

AVM 정지선인지기반 도심환경 종방향 측위보정 알고리즘

김종호* · 이현성* · 유진수* · 이경수**†

AVM Stop-line Detection based Longitudinal Position Correction
Algorithm for Automated Driving on Urban Roads

Jongho Kim*, Hyunsung Lee*, Jinsoo Yoo*, Kyongsu Yi**†

Key Words : Automated Driving(자율주행), Position Correction(위치 보정), Stop-line Detection(정지선 인지), ROS(로봇 운영체제)

ABSTRACT

This paper presents an Around View Monitoring (AVM) stop-line detection based longitudinal position correction algorithm for automated driving on urban roads. Poor positioning accuracy of low-cost GPS has many problems for precise path tracking. Therefore, this study aims to improve the longitudinal positioning accuracy of low-cost GPS. The algorithm has three main processes. The first process is a stop-line detection. In this process, the stop-line is detected using Hough Transform from the AVM camera. The second process is a map matching. In the map matching process, to find the corrected vehicle position, the detected line is matched to the stop-line of the HD map using the Iterative Closest Point (ICP) method. Third, longitudinal position of low-cost GPS is updated using a corrected vehicle position with Kalman Filter. The proposed algorithm is implemented in the Robot Operating System (ROS) environment and verified on the actual urban road driving data. Compared to low-cost GPS only, Test results show the longitudinal localization performance was improved.

1. 서론

자율주행 자동차의 상용화에 있어서, 측위 결과의 정확도와 정밀도는 매우 중요하다. 특히 목표 주행경로를 설정하고 이를 따라 주행하는 경로 추종주행은 도심도로에서의 자율주행 상황에서 대부분을 차지하는 시나리오이며, 차량제어를 측위결과에 크게 의존한다.

Real Time Kinematic(RTK) GPS는 실시간 기준의 GPS가 사용하는 위성신호와 더불어 지상의 여러 기준국 안테

나 신호를 받으며 측위 결과를 보정하는 고정밀 측위장비이다. 오차가 수 cm 이하로 높은 측위 정확도를 나타내는 RTK GPS는 많은 자율주행 연구에 측위장비로써 사용되고 있다. 하지만 매우 높은 가격에 판매되고 있기 때문에, 양산을 위한 자율주행차량에 장착되기에 어려움이 존재한다. 그에 비해 저가형 GPS(Low-cost GPS, LC GPS)는 현재 양산되고 있는 많은 승용 및 상용차량에 장착되어 있다. 이는 네비게이션 및 긴급구조신호전달 등에 사용되기 위한 목적으로 양산차량에 적용될 수 있는 가격적 이점을 가지고 있지만 1~3m 수준의 높은 오차범위를 갖기 때문에 자율주행연구, 개발에 직접적인 측위장비로 사용되기에 큰 어려움이 있다.

