

속도를 이용한 위치 측정 오류 검출 및 교정

이준석*, 송하윤

홍익대학교 컴퓨터공학과

*rolunoa@gmail.com, hayoon@hongik.ac.kr

Detecting and Correction Errors of Positioning Data Using The Speed

Jun-Seok Lee*, Ha Yoon Song

Dept of Computer Engineering, Hongik University

요 약

스마트 모바일 기기 즉, 스마트폰을 이용한 위치 기반 서비스(LBS)가 각광 받고 있는 가운데, 사용자 관점에서의 위치 측정에 대한 오류를 검출과 위치를 교정하는 서비스가 필요로 하게 되었다. GPS를 이용한 위치 측정의 방법을 예로 들어 GPS 신호를 수신하는 기기에서의 위치 측정 오류 감지는 어려운 실정이고 약간 더 정밀하게 측정하는 방법만이 존재하지 않는다. 이 논문에서는 GPS 수신기와 같은 사용자 관점에서 위치 데이터들의 속도를 계산하여 속도의 변화량을 이용해 multipath와 같은 위치 데이터의 오류를 검출하고 교정한 결과에 대해 다룬다. 단순 속도의 변화량이 아닌 인간의 이동 속도가 지수 확률 분포를 따른다는 사실을 기반으로 이동 윈도우(Moving Window)의 개념을 도입해 매 윈도우마다 한계 속도를 결정하고 한계 속도 이상의 속도가 발생하였을 때를 위치 오류라 검출 할 것이다.

1. 서론

스마트폰의 등장 이전 시기엔 GPS, GLONASS와 같은 사용자의 위치를 측정하는 시스템은 거의 사용되지 않았다. 실제로 이 시기에 사용자의 위치 파악을 필요로 하는 분야는 자동차에 부착된 네비게이션이 대부분이었고 위치를 필수적으로 사용해야하는 산악인, 항해사, 비행기 조종사 등이 GPS와 같은 위치 측정 시스템을 사용하였다. 하지만 현재 iPhone, Galaxy 같은 GPS 신호 수신기가 부착된 스마트폰의 등장 이후 모바일 유저들은 언제 어디서든 손쉽게 자신의 위치를 파악할 수 있게 되었다. 이로 인하여 스마트폰 사용자는 세계 어느 곳을 가더라도 자신의 위치를 파악하고 무선 네트워크를 이용해 현재 위치 주변에 대한 정보를 얻고 원하는 곳까지 이동하거나, 근처 좋은 식당을 알 수 있거나, 자신이 운동한 운동량을 알 수 있게 되었다. 이러한 위치 기반 서비스(LBS)가 탑재된 모바일 어플리케이션들이 나날이 각광받고 있는 가운데 여러 환경요소로 인해 발생하는 위치 측정 오류를 감지하고 또한 교정하는 연구가 필요로 하게 되었다.

현재 GPS의 위치 측정의 이론적인 정확도는 7.8미터이며 95%의 신뢰도를 갖고 있고[1], 날씨 또는 지형등과 같은 환경요소가 완벽하다면 실제 민간인이 쓰는 GPS의 정확도는 1 ~ 2 미터에 가깝다. 하지만 여전히 GPS 위성파 GPS 신호 수신기 간의 시각 동기 오류, 주변 지형으로 인해 GPS 신호의 반사로 발생하는 multipath와 같은 위치 측정 오류들이 존재하고 있다. 이와 같은 위치 측정 오류

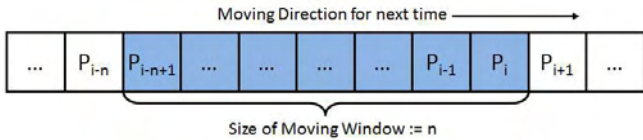
는 위치 측정 시스템의 사용자 즉, GPS 신호 수신기만으로는 감지해내기 어려우며 하이브리드 위치 측정 시스템과 같이 Wi-Fi 신호를 이용하여 위치를 약간 더 정밀하게 측정하는 방법만이 존재하지 않는다[2].

