching 방법이 [15]에, 디지털 도로 지도와 프로그램이 계산한 궤적간의 일반적인 offset을 계산하고, 프로그램이 계산한 현재 궤적에서 offset을 제거하여 현재 궤적으로 결정하여 주는 방법으로 교차로 내에서 자동차의 정확한 위치를 구하는 알고리즘이 [16]에 소개된다.

Simple Distance Map Matching (SDMM) 전략은 측위 결과와 가장 거리가 가까운 도로를 찾아 교정함으로써 궤적과 수직인 도로상의 점으로 교정 결과를 보이는 오류를 나타낸다. 또한 평행인 도로가 여럿일 때에는 엉뚱한 길로 교정하는 결과를 보이기 쉽다. 이러한 SDMM 전략의 단점을 극복하기 위하여, [17]은 지금까지의 궤적을 바탕으로 가능한 미래의 경로들을 생성하고 GPS 데이터와 이들과 비교하여 궤적을 결정하는 map-matching with look-ahead라는 방법을 제안한다.

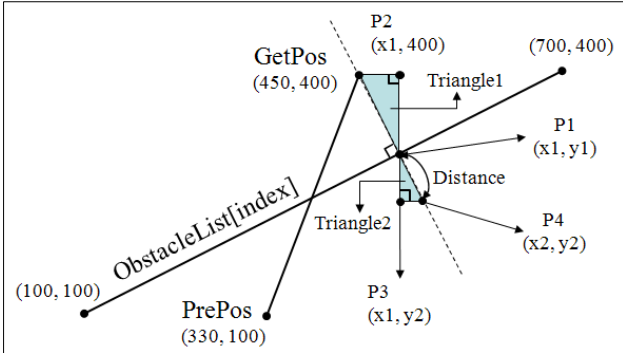
초기의 map-matching은 측위결과로 얻은 점을 가장 가까운 도로로 이동하여 주는 방법으로 간단하고 빠르지만 오류가능성이 높은 단점이 있었다. 현재의 map-matching은 여러 차례의 측위결과로 얻은 궤적을 가장 가까운 도로로 정합하는 방법으로 정확도는 제고되었지만 처리 시간이 길어진다. 그래서 본 논문은 간헐적으로 궤적을 이용하는 map-matching을 실시하여 현재의 궤적을 도로로 정합시킨 다음 한동안은 측위 결과로 얻은 현재위치를 이 궤적의 연장선으로 이동하는 방법을 제안한다. 이동 거리는 현재 궤적을 도로로 정합시키기 위하여 이동한 거리로 하고, 이동결과가 장애물을 통과하는 것과 같은 비현실적인 진행을 나타내면 즉시 보정한다.

3. 실시간보정방법

본 논문은 측정치로 구축한 궤적을 관련연구에 소개된 바와 같은 map-matching 방법으로 도로에 정합하는 작업을 간헐적으로 수행하고, 측위점을 실시간으로 보정하는 방법을 제안한다. 따라서 본 절은 실시간 보정 방법을 소개한다.

실시간 보정 방법은 기존의 map-matching 방법으로 측정 궤적을 이동하여 도로에 정합할 때 이동방향과 거리를 오프셋이라는 변수에 보관한다. 그리고 다음 궤적 map-matching 시까지는 측정점을 오프셋만큼 이동하여 실시간으로 교정한다. 측정점을 오프셋만큼 이동한 점을 교정점이라 한다. 실시간 보정 방법은 직전 교정점과 현재 교정점 사이에

장애물이 존재하는 비현실적인 측정 결과도 실시간으로 보정한다. 직전 교정점을 PrePos, 현재 교정점을 CurPos라고 하고, 도면의 선의 시작점과 끝점을 수록한 배열을 ObstacleList[]라 하면, 현재 교정점을 교정하는 방법을 설명하는 방법은 그림 1과 같다.

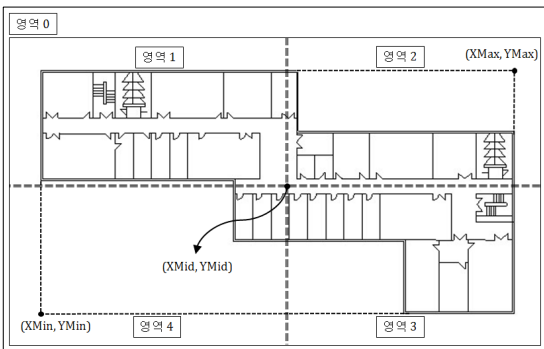


(그림 1) CurPos를 (x2, y2)로 교정함.

그림 1은 PrePos로부터 벽을 통과하여 CurPos로 이동한 결과를 보인다. 그러나 벽을 뚫고 이동하는 것은 불가능한 현상이므로 벽 내부로 CurPos를 재차 교정한다. 제한하는 방법은 교정점에서 벽으로 내린 수직선 방향으로, 벽에서 일정 거리(그림 1에서 Distance라 함) 떨어진 지점을 새로운 교정점으로 결정한다. 벽을 나타내는 직선의 시작점과 끝점, PrePos와 CurPos, 그리고 Distance가 주어지면 교정점을 구하는 방법은 간단함으로 설명은 생략한다.