현대사회의 차량기술의 화두는 운전자의 주행안전과

* 서울대학교 기계항공공학부, 학생

** 서울대학교 기계항공공학부, 교수

†교신저자 : kyi@snu.ac.kr

E-mail : kimjhmj@snu.ac.kr

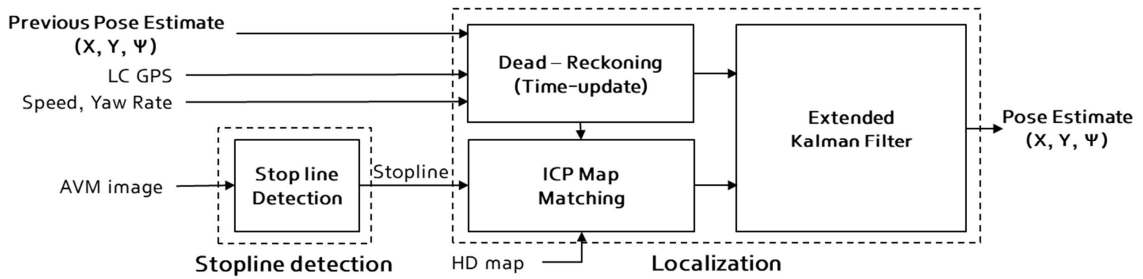


Fig. 1 Architecture of Longitudinal position correction algorithm

편리를 돕기 위한 ADAS(첨단 운전자 보조 시스템)이다. 이미 많은 차량에 ADAS를 구성하는 Vision 센서들이 장착 되어있다. 따라서 이러한 Vision센서와 Map을 이용한 많은 관련연구⁽¹⁻⁴⁾가 진행중이다. 이 중 Around View Monitoring(AVM) Camera는 Vision 센서의 일종으로 4 개 카메라를 사용한 영상합성으로 차량 주변 시야를 제공 하여 운전자의 주차편의를 돕는다.

본 연구에서는 Fig. 1과 같이 AVM카메라를 사용하여 전방 정지선을 인지하고, 이를 고정밀 지도(HD map)의 정지선 정보와 비교, 매칭하여 보정위치를 계산한다. 또한 보정위치와 LC GPS를 칼만필터를 사용하여 결합하여 LC GPS의 종방향 측위성능을 향상시키는 알고리즘을 제시한다. 알고리즘은 Robot Operating System(ROS)환경에서 구현되며 실제 도로주행 데이터를 사용하여 알고리즘에 대한 성능검증을 진행하고 그 결과를 확인한다.

2. 센서 구성 및 데이터 취득 방법

2.1. 차량 및 센서구성

본 연구에서 제시하는 알고리즘 검증을 위한 주행데이터 수집에 사용된 실험차량은 현대 Solati이다. 또한 차량 센서로 Omnivue 사의 AVM Camera와 U-blox사의 Low-cost GPS가 사용되었으며 휠속도 및 각속도는 실험차량에 기본 장착된 센서신호를 CAN 통신으로 수신하여 사용하였다. 실험결과의 성능확인을 위한 Reference GPS로 Inertial Labs사의 RTK GPS가 사용되었다. 실험차량의 제원 및 센서 정보는 Table 1과 같다.

2.2. 데이터 취득 방법

데이터 취득을 위한 주행은 상암 자율주행 테스트베드(Fig. 2)에서 진행하였다. 상암 자율주행 테스트베드는

Table 1 Sensor specifications for a test car

Components/Model	Specification
Test Car / Hyundai Solati	- Type: commercial 4-door van - Length: 6195mm - Width: 2038mm
Low-cost GPS / U-blox	- Update rate: 1Hz - Position accuracy: 2m
AVM Camera / Omnivue 360	- 4 Camera - Horizontal field of view: 185° - Vertical field of view: 142° - Total pixel: 120M pixel
Reference GPS / Inertial Labs INS.P	- Update rate: 200Hz - Position accuracy: 0.4m (DGPS) / 0.01m (RTK) - Velocity accuracy: 0.03m/s



Fig. 2 Sangam automated driving testbed & HD map

국토지리정보원을 통해 HD map이 제공되고, 24시간 민간에 개방되어 자율주행 개발차량과 일반차량이 모두 주행이 가능한 지역이다.

실험차량에 장착된 PC에서는 ROS기반의 센서 처리 패키지 통해 ROS topic이 발행된다. 데이터 로깅은 Rosbag 패키지를 사용하여 bag 파일로 저장하였다.

3. 정지선 인지

3.1. 허프직선검출(Hough line detection)

허프직선검출 방법은 Fig. 3과 같이 Image Space의 직선에 포함되는 점들이 Hough Space에서 한점에서 교차하는 선들로 변환되는 허프변환(Hough transform)의 특징을 이용하여 허프영역의 교차점의 수로 이미지영역의 직선을 검출하는 방법이다.⁽⁵⁾ 허프직선검출은 곡선을 검출할 수 없지만 RANSAC과 같은 모델기반 인지방법에 비해 알고리즘이 간단하다는 장점이 있다.

3.2. AVM 직선검출 프로세스

AVM Camera 이미지 상에서 정지선으로 추정할 수 있는 직선을 검출하는 과정은 다음과 같다.