그리하여 이 논문에서는 GPS 신호 수신기만으로 위치 측정 오류를 속도를 이용하여 확률적으로 감지하고 위치를 대략적으로 교정하는 방법에 대해 소개할 것이다. 2장에서는 위치 오류 검출에 대한 연구의 선행 연구를 소개하고 3장에서는 위치 측정 오류를 확률적으로 검출할 수 있는 컴퓨팅 알고리즘에 대해 살펴 볼 것이다. 우리는 이 알고리즘을 이동 윈도우 필터링이라 하겠다. 그리고 4장에서는 이동 윈도우 필터링으로 실제 측정된 위치 데이터에 대해 실험한 결과를 보여줄 것이고 마지막 5장에서는 그에 따른 결론과 향후 연구 방향에 대해 다룰 것이다.

2. 이전 연구

이동 윈도우를 이용하여 과거 속도를 분석하여 위치 측정 오류 검출을 실험한 연구가 있었다[3]. 이 연구에서는 인간의 이동 속도가 정규 분포를 따른다는 가정 하에 이동 윈도우 내의 속도를 정규 확률 분포(Normal Distribution)로 모델링하여 위치 오류를 판단하였다.

그리고 특정 속도 구간에서 인간의 이동 속도가 의미 있는 관계로 지수 확률 분포를 따른다는 사실이 있다[4]. 비교적 속도 모델링 계산이 간단한 지수 확률 분포의 이용은 모바일 기기의 연산을 줄여줌으로 소비 하는 전력이



<그림 1> 이동 윈도우의 도식

적다는 장점이 있다.

위 연구를 종합적으로 이용해 기록된 위치 데이터에 대한 최근 속도 표본을 이동 윈도우 안에서 지수 확률 분포로 모델링 하고, 현재 측정되는 위치의 속도가 정상 확률로서 나오는 속도인지를 판단하여 해당 위치 오류 여부를 결정 할 것이다.

3. 이동 윈도우 필터링

이 논문에서 소개하는 연구에서는 속도를 이용하여 확률적으로 위치 오류를 검출하는 것뿐만 아니라 백트래킹(Backtracking)기법을 사용하여 잘못된 오류 검출, 즉 오류가 아님에도 불구하고 오류로 검출된 위치를 정상적으로 측정된 위치로서 복구한다. 1절과 2절에서는 이동 윈도우와 모델링의 방법에 대해 소개하고 3절에서는 백트래킹에 대한 방법을 소개한다.

3.1 이동 윈도우

<그림 1>은 이동 윈도우를 도식화하여 표현한 것이다. 시간 순서에 따라 측정된 과거의 위치 데이터는 바탕색으로 표시된 이동 윈도우 안에 입력된다. 입력된 위치 데이터의 속도를 바탕으로 지수 확률 분포로 모델링 한 뒤 다음 입력되는 위치데이터 P_{i+1}의 속도가 정상적인 속도인지 확인을 한다. 만약 P_{i+1}의 속도가 정상이라면 이동 윈도우는 오른쪽으로 이동한다. 명확히 말하자면 위치 데이터 P_{i-n+1}는 빠지고 P_{i+1}가 윈도우에 입력된다. 그리고 위치 P_{i+1}가 비정상적인 속도가지고 있다면 이동 윈도우를 오른쪽으로 이동시키고 이전 위치 P_i와 다음 위치 P_{i+2}의 선형 보간으로 교정 한다. 이동 윈도우의 이동은 다음 위치의 입력이 없을 때까지 계속 된다.

3.2 지수 확률 분포 모델링

속도를 지수 확률로서 모델링하기 위해 먼저 이동 윈도우 내에 있는 속도들의 평균과 표준편차를 계산한다. 그 후 평균(μ)과 표준편차(σ)가 같다는 지수 확률 분포의 특징을 이용해 우리는 표준편차를 지수 확률 분포의 척도 모수(Scale Parameter, $1/\lambda$)로 결정하고 평균과 표준편차의 차이는 위치 모수로 사용하였다. 위치 모수(Location Parameter) γ 를 식(1)과 같이 계산한다.

$$\gamma = \begin{cases} \mu - \sigma & \mu > \sigma \\ 0 & otherwise \end{cases} \quad (1)$$

이렇게 모델링된 지수 확률 분포의 누적 확률 함수를

<표 1> 단순 백트래킹 알고리즘

Backtracking of Moving Window Filtering
1: Input P(i) in Moving Window
2: if(P(i) is not error)
3: then {
4: Update Moving Window Model excluding P(i-1)
5: if(P(i-1) was error && P(i-1) is not error)
6: then { P(i-1) = Original P(i-1) }
7: }

이용해 미리 정한 일정 수준의 유의 구간을 벗어난 속도를 오류라 판단 할 것이다.

