

Performance Improvement of Map Matching Using Compensation Vectors

安 禱 郎* · 李 東 旭†
(Do-Rang Ahn · Dong-Wook Lee)

Abstract - Most car navigation systems(CNS) estimate the vehicle's location using global positioning system(GPS) or dead reckoning(DR) system. However, the estimated location has undesirable errors because of various noise sources such as unpredictable GPS noises. As a result, the measured position is not lying on the road, although the vehicle is known to be restricted on the road network. The purpose of map matching is to locate the vehicle's position on the road network where the vehicle is most likely to be positioned. In this paper, we analyze some general map matching algorithms first. Then, we propose a map matching method using compensation vectors to improve the performance of map matching. The proposed method calculates a compensation vector from the discrepancy between a measured position and an estimated position. The compensation vector and a newly measured position are to be used to determine the next estimation. To show the performance improvement of the map matching using compensation vectors, the real time map matching experiments are performed. The real road experiments demonstrate the effectiveness and applicability of the proposed map matching.

Key Words : GPS(Global Positioning System), CNS(Car Navigation System), Map Matching, Compensation Vectors

1. 서 론

과학기술의 발전으로 고성능 장비를 저렴한 가격으로 활용할 수 있게 됨에 따라서 자동차의 자동항법에 대한 관심과 연구가 확산되어 가는 추세이다. 그동안 산발적으로 진행되었던 디지털 지도제작, CAD 시스템, 자동운송위치시스템(AVLS), 차단관리(Fleet Management), 차량항법시스템(CNS), 위치측정시스템(GPS) 등을 통합한 지능형 교통시스템(ITS: Intelligent Transportation Systems)이라는 새로운 관심분야가 등장하였고, 이러한 ITS 관련 신기술들은 충돌방지시스템, 일반적인 운송정보서비스, 도로 기반시설의 관리와 같은 많은 분야에서 응용되고 발전되고 있다. 이러한 기술들이 널리 쓰이고 발전하기 위해서는 무엇보다도 먼저 도로상에서 주행하는 차량 및 운송수단의 정확한 위치파악이 필수적이다. 일반적으로 알려진 바와 같이 GPS나 셀룰러 통신을 이용한 위치파악 시스템은 다양한 장비의 잡음과 통신 잡음으로 인하여 차량의 정확한 위치파악이 불가능하게 된다. 결과적으로 차량이 도로위에서 주행 중임에도 불구하고, 주행 경로를 벗어난 것과 같은 측정결과를 나타내게 된다.

그래서 차량의 위치를 보다 정확히 파악하기 위해서 GPS와 추측항법(DR: Dead Reckoning)을 이용하기도 한다. GPS

는 지구 상공에서 주기궤도를 선회하고 있는 저궤도 인공위성들과 삼각측량 원리를 이용하여 차량의 절대 위치를 측정해내는 시스템으로 SPS(Standard Positioning Service)와 PPS(Precise Positioning Service)로 나눈다. PPS는 군사용 목적으로 개발된 서비스로 특수한 암호해독장비를 갖출 수 있는 권한이 부여된 사람에게만 사용권한이 부여되는 것이고, SPS는 일반인에게 제공되는 것으로 SA(Selective Availability)에 의해서 PPS보다 정확도가 떨어지게 된다. 그리고 GPS는 위성시계의 오차, 위성궤도의 오차, 대기권에서의 전파지연, 수신기에서 발생하는 오차, 그리고 위성들의 궤도면 배치 상황에 따라라도 오차가 발생하게 된다. DR시스템은 관성항법장치 원리를 이용한 것으로 자이로(GYRO)센서, 휠(Wheel)센서, 스피드센서, 가속도센서 등을 이용하여 차량의 상대위치를 측정해내는 시스템이다. DR시스템 역시 미사일 추적 등과 같은 군사목적으로 개발된 것이지만 GPS보다 훨씬 이전부터 일반 항법분야(항공기나 선박)에 활발히 이용된 기술이다. GPS와 DR 시스템을 비교할 때 어느 것이 더 좋은 위치측정 시스템인지 단언하기는 어렵다. 이들 시스템은 각각 절대 측위시스템과 상대 측위시스템으로 근본 성격이 다르기 때문이다. 또한 GPS와 DR시스템의 단점을 상호 보완할 수 있는 하이브리드형 위치확인시스템도 활발히 연구, 적용되고 있다. GPS나 시스템오차보정 위치측정시스템(DGPS: Differential GPS) 또는 DR시스템이 아무리 정확한 위치를 파악한다 하더라도 오차가 발생하는 것은 어쩔 수 없다. 또한 항법용으로 특수 제작된 지도 역시 여러 가지 이유 때문에 기본적인 오차가 발생하게 된다. 이러한 부정확한 차량의 위치 값을

† 교신저자, 正會員 : 東國大 電氣工學科 副教授 · 工博
E-mail : dlee@dgu.edu

* 正會員 : 東國大 電氣工學科 博士課程
接受日字 : 2004年 6月 21日
最終完了 : 2004年 12月 23日

맵 데이터 베이스를 참조하고 에러를 보정하여 차량의 위치를 표시해 주는 것이 맵매칭이고, 맵매칭 알고리즘은 종종 차량이 가장 있을만한 도로의 위치를 찾아서 이러한 불일치를 수정해 주는 방법으로 사용된다. 맵매칭의 가장 편한 방법은 연속적인 측정값들과 도로 지도 정보에서 주어진 차량의 가능한 경로 중 가장 유사한 경로를 비교 하는 것과 같은 패턴매칭이다. 이 방법은 측정값을 저장하는 것이 필요하지만 측정값이 오차가 있는 경우에는 측정오차나 성능개선이 되지 않는다. 최근의 맵매칭은 track-splitting 필터, multiple hypothesis 추적[1], Markov localization[2], C-Measure[3], 그리고 다양한 구조의 Multiple Model 추적 [4] 등과 같이 다양한 방법을 사용하고 있다.