- 1) AVM Camera image로부터 정지선이 인지되는 전방 관심영역(Regions of Interests, ROI)를 분리한다.
- 2) 흑백 필터로 이미지를 단순화하고, 가우시안 필터를 (Fig. 4) 사용하여 이미지 내 노이즈를 smoothing한다.
- 3) Sobel mask filter(Fig. 4)를 사용하여 이미지 상의 경계점(edge)을 이진화(binary)한다.
- 4) 상기한 Hough line detection method를 사용하여 이미지내에서 직선정보(기울기, y절편)를 검출한다.

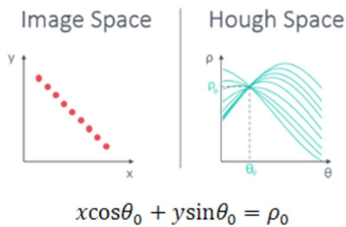


Fig. 3 Hough transform

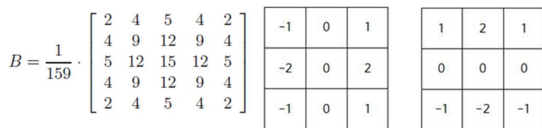


Fig. 4 Gaussian filter & Sobel mask (x filter, y filter)

일정 기준을 넘은 모든 직선들을 포함한다. 이는 도로의 크랙, 정지선 이외의 노면표시 등을 정지선으로 오인할 수 있게 하므로 정지선에 대한 판단과정이 필요하다.

먼저 기울기 파라미터(Slope threshold)를 설정하고, 기울기 파라미터를 기준으로 작은 기울기를 갖는 직선을 Valid line, 큰 기울기를 갖는 직선을 Invalid line으로 분리한다. 이는 1차적으로 정지선과 이외의 노면표시를 분리하는 효과를 가진다. 또한 Valid line과 Invalid line의 수에 따른 정지선판단을 위한 파라미터인 Valid_threshold와 Invalid_threshold 설정한다.

정지선판단 순서도(Fig. 5)를 통해 정지선 미인지 상황을 3가지로 분류하였다. (a)에서 Invalid line의 수가 Invalid_threshold이상일 경우 이미지 잡음이 많은 상황과 (b)에서 Valid line의 수가 Valid_threshold 이하일 경우, (c)에서 직선 간의 폭이 실제 정지선폭 이하일 경우 오인지의 가능성이 높은 상황으로 정지선 인지에서 제외 하도록 하였다.

정지선 인지의 경우 모든 Valid line의 y값의 평균을 이미지상 정지선과의 거리(Stopline_y)로 판단하고, 식 (1)과 같이 이미지의 픽셀 당 실제 거리와 차량 길이를 더해 실제 정지선과의 거리를 계산하였다.

$$\text{Stopline_distance (m)} = 2.15 + (\text{Stopline_y}) * 0.02816 \quad (1)$$

3.3. 정지선 판단(Stopline decision)

3.2의 과정에서 검출된 직선들은 전방 이미지 상에서

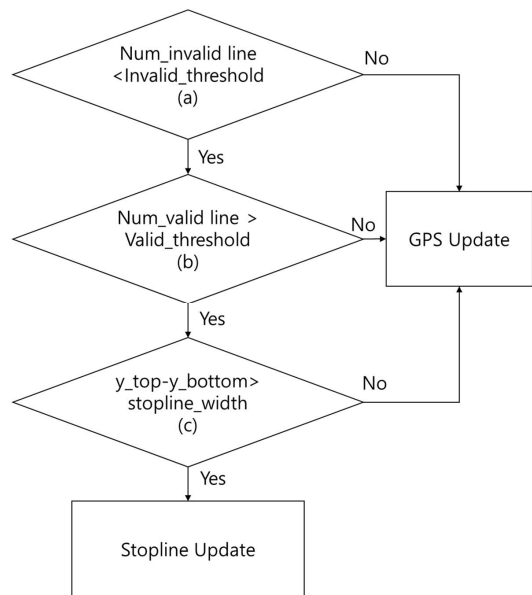


Fig. 5 Stopline decision flowchart

4. HD map 기반 정지선 매칭

4.1. Iterative Closest Point (ICP)

ICP 알고리즘이란 하나의 데이터셋을 다른 데이터셋에 매칭하는 방법으로 두가지 단계로 요약될 수 있다. 첫 번째로 하나의 데이터셋에서 다른 데이터셋과 가장 가까운 거리의 대응점을 찾는다. 두 번째로 대응점과의 거리를 최소화하는 변환 행렬을 계산하고 변환을 수행한다. 이 과정을 반복하게 되면 구하고자 하는 목표 변환행렬을 구할 수 있다.⁽⁶⁾