3.3 백트래킹

유의 구간을 벗어난 속도, 즉 오류로 검출된 속도지만 충분히 발생할 수 있는 속도인 경우가 존재 한다. 이러한 이유 때문에 백트래킹 개념을 도입하여 오류라 판단된 속도를 다시 확인하는 부분이 필요하다. 다음 알고리즘 <표 1>은 이동 윈도우 필터링에서의 백트래킹에 대한 기술이다.

<표 1>은 위치 P_{i-1}가 오류로 판단되어 교정되고 난 상태에서 이동 윈도우에 입력되는 위치 P_i가 오류가 아닐 경우 실행되는 알고리즘 나타내고 있다. 다시 말해 다음 시각의 위치인 P_i를 이동 윈도우에 입력하고 모델링하였더니 오류로 판정되었던 위치 P_{i-1}가 정상 위치였던 것으로 판정되어 교정된 위치에서 본래의 위치로 되돌려 놓는 알고리즘이다.

하지만 백트래킹은 이렇게 이상적인 조건에서만 존재 하지 않는다. 연속적인 오류 즉, 위치 모두 오류 일 경우에서도 백트래킹을 구현해야하기 때문에 우리의 실험에서는 <표 1> 의 백트래킹에서 진보된 백트래킹을 포함한 이동 윈도우 필터링을 사용하였다.

<표 2>는 위치 P_i와 P_{i+1} 모두 오류로 판정 되었을 경우를 대비하여 실제 실행되는 이동 윈도우 필터링 알고리즘이다. 예를 들어 발생 할 수 있는 모든 경우를 나눠 생각해보자. 백트래킹이 실행되기 위하여 위치 P_{i-1}가 오류로 검출되었다 가정하고,

1) P_i가 오류, P_{i+1}는 정상일 경우

P_i는 교정된 위치가 아닌 측정된 위치 P_{i-1}와 위치 P_{i+1}의 선형 보간 교정(줄번호 7)된다. 이와 같은 단순한 교정은 P_{i+1}이 정상이라는 가정을 하였기 때문에 가능하다. 그리고 이동 윈도우에서 P_{i-1}의 속도를 제외한 뒤, 속도의 유의 구간을 계산하고(줄번호 8) P_{i-1}의 속도가 이 유의 구간 내에 속하게 되면 오류로 판정되었던 위치 P_{i-1}를 정상인 위치 데이터로 되돌려 놓는다(줄번호 10). 백트래킹이 모두 실행되고 나면 P_{i-1}를 포함한 이동 윈도우 내의 속도들을 다시 모델링하여 위치 P_{i+1}의 속도가 확률적으로 정상인 속도인지 확인한다(줄번호 15). 이러한 작업을 다음 위

<표 2> 이동 윈도우 필터링 알고리즘

Moving Window Filtering
1: Repeat
2: Input P(i) in Moving Window
3: if(P(i) is error)
4: then {
5: if(Acceleration of P(i+1) >= 10.8m/s ²)
6: then { Estimate P(i+1) by previous Moving Window }
7: P(i) = (Original P(i-1) + P(i+1)) / 2 }
8: Update Moving Window Model excluding P(i-1)
9: if(P(i-1) was error && P(i-1) is not error)
10: then { P(i-1) = Original P(i-1) }
11: else if(P(i-1) was error && P(i), P(i-1) are error)
12: then { P(i) = (P(i-1) + P(i+1)) / 2 }
13: Update Moving Window Model including P(i-1)
14: P(i+1) = Original P(i+1)
15: Check whether P(i+1) is error
16: i++

