

도심 자율주행을 위한 라이다 정지 장애물 지도 기반 위치 보정 알고리즘

노한석* · 이현성** · 이경수***,†

LiDAR Static Obstacle Map based Position Correction Algorithm for Urban Autonomous Driving

Hanseok Noh*, Hyunsung Lee**, Kyongsu Yi***,†

Key Words: *Urban Autonomous Driving*(도심 자율 주행), *Position Correction*(위치 보정), *ROS*(로봇 운영체제), *LiDAR point cloud*(라이다 포인트 클라우드), *Normal Distribution Transformation*(정규 분포 변환), *Static Obstacle*(정지 장애물), *Occupancy Grid Map*(점유 격자 지도)

ABSTRACT

This paper presents LiDAR static obstacle map based vehicle position correction algorithm for urban autonomous driving. Real Time Kinematic (RTK) GPS is commonly used in highway automated vehicle systems. For urban automated vehicle systems, RTK GPS have some trouble in shaded area. Therefore, this paper represents a method to estimate the position of the host vehicle using AVM camera, front camera, LiDAR and low-cost GPS based on Extended Kalman Filter (EKF). Static obstacle map (STOM) is constructed only with static object based on Bayesian rule. To run the algorithm, HD map and Static obstacle reference map (STORM) must be prepared in advance. STORM is constructed by accumulating and voxelizing the static obstacle map (STOM). The algorithm consists of three main process. The first process is to acquire sensor data from low-cost GPS, AVM camera, front camera, and LiDAR. Second, low-cost GPS data is used to define initial point. Third, AVM camera, front camera, LiDAR point cloud matching to HD map and STORM is conducted using Normal Distribution Transformation (NDT) method. Third, position of the host vehicle position is corrected based on the Extended Kalman Filter (EKF). The proposed algorithm is implemented in the Linux Robot Operating System (ROS) environment and showed better performance than only lane-detection algorithm. It is expected to be more robust and accurate than raw lidar point cloud matching algorithm in autonomous driving.

1. 서론

자율주행 자동차 연구에 있어서, 차량의 위치를 추정하

는 측위 기술의 정확도와 정밀도는 매우 중요하다. 차량의 추정위치를 기반으로 목표 주행경로가 생성되고 이러한 주행 경로를 추종하는 시나리오는 도심도로 자율주행에 서 대부분이다.

Real Time Kinematic(RTK) GPS는 GPS, BeiDou, GLONASS, Galileo, NavIC와 같은 Global Navigation Satellite System(GNSS)에서 얻은 위치 데이터에 이동국(rover)이 기준국(base)에서 전송된 반송파 위상 보정

* 서울대학교 기계공학부, 학생
 ** 서울대학교 기계공학부, 학생
 *** 서울대학교 기계공학부, 교수
 †교신저자: kyi@snu.ac.kr
 E-mail: nhs0601@snu.ac.kr

치를 수신하고, 이를 이동국에서 관측된 값으로 보정하여 실시간으로 1~3cm정도의 정밀도를 향상시킨 측량 방식이다.⁽¹⁾ 고정밀 RTK GPS는 높은 정확도를 가지고 있기 때문에 level 3 이하의 자율주행 기술 연구에 활용되어왔다. 하지만 매우 높은 가격에 판매되고 있어 양산을 위한 자율주행 자동차에 장착하기에는 적합하지 않다. 또한 차량과 건물들이 밀집한 도심지역에서 수행되는 level4 이상의 자율주행에서 계속 사용하기에는 위성신호 방해의 우려가 있다. 반면 저가형 GPS(Low-cost GPS, LC GPS)는 현재 양산되고 있는 많은 승용차량에 장착 되어 있으며 RTK GPS에 비해 훨씬 싸다는 가격적인 이점을 가지고 있다. 하지만 보통 1~3m 수준의 오차범위를 갖기 때문에 LC GPS의 데이터를 바로 측위 장비로 사용하기에는 개선되어야 할 부분이 많다.

