

데이터베이스 기반 GPS 위치 보정 시스템

Database based Global Positioning System Correction

문 준 호¹, 최 혁 두¹, 박 남 훈², 김 중 희³, 박 용 운³, 김 은 태[†]

Junho Moon¹, Hyuk-Doo Choi¹, Nam-Hun Park², Chong-Hui Kim³,
Yong-Woon Park³, Euntai Kim[†]

Abstract A GPS sensor is widely used in many areas such as navigation, or air traffic control. Particularly, the car navigation system is equipped with GPS sensor for locational information. However, when a car goes through a tunnel, forest, or built-up area, GPS receiver cannot get the enough number of satellite signals. In these situations, a GPS receiver does not reliably work. A GPS error can be formulated by sum of bias error and sensor noise. The bias error is generated by the geometric arrangement of satellites and sensor noise error is generated by the corrupted signal noise of receiver. To enhance GPS sensor accuracy, these two kinds of errors have to be removed. In this research, we make the road database which includes Road Database File (RDF). RDF includes road information such as road connection, road condition, coordinates of roads, lanes, and stop lines. Among the information, we use the stop line coordinates as a feature point to correct the GPS bias error. If the relative distance and angle of a stop line from a car are detected and the detected stop line can be associated with one of the stop lines in the database, we can measure the bias error and correct the car's location. To remove the other GPS error, sensor noise, the Kalman filter algorithm is used. Additionally, using the RDF, we can get the information of the road where the car belongs. It can be used to help the GPS correction algorithm or to give useful information to users.

Keywords: Road database file (RDF), Kalman Filter, GPS Sensor correction, Mobile Robot Localization

1. 서론

최근 위치 인식 기술은 자동차 네비게이션이나 항공교통 제어 시스템 등 많은 분야에서 중요한 기술로 사용되고 있다. 이러한 위치 인식 기술로는 IMU 센서를 이용하여 위치를 예측하는 기술^[9]과 지상에 pseudo-lite를 설치하여 위치를 계산하는 기술^[10], 인공위성의 신호를 이용한 GNSS 기술 등이 있다. 하지만 IMU나 pseudo-lite와 같은 기술은 오차가 축적되거나 비싼 설

치비로 인하여 주로 사용되지 않고 있으며, 가격대 성능비가 좋은 GNSS 중 미국에서 개발한 GPS 시스템이 널리 사용되고 있다. 하지만 GPS 시스템 역시 단점을 가지고 있다. GPS 수신기가 위성의 신호를 받는 경우 잡음이 추가되기 때문에 정확한 위치 정보를 수신하기 어려우며, 특히 위성과 수신기 사이에 나무나 건물과 같은 장애물이 있거나 위성 위치가 넓게 퍼져 있지 않을 경우, DOP(Dilution of Precision)값이 증가하며, GPS 수신기는 편중 오차(bias error)를 생성시킨다^[11]. 따라서 GPS 센서는 위성의 신호를 제대로 받을 수 없는 특정한 상황에서 허용할 수 없을 만한 오차를 발생시킨다.

이러한 문제를 해결하기 위하여 지금까지 많은 연구자들이 다양한 연구를 수행하였다. GPS 센서의 위치를 보정하기 위해 일반적으로 이용하는 방법 중 하나는 IMU 센서 정보를 이용하여 위치를 보정하는 방법이다. ^{[11-13][12]} 자동차의 선속도나 각속도 등의 IMU 센

Received : May 24, 2012; Reviewed : Jun. 19, 2012; Accepted : Aug. 10, 2012

※ This project was funded by the Agency for Defense Development (ADD) under Grant UD100061HD.

[†] Corresponding author: Electrical & Electronic Engineering, Yonsei University, Shinchon-Dong, Seodaemun-Gu, Seoul, Korea (etkim@yonsei.ac.kr)

¹ Electrical & Electronic Engineering, Yonsei University (Junho87@yonsei.ac.kr, goodgodgd@yonsei.ac.kr)

² Computer Science, Anyang University (nmhnpark@anyang.ac.kr)

³ Agency for Defense Development (ADD) (chonghui.chkim@gmail.com, yongwoon5901@gmail.com)

서 정보와 GPS 센서 정보를 얻은 후, 파티클 필터를 사용한 MCL(Monte Carlo Localization) 알고리즘^[2]이나 칼만 필터 알고리즘^[1-3]을 이용하여 GPS 위치를 보정하는 연구가 많이 수행되었다. 이러한 방법은 GPS 센서만을 이용한 결과보다 정확도가 높지만 GPS 센서 정보에 많이 의존하게 되므로 GPS의 편중 오차를 효과적으로 제거하지 못한다. GPS, IMU, Pseudo-lite와 TLS(Terrestrial Laser Scanning)를 자동차에 모두 장착하여 위치를 보정하는 연구도 진행되어왔지만^[2], GPS 수신기만을 장착하는 방법보다 비용도 많이 들고 비효율적이다. 이 외의 방법으로는 이미지 센서나 LiDAR 센서, Radar 센서 등을 이용하여 위치를 인식하는 방법도 연구되어왔다^[4-6,8]. 하지만 센서 정보를 매칭하기 위해서는 실험할 지역의 센서 정보들을 미리 수집해 놓아야 하며, 매칭하기도 쉽지 않다는 단점이 존재한다.

이러한 문제를 해결하기 위하여 본 논문에서는 도로정보가 저장된 RDF (Road Database File)의 특징점을 이용하여 GPS의 편중 오차를 보정하려 한다. 연구를 수행하기 전에 실험할 지역의 도로 정보와 도로의 특징점 정보를 RDF로 만들어 저장한다. 이후 자동차가 GPS 센서를 가지고 이동하면서 칼만 필터를 이용하여 GPS 센서의 잡음을 제거하고, 주행하다가 정지선을 감지하면 정지선의 좌표를 RDF에서 검색하여 절대 좌표를 찾아 편중 오차를 보정하게 된다. 추가로 RDF의 다양한 정보들을 이용하여 사용자에게 현재 도로의 정보나 연결된 도로 정보, 주변의 건물이나 특징 정보들을 제공해 주게 된다.