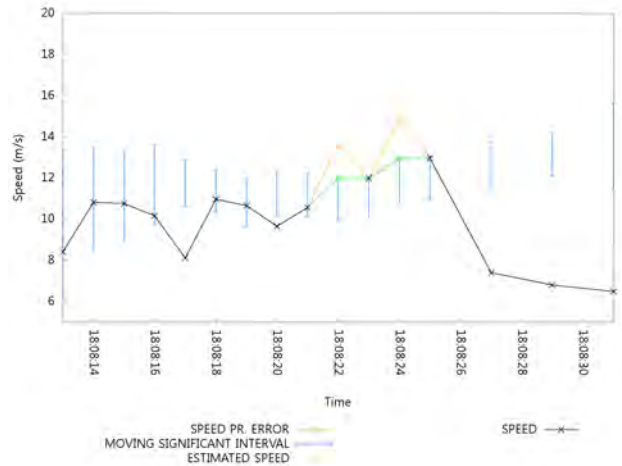
치 데이터가 더 이상 들어오지 않을 때 까지 반복한다.

2) P_i는 정상, P_{i+1}가 오류일 경우

위치 P_i는 교정이 필요 없으므로 곧바로 P_{i-1}를 제외한 이동 윈도우 내의 속도들을 지수 확률 분포로 모델링한다. 그리고 P_{i-1}의 속도가 정상적으로 발생할 수 있는 속도였는지 다시 확인한다. 만약 P_{i-1}가 정상 속도로 판정되면 이전에 측정되었던 원래의 위치로 되돌려 놓고(10번째 줄), P_{i-1}가 백트래킹에 의해서도 오류로 판정되면 교정된 위치 P_{i-1}와 P_{i+1}를 이용해 위치 P_i에 대해 선형 보간 교정을 한다(줄번호 12).

3) P_i와 P_{i+1}가 모두 오류일 경우

P_i가 오류라 판정되었고 P_{i+1}가 오류라는 것을 미리 알고 있는 상태라면 P_i의 위치를 교정하는데 있어 매우 무리가



<그림 2> 위치 데이터 오류 검출과 교정 (그래프)

있다. 또한 이동 윈도우를 업데이트하기 전 P_{i+1}가 오류라고 판정하는 것은 불가능하므로 우리는 P_{i+1}가 오류가 아니라 판단 할 수 있는 방법으로 가속도를 이용하였다. 위치 P_{i+1}에서 발생한 가속도가 미리 설정한 가속도 값을 넘지 않는다면 P_{i+1}는 정상인 위치 데이터라 판단(줄번호 5)하였다. 만약 설정된 가속도를 넘는다면 이는 확실한 오류로 보고 위치 P_i가 입력되기 전 이동 윈도우에서 계산된 평균 위치 변화량과 위치 P_{i+1}가 측정된 방향을 통해 P_{i+1}의 위치를 추정하여 임시로 교정한다. 그 뒤 1)의 경우에서와 같이 진행된다.

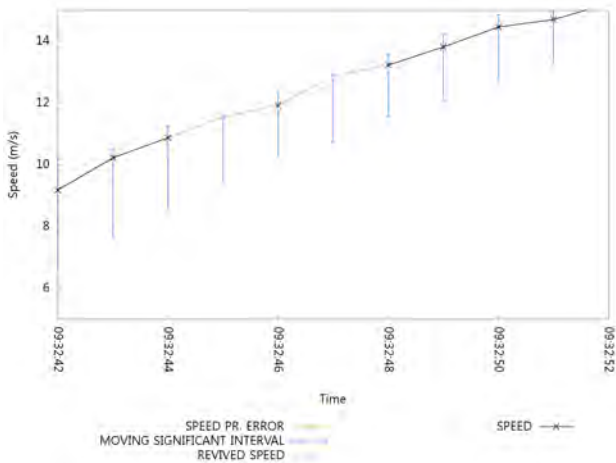
4. 실험 결과

이번 장에서는 3장에서 소개한 이동 윈도우 필터링을 이용해 실험한 결과를 소개한다. 위치 데이터 제공자의 실제 위치데이터에서의 실험한 결과이다. 위치와 속도를 직관적으로 확인하기 위해 속도 그래프와 위치에 대한 지도를 이용하였다.