PrePos와 CurPos 사이에 장애물이 있을 경우에 CurPos가 오류일 확률이 높지만 사용자가 실제로 장애물 저편에 위치할 가능성도 배제할 수 없다. 그래서 제안하는 실시간 보정 알고리즘은 연속적으로 일정 횟수 이상 장애물 저편이라는 판정이 나오면 현재 위치가 실제로 장애물 저편이라고 판정한다.

장애물을 통과하는지 판정하기 위하여 도면상의 모든 직선들과 교점을 구하고, 그 교점이 PrePos와 CurPos 사이에 있는지 판정해야 한다. 이때 비교할 직선들의 수를 줄이기 위하여 도면을 그림 2에 보이는 바와 같이 작은 영역으로 구분한다. 그림 2에서 XMax, XMin, YMax, YMin은 각각 도면을 구성하는 모든 점들의 좌표 값에서 가장 큰 X좌표, 가장 작은 X좌표, 가장 큰 Y좌표 그리고 가장 작은 Y좌표 값이다. 그림 2는 전체 영역을 4 개의 부분영역으로 구분한 예를 보인다.



(그림 2) 도면 지도의 중심점(XMid, YMid)을 기준으로한 각 영역 구분

이상과 같은 실시간 보정 알고리즘을 표 1에 요약한다. 표 1에서 ObstacleList[] 배열은 직선(PrePos, CurPos)가 모두 속한 부분영역의 장애물이 저장된 배열을 의미한다. 예를 들어, PrePos가 영역1에 속하고 CurPos가 영역2에 속하면 ObstacleList[]는 영역1의 장애물과 영역2의 장애물 모두의 리스트이다.

<표 1> 실시간 보정 알고리즘

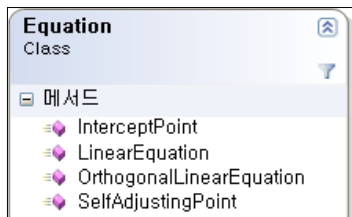
```
Algorithm RT_Adjusting (PrePos, CurPos, ObstacleList[])
/* PrePos는 직전 교정점 (측위점에 오프셋을 더하여 얻은 점), CurPos는 현재 교정점, ObstacleList[] 배열은 도면의 layer들 중에서 직선(PrePos, CurPos)가 내포된 영역의 장애물이 저장된 배열이다.
```

```
직선(PrePos, CurPos)가 ObstacleList[] 배열의 요소 (장애물)와 교차할 경우 보정된 위치를 계산하여 반환하고, 교차하지 않을 경우 CurPos를 반환한다. */
```

1. 직선(PrePos, CurPos)를 구한다.
 2. bIntercept = False
// bIntercept는 직선(PrePos, CurPos)와 직선(ObstacleList[index-1], ObstacleList[index])의 교차 여부를 의미한다. 최초 False(교차하지 않은 상태)로 설정한다.
 3. index=1
 4. loop ((index < size of ObstacleList) and (bIntercept == False))
 - 4.1 직선(ObstacleList[index-1], ObstacleList[index])를 구한다.
// 직선(PrePos, CurPos)와 직선(ObstacleList[index-1], ObstacleList[index])가 교차할 경우
 - 4.2 if (직선(PrePos, CurPos)와 직선(ObstacleList[index-1], ObstacleList[index])의 기울기가 다르다.)
 - 4.2.1 직선(PrePos, CurPos)와 직선(ObstacleList[index-1], ObstacleList[index])의 교점을 구한다.
 - 4.2.2 if (교점의 X 좌표가 PrePos의 X 좌표와 CurPos의 X 좌표 사이에 있다.)
 - 4.2.2.1 bIntercept = True
 - 4.2.2.2 gCntIntercept 증가 // gCntIntercept는 전역변수로 Algorithm을 수행할 때마다 두 직선방정식이 교차하면 값을 1 증가시킨다.
 - 4.2.3 end if
 - 4.3 end if
 - 4.4 index 증가
 5. end loop
 6. if (bIntercept == True)
 - 6.1 if ((gCntIntercept != 0) and (gCntIntercept <= 임계치))
 - 6.1.1 return (그림 1처럼 구한 보정 위치)
 - 6.2 else
 - 6.2.1 gCntIntercept = 0
 - 6.2.2 return (CurPos) // CurPos가 최종 위치이다.
 - 6.3 end if
 7. else
 - 7.1 gCntIntercept = 0
 - 7.2 return (CurPos) // CurPos가 최종 위치이다.
 8. end if
- End Algorithm

4. 구현전략

이동 단말기의 일련의 측위 결과를 바탕으로 칼만필터 방법으로 사용자의 위치를 추적하게 된다. 이 찾은 위치를 직선에 찾은 위치와 비교함으로써 추적 궤적이 장애물을 통과하는지를 판단하게 된다. 이 경우 최종 위치는 본 연구에서 제안하는 실시간 보정 알고리즘을 이용하여 구하게 된다. 실시간보정 알고리즘에서는 주어진 두 점을 지나는 직선의 방정식을 구하는 함수와 두 직선의 교점을 구하는 함수, 그리고 그림 1처럼 보정된 위치를 계산하는 함수들이 사용되었다. 본 연구에서는 이러한 작업들을 처리하기 위해 그림 3과 같은 Equation Class를 작성하였다.