본 논문의 구조는 다음과 같다. 2장에서는 GPS의 문제를 정의하며, 이 문제를 해결하기 위해 사용될 데이터베이스를 설계한다. 3장에서는 칼만 필터를 이용하여 GPS의 잡음을 제거하는 방법을 설명하며, 4장에서는 데이터베이스의 특징점 좌표를 이용하여 편중 오차를 보정하는 방법을 설명한다. 5장에서는 GPS 위치 보정 알고리즘을 이용하여 시뮬레이션을 진행한 영상과 그 결과를 분석하여 알고리즘의 성능을 검증한다. 끝으로 6장에서는 결론과 향후 연구 방향에 대하여 기술한다.

2. 문제 정의 및 데이터베이스 설계

2장에서는 GPS 신호에 추가되는 오차의 구성과 원인에 대해서 설명한다. 또한 이러한 오차를 제거하기

위해 본 논문에서 사용되는 도로 데이터베이스를 설계한다.

2.1 문제 정의

GPS는 위성의 신호를 수신하여 수신기의 위치 정보를 계산하는 시스템이다. GPS 위성은 신호를 지구상에 보내기만 하며 GPS 수신기는 그 신호를 받아서 처리하기만 하면 되기 때문에, GPS 시스템은 사용자의 숫자에 제한이 없다는 큰 장점을 가지고 있다. 이러한 장점 때문에 GPS 시스템은 항해나 국방 관련 분야 외에도 자동차 네비게이션 분야에서 특히 많이 사용되고 있다. GPS가 위치를 비교적 정확하게 인식하기 위해서는 4개 이상의 위성 신호를 전송받아야 한다. 하지만 실제 상황에서는 나무나 건물과 같은 장애물이 존재하거나 위성이 고르게 배열되지 않는 경우가 있다. 이러한 경우, GPS 위치는 편중 오차를 생성한다. 또한 신호 송수신시 잡음이 추가된다. 따라서 GPS 위치는 식 (1)과 같이 나타낼 수 있다.

$$GPS\ signal = location + bias\ error + noise \quad (1)$$

본 논문에서는 이러한 GPS 센서의 편중 오차와 잡음을 제거하기 위한 연구를 진행하였다. 다음 장에서는 이 문제를 해결하기 위해 필요한 도로 데이터베이스를 설계한다.

2.2 도로 데이터베이스 설계

RDF는 실제 도로의 정보, 즉 도로의 좌표나 차선의 좌표, 도로의 특성, 도로 간의 연결관계, 정지선 등 특징점의 위치 등을 데이터베이스에 저장하기 위해 설계한 모델이다. 이러한 정보들을 보다 효율적으로 관리하기 위하여 RDF의 구조를 설계하며, 도로 데이터의 저장, 관리뿐만 아니라 필요한 정보를 검색하여 제공하기 위하여 사용한다.

본 논문에서 사용하는 도로 데이터베이스는 그림 1과 같이 구성되어 있다. 도로는 Element라는 단위로 구성되어 있으며 이 테이블은 도로의 정보를 저장하고 있다. 또한 어느 도로에 자동차가 속해있는지를 검색하기 위하여 Polygon이라는 작은 단위로 나누었으며, 이 테이블은 좌표정보를 저장한다. 마찬가지로 차선 정보 검색을 위하여 Lane 테이블을 만들었으며, 차선의 좌표 정보를 저장하기 위하여 LineString 테이블을 구성

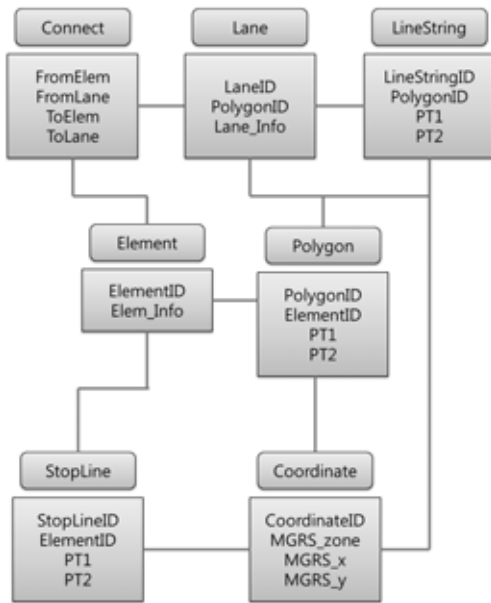


Fig. 1. The structures of RDF

하였다. 이 외에도 차선의 연결 정보를 저장하기 위한 Connect 테이블과 편중 오차를 보정할 특징점이 될 정지선 테이블을 추가하였다.

이러한 과정을 통하여 구성된 RDF에 도로 정보를 저장하여 도로 데이터베이스를 구성할 수 있다. 그림 2의 (a)와 같은 실제 도로를 그림 1과 같은 구조로 데이터베이스화하여 저장한 후, 도로 좌표와 차선의 좌표, 정지선의 좌표정보를 이용하여 그림 2의 (b)와 같이 출력할 수 있다.

이와 같이 설계된 RDF는 GPS의 편중 오차 보정 뿐만 아니라 사용자에게 유용한 정보를 제공하는 데 사용될 것이다.

3. 칼만 필터를 이용한 위치 추정

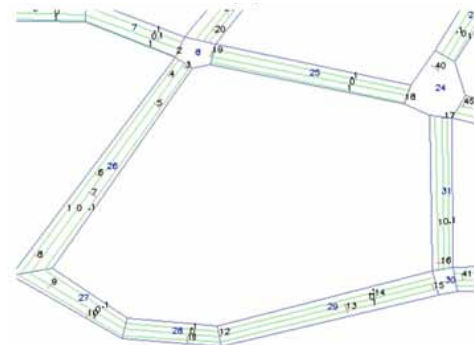
GPS 센서 정보는 크게 편중 오차와 잡음을 포함하고 있다. 본 논문에서는 잡음이 존재하는 시스템에서 1차적으로 현재 위치를 추정하기 위하여 칼만 필터 알고리즘을 사용하고, 그 후에 도로 데이터베이스의 특징점을 이용하여 추정한 위치에 포함된 편중 오차를 보정하는 방법을 사용한다. 본 장에서는 칼만 필터를 이용하여 잡음을 보정하는 방법을 먼저 설명하도록 하겠다.