국토지리정보원에서 제공하는 High Definition Map (HD map)는 자율주행차량에 도로 형상과 도로 노면에 있는 차선 및 마킹에 대한 위치 정보를 오차범위 10cm이내의 정확도로 전달해준다. 본 실험에서는 RTK GPS와 차선 인지 알고리즘 및 위성 사진 등을 활용하여 자체 제작한 map을 HD map 대신 사용하였다. 또한 정지 장애물들을 이용하여 생성한 정지 장애물 기준 지도(STORM)를 사용하여 차선 정보 외에도 정지 장애물에 대한 위치를 나타내는 map을 추가로 사용하였다.

AVM 및 전방카메라를 이용하여 차선인지를 하여 HD map과 NDT(Normal Distribution Transformation) 알고리즘을 이용하여 매칭하면 차량의 현재 위치 및 자세에 대한 정보를 얻을 수 있다. 하지만 교차로에는 차선에 대한 정보를 얻을 수 없기 때문에 자율주행차량이 교차로를 통과할 시에 현재 위치 및 자세에 대한 추정 시 오차가 발생할 수 있다.

Light Detection And Ranging(LiDAR)은 레이저 펄스를 발사하고, 그 빛이 주위의 물체에서 반사되어 돌아오는 것을 받아 물체까지의 거리에 대한 높은 정확도를 가진 데이터를 준다. LiDAR sensor는 Vision sensor과 더불어 차량의 인지 정보형성을 위해 가장 많이 연구되고 있다.⁽²⁾ 앞서 언급한 AVM 및 전방카메라와 더불어 현재 차량에서 인지되는 정지 장애물들을 사전에 생성한 LiDAR map을 NDT알고리즘을 이용하여 매칭하여 차선이 없는 교차로 구역에 대하여 차량의 위치를 추정할 수 있게 하였다.

Odometry란 출발 지점으로부터 얼마나 떨어져 있는지를 파악하는 상대적인 위치를 추정하는 개념이다. 본 연구에서는 자율 주행 차량의 LiDAR sensor를 통한 SLAM (Simultaneous Localization And Mapping)와 Chassis CAN 정보를 이용하여 차량의 동적상태(속도 및 각속도)

에 대해 확장칼만필터(EKF)에 기반하여 추정한다.

추정한 동적상태를 바탕으로 확장칼만필터(EKF)에 기반해 process update를 하고 맵매칭결과를 measurement update에 활용하여 차량의 추정위치를 보정하는 알고리즘을 작성하였다. 위치 보정 알고리즘은 Robot Operating Systems(ROS)에서 구현되어 LiDAR 및 Vision 센서, Low cost GPS가 장착된 자율주행 차량으로 서울대학교 시흥캠퍼스 미래모빌리티센터의 테스트베드에서 실험한다. 기존의 Vision 기반 차선 인지 알고리즘의 취약한 부분인 교차로 대응 성능 비교를 위해 교차로가 있는 코스를 test용 코스로 선정하였다. Vision 기반 차선 인지 알고리즘을 기본 알고리즘으로 하여 이 논문에서 제시한 알고리즘을 추가적으로 적용한 알고리즘과 비교하였다.

2. 실험 차량 센서 구성 및 데이터 취득 방법

2.1. 차량 및 센서구성

본 연구에서는 알고리즘의 올바른 성능검증을 위해 map을 제작하기 위한 차량과 알고리즘 검증용 차량을 각각 다른 차량으로 수행하였다. Map을 제작하기 위한 차량은 Hyundai Sonata DN8을 이용하였고 OxTs사의 RTK GPS RT3000를 장착, AVM및 Logitech 사의 streamcam을 이용해 차선인지에 사용하였으며 LiDAR sensor로는 Velodyne사의 VLP-16 4개를 장착하여 사방을 모두 인지 할 수 있게 하였다.

알고리즘 test용 차량으로는 Hyundai IONIQ을 이용하였고 초기위치 초기화를 위해 사용한 Low cost GPS는 u-blox사의 C100-F9K 제품을 이용하였고 오차분석을 위한 reference RTK GPS로는 Inertial labs 사의 INS-P를 사용, AVM 및 logitech 사의 streamcam을 이용해 차선인지를 하였으며 LiDAR sensor로는 Velodyne사의 VLP-16 4개를 장착하였다. 알고리즘 검증용 차량에 들어간 센서 및 장비들의 정보는 Table 1과 같다.