4.2. HD map 매칭 및 보정위치 계산

HD map은 차선, 정지선, 노면표시와 같은 도로의 정보를 수 cm의 정확도로 갖춘 지도를 말하며, 본 연구에서 제시하는 알고리즘의 실험환경인 상암 테스트베드 지역의 HD map 정보는 shp 파일형식으로 국토지리정보원에서 제공되고 있다.

ICP 알고리즘을 적용하기 위해 Fig. 6과 같이 차량 중심으로부터 정지선과의 거리만큼 떨어진 6m 길이의 pointcloud

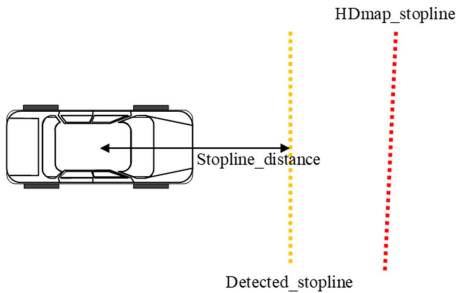


Fig. 6 Stopline pointcloud generation

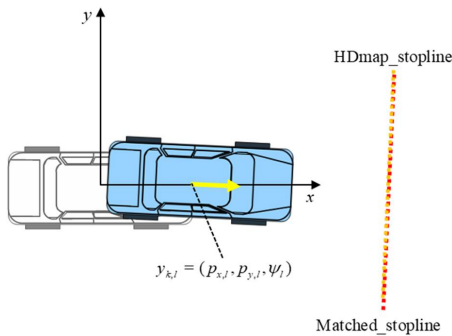


Fig. 7 ICP matching result & Correction pose

데이터(Detected_stopline)를 생성하였다. 또한 HD map 의 shp파일로 구성된 정지선 정보를 pointcloud 데이터(HDmap_stopline)로 변환하였다.

정지선 인지시에 ICP 알고리즘을 사용하여 차량중심 10m 반경의 정지선 HD map과 매칭되고 변환행렬을 얻는다. 그리고 Fig. 7과 같이 변환행렬을 이용하여 차량 좌표계에서의 보정위치를 계산한다.

5. 위치 업데이트

5.1. LC GPS Time & Measurement update

보정위치를 반영하기에 앞서, 위치정보의 연속성을 위해 차량신호(휠속도, 각속도)를 이용한 위치추정이 필요하다.

따라서 위치정보에 확장칼만필터(EKF)를 사용하기 위해 식 (2)와 같이 상태, 입력 벡터를 정의하고, Time update 를 수행하였다.

$$x_k = \begin{pmatrix} p_{x,k} \\ p_{y,k} \\ \psi_k \end{pmatrix}, u_k = \begin{pmatrix} v_k \\ \gamma_k \end{pmatrix}, Q = \begin{pmatrix} 0.03^2 & 0 & 0 \\ 0 & 0.03^2 & 0 \\ 0 & 0 & \left(\frac{5.0}{\pi} * 180\right)^2 \end{pmatrix}$$

$$\hat{x}_k^- = \begin{pmatrix} \hat{x}_{1,k-1} + \Delta t \cdot u_{1,k-1} \cdot \cos(\hat{x}_{3,k-1} + \Delta t \cdot u_{2,k-1}) \\ \hat{x}_{2,k-1} + \Delta t \cdot u_{1,k-1} \cdot \sin(\hat{x}_{3,k-1} + \Delta t \cdot u_{2,k-1}) \\ \hat{x}_{3,k-1} + \Delta t \cdot u_{2,k-1} \end{pmatrix}$$

$$F_{k-1} = \left. \frac{\partial f_{k-1}}{\partial x} \right|_{\hat{x}_{k-1}^-} = \begin{pmatrix} 1 & 0 & -\Delta t \cdot u_{1,k-1} \cdot \sin(\hat{x}_{3,k-1} + \Delta t \cdot u_{2,k-1}) \\ 0 & 1 & \Delta t \cdot u_{1,k-1} \cdot \cos(\hat{x}_{3,k-1} + \Delta t \cdot u_{2,k-1}) \\ 0 & 0 & \Delta t \end{pmatrix}$$