GPS의 잡음을 제거하는 대표적인 방법은 칼만 필터 알고리즘을 사용하는 것이다⁷⁾. 칼만 필터 알고리즘은 잡음이 존재하는 시스템 내에서 현재 상태를 보다

정확하게 예측하기 위한 재귀적 알고리즘으로, 크게 예측 단계(Prediction Step)와 보정 단계(Correction Step)의 두 부분으로 구성되어 있다. 예측 단계에서는 이전 상태와 현재 속도, 각속도를 이용하여 현재 위치를 예



(a)



(b)

Fig. 2. (a) The real road and (b) road reconstructed by RDF

Table 1. Notation of variables used in the Kalman Filter algorithm

변수	설명
$x_t = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix}$	로봇의 현재 상태
$\mu_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$	칼만 필터를 통해 예측한 로봇의 현재 상태: 좌표 및 방향
$\bar{\mu}_t$	예측 단계를 통해 예측한 상태
$u_t = \begin{bmatrix} v_t \\ w_t \end{bmatrix}$	로봇의 선속도 및 각속도
$z_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix}$	센서의 측정 좌표 값
$\hat{z}_t = \begin{bmatrix} \hat{x}_t \\ \hat{y}_t \end{bmatrix}$	센서의 측정 예측 값

측하고, 보정 단계에서는 GPS 센서 값을 측정 값으로 사용하여 현재 위치를 보정하게 된다. 칼만 필터 알고리즘을 설명하기 위한 변수들의 표기법은 표 1에서 제시한다.

3.1 예측 단계

칼만 필터의 예측 단계에서는 선속도와 각속도를 입력 값으로 사용하여 현재 상태를 예측한다. 하지만 이 때 입력 값에는 잡음이 추가되기 때문에 식 (2)와 같이 실제 속도와 각속도는 값은 노이즈가 추가된 형태로 표현할 수 있다. 이 때 입력 값에 추가되는 가우시안 잡음의 평균은 0, 공분산은 M_t 로 가정한다. 본 논문의 시뮬레이션 환경에서는 $M_t = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$ 로 설정하였다.

$$\hat{u}_t = \begin{pmatrix} \hat{v}_t \\ \hat{w}_t \end{pmatrix} = \begin{pmatrix} v_t \\ w_t \end{pmatrix} + N(0, M_t) \quad (2)$$

또한 실제 로봇의 위치 x_t 는 실제 입력 값과 이전의 실제 위치를 이용하여 (3)과 같이 표현할 수 있으며, (4)와 같이 식을 전개할 수 있다.

$$x_t = g(x_{t-1}, \hat{u}_t) \quad (3)$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}_t}{\hat{w}_t} \sin \theta + \frac{\hat{v}_t}{\hat{w}_t} \sin(\theta + \hat{w}_t \Delta t) \\ \frac{\hat{v}_t}{\hat{w}_t} \cos \theta - \frac{\hat{v}_t}{\hat{w}_t} \cos(\theta + \hat{w}_t \Delta t) \\ \hat{w}_t \Delta t \end{pmatrix} \quad (4)$$

이 때 예측한 로봇의 현재 상태는 (5)와 같이 잡음을 분리할 수 있다.

$$x_t = g(x_{t-1}, u_t) + \varepsilon_t \quad (5)$$

이 때 식 (5)의 로봇의 상태에 대한 자코비안 G_t 는 (6)과 같이 계산할 수 있다.

$$G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}} = \begin{pmatrix} 1 & 0 & \frac{v_t}{w_t} (-\cos \mu_{t-1, \theta} + \cos(\mu_{t-1, \theta} + w_t \Delta t)) \\ 0 & 1 & \frac{v_t}{w_t} (-\sin \mu_{t-1, \theta} + \sin(\mu_{t-1, \theta} + w_t \Delta t)) \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

같은 방법으로 속도 입력에 대한 자코비안 V_t 는 식 (7)과 같이 표현할 수 있다.

$$V_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial u_t} = \begin{pmatrix} \frac{\sin \theta + \sin(\theta + w_t \Delta t)}{w_t} & \frac{v_t (\sin \theta - \sin(\theta + w_t \Delta t)) + v_t \cos(\theta + w_t \Delta t) \Delta t}{w_t^2} & \frac{v_t \cos(\theta + w_t \Delta t) \Delta t}{w_t} \\ \frac{\cos \theta - \cos(\theta + w_t \Delta t)}{w_t} & -\frac{v_t (\cos \theta - \cos(\theta + w_t \Delta t)) + v_t \sin(\theta + w_t \Delta t) \Delta t}{w_t^2} & \frac{v_t \sin(\theta + w_t \Delta t) \Delta t}{w_t} \\ 0 & \Delta t & 0 \end{pmatrix} \quad (7)$$

따라서 예측 단계에서 예측한 현재 로봇의 상태는 식 (8), (9)와 같이 계산할 수 있다.

$$\bar{\mu}_t = \mu_{t-1} + \begin{pmatrix} -\frac{v_t}{w_t} \sin \theta + \frac{v_t}{w_t} \sin(\theta + w_t \Delta t) \\ \frac{v_t}{w_t} \cos \theta - \frac{v_t}{w_t} \cos(\theta + w_t \Delta t) \\ w_t \Delta t \end{pmatrix} \quad (8)$$