본 논문에서는 일반적으로 사용되고 있는 맵매칭 알고리즘을 분석하고 이러한 맵매칭에서 발생하는 단점을 보완하기 위한 보정벡터를 이용한 새로운 기법을 제안한다. 또한 이를 실시간 맵매칭에 적용시켜 구현함으로써 보정벡터를 이용한 맵매칭의 성능을 비교 검증하였다.

2. 맵 매칭 알고리즘

맵매칭 알고리즘은 일반적으로 기하학적인 맵 매칭 알고리즘과 맵 데이터의 토폴로지를 고려한 맵 매칭 방법으로 나뉘어 진다. 기하학적인 맵 매칭 알고리즘으로는 Point-to-Point 매칭 방법, Point-To-Curve 매칭 방법 그리고 각을 고려한 매칭 방법 등이 있다. 토폴로지를 고려한 알고리즘으로는 확률에 기반한 알고리즘과 퍼지이론에 기반한 알고리즘 등이 연구되고 있다.

2.1 Point-to-Point 매칭

Point-to-Point 매칭은 그림 1과 같이 GPS로부터의 측정점 P1과 맵 데이터(A1, A2, A3, B1, B2, B3, ...) 중 최소 거리를 갖는 맵 데이터를 매칭 시키는 방법이다. 즉, 그림에서 나타난 바와 같이 P1과 B2간의 거리가 최소거리이므로 P1입력은 도로B에 위치한 B2로 매칭 되게 된다. 이 방법의 가장 큰 특징은 구현하기 쉽고 연산속도가 빠른 장점을 갖는 것이다. 그러나 다음과 같은 문제점을 가지고 있다. 아래 그림에서 맵 데이터 B2가 존재하지 않는다면 P1과 A2간의 거리가 최소 거리가 되므로 차량의 실제위치가 도로B에 위치하는 경우에도 P1은 도로 A에 위치한 A2로 매칭 되게 된다. 결과적으로 차량의 정확한 위치파악에 어려움을 주게 된다.

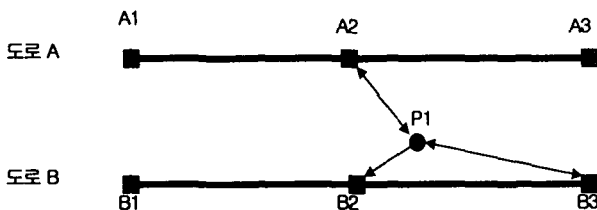


그림 1. Point-to-Point 매칭
Fig. 1. Point-to-Point matching

2.2 Point-to-Curve 매칭

Point-to-Curve 매칭 방법은 Point-To-Point 매칭 방법의 단점을 보완한 방법이다. GPS입력 좌표와 도로 간의 최소거리를 갖는 점을 생성하고 그 점에 매칭 시키는 방법이다. 그림 2와 같이 P1에 대하여 도로A와 B에 수직인 직선을 그어 수직선과 만나는 점들을 생성하고 최소거리를 갖는 도로를 선택하여 매칭 시킨다.

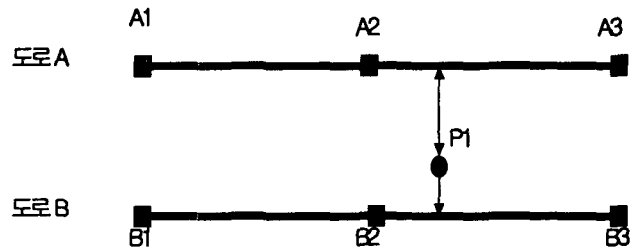


그림 2. Point-to-Curve 매칭
Fig. 2. Point-to-Curve matching

그러나 그림 3과 같은 상황에서 P1과 P3점은 도로B에 매칭이 되고 P2는 도로A에 매칭이 되어 매칭점들은 도로A와 도로 B를 오가게 되는 단점을 가지고 있음을 알 수 있다. 이러한 문제점은 Point-To-Point 매칭 방법에서도 같은 결과를 갖는다.

