

정류장 및 노선 정보를 이용한 GPS기반 버스 노선 추측 알고리즘

윤용상, 김경백

전남대학교 전자컴퓨터공학부

e-mail:tiah.florens@gmail.com, kyungbaekkim@jnu.ac.kr

GPS based Bus Route Estimation Algorithm by Using Information of Bus Stops and Routes

Yongsang Yoon, Kyungbaek Kim

Dept of Electronics and Computer Engineering

Chonnam National University

요 약

일반적인 이동 수단은 도보, 자동차, 버스가 있으며 그 외에 기차, 자전거 등이 있다. 사용자가 이용하는 이동 수단 상태는 개개인의 특징을 포함하며 이를 파악한다면 다양한 면에서 유용하게 사용될 수 있다. 사용자의 이동 수단의 상태를 파악하기 위해서 GPS 정보가 사용될 수 있고, 위치정보 기반의 평균속도를 계산함으로써 대표적인 이동수단인 도보, 자동차, 버스를 구별할 수 있다. 이 논문에서는 사용자가 버스를 탑승하고 있을 경우, 위치 정보와 GIS정보(버스 정류장 정보와 버스 노선 정보)를 통해 해당 탑승 버스의 노선 정보를 예측하기 위한 알고리즘을 제안한다. 알고리즘의 수행 가능성을 파악하기 위해, 약 1개월간 수집된 버스 이동 GPS 로그와 광주지역 버스 노선 및 정류장 정보를 이용한 알고리즘 검증을 수행하였고, 알고리즘이 정상적으로 동작하는 것을 확인하였다.

1. 서론

이동 기기의 고도화 및 소프트웨어의 발전과 더불어, 2007년 아이폰 출시 이후 스마트폰 붐이 일어나면서 한국에서의 안드로이드 폰 공급 및 수요가 폭발적으로 증가하게 되었고, GPS Sensor가 장착된 이 기기들을 통해 누구든지 손쉽게 자신의 위치 정보를 기록할 수 있는 환경이 구성되었다. 이에 따라 누구나 손쉽게 위치 정보를 활용할 수 있게 되었다. 위치 기반 서비스의 폭이 점차 확대되면서 페이스북, 트위터 같은 SNS는 기본적으로 위치 정보를 활용한 기능을 제공한다.

위치 정보를 기반으로 파악할 수 있는 정보는 단순히 위치 뿐 만이 아니라 다른 정보와 결합하여 분석한다면 정확한 이동 수단까지 파악할 수 있다. 이로써 얻을 수 있는 가장 큰 효과는 대표적으로 사용자의 편의성 증대 그리고 자원의 효율적인 이용이 있다. 아주 간단한 예로는 스마트폰이 사용자의 이동 수단의 상태를 자동적으로 인지하여, 해당 이동 상태에 알맞은 사용자 인터페이스를 제공하는 것을 생각할 수 있다. 좀 더 자세한 시나리오를 생각해보면, 사용자가 지도 정보를 사용하여 임의의 목적지로의 경로를 찾는 경우를 들 수 있다. 만약 사용자의 현재 이동 수단 상태가 버스 노선 17번을 타고 있는 경우라면, 지도 정보 서비스는 앞으로 몇 번째 정류장에서 내려

서 움직여야 하는지에 대한 정확한 정보를 사용자에게 제공할 수 있게 된다.

이와 같이 자세한 사용자의 이동 수단 상태를 파악할 수 있다면 얻을 수 있는 효과가 큰 반면, 위치 정보만으로 이와 같은 자세한 상태를 얻는 것은 쉽지 않다. 단순히 도보 상태와 차량 상태를 구분 짓는 정도는 위치정보와 평균속도 정보를 사용하여 파악할 수 있다. 하지만, 간단한 예로, 버스와 자동차를 구분하는 것은 정차주기의 변화 및 도로 교통 상황 등에 의한 다양한 변수로 인해 구분을 명확히 짓기 힘들게 된다.