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T \quad (9)$$

3.2 보정 단계

칼만 필터의 보정 단계에서는 측정된 GPS 센서의 값을 사용하여 예측했던 현재 상태를 보정한다. 이 단계에서도 측정 값에는 잡음이 추가되기 때문에 GPS 센서의 값은 식 (10)과 같이 표현할 수 있다. 이 때 GPS 센서 값에 추가되는 가우시안 잡음의 평균은 0, 공분산은 Q_t 로 설정하였다. 본 논문의 시뮬레이션 환경에서는 $Q_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ 로 설정하였다.

$$z_t = h(x_t) + \varepsilon_t = \begin{pmatrix} x \\ y \end{pmatrix} + N(0, Q_t) \quad (10)$$

이 때 x_i 에 대한 $h(x_i)$ 의 자코비안 H_i 는 식 (11)과 같이 계산된다.

$$H_i = \frac{\partial h(\bar{\mu}_i)}{\partial x_i} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (11)$$

따라서 칼만 필터 알고리즘의 칼만 이득은 (12)와 같이 계산할 수 있으며, 이 값을 이용하여 (13), (14)와 같이 현재 상태를 보정할 수 있다.

$$K_i = \bar{\Sigma}_i [H_i]^T [H_i \bar{\Sigma}_i [H_i]^T + Q_i]^{-1} \quad (12)$$

$$\mu_i = \bar{\mu}_i + K_i (z_i - \hat{z}_i) \quad (13)$$

$$\Sigma_i = (I - K_i H_i) \bar{\Sigma}_i \quad (14)$$

4. GPS 편중 오차의 보정

이번 장에서는 RDF에 저장된 정지선의 위치를 특징점으로 사용하여 GPS의 편중 오차를 보정하는 알고리즘을 설명한다. 자동차에 설치된 센서에서 정지선의 양 끝점을 인식할 수 있다고 가정하며, 이 때 RDF에 저장된 많은 정지선 좌표 중 인식한 정지선과 맞는 정지선을 검색한다. 위의 과정을 통하여 얻은 자동차의 상대 좌표계상의 정지선 좌표와 절대 좌표계상의 정지선 좌표를 이용하여 실제 자동차의 위치를 알아낸 후, GPS 좌표에 추가된 편중 오차를 계산하여 보정하게 된다.

4.1 정지선 매칭 알고리즘

자동차에 장착된 이미지 센서 등으로 감지한 정지선과 데이터베이스 내의 정지선을 매칭하기 위해서, 기본적으로 주위 정지선과 이전 단계에 예측한 자동차와의 거리를 계산하여 가장 가까운 정지선을 선택하게 된다. 하지만 바로 옆 차선의 정지선이 앞의 정지선보다 더 가까이 있거나 예측한 자동차 위치에 잡음이 추가되어 제대로 정지선이 매칭되지 않을 경우 더 큰 오차가 발생하며, 제대로 알고리즘이 진행되지 않게 된다. 따라서 본 논문에서는 정지선 매칭 알고리즘에 반대 차선의 정지선을 배제하는 알고리즘을 추가하였다.

반대 차선을 배제하기 위해서는 먼저 기준이 되는

중앙선의 두 점을 자동차의 방향과 맞추는 것이 중요하다. 그림 3과 같이 자동차의 방향(θ)과 정지선의 방향(θ_c)을 비교하여 만약 $|\theta_c - \theta| < \pi/2$ 라면 같은 방향이므로 P1과 P2의 순서를 그대로 두고, 아니면 순서를 바꾸어, 어느 상황에서도 자동차 앞의 방향에 점 P1이 오도록 저장한다.

중앙선 좌표의 순서를 고정하였으면, 자동차는 중앙선의 오른쪽 방향으로 이동하므로 중앙선을 기준으로 오른쪽 방향의 정지선만 고려하면 된다. 정지선이 차선의 오른쪽에 있는 경우는 그림 4 (a)와 같이 자동차가 오른쪽 방향으로 진행하는 경우와, (b)와 같이 왼쪽으로 진행하는 경우를 고려하면 된다.

먼저 그림 4 (a)와 같이 $P1_x > P2_x$ 인 경우에는 중앙선의 아래쪽 방향에 존재하는 정지선 S를 고려해야 하므로 부등식 (15)를 만족하는 정지선을 찾아야 한다.

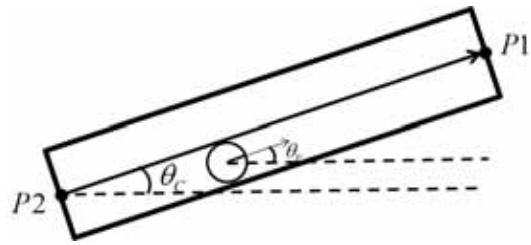


Fig. 3. The process of ordering center points

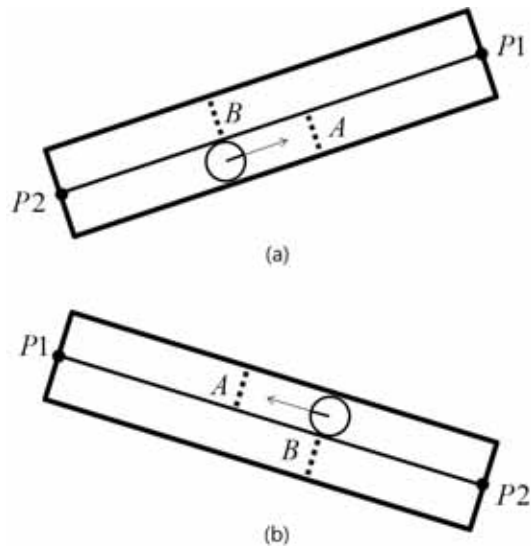


Fig. 4. The 2 kinds of situations for a driving vehicle. (a) A vehicle driving from left side to right side and (b) the vehicle driving from right side to left side

$$S_y - P1_y < \frac{P2_y - P1_y}{P2_x - P1_x} (S_x - P1_x) \quad (15)$$

$P1_x > P2_x$ 의 조건을 이용하여 부등식 (15)를 정리하면 부등식 (16)과 같은 결과를 얻을 수 있다.