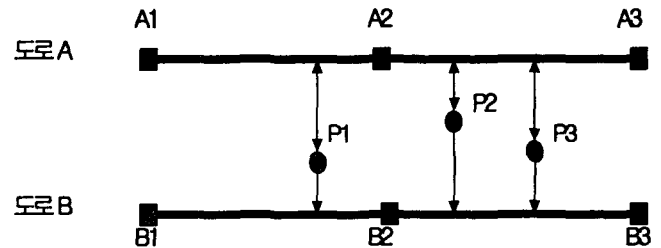


그림 3. Point-to-Curve 매칭의 문제점
Fig. 3. Problem in Point-to-Curve matching

2.3 도로와의 각을 고려한 매칭

도로와의 각을 고려한 매칭 방법은 GPS입력 P1, P2 두 점을 잇는 직선과 도로간의 각도 차를 고려하여 도로를 판별하는 방법이다. 그림 4와 같이 P1, P2를 잇는 직선과 도로A, 도로B와의 각도 차인 α 와 β 중 작은 값을 가지는 도로인 도로 A에 Point-To-Point 나 Point-To-Curve 매칭 방법으로 매칭 시킨다. 그림 5와 같은 상황에서 도로와의 각도만을 가지고 도로를 판단하기에는 쉽지 않다. 그리고 심한 굴곡을 갖는 형태의 도로나 비슷한 형태를 가지는 도로가 다수 존재할 때에는 매칭이 쉽지 않다. 그래서 각을 고려한 매칭은 Point-To-Point 나 Point-To-Curve 매칭 방법 등과 같이 사용된다.

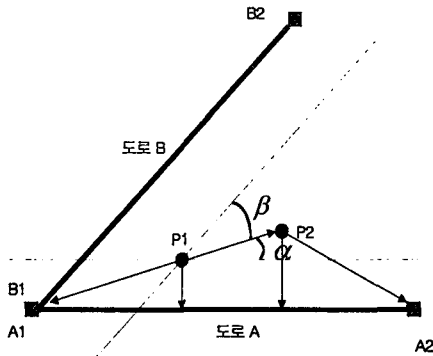


그림 4. 각을 고려한 매칭
Fig. 4. Matching using angle

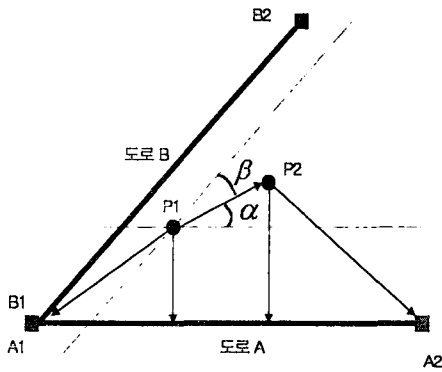


그림 5. 각을 고려한 매칭 문제점
Fig. 5. Problem in matching using angle

2.4 토폴로지를 고려한 매칭

토폴로지를 고려한 매칭 방법은 도로 네트워크의 토폴로지 정보를 바탕으로 매칭 가능한 후보 도로를 제한하고 매칭하는 방법이다. 이는 실제 도로에서 교차로에서 좌회전 불가능한 도로나 막혀 있는 도로 등 맵 데이터로부터 이미 알고 있는 정보를 바탕으로 응용될 수 있다.

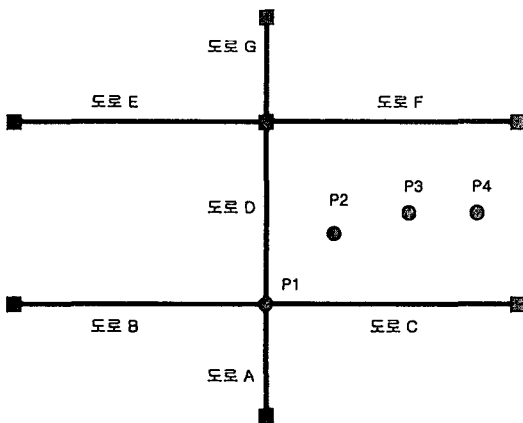


그림 6. 토폴로지를 고려한 매칭
Fig. 6. Matching using topology

그림 6에서 초기에 P1좌표에 있다는 것을 확인한다면 P2, P3, P4 GPS좌표는 교차로를 통과하지 않고 도로E, 도로F, 도로G로 이동할 수 없다는 것을 알고 있으므로 P2, P3, P4는 도로A, 도로B, 도로C, 도로D 중 하나의 도로에 매칭 되어야 한다.

토폴로지를 고려한 매칭 방법 중 C-measure 알고리즘[3]은 특정한 도로상에 차가 존재할 가능성이나 확률을 나타내는 값을 계산하여 모든 도로의 그 값을 비교함으로써 차가 존재하는 도로를 구별한다. 즉, *i*번째 도로의 C-measure는 *i*번째 도로상에 차가 위치할 가능성을 나타내고 *k*번째 입력에서 *i*번째 도로의 C-measure를 $C_i(k)$ 로 정의한다면 차가 위치하는 도로는 아래와 같이 결정된다.

IF $C_i(k) \geq C_i$,
THEN *i* 번째 도로에 차가 위치한다.
ELSE *i* 번째 도로에 차가 위치하지 않는다.
여기서 C_i 는 Threshold 값이다.