(그림 3) Equation Class의 Diagram

5. 결론

본 논문은 옥내 측위 결과를 보정하는 방법을 제안한다. 제안하는 방법은 기존의 GPS 기반의 자동차 항법 시스템에 적용되는 map-matching 방법에 기반을 둔다. 즉, 이동 단말기의 일련의 측위 결과를 바탕으로 칼만필터 방법으로 궤적을 구하고, 디지털 지도의 도로들과 비교하여 가장 가까운 도로를 현재 이동단말기의 이동 route로 결정하고 이를 교정 route라 칭한다. 이때 칼만필터 방법으로 구한 궤적을 교정 route로 이동하기 위하여 이동한 방향과 거리를 오프셋트라 한다.

실시간 보정은 측위 결과에 오프셋트를 더하여 교정점을 구하고, 이전의 교정점과 현재 교정점이 장애물을 통과하는 비현실적인 현상을 보일 때 이를 실시간으로 교정한다. 본 논문은 제안하는 방법의 알고리즘을 제공하며, 실시간 보정을 위하여 사용될 각종 함수를 구현한 C# 클래스도 제공한다.

참고문헌

[1] Marwa Mabrouk, OpenGIS Location Services (OpenLS): Core Services OGC 03-006r <http://www.opengis.org/>

[2] Virrantaus, K., J. Veijalainen, and J. Markkula, "Developing GIS-Supported Location-Based Services," Proceedings of the Second International Conference on Web Information Systems Engineering, 2001, Vol. 2, Dec. 3-6 2001, pp.66-75.

[3] Adam Smith, HariBalakrishnan, Michel Goraczko, and Nissanka Priyantha, "Tracking Moving Devices with the Cricket Location System", Proc. of MobisSYS'04, June 2004.

[4] R. Want, A. Hopper, V. Falco and J. Gibbons, "The Active Badge Location System", ACM Transactions on Information Systems 10, pp. 91-102, Jan. 1992.

[5] Wann, C. & Lin, M. (2004), "Data fusion methods for accuracy improvement in wireless location

systems," Proceeding of the IEEE Wireless Communications and Networking Conference, 2004. (WCNC 2004) Vol. 1, 21-25 March 2004, pp. 471-476

[6] Youssef, M., Agrawala, A. & Shankar, A.U. (2003), "WLAN Location Determination via Clustering and Probability Distributions," Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom), 2003, 23-26 March 2003, pp. 143-150.

[7] Youssef, M. & Agrawala, A. (2004), "Continuous Space Estimation for WLAN Location Determination Systems," Proceedings of 13th International Conference on Computer Communications and Networks, 2004 (ICCCN 2004) pp. 161-166

[8] Yim, J., "Introducing a decision tree-based indoor positioning technique," Expert Systems with Applications, Vol. 34, Issue 2, 2008, pp. 1296-1302.

[9] Ciurana, M., Barcelo-Arroyo, F., Izquierdo, F., "A ranging system with IEEE 802.11 data frames," 2007 IEEE Radio and Wireless Symposium, 9-11 Jan. 2007, pp. 133 - 136.

[10] Ciurana, M., Barcelo-Arroyo, F., Cugno, S., "A novel TOA-based indoor tracking system over IEEE 802.11 networks," 2007 16th Mobile and Wireless Communications Summit, IST, 1-5 July 2007, pp. 1-5.

[11] Hwang, P.Y.C. and R.G. Brown (1992). Introduction to Random Signals and Applied Kalman Filtering, John Wiley & Sons.

[12] Morisue, F., Ikeda, K., "Evaluation of map-matching techniques," 1989 Vehicle Navigation and Information Systems Conference, 11-13 Sept. 1989, pp. 23-28.

[13] Collier, W.C., "In-vehicle route guidance systems using map-matched dead reckoning," IEEE 1990 Position Location and Navigation Symposium, 20-23 March 1990 pp. 359-363.

[14] Pyo, J, Shin, D, Sung, T, "Development of a map matching method using the multiple hypothesis technique," Proceedings of IEEE 2001 Intelligent Transportation Systems, 25-29 Aug. 2001, pp. 23-27.

[15] Wenk, C., Salas, R., Pfoser, D., "Addressing the Need for Map-Matching Speed: Localizing Global Curve-Matching Algorithms," 18th International Conference on Scientific and Statistical Database Management, 2006, pp. 379-388.

[16] Ardeshiri, T., Kharrazi, S., Thomson, R., Bargman, J., "Offset Eliminative Map Matching Algorithm for Intersection Active Safety Applications," IEEE 2006 Intelligent Vehicles Symposium, 13-15 June 2006, pp. 82-88.

[17] Jones, K., Liu, L., Alizadeh-Shabdiz, F., "Improving Wireless Positioning with Look-ahead Map-Matching," 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, (MobiQuitous 2007) 6-10 Aug. 2007, pp.1-8.