4.1 위치 데이터 오류 검출 및 교정



<그림 3> 위치 데이터 오류 검출과 교정 (지도)



<그림 4> 오류에 대한 백트래킹(그래프)

백트래킹에 의한 위치 오류 검출과 교정에 앞서 가장 많이 발생하는 경우로 백트래킹이 필요 없는 위치 측정 오류가 발생 하였을 때 그 데이터에 대한 오류 검출과 교정을 보여 줄 것이다. <그림 2>는 이러한 위치에 대한 시간과 속도의 그래프이다. 검정 그래프는 오류가 아닌, 즉 정상적인 속도(SPEED), 노란 그래프는 속도 유의구간에 의해 오류로 판정된 속도(SPEED PR. ERROR), 초록 그래프는 위치의 교정으로 계산된 속도(ESTIMATED SPEED)이다. 파란 구간 형태의 수직선은 해당 시각에서 충분히 나올 수 있는 속도의 구간, 즉 유의구간(MOVING SIGNIFICANT INTERVAL) 이다. <그림 2>에서 18:08:22인 시각과 18:08:24인 시각에서의 속도는 속도의 유의구간을 초과했기 때문에 해당 위치들은 교정되고 교정된 위치의 속도는 초록색 점과 같이 계산되었다. <그림 3>은 같은 위치 데이터를 지도상에 나타낸 것으로 위치 데이터는 왼쪽으로부터 오른쪽으로 이동하는 방향이다. 하얀색 점은 정상적으로 측정된 위치, 노란색 점은 오류로 판정된 측정된 위치이고 초록색 점은 해당 위치 오류에 대해 교정된 위치이다.

4.2 백트래킹이 적용된 오류 검출 및 교정

<그림 4>와 <그림 5>는 급가속 시 발생할 수 있는 상황의 속도에 대한 그래프와 해당 위치 데이터를 표시한 지도이다. 09:32:45인 시각에서의 속도를 보면 유의구간을 살짝 넘어갔지만 백트래킹에 의해 교정된 속도가 아닌 측정된 속도(REVIVED SPEED)로 되돌려 났다. 09:32:47인 시각에서도 마찬가지로 백트래킹에 의해 오류로 검출된 위치가 측정된 위치로 었다. <그림 5>는 이에 대한 지도상의 위치이고 <그림 3>에서와 마찬가지로 하얀색 점은 정상적으로 측정된 위치이고 점 안에 검은 점으로 표시된 위치는 오류로 검출되었다가 백트래킹에 의해 정상으로 판정된 위치이다. 위치는 왼쪽에서 오른쪽으로 이동하는 방향이다.

5. 결론 및 향후 연구

이 논문에서는 GPS와 같은 위치 측정 시스템에서 위



<그림 5> 오류에 대한 백트래킹(지도)

치 측정 도중 시각 동기화 오류, Multipath 오류와 같은 위치 측정 오류를 발생 하였을 때 사용자의 입장, 즉 신호 수신기의 입장에서 오류를 검출하고 교정하는 실험을 행하였다. 또한 백트래킹 개념을 도입하여 정상인 속도를 오류로 판정하는 잘못된 검출을 피하였다.

우리는 이동 윈도우 필터에서의 알고리즘은 최근 위치 일수록 민감하게 반응하지 않고 윈도우 내의 모든 속도들에 대해 같은 가중치를 두고 확률 분포로서 모델링하였다. 그리하여 잘못된 오류 판정 또는 실제로 오류이지만 정상인 위치로서의 판정이 발생하였다. 이에 대해 우리는 시간 간격, 속도의 변화량을 분석하여 이 2가지 변수에 대해 확률 분포를 만들어 더욱 정밀한 위치 오류 검출을 연구하고 실험할 계획이다.

6. Acknowledgement

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2012R1A2A2A03046473)

참고문헌

- [1] Misra, Pratap, and Per Enge. Global Positioning System: Signals, Measurements and Performance Second Edition. Lincoln, MA: Ganga-Jamuna Press, 2006..
- [2] Miller, John M, and Himanshu S. Amin. "System for dynamically pushing information to a user utilizing global positioning system." U.S. Patent No. 6,741,188. 25 May 2004.
- [3] Song, Ha Yoon, and Kyoung-woon On. "Detection and Estimation of Erroneous Positioning Data." Procedia Computer Science 18 (2013): 2533-2536.
- [4] 이준석, 송하윤, "인간 이동속도의 분포 추정", 2014년 추계학술발표대회 논문집, 제 21권, 제 1호. 2014.