Threshold 값을 초과하는 도로가 한 개 이상 존재한다면 가장 큰 값을 가지는 도로를 선택한다. 그리고 일단 차가 위치하는 도로가 결정되면 Point-to-Curve 방법으로 도로에 매칭되는 알고리즘이다. C-measure 알고리즘은 식(1)과 같이 표현된다. 식(2)는 GPS입력좌표와 도로간의 거리에 관한 factor 이고 식(3)은 도로와 입력좌표가 이루는 각에 관한 factor이다.

$$C(k+1) = \alpha \cdot D(k) + \beta \cdot V(k) + \gamma \cdot C(k) \quad (1)$$

$$D(P_c, P_r) = \frac{1}{1 + \frac{\|P_c - P_r\|^2}{\sigma^2}} \quad (2)$$

$$V(V_c, V_r) = \frac{(V_c \cdot V_r)^2}{\|V_c\|^2 \|V_r\|^2} = \cos^2 \theta_\delta \quad (3)$$

$P_c = (x_c, y_c)$ 는 GPS입력 좌표, $P_r = (x_r, y_r)$ 은 도로상에 수직으로 투영된 좌표, V_c 는 GPS입력 좌표의 속도벡터, V_r 은 도로상에 수직으로 투영된 좌표의 속도벡터, σ 는 GPS입력 좌표의 표준편차 그리고 θ_δ 는 V_c 와 V_r 사이의 각도를 나타낸다. $C(k)$ 값은 모든 도로에 대해 계산되며 가장 높은 값을 가진 도로를 선택한다. 그림 7에서 Position fixing mode는 차가 위치하는 도로를 결정하는 과정이고 Tracking mode는 결정된 도로에 매칭시키는 과정이다. Position fixing mode에서는 선택된 도로에서 MaxCk 값이 3보다 크게 되면 현재의 도로에서 벗어났다고 판단하여 Tracking mode로 들어가 여러개의 도로중에서 매칭할 도로를 찾는 구조로 되어있다. 그리고 Tracking mode에서는 각각의 도로마다 $C(k)$ 를 구하고 각 도로중에서 $C(k) < C_i$ 를 만족하는 도로가 나타나면 매칭할 도로로 판단하여 Position fixing mode로 들어간다.

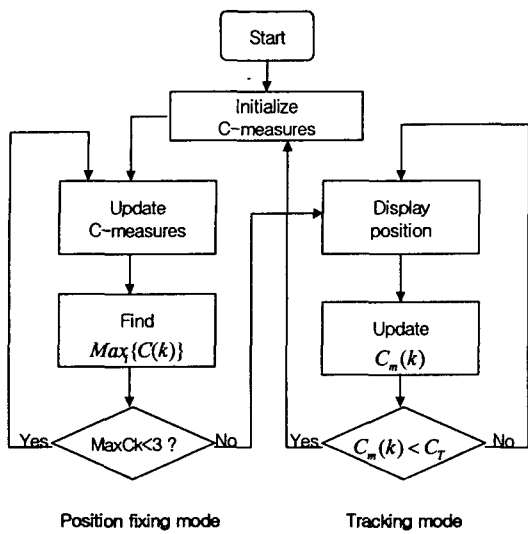


그림 7. C-measure 알고리즘 구조
Fig. 7. C-measure algorithm structure

3. 보정벡터

GPS 수신기에서 입력되는 GPS 좌표를 맵에 매칭하면 GPS 수신 좌표와 맵과의 오차가 발생하게 된다. 발생한 오차가 다음에 GPS 좌표로부터 입력될 데이터도 비슷한 오차를 가질 것을 알 수 있다. 본 논문에서 제안한 보정벡터는 먼저 매칭한 좌표와 입력 좌표 사이에 오차를 고려하여 보정벡터를 얻을 수 있다. 이렇게 얻어진 보정벡터는 새로운 입력 데이터를 맵에 매칭할 때 기하학적인 방법으로 도로에 매칭한다. 보정벡터를 구하는 방법의 예로 그림 8에서 GPS 입력 P_k 에 보정벡터 \vec{b}_{k-1} 를 더하고 이 값에 Point-to-Curve 방법으로 도로에 수직으로 투영한다. 투영하여 만나는 점 M_k 이 매칭점이 된다. 이때 보정벡터 \vec{b}_k 는 $M_k - P_k$ 가 된다. 다음 GPS입력 P_{k+1} 은 얻어진 보정벡터 \vec{b}_k 와 더하고 도로에 수직으로 투영하여 만나는 점 M_{k+1} 을 도로와 매칭점으로 결정한다. 이러한 과정을 계속 반복하면서 보정벡터는 계속 수정되고 오차를 줄여 나가게 된다.