이 논문에서는 보다 자세한 사용자의 이동 수단 상태를 파악하기 위해서 GIS정보를 사용하는 경우중 하나로 버스 정류장 및 노선 정보를 이용한 위치기반 버스 노선 예측 알고리즘을 제안한다. 이 알고리즘을 사용하면, 수집되는 위치정보를 통해 버스와 차량상태를 확실하게 구분하고, 버스의 경우 사용자가 탑승한 버스 노선 정보 까지 정확히 파악할 수 있다.

2장에서는 알고리즘 수행을 위해서 필요한 데이터 모델에 대해서 기술하고, 3장에서는 버스 노선 예측 알고리즘을 소개한다. 4장에서는 버스 이용 시 수집된 GPS 데이터를 사용하여 알고리즘의 성능 여부를 검증하고, 5장에서 결론을 내린다.

⁺ "본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터육성지원사업의 연구결과로 수행되었음" (NIPA-2014-H0301-14-1014)

2. 데이터 모델

2.1 위치 정보

GPS Sensor가 장착된 스마트폰과 같은 기기를 통해서 위치 정보를 수집한다. 각각의 위치 정보들은 P(i)로 표현하고 lat ,lng , t를 포함한다. lat는 위도를 lng는 경도 그리고 마지막 t 는 GPS 좌표가 기록된 시간을 의미한다.

$$P(i) = \{ \text{lat} , \text{lng} , t \}$$

2.2 위치 정보 변환

위치 정보가 2개 이상 수집 되었을 때 두 개의 연속된 위치 정보 P(i)와 P(i+1) 사이의 거리, 시간 차이를 구하여 속력 V(i)을 계산한다. (lat(i),lng(i)),(lat(i+1),lng(i+1)) 두 점 사이의 거리를 계산할 때 사용 가능한 공식은 Spherical law of cosine이다. 이 법칙은 임의의 Sphere표면상의 두 위치들에 대한 radian표현법을 알고 있다면, 두 위치 사이의 arc의 길이를 구하는 법을 정의한다. 즉, $\text{dist} = \text{acos}(\sin(\phi_1) \times \sin(\phi_2) + \cos(\phi_1) \times \cos(\phi_2) \times \cos(\Delta\lambda)) \times R$ 과 같은 식을 적용하여 거리를 구한다. 이식을 지구에 적용할 경우, ϕ 는 위도를 λ 는 경도($\Delta\lambda$ 은 두 경도의 차이)를 나타내고, R은 지구의 반지름의 길이를 의미한다. 이 식을 64비트기반의 소수점 15자리까지 표현 가능한 플로팅 포인트 수를 사용하여 계산 할 경우, 약 1m 이하의 오차를 보이게 된다. 시간 차이는 second 단위인 T(i) , T(i+1)를 비교함으로써 쉽게 계산이 가능하다. 이렇게 구해진 시간과 거리차이로 m/s 단위인 V(i) 속력을 계산한다.

$$V(i) = \text{dist}(i, i+1) / (T(i+1) - T(i))$$

2.3 GIS 정보 - 버스 정류장 및 노선

정류장 정보 busStop(i)는 id, sname, through Bus 3가지 항목들로 구성되어 있다. Id는 각 버스 정류장의 고유 번호이며 sname은 각 정류장의 명칭이다. 마지막으로 throughBus는 해당 정류장에 정차하는 모든 버스들을 상,하행을 구분하여 담고 있다.

버스 노선 정보 busLine(i)는 id, bname, topDown, downTop 4가지 항목들로 구성되어 있다. Id는 각 버스의 고유 번호이며 bname은 버스의 명칭이다. topDown과 downTop은 해당 버스의 상행, 하행 경로 상의 정류장을 순서대로 모아 놓은 것이다. 현재 사용된 버스 노선의 개수는 101개이다.

$$\text{BusStop}(i) = \{ \text{id}, \text{sname}, \text{throughBus} \}$$

$$\text{BusLine}(i) = \{ \text{id}, \text{bname}, \text{topDown}, \text{downTop} \}$$