$$(S_y - P1_y)(P2_x - P1_x) > (P2_y - P1_y)(S_x - P1_x) \quad (16)$$

같은 방법으로 그림 4 (b)와 같이 $P1_x > P2_x$ 인 경우, 중앙선의 위쪽 방향에 존재하는 정지선 \blacktriangleleft 를 고려해야 하므로 부등식 (17)을 만족하는 정지선을 찾아야 한다.

$$S_y - P1_y > \frac{P2_y - P1_y}{P2_x - P1_x} (S_x - P1_x) \quad (17)$$

$P1_x > P2_x$ 의 조건을 이용하여 부등식 (17)을 정리하면 부등식 (18)과 같은 결과를 얻을 수 있다.

$$(S_y - P1_y)(P2_x - P1_x) > (P2_y - P1_y)(S_x - P1_x) \quad (18)$$

그러므로 두 부등식 (16)과 (18)은 (19)와 같이 하나의 부등식으로 표현할 수 있다.

$$(S_y - P1_y)(P2_x - P1_x) > (P2_y - P1_y)(S_x - P1_x) \quad (19)$$

따라서 위의 식 (19)를 정지선 검색 알고리즘에 적용하면 현재 자동차가 지나고 있는 도로의 정지선을 기준으로 반대편의 정지선을 고려하지 않을 수 있다. 또한 부등식 (19)는 예측한 자동차의 위치와는 관계 없이 자동차의 방향을 중앙선에 맞춘 뒤 중앙선 좌표를 이용하기 때문에 예측한 자동차의 위치가 오차로 인하여 실제 위치에서 벗어나더라도 알고리즘에는 문제가 없다는 장점이 있다.

4.2 편중 오차 보정 알고리즘

도로 데이터베이스에서 원하는 정지선의 좌표를 검색하였으면, 그 좌표와 자동차가 인식한 정지선 좌표를 이용하여 자동차의 위치를 보정하여야 한다. 하지만 문제는 자동차가 인식한 정지선의 좌표계와 도로 데이터베이스 내의 정지선 좌표계가 다르다는 점과, 정지선의

두 점의 순서가 다를 수 있다는 점이다. 따라서 다른 두 좌표계를 고려하고, 두 점의 기울기만을 이용하여 위치를 보정하고자 한다. 편중 오차 보정 알고리즘을 설명하기에 앞서 먼저 알고리즘에 사용되는 변수들을 표 2에 정리하였다.

자동차의 위치 좌표를 보정하기에 앞서 먼저 각도를 보정한다. 그림 5는 각도를 보정하는 과정을 그림으로 표현해주고 있다. 그림과 같이 인식한 정지선의 절대 좌표의 기울기를 α , 자동차의 상대 좌표계에서 정지선의 기울기를 β 라 하면, α 와 β 는 식 (20)과 같이 표현된다. 이 때 $P1, P2$ 는 절대 좌표계에서 정지선 양 끝의 좌표를 순서 없이 저장한 좌표이며, $P1', P2'$ 는 상대 좌표계에서 정지선 양 끝의 좌표를 순서 없이 저장한 좌표이다.

$$\begin{cases} \alpha = \arctan\left(\frac{P2_y - P1_y}{P2_x - P1_x}\right) \\ \beta = \arctan\left(\frac{P2'_y - P1'_y}{P2'_x - P1'_x}\right) \end{cases} \quad (20)$$

Table 2. Notation of variables used in the bias correction algorithm

변수	설명
α	절대 좌표계에서 정지선의 기울기
β	상대 좌표계에서 정지선의 기울기
ρ	상대 좌표계에서 정지선 중심과 로봇 사이의 거리
ϕ	상대 좌표계에서 로봇의 방향을 기준으로 정지선 중심 좌표와의 각도 (그림 6 참조)
$P1, P2$	절대 좌표계에서 정지선 양 끝점 좌표
$P1', P2'$	상대 좌표계에서 정지선 양 끝점 좌표
C_x, C_y	절대 좌표계에서 정지선 중심 좌표
C_x', C_y'	상대 좌표계에서 정지선 중심 좌표
$\bar{\mu} = \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{\theta} \end{pmatrix}$	칼만 필터 알고리즘에 의해 예측한 현재 로봇의 상태 정보
$V = \begin{pmatrix} V_x \\ V_y \\ V_\theta \end{pmatrix}$	편중 오차 보정 알고리즘에 의해 계산된 현재 로봇의 상태 정보
\bar{x}	칼만 필터의 결과에 편중 오차 보정값을 더해 최종적으로 예측한 로봇의 상태 정보

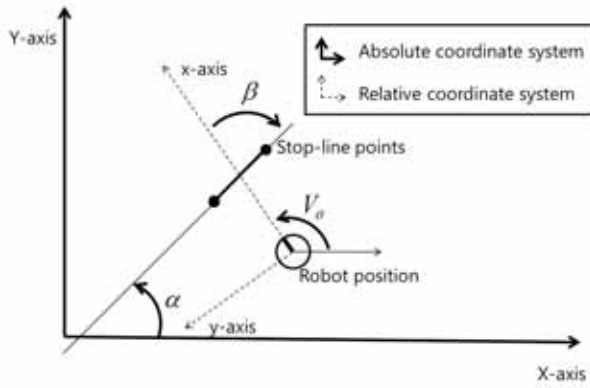


Fig. 5. The vehicle direction calculation process

식 (20)에 의해 계산된 α, β 의 좌표를 이용하면 자동차의 각도(θ)를 식 (21)과 같이 계산할 수 있다.

$$V_{\theta} = \alpha - \beta \quad (21)$$

자동차의 방향이 결정되면 이를 이용하여 위치를 결정한다. 그림 6은 자동차의 위치를 보정하는 과정을 그림으로 표현하고 있다. 먼저 센서에서 얻은 상대 좌표계의 정지선 좌표를 이용하여 그림 6의 $C'(x,y)$ 와 같이 정지선의 중심점을 계산하고, 식 (22)를 이용하여 정지선의 중심점 좌표와 자동차 위치와의 거리인 ρ 와 자동차 방향과의 각도차이인 θ 를 계산한다.