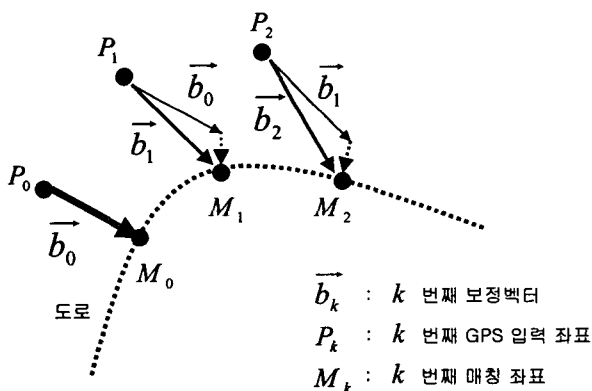


그림 8. 보정벡터의 원리
Fig. 8. Principle of compensation vector

그림 9와 그림 10은 갈림길에서 맵 매칭을 보여준다. 그림 9는 Point-to-Curve 매칭의 예를 보여주었고 그림 10은 보정벡터를 적용한 매칭을 보여준다. 일반적으로 맵 매칭은 GPS에서 수신된 위치 데이터와 맵과의 거리 그리고 도로와 이루는 각도에 대한 정보를 이용하여 매칭하게 된다. 그림 10과 같이 보정벡터를 적용하였을 때 구간 A에서는 보정벡터를 사용하지 않았을 때와 동일하다. 하지만 B구간과 같이 도로의 형태가 도로의 수직방향으로 평행이 되지 않으면 보정벡터는 기울어지게 된다. 따라서 구간 A에서 보정벡터는 도로의 수직성분 오차만을 보정하게 되지만 B구간을 통과하면서 도로의 수평성분 오차까지 보정벡터에 포함되게 됨을 알 수 있다. 이렇게 곡선도로를 통과하면서 생성된 보정벡터는 도로1에서 도로3으로 진입하는 구간C와 같이 인접한 갈림길에서의 도로결정 오류를 줄일 수 있다. B구간에서 보정벡터를 사용하지 않는 경우라면 도로와의 거리와 도로와의 이루는 각도 중에서 거리에 더 많은 가중치를 주는 경우는 차량이 진행하고 있지 않은 도로에 매칭 시킬 것이다. 그러나 보정벡터를 사용하는 경우에는 도로에 매칭시켜 오면서 오차성분을 알고 거기에 맞게 거리를 고려할 수 있음으로 더 효과적으로 매칭을 할 수 있다. 그림 10의 P16 GPS좌표는 도로3보다는 도로2에 더 가까워서 도로2에 매칭될 가능성이 높다. 그러나 보정벡터로 인하여 GPS 수신기로부터 입력된 데이터가 도로3에 더 가깝기 때문에 도로3에 매칭될 가능성이 높다. 따라서 그림 9의 C구간에서의 매칭에서는 도로 2와 도로3을 진동하지만 그림 10의 C구간에서의 매칭에서는 도로3으로 매칭된다.

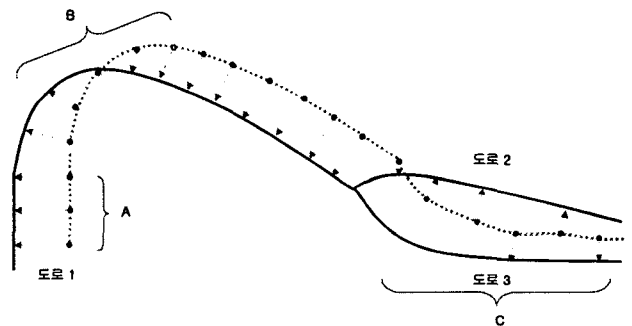


그림 9. Point-to-Curve 매칭
Fig. 9. Point-to-Curve matching

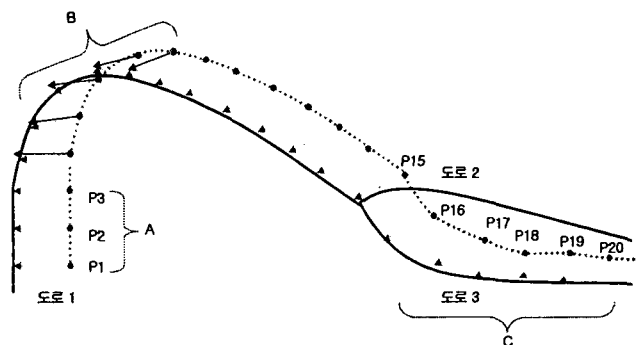


그림 10. 보정벡터를 적용한 매칭
Fig. 10. Matching using compensation vector

그림 11과 그림 12는 인접한 곡선 도로에서의 맵 매칭을 보여준다. 그림 11에서 보정벡터를 사용하지 않는 경우 구간 A에 매칭 되어야 할 GPS로부터 입력좌표가 구간 B로 매칭 되는 오류를 가지고 있음을 알 수 있다. 또한 구간B에서 물체의 이동방향과 매칭된 점들이 도로를 옮겨다니는 현상을 알 수 있다. 그러나 그림 11에서처럼 보정벡터를 적용하게 되는 경우는 이러한 문제를 해결할 수 있다.