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q \quad (2)$$

또한 LC GPS로부터 위치정보가 전달되었을 때 Measurement update를 통해 식 (3)과 같이 반영하였다.

$$y_k = \begin{pmatrix} p_x \\ p_y \\ \psi_k \end{pmatrix}, v_k \sim \mathcal{N}(0, R), R = \begin{pmatrix} 5.0^2 & 0 & 0 \\ 0 & 5.0^2 & 0 \\ 0 & 0 & \left(\frac{5.0}{\pi} * 180\right)^2 \end{pmatrix}$$

$$y_k = H_k \cdot x_k + v_k, H_k = I_{3 \times 3}$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} = P_k^- (P_k^- + R)^{-1}$$

$$P_k^+ = (I - K_k H_k) P_k^- = (I - K_k) P_k^-$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) = (I - K_k) \hat{x}_k^- + K_k y_k \quad (3)$$

5.2. 정지선 인지 보정위치 반영(Stopline update)

정지선 인지 및 맵 매칭을 통해 획득한 보정위치를 칼만필터의 업데이트 과정에 포함하기 위해 다음의 과정으로 추가적인 measurement update를 수행하였다.

$$\hat{x}_{k,l}^- = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, y_{k,l} = \begin{pmatrix} P_{x,l} \\ P_{y,l} \\ \psi_l \end{pmatrix}$$

$$v_k \sim N(0, R_l), R_l = \begin{pmatrix} 0.1^2 & 0 & 0 \\ 0 & 30.0^2 & 0 \\ 0 & 0 & \left(\frac{10.0}{\pi} * 180\right)^2 \end{pmatrix} \quad (4)$$

$$T = \begin{pmatrix} \cos(\hat{\psi}_{k,g}^-) & -\sin(\hat{\psi}_{k,g}^-) & 0 \\ \sin(\hat{\psi}_{k,g}^-) & \cos(\hat{\psi}_{k,g}^-) & 0 \\ 0 & 0 & 1 \end{pmatrix}, P_{k,l}^- = T^{-1} \cdot P_{k,g}^- \cdot (T^{-1})^T \quad (5)$$

차량 좌표계에서 종방향 위치정보를 보정하기 위하여 식 (4)와 같이 $\hat{x}_{k,l}^-$ 과 R_l 를 설정하였으며, 식 (5)를 통해 좌표계 간의 회전변환 행렬을 구하고, 오차 공분산행렬을 회전시킨다.

$$y_{k,l} = H_k \cdot x_{k,l} + v_k, H_k = I_{3 \times 3}$$

$$K_k = P_{k,l}^- H_k^T (H_k P_{k,l}^- H_k^T + R_l)^{-1} = P_{k,l}^- (P_{k,l}^- + R_l)^{-1}$$

$$P_{k,l}^+ = (I - K_k H_k) P_{k,l}^- = (I - K_k) P_{k,l}^-$$

$$\hat{x}_{k,l}^+ = \hat{x}_{k,l}^- + K_k (y_{k,l} - H_k \hat{x}_{k,l}^-) = K_k y_{k,l} \quad (6)$$

$$\hat{x}_{k,g}^+ = T \cdot \hat{x}_{k,l}^+ + \begin{pmatrix} \hat{p}_{x,k,g}^- \\ \hat{p}_{y,k,g}^- \end{pmatrix}, P_{k,g}^+ = T \cdot P_{k,l}^+ \cdot T^T \quad (7)$$

식 (6)을 통해 차량 좌표계에서 위치와 오차공분산을 추정하며, 식 (7)에서 위치와 오차공분산을 전역 좌표계로 되돌린다.