2.4 GIS 정보 변환 - 공유 정류장 개수

각 버스 노선의 전체 경로는 유일하지만 버스 정류장

은 여러 버스 노선과 공유하기 때문에, 각 노선의 일부는 다른 버스 노선과 겹칠 수 있다. 총 101개의 버스들이 자기 자신을 제외한 나머지 100개의 버스와 짝을 이루어 각각 버스의 대해 연속적으로 정류장을 공유하는 정류장 개수의 최대값을 구하여 기록한다. 이는 사용자가 탑승했을 거라고 추정되는 여러 개의 후보 버스 중 사용자가 탑승한 버스를 구분하는데 사용된다.

$$(\text{Busline}(i) , \text{BusLine}(j)) = \text{count}$$

3. 버스 노선 예측 알고리즘

사용자의 위치정보와 속도정보를 이용하여 대략적인 사용자의 이동 수단 상태를 파악한 후, 만약 이동 수단 상태가 버스일 경우 제안하는 알고리즘을 수행하여 현재 이동 중인 사용자가 탑승하고 있는 버스를 예측한다.

3.1. 알고리즘의 기본 흐름

사용자의 새로운 위치가 수신 될 때 마다 일정 반경 내의 모든 버스 정류장을 검색하여 해당 정류장들과 정류장을 거치는 버스 노선을 획득한다. 새로 검색된 정류장이 기존에 획득한 버스 노선 경로 상의 다음 정류장 인지 확인하는 과정을 반복하여 경로상의 정류장을 연속적으로 가장 많이 거친 버스를 후보 선택한다. 후보 버스들에서 다음의 두 가지 기준을 충족한 버스를 선택한다.

1. 경로 상 정류장을 지나친 횟수가 가장 큰 버스
2. 자신의 경로 상 정류장을 지나친 횟수가 다른 모든 후보 버스들과 공유 정류장 수보다 클 때



(그림 1) 사용자 위치에 따른

정류장 검색 범위의 예

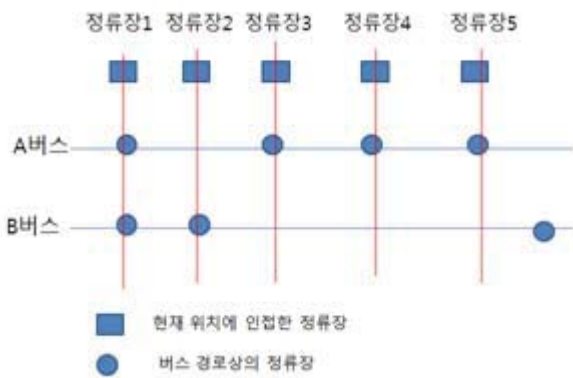
3.2. 위치에 따른 정류장 및 버스 노선 정보 획득

임의의 위치정보 P(i)가 들어오면, 현재 좌표를 중심으로 (lat-0.001)~(lat+0.001), (lng-0.001~lng+0.001) 범위의 모든 정류장을 검색한다. 위도나 경도에서 0.001의 차이는 m로 환산하면 약 109m정도가 된다. 정류장들이 검색되면, 이미 등록된 정류장을 제외한 새로운 정류장들을 정류장 리스트(passedBusStopList)에 추가한다.

새로운 정류장 리스트가 확보되면, 이 정류장들과 관련된 버스 노선들을 정리한다. 이때, 기존에 등록되어 있던 버스 노선을 제외한 새로운 버스 노선들을 버스 노선 리스트(BussLineSet)에 추가한다. 이때 각 버스 노선 항목은

현재까지 사용자가 지나온 정류장 정보를 저장한다. 또한 버스 노선이 속한 정류장을 노선상의 현재 정류장으로 설정한다.