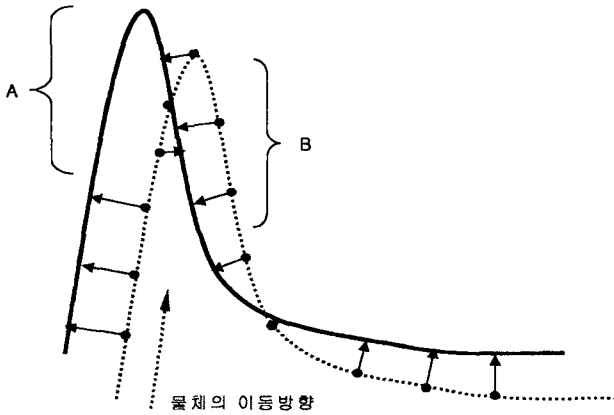


그림 11. Point-to-Curve
Fig. 11 Point-to-Curve

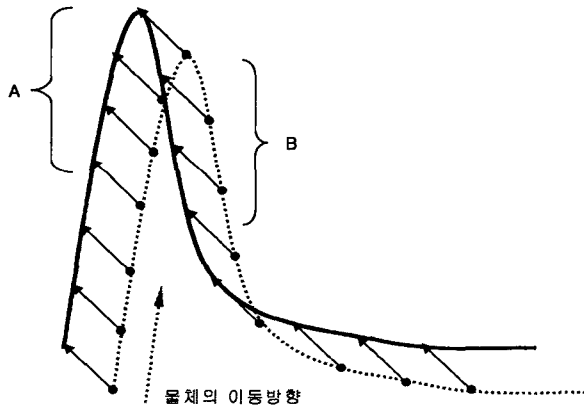


그림 12. Point-to-Curve 보정벡터를 적용한 매칭
Fig. 12 Point-to-Curve matching using compensation vector

4. 보정벡터를 이용한 맵 매칭

4.1 구현 알고리즘

구현 알고리즘은 여러 후보도로들 중에서 매칭할 도로를 결정하는 position fixing 모드와 결정된 도로에서 매칭을 하는 tracking 모드로 구성되어 있다. position fixing 모드에서는 입력된 GPS 좌표로부터 도로상에 차가 존재할 가능성이 나 확률을 나타내는 값을 계산하여 모든 도로의 그 값을 비교함으로써 차가 존재할 가능성이 있는 도로를 찾는다. 그리고 threshold 값을 초과하는 도로가 한 개 이상이 존재할 경우 가장 큰 값을 가지는 도로를 선택한다. 도로가 선택되면

tracking 모드에서는 position fixing 모드에서 찾아진 도로상에 매칭을 하는 부분이다.

$$D(M_c, M_r) = \frac{1}{1 + \frac{\|M_c - M_r\|^2}{\sigma^2}} \quad (4)$$

$$V(V_{mc}, V_{mr}) = \frac{(V_{mc} \cdot V_{mr})^2}{\|V_{mc}\|^2 \|V_{mr}\|^2} = \cos^2 \theta_\delta \quad (5)$$

$M_c = (x_c, y_c)$ 는 GPS입력 좌표에 보정벡터를 적용한 좌표, $M_r = (x_r, y_r)$ 은 입력 좌표에 보정벡터를 적용하여 도로상에 수직으로 투영된 좌표, V_{mc} 는 GPS입력 좌표에 보정벡터를 적용한 속도벡터, V_{mr} 은 입력 좌표에 보정벡터를 적용하여 도로상에 수직으로 투영된 좌표의 속도벡터, σ 는 GPS입력 좌표의 표준편차 그리고 θ_δ 는 V_{mc} 와 V_{mr} 사이의 각도를 나타낸다. 식(4), 식(5)식을 식(1)에 대입하여 $C(k)$ 값이 모든 도로에 대해 계산되며 가장 높은 값을 갖는 도로가 차량이 있을 확률이 가장 높은 것을 의미한다. 그림 13은 시스템 초기화, GPS 좌표 수신과 맵 매칭하는 과정을 보여주는 순서도이다.

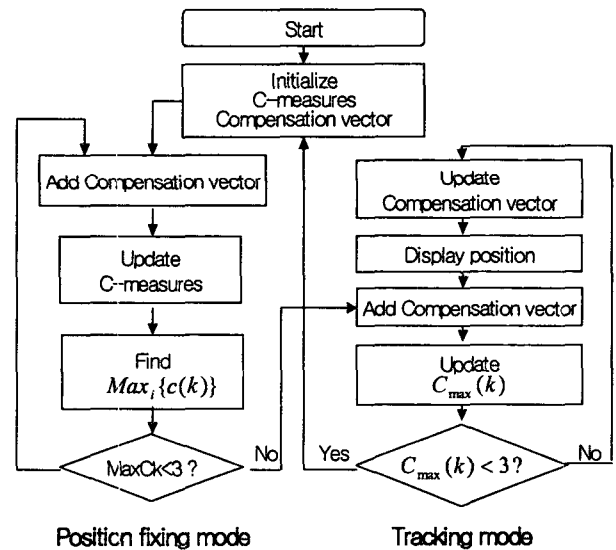


그림 13. 구현된 맵 매칭 알고리즘
Fig. 13. Proposed map matching algorithm

4.2 시스템 구현

시스템은 그림 14(a)와 같이 GPS 수신기, GPS 안테나, 전원, 그리고 컴퓨터로 구성되어 있다. 그림 14(b)는 시스템 구현에 사용된 GPS 수신기의 내부를 보여주고 있다. 구현 시스템에서 사용된 GPS 수신기는 여러 가지 GPS 데이터 포맷을 제공하는데, 본 실험에서 사용한 데이터 포맷은 NMEA(National Marine Electronics Association)의 GPRMC 포맷으로 차량의 위치정보, 속도, 그리고 방향정보를 나타내 주는 것을 사용하였다.