$$\begin{cases} \rho = \sqrt{C'_x{}^2 + C'_y{}^2} \\ \phi = \arctan\left(\frac{C'_y}{C'_x}\right) \end{cases} \quad (22)$$

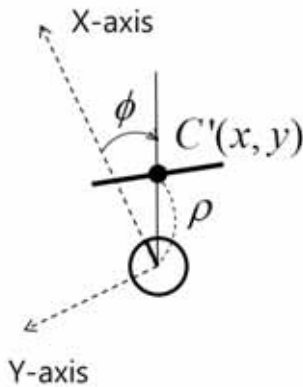


Fig. 6. Definition of ρ and ϕ

ρ, ϕ 을 계산하면 현재 자동차의 위치는 식 (23)과 같이 계산된다. 이 때 C_x, C_y 는 절대좌표계에서 정지선의 중심점이며, θ 는 자동차의 각도이다.

$$\begin{cases} V_x = C_x - \rho \cos(\phi + V_{\theta}) \\ V_y = C_y - \rho \sin(\phi + V_{\theta}) \end{cases} \quad (23)$$

따라서 편중 오차 보정 알고리즘을 통하여 칼만 필터를 이용한 위치 정보와 정지선 인식을 통한 위치 정보가 얼마나 차이가 나는지를 식 (24)와 같이 계산할 수 있다.

$$\begin{cases} e_x = V_x - \bar{x} \\ e_y = V_y - \bar{y} \\ e_{\theta} = V_{\theta} - \bar{\theta} \end{cases} \quad (24)$$

식 (24)에서 계산한 자동차의 위치 보정값(e)은 다음 정지선이 인식되기 전까지 칼만 필터 알고리즘에 의하여 계산된 자동차 상태 값에 추가되어 식 (25)와 같이 위치를 보정하게 된다. 다음 정지선을 인식하면 위치가 다시 업데이트된다.

$$\bar{x} = \bar{\mu} + e \quad (25)$$

5. 시뮬레이션 결과

이번 장에서는 GPS의 오차를 제거하는 알고리즘을 이용하여 시뮬레이션한 후 결과를 통하여 GPS 위치가 보정됨을 증명할 것이다. 그림 7은 전체 시뮬레이션 과정을 보여준다.

시뮬레이션 과정은 크게 3가지(Simulator, RDF, Output)로 나누어진다. 먼저 사용자가 자동차가 이동할 경유점을 정해주면 Simulator에서 각 Step마다 로봇이

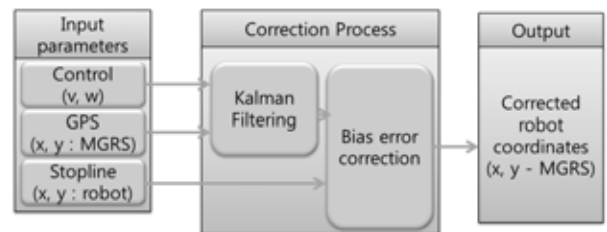


Fig. 7. The simulation process

움직인 위치를 계산하며, 잡음이 추가된 속도와 각속도를 계산한다. 추가로 이 위치에 편중 오차와 잡음을 추가하여 가상 GPS 신호를 만든다. 만약 자동차가 이동하다가 정해진 범위 내에서 정지선을 감지하면 그 상대 좌표 역시 계산하게 된다. 끝으로 계산한 속도, 각속도, GPS좌표, 정지선 좌표를 RDF단으로 전송한다.

그림 8은 MFC환경에서 출력된 시물레이션 화면을 보여준다. 그림 8의 왼쪽 지도에서 사용자는 도로에 원하는 경유점을 설정할 수 있으며, 그림 오른쪽의 Start 버튼을 클릭하면 설정한 경유점을 따라 자동차가 움직이면서 실제 좌표와 GPS좌표, 속도, 정지선과의 거리 등을 계산하여 RDF단으로 전송한다.

RDF 단에서는 3장에서 소개한 칼만 필터 알고리즘과 편중 오차 보정 알고리즘을 이용하여 Simulator로부터 받은 GPS좌표를 보정하게 된다. 이 결과 GPS의 잡음과 편중 오차는 보정되며 보정된 좌표는 출력단으로 보내진다. RDF 부분에서는 추가적으로 도로의 연결 정보를 검색하거나 주위의 특징 좌표를 검색하는 역할도 수행한다.

출력단에서는 RDF로부터 받은 보정된 GPS좌표를 사용자에게 보여준다. 추가로 RDF에서 검색한 연결된 도로 정보 등의 유용한 정보 역시 그림 9의 오른쪽 화면에 보여준다. 사용자는 이 정보를 이용하여 현재 도로의 상태와 어느 도로로 이동할 수 있는지 등의 정보

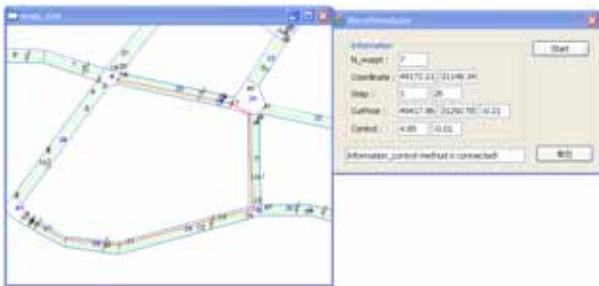


Fig. 8. The simulation window

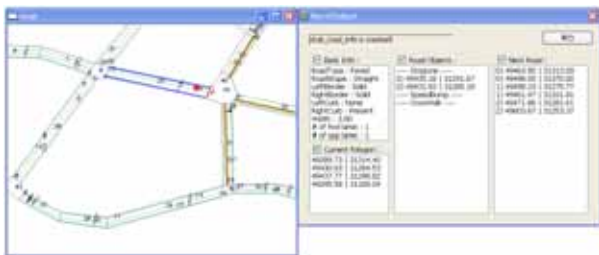


Fig. 9. The output window

를 쉽게 얻을 수 있다.