이때, 기존에 버스 노선 리스트에 등록되어 있던 노선들에 대해서는 새로 들어온 정류장에 따라서 이 노선의 유효성 여부를 판단한다. 임의의 기존 등록된 하나의 노선이 현재 정류장 위치를 알고 있다고 하자. 이 때, 새로 들어온 정류장들 중의 하나라도 이 노선의 다음 정류장과 일치한다면, 해당 노선의 pass count를 1 증가시키고 새로 들어온 해당 정류장을 그 노선의 현재 정류장으로 변경한다. 예를 들어 그림 2에서와 같이 임의의 사용자의 위치정보에 따라 정류장 1부터 정류장 5까지의 정보가 연속적으로 획득되었을 경우를 생각해보자. 정류장 2번이 검색될 때, A 버스의 경우엔 현재 정류장이 정류장 1이고 다음 정류장이 정류장 3이므로 pass count가 증가 하지 않고, B 버스의 pass count가 증가한다. 이런 방식으로 정류장 5번까지 정보를 획득한 후 살펴보면 A 버스의 pass count는 4, B 버스의 pass count는 2가 된다.



(그림 2) 버스 정류장 인지에 따른 노선 정보 업데이트의 예

3.3. 후보 버스 선별 및 버스 노선 예측

후보 버스 리스트(Candidate list)에는 버스 노선 리스트 중에서 pass count가 일정 기준 값 (Threshold) 이상인 것들을 추가한다. 버스 노선 예측은 이 후보 버스 리스트 중에서 하나를 선택하는 것으로 수행된다. 우선 후보 버스 리스트 중에서 가장 큰 pass count를 가지는 버스 노선을 찾고, 이 노선의 pass count보다 1이 작은 값을 가지는 버스 노선들도 모두 찾아낸다. 단순히 pass count가 큰 값을 가지는 노드만 선택하지 않는 이유는, 사용자의 위치정보기반의 정류장을 획득할 때의 오차를 보정하기 위해서이다.

이와 같이 수집된 노선 정보들과 공유정류장 개수 정보를 함께 고려하여 최종적으로 버스 노선을 예측한다. 즉 수집된 노선 정보들 간의 공유정류장 개수를 체크한 후, 임의의 노선 정보가 공유정류장 개수를 넘어갈 경우 이 노선을 최종적으로 예측된 버스 노선으로 정한다.

4. 알고리즘 검증

알고리즘의 검증을 위해 제안된 알고리즘 수행을 위한 프로그램을 java를 이용해 작성하였다. 알고리즘에 사용될 입력 위치정보 수집을 위해, 17번에 걸쳐 광주지역 버스를 탑승하여 이동 중 매 1초마다 위치를 저장하여 위치정보 입력 로그를 생성하였다. 또한 GIS 입력정보로는 광주지역 버스 노선과 정류장 위치 정보를 사용하였다. 검증 시 사용된 알고리즘의 pass count threshold를 6으로 한 경우

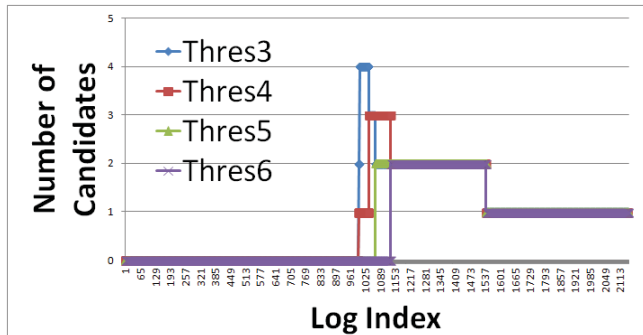
Index	예측여부	예측시간	예측 후보 수
1	Yes	19분33초	1
2	Not Yet	-	2
3	Not Yet	-	2
4	Yes	18분47초	1
5	Yes	18분57초	1
6	Not Yet	-	5
7	Yes	11분57초	1
8	Yes	11분37초	1
9	Yes	20분55초	1
10	Yes	13분03초	1
11	Yes	18분47초	1
12	Not Yet	-	3
13	Yes	11분50초	1
14	Yes	21분39초	1
15	Yes	15분29초	1
16	Yes	12분13초	1
17	Yes	19분44초	1

<표 1> 버스 예측 알고리즘 수행 결과의 결과는 표 1과 같다. 표 1에서 예측 시간이란 알고리즘에 위치정보가 들어오는 시점에서 단일 버스를 예측하는데 까지 걸린 시간을 의미한다. 예측 후보 수는 예측 알고리즘을 통해 최종으로 선택된 후보 버스의 수를 나타낸다.