6. 실차 주행데이터 기반 알고리즘 성능검증

6.1. 알고리즘 구현 환경 및 방법

Robot Operating System(ROS)는 미들웨어 소프트웨어

어로서 하드웨어 추상화, 하위제어, 메시지전달, 패키징을 제공한다. 각 패키지는 Python, C++ 등의 다른 언어로 작성될 수 있으며 병렬동작을 지원하고 센서 인터페이스 방법이 간편하다는 장점이 있다.

본문에서 제시하는 알고리즘은 정지선인지(Stopline detection), 측위(Localization)의 두가지 패키지로 구분되어 구현되었으며, 정지선 인지 패키지는 OpenCV라이브러리를 사용한 Python언어, 측위 패키지는 Eigen 및 PCL라이브러리를 사용한 C++로 작성 및 구현되었다.

6.2. 알고리즘 적용 결과

상암 자율주행 테스트베드에서 14번의 정지선이 존재하는 주행경로를 설정하였으며 정밀도로지도는 모든 정지선정보를 포함하고 있다. 이 구간을 주행하며 LC GPS, RTK GPS, AVM Camera 데이터를 Rosbag형태로 취득하였다.

알고리즘 적용은 취득한 데이터를 기반으로 Offline 상황에서 이루어졌다. 이 때 정지선 인지관련 파라미터(Threshold)를 도로환경에 맞게 조정하였으며, 정지선 오인지로 인한 잘못된 측위 업데이트가 일어나는 상황을 줄이기 위해 검출율(Recall)보다 정확도(Precision)을 중심으로 하는 파라미터 조정이 이루어졌다.

Rosbag 주행데이터에 대한 알고리즘 적용결과 정지선 업데이트가 발생하기 전후시점을 ROS환경에서 제공하는 Rviz 패키지를 이용하여 가시화하였다(Estimation: white car, Reference: red car).

Fig. 8에서 추정 차량 위치는 Reference data와 비교하였을 때 종방향 위치오차가 존재하는 것을 확인할 수 있으며, Fig. 9에서 AVM Camera 이미지에 정지선이 인지되어 측위 업데이트를 수행한 결과 위치오차를 보정되었음을 확인하였다.

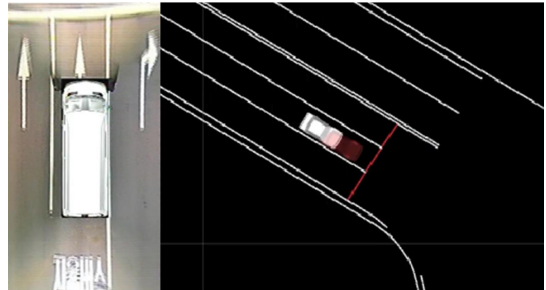


Fig. 8 Before stopline detect & update

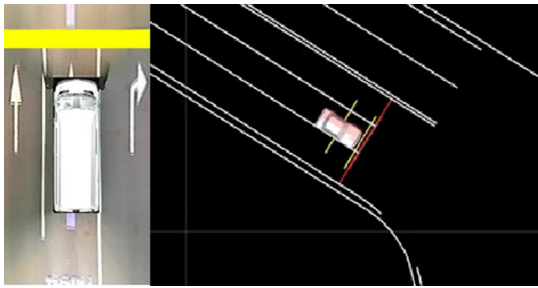


Fig. 9 Stopline detect & update

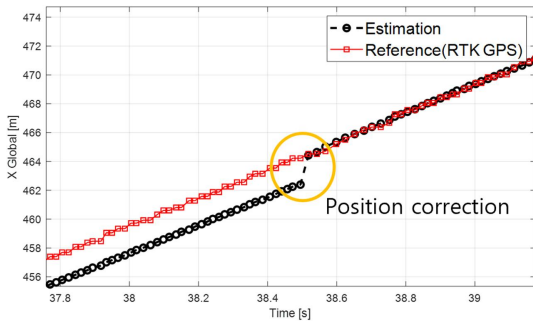


Fig. 10 Longitudinal position correction algorithm test result (Global X direction)