그림 10은 위의 시물레이션 프로그램을 통하여 얻은 자동차의 실제 경로와 GPS 경로, 보정한 결과의 경로를 표시하였다. 그림 10 (a)의 파란색 경로는 사용자가 임의로 지정한 경유점을 따라 움직인 경로이며, 진한 검정색 경로는 자동차의 위치에 편중 오차와 잡음을 추가시킨 GPS 경로이다. 본 실험에서는 편중 오차를 각 축마다 증가다가 감소하도록 정하였으며 잡음은 가우시안으로 정하였다. 따라서 실험에서 만든 GPS 좌표는 그림 10 (a)의 진한 검정색 경로와 같이 도로를 벗어나 움직이게 된다. 하지만 복원한 좌표는 그림 10 (b)의 진한 빨간 점선과 같이 칼만 필터 알고리즘에 의해 노이즈가 보정되고, 정지선을 감지할 때마다 편중 오차가 보정되어 (a)의 GPS 좌표와 같이 차선을 벗어나지 않고 따라가는 것을 확인할 수 있다. 그림 10 (b)의 보라색 원은 10 step마다 공분산 행렬을 가시화한 것이다.

그림 11은 보정 전의 GPS 좌표 오차(빨강)와 보정 후의 GPS 좌표 오차(파랑)를 보여준다. 보정 전에는 빨간 그래프와 같이 오차를 증가하다가 감소하도록 설



(a)



(b)

Fig. 10. The simulation results

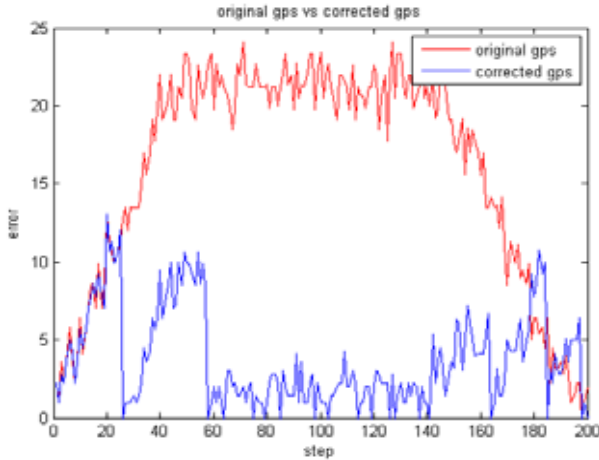


Fig. 11. The error between original and corrected GPS

정하였지만, 보정 후의 좌표는 파란 그래프와 같이 정지선을 만날 때마다 위치가 보정되어 주기적으로 에러가 감소하는 것을 쉽게 확인할 수 있다. 이는 정지선을 인식하면 수행되는 편중 오차 보정 알고리즘이 확실하게 동작하는 것을 의미한다.

다음으로 실제 GPS 신호에서도 본 논문에서 제시한 알고리즘이 동작하는지 확인하기 위하여 같은 지역에서 GPS를 장착한 자동차를 주행시켜 얻은 GPS 좌표를 이용하여 실험하였다. 본 실험에서 사용한 GPS 좌표는 국방과학연구소에서 제공하였다. 그림 12 (a)의 경로는 실제로 얻은 GPS 좌표를 화면에 표시한 결과이다. 주변에 장애물이 많지 않아서 노이즈는 심하지 않지만, 편중 오차에 의하여 도로 경로를 벗어나거나 반대 차선에서 나타나는 것을 확인할 수 있다. 하지만 본 논문에서 제시한 알고리즘에 의하여 보정한 결과인 그림 12 (b)의 경로에서는, 정지선이 인식되는 경우 그 위치가 보정되어서 올바른 차선으로 이동하는 것을 확인할 수 있다. 그림 12 (b) 내에 표시한 원의 중심에 나타난 경로를 보면, 자동차가 진행하면서 인식한 정지선의 좌표 정보에 의해 편중 오차가 보정되는 것을 쉽게 확인할 수 있다.

위의 두 시뮬레이션 결과를 통하여 본 논문에서 제시한 GPS 보정 알고리즘을 이용하면, 신호의 노이즈나 위성의 배열 등으로 생기는 에러에 의해 도로를 벗어나거나 반대 차선에 나타나는 GPS 신호를 특징점인 정지선의 좌표를 인식하여 보정할 수 있다는 것을 확인할 수 있다.