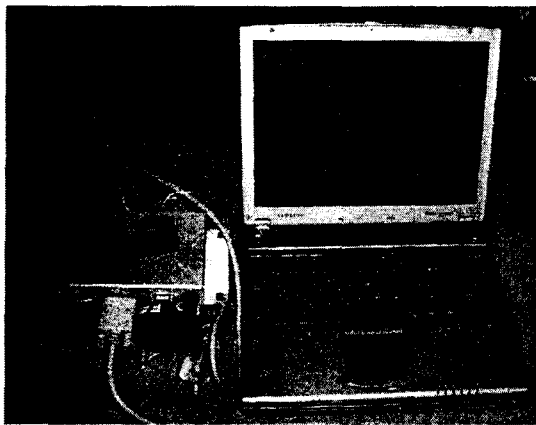
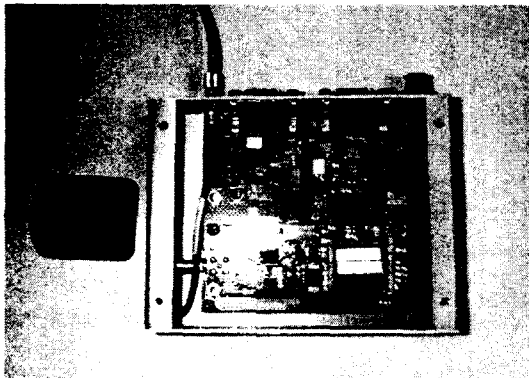
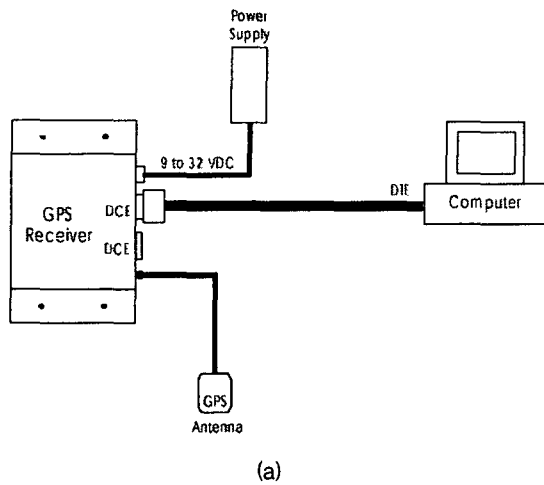


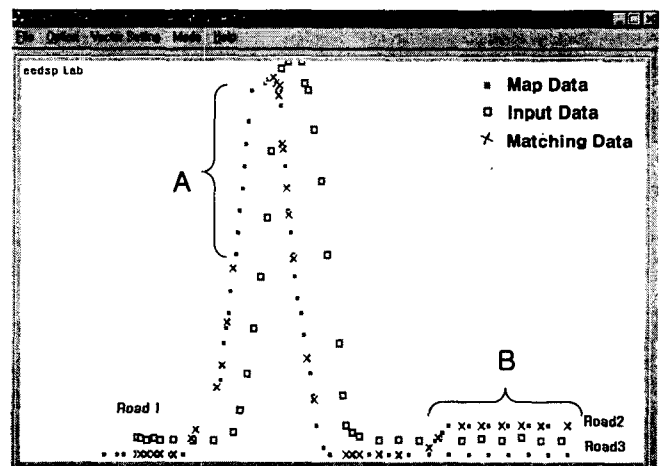
그림 14. GPS 수신기와 구현시스템
 (a) 시스템 블록도
 (b) GPS 수신기 및 안테나
 (c) 구현 시스템

Fig. 14. GPS receiver and implemented system
 (a) System block diagram
 (b) GPS receiver and antenna
 (c) Implemented system

알고리즘은 Visual C++를 사용하여 구현하였다. 그림 14(c)은 구현 시스템을 보여 주고 있다. 실험 방법은 그림 14의 시스템을 차량에 장착하여 차량을 운행하면서 실험하였다. 디지털 지도를 구할 수 없어서 실험은 2가지 단계로 실행되었다. 먼저 같은 도로를 여러번 운행하여 디지털 지도를 만들고 이 지도를 이용하여 맵매칭을 수행하는 과정으로 수행되었다.

4.3 실험결과

일반적으로 도로의 간격이 멀리 떨어져 있는 경우에는 매칭 알고리즘이 효과를 나타내지 못한다. 도로가 GPS 좌표 오차 범위 밖에 있을 경우에는 가까운 도로에 매칭을 하던 된다. 따라서 실험은 하나의 도로이지만 인접한 곳에서 커브가 있는 상황과 인접한 2개의 도로가 있는 갈림길 상황에서 실험하였다. 그림 15(a)는 C-Measure 알고리즘에 보정벡터를 적용하지 않았을 경우, 그림 15(b)는 C-Measure 알고리즘에 보정벡터를 적용하였을 경우에서 실험한 결과를 나타낸 것이다. 여기에서 '■'는 도로의 데이터 베이스에 있는 디지털 맵의 좌표를 보여주고 있고, '□'는 GPS로부터 입력된 좌표들을 표시하고, 'x'는 입력된 좌표로부터 매칭된 결과를 보여주고 있다. 여기에서 도로의 맵 데이터는 분기점을 기준으로 하여 3개의 맵 데이터로 구성 되어있다. 그림 15(a), 그림 15(b)에서 도로1의 구간 A를 비교해보면 보정벡터를 적용하였을 경우와 적용하지 않았을 경우에서 보정벡터를 사용하지 않았을 경우는 커브길의 오른쪽 방향에 매칭이 되었고 보정벡터를 적용하였을 경우에는 도로의 진행 방향에 매칭이 되어 보정벡터를 적용하였을 경우와 보정벡터를 적용하지 않았을 경우에 차이가 있음을 알 수 있다. 즉, 도로에 매칭을 하는 Tracking 모드 상황에서 보정벡터의 이점을 보여주고 있다. 또한 인접한 도로의 거리가 가까울 때 구간 B에서는 차량이 도로3으로 진행하고 있음에도 불구하고 도로3에 매칭 되지 않고 도로2에 매칭되는 현상이 나타났다. 하지만 보정벡터를 적용할 경우 실제 진행하고 있는 도로에 매칭이 되었음을 보여 주고 있다. 즉, 후보 도로들 중에서 매칭할 도로를 찾는 Position fixing 모드 상황에서 보정벡터의 이점이 있음을 알 수 있다.