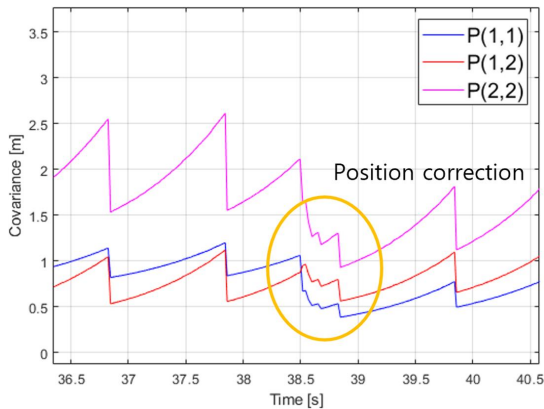


Fig. 11 Covariance matrix P

알고리즘 실행 과정에서 기록한 위치데이터를 분석한 결과는 다음과 같다. Fig. 10은 Reference(RTK GPS), LC GPS, 알고리즘을 적용한 측위 결과를 도시한 그래프이다. 그래프에서 정지선 인지시점(38.5s) 이전까지 1~2m 수준의 종방향 위치오차가 존재하지만 측위 업데이트 이후 0.1~0.2m 수준으로 감소하는 것을 확인하였다. 또한

오차공분산 값을 도시한 Fig. 11에서는 정지선 인지 및 업데이트가 발생하고 p_x, p_y 에 대한 오차공분산이 감소하는 것을 확인할 수 있었다.

7. 결 론

본 연구에서는 도심도로에서AVM 카메라의 전방 이미지로부터 Hough line detection 방법을 사용하여 정지선을 인지하고, HD map의 정지선 정보와 ICP 매칭하여 대응하는 정지선을 찾고 수정 위치를 계산하였다. 그리고 이를 EKF를 사용하여 추정위치를 Measurement update 하여 종방향 측위정확도를 향상시키는 알고리즘을 제시하였다.

제시한 알고리즘의 성능을 검증하기 위해 ROS환경에서 Python 및 C++언어로 구현하였으며, 상암 자율주행 테스트베드 지역을 실차로 주행하고 Rosbag형태로 획득한 AVM 및 LC GPS의 데이터에 적용하여 Reference data(RTK GPS)와 측위 결과를 비교하였다.

검증의 결과 측위 정확도가 연구에서 제시한 알고리즘을 적용하였을 때 향상되는 것을 확인하였으며, 정지선인지 성능과 알고리즘 성능이 직결되는 것을 확인하였다.

후 기

본 연구는 국토교통부 및 국토교통과학기술진흥원의 2018년 교통물류연구사업(18TLRP-B146733-01, 자율주행기반 대중교통시스템 실증 연구)의 지원을 받아 연구되었음을 밝히며, 이에 감사드립니다.

참고문헌

- (1) Markus, S., Carsten, K., Uwe, F., 2013, "LaneLoc: Lane Marking based Localization using Highly Accurate Maps", IEEE Intell. Veh. Symp(IV), pp. 449~454.
- (2) Dominique, G., Rachid, B., Marc, R., 2014, "Map-Aided Localization with Lateral Perception", IEEE Intell. Veh. Symp(IV), pp. 674~680.
- (3) Z. Tao, Ph. Bonnifait, V. Fremont, J. Ibanez-Guzman, 2013, "Lane marking aided vehicle localization", IEEE 16th Int. Conf. Intell. Transp. Syst., pp. 1509~1515.

- (4) Dongwok, K., Beomjun, K., Taeyoung, C., Kyongsu, Yi., 2017, "Lane-Level Localization Using an AVM Camera for an Automated Driving Vehicle in Urban Environments", IEEE/ASME Transactions on Mechatronics, Vol. 22, No. 1, pp. 280~290.
- (5) N. Kiryati, Y. Eldar, A. M. Bruckstein, 1991, "A probabilistic Hough transform", Patter Recognition, Vol. 24, No. 4, pp. 303~316.
- (6) P.J. Besl, N. D. McKay, 1992, "Method for registration of 3-D shapes", Robot.-DL Tentative. Int. Soc. Opt. Photon., pp. 586~606.