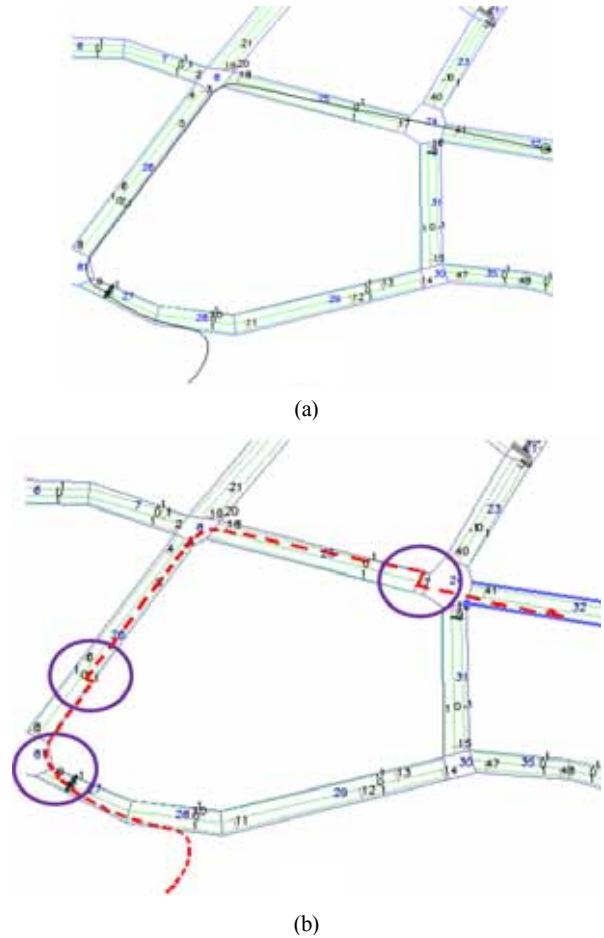


Fig. 12. The simulation results using real GPS signal

6. 결론

GPS 센서 정보는 편중 오차와 잡음을 가지고 있기 때문에 건물 사이로 이동하거나 주위에 장애물이 존재하는 경우 위치를 정확히 추정하기 어렵다. 이러한 문제를 해결하기 위하여 본 논문에서는 칼만 필터와 RDF의 정지선 좌표를 특징점으로 사용하여 편중 오차와 잡음을 제거하고자 하였다. 칼만 필터 알고리즘은 매 스텝마다 수행되며 편중 오차 보정 알고리즘은 정지선을 인식하면 수행된다. 두 가지 실험을 통하여 제안한 알고리즘을 이용하여 GPS의 오차를 제거할 수 있음을 확인하였으며, 데이터베이스를 확장하여 사용자에게 유용한 정보를 추가로 제공할 수도 있음을 시뮬레이션을 통하여 보여주었다.

본 논문에서 제시한 알고리즘은 특징점이 제대로 인식되어야 한다는 가정이 존재하며, 실험 전에 도로 정보가 저장된 RDF를 만들어야 한다는 단점이 존재한다.

다. 하지만 이 두 가지 조건이 충족된다면, 현재 네비게이션에서 GPS 위치를 보정하기 위해서 사용하는, 자동차 바퀴의 회전을 계산하여 위치를 추정하는 기술의 단점인 오차가 갈수록 누적된다는 점을 해결해 줄 알고리즘이 될 수 있을 것이다.

참고문헌

[1] R. L. Greenspan, "GPS and inertial integration", Global Positioning System: Theory and Applications, vol. 164, pp.187-218, 1996.

[2] D. A. Grejner-Brzezinska, R. Da, and C. Toth, "GPS error modeling and OTF ambiguity resolution for high-accuracy GPS/INS integrated system", Journal of Geodesy, vol. 72, no.11, pp.628-638, Nov, 1998.

[3] J. Wendel, O. Meister, C. Schlaile, and G. F. Trommer, "An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter", Aerospace Science and Technology, vol. 10, pp.527-533, Sep, 2006.

[4] Patrick J. F. Carle, Paul T. Furgale, Timothy D. Barfoot, "Long-range rover localization by matching LIDAR scans to orbital elevation maps", Journal of Field Robotics, vol. 27, pp.344-370, June, 2011.

[5] F. Viani, P. Rocca, G. Oliveri, D. Trincherro, A. Massa, "Localization, tracking, and imaging of targets in wireless sensor networks : An invited review", Radio Science, vol. 46, 2011.

[6] S. Cho, J. Lee, "Localization of a high-speed mobile robot using global features", Robotica, vol. 29, pp.757-765, September, 2011.

[7] H. Bian, Z. Jin, W. Tian "IAE-adaptive Kalman filter for INS/GPS integrated navigation system", BIAI Journals & Magazines, vol. 17, pp.502-508, 2006.

[8] J. Moon, H. Choi, N. H. Park, Y. W. Park, C. H. Kim, and E. Kim, "LiDAR sensor matching system using database", International Conference on Automation and Control Engineering, pp.995-998, January, 2012.

[9] 김상섭, 진용, 조성운, 박찬국, 지규인, 이영재, "저

급 IMU의 오차 보정", 항공우주학회 춘계학술발표회 논문집, pp.197-200, 2000.

[10] 김종원, 이택진, 소형민, 전상훈, 김강호, 기창돈, "멀티 안테나와 반송파위상 측정치를 이용한 의사 위성 위치 측정 방법", 한국항공우주학회 춘계학술대회, pp.614-620, 2011.

[11] 차득기 저, "GPS 측량이해", 성림출판사, May, 2006.

[12] 노성우, 김태균, 고낙용, 배영철, "이동로봇의 GPS 위치 정보 보정을 위한 파티클 필터 방법" 한국전자동신학회 논문지 7권 2호, pp.381-389, 2012.



문준호

2010 연세대학교 전기전자공학(공학사)
2010~현재 연세대학교 전기전자공학과 석사과정
관심분야: SLAM, 데이터베이스



최혁두

2009 연세대학교 전기전자공학(공학사)
2009~현재 연세대학교 전기전자공학과 통합과정
관심분야: 로봇틱스, 인공지능, SLAM



박 남 훈

2002 연세대학교 컴퓨터과학
과(공학석사)
2007 연세대학교 컴퓨터과학
과(공학박사)
2007~2008 연세대학교 데이
터베이스 국가지정 연구
실 연구원

2008~2010 Worcester Polytechnic Institute Research
Associate

관심분야: 데이터베이스, 데이터마이닝, 실시간 데
이터 스트림



박 용 운

1994 University of Utah 기계공
학과 박사
1982~현재 국방과학연구소
책임연구원 무인자율화
기술 부장
관심분야: 자율주행, SLAM,
차량제어



김 종 희

2007 한국과학기술원 전자전
산학부 전기 및 전자공학
전공 박사
2008~현재 국방과학연구소
선임연구원
관심분야: 최적제어, 자율주행



김 은 태

1992 연세대학교 전자공학과
(공학사)
1994 연세대학교 전자공학과
(공학석사)
1999 연세대학교 전자공학과
(공학박사)

1999~2002 국립환경대학교 제어계측공학과 조교수
2008 BISC (Berkeley Initiative in Soft Computing), UC
at Berkeley, USA

2002~현재 연세대학교 전기전자공학부 교수
관심분야: Computational Intelligence, 지능형 로봇