알고리즘 수행 결과, 총 17개 로그 중 13개의 로그에서 단일 버스 노선을 정확히 예측하였고 그 평균 예측 시간은 16분 29초였다. 나머지 4개의 노선에서는 알고리즘에서 최종적으로 선택된 후보 버스 중엔 해당 버스 노선이 들어있었지만, 알고리즘에서 제안된 기준을 통과하지 못해 단일 후보로 선정이 되지 못하였다. 단일 후보를 선정하지 못한 이유는 크게 두 가지이다. 첫째는, 버스 예측을 위한 입력 위치정보의 길이가 충분치 못한 경우로, 12번 로그의 경우 로깅 시간이 13분 정도로 확실한 예측을 얻기 위한 시간이 약간 부족했다. 두 번째 이유는 버스간의 공유 노선 구간이 긴 경우이다. 2번, 3번, 6번 로그의 경우, 해당 버스 노선과 겹치는 버스들이 많이 운행되는 구간에서 사용자 위치가 저장되어 있었다.

제안된 알고리즘에서 사용되는 pass count threshold의 영향에 대해 이해하기 위해 pass count threshold값을 3에서 6까지 변화 시켜가며 노선 예측 상황을 확인한 결과를 그림 3에서 나타내고 있다. 이 그림에서 threshold값이 작아질수록 같은 시간의 예측 후보의 수가 많은 것을 알 수

있다. 하지만, 일정 시간이 지나면 예측 후보의 수가 수렴하는 것을 볼 수 있다. 또한 이 그림에서 threshold가 3일 경우 예측 후보의 최대값은 4이고, 이는 threshold가 작게 설정되어 예측 후보의 수가 많아지더라도 그 수가 그리 크지 않음을 의미한다.



5. 결론

이동기기의 고도화에 따라 사용자의 행동 양식을 이동 기기에서 인지하기 위한 연구가 활발히 진행되고 있다. 이 논문에서는 이동기기의 GPS정보와 지상의 GIS정보, 특히 버스 정류장 및 버스 노선 정보를 함께 사용하여 사용자의 이동수단을 판단하여 탑승하고 있는 버스의 노선 정보를 예측하는 알고리즘을 제안하였다. 실제 버스 탑승을 통해 수집된 GPS정보를 사용하여 알고리즘의 성능을 검증하였다. 버스의 잦은 정차 및 정류장간의 간격의 다양성에 따라 버스 노선 예측에 필요한 소요 시간은 평균 16분 29초로 약간 길고 그 편차도 큰 것을 확인하였다.

현재, 이 알고리즘과 함께 사용자의 입력을 받아들여 현재 탑승 노선을 쉽게 확인 할 수 있는 실시간 버스 노선 예측 어플리케이션을 구상중이다.

참고문헌

[1] 윤용상, 김경백, 사용자 이동 방식 인지를 위한 위치정보 수집 및 분석, 2013년도 한국정보처리학회 춘계학술대회.

[2] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, Wi Ying Ma, Understanding Mobility Based on GPS data, In Proceedings of Ubicomp 2008.

[3] Yu Zheng, Like Liu, Longhao Wang, Xing Xie, Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web, In Proceedings of WWW 2008.

[4] Leon Stenneth, Ouri Wolfson, Philip S. Yu Bo Xu, Transportation Mode Detection using Mobile Phones and GIS Information, In Proceedings of ACM SIGSPATIAL GIS 2011

[5] Pengfei Zhou, Yuanqing Zheng, Mo Li, How Long to Wait?: Predicting Bus Arrival Time with Mobile Phone based Participatory Sensing, In Proceedings of MobiSys 2012