(a)

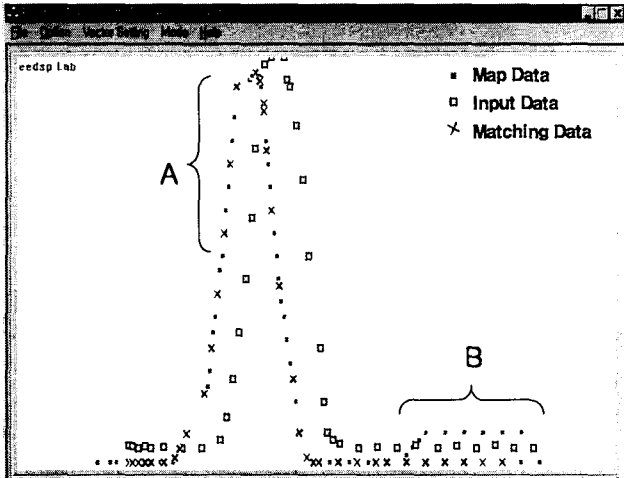
참 고 문 헌

[1] Jong-Sung Pyo, Dong-Ho Shin, and Tae-Kyung Sung, "Development of a map matching method using the multiple hypothesis technique", in Proc. IEEE ITS Conf., Oakland(CA), USA, pp. 23-27, August 2001.

[2] S. Thiebaux and P. Lamb, "Combining Kalman filtering and Markov localization in network-like environments", in Pacific Rim Intl. Conf. AI, pp. 756-766, 2000.

[3] S. Kim and J-H Kim, "Adaptive fuzzy-network-based C-measure map-matching algorithm for car navigation system", IEEE Trans. on industrial electronics, vol. 48, no. 2, pp. 442-441, April. 2001.

[4] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with variable structure IMM estimator", IEEE Trans. AES, vol 36, no. 1, pp. 26-46, 2000.



(b)

그림 15. 보정벡터 실험

- (a) 보정벡터를 적용하지 않는 C-Measure 알고리즘
- (b) 보정벡터를 적용한 C-Measure 알고리즘

Fig. 15. Compensation vector experiment

- (a) C-Measure algorithm without compensation vector
- (b) C-Measure algorithm with compensation vector

5. 결 론

본 논문에서는 일반적인 맵 매칭 알고리즘의 성능 향상을 위해 적용할 수 있는 보정벡터를 제안하였고, 실험을 통하여 타당성을 검증하였다. 검증하기 위한 실험은 C-Measure 알고리즘에 보정벡터를 적용하였을 경우와 적용하지 않았을 경우를 비교하여 실시하였다. 실험 결과로부터 도로와 도로 사이의 간격이 먼 곳에서는 보정벡터를 적용하지 않을 경우와 비슷한 결과를 보이는데 인접한 도로가 있는 부분에서 보정 벡터를 적용할 경우 맵 매칭에 더욱 효과적임을 알 수 있었다. 보정벡터 방법은 GPS 데이터 입력부분에 삽입하는 방법으로 일반적인 맵 매칭 알고리즘에 적용하기 쉽고 또한 선택적으로 적용될 수 있을 것이다. 또한 보정벡터는 연산량이 많지 않아 실시간으로 업데이트 되기 때문에 실시간 맵 매칭 알고리즘에 적용되기 쉬운 특성을 가지고 있다.

저 자 소 개



안 도 랑(安 壽 郎)

1972년 3월 3일생.
 1997년 동국대 공대 전기공학과 졸업.
 1999년 동 대학원 전기공학과 졸업(석사).
 1999년~ 현재 동 대학원 전기공학과
 박사과정
 Tel : (02)2260-3350
 E-mail : ahndr@dgu.edu



이 동 옥(李 東 旭)

1960년 10월 2일생.
 1983년 서울대 공대 전기공학과 졸업.
 1985년 동 대학원 전기공학과 졸업(석사).
 1992년 미국 조지아 공대 전기공학과
 졸업(공학박).
 1992년~1993년 삼성SDS(주) 선임연구원.
 1993년~현재 동국대 공대 전기공학과 부교수.
 Tel : (02)2260-3350
 E-mail : dlee@dgu.edu