2.2. 데이터 취득 방법 및 HD map 제작

알고리즘 검증을 위한 자율주행 테스트는 서울대학교 시흥캠퍼스 미래모빌리티기술센터(FMTC)의 테스트베드에서 진행하였다. 알고리즘은 ROS환경에서 구현하였고 주행코스는 차선이 있는 곳과 차선이 없는 곳을 비교하기 위하여 교차로가 있는 코스로 선정하여 실험을 진행했다. Fig. 1에서 파란색선으로 표시한 코스에 대하여 알고리즘

Table 1 Sensor specification for a test car

Components/Model	Specification
Test Car / Hyundai IONIQ	- Type: 4-door sedan - Length: 4470mm - Width: 1820mm
3D-LiDAR / Velodyne VLP-16	- Channels: 16 - Update rate: 5-20Hz - Position accuracy: 3cm - Angular resolution (horizontal): 0.2 degree
Reference GPS/ Inertial Labs INS-P	- Update rate: 200Hz - Position accuracy: 0.03m - Velocity accuracy: 0.03m/s
AVM Camera/Omnivue 360	- 4 Camera - Horizontal field of view: 185° - Vertical field of view: 142° - Total pixel: 120M pixel
Front camera/logitech stream cam	- FOV:78 - 60fps (MJPEG)
Low cost GPS/ U-blox EVK-M8L	- Update rate: 1Hz - Position accuracy: 2m



Fig. 1 FMTC automated driving testbed

검증을 수행하였다. 알고리즘 성능 검증을 위하여 Reference GPS로써 RTK GPS를 사용하였으며 수동으로 주행하며 알고리즘 적용 결과를 ROS topic 형태로 추출하였다. 추출한 ROS topic은 Rosbag패키지를 사용하여 bag 파일로 저장하였다.

FMTC 테스트베드의 HD map은 국토지리정보원에서 제공하지 않으므로 직접 RTK GPS와 차선 인지 알고리즘, 드론으로 촬영한 사진을 사용하여 차선정보가 기록된 HD map을 제작하였다. RTK GPS와 front camera, LiDAR,

AVM 등을 장착한 차량으로 차선의 증상을 따라 수동 주행하였다. 차량 좌표계로 인지되는 차선의 위치를 RTK GPS에서 나오는 차량의 위치를 이용하여 UTM 좌표계로 기록하였다. Smoothing 과정과 드론으로 촬영한 사진과의 비교를 통해 자체제작한 HD map을 완성하였다.

3. 정지 장애물 지도 생성

3.1. 정지 장애물 지도 생성 및 차량 오도메트리 추정

정지 장애물 지도(STOM)는 차량 좌표계를 기준으로 하고 사전에 지정된 높이와 너비, 해상도를 갖는 2D 형태의 점유격자지도 형태로 나타내어진다. 차량 센서로부터 얻어진 LiDAR point cloud들을 점유격자지도의 해상도에 따라 격자로 분리한 뒤 각 격자안의 point cloud에 대하여 정지된 point cloud 가 존재할 확률을 계산해 0과 1사이의 값을 가진 점유격자 지도 형태로 가공된다. 가공된 점유격자지도에서 특정 임계점(Threshold)값 이상의 값을 갖는 격자로 재구성하여 정지 장애물 지도를 재정의하였다. 본 알고리즘에서는 threshold를 0.85로 두었으며 생성된 정지 장애물 지도를 Static Obstacle Map(STOM)으로 명명하였다. 차량의 Chassis IMU 센서로부터 얻은 정보를 이용해 process update를 하고 LiDAR 센서로부터 들어오는 point cloud의 입력으로 베이지안(Bayesian) 룰에 기반하여 사후 확률(Posteriori)로 STOM을 업데이트 한다.⁽³⁾

3.2. 차량 오도메트리 추정

Normal Distribution Transformation(NDT) 알고리즘은 Point cloud 간의 매칭을 통해 변환행렬을 계산하는 알고리즘 중 하나이다. NDT는 두 개의 point cloud가 있을 경우, 두 점군이 겹칠 때 각 점들의 오차를 정규분포로 계산해 최소화하는 알고리즘으로, ICP(Iterative Closest Points)와 같이 모든 점들에 대하여 오차를 계산하지 않고 분포양상만을 이용하여 더욱 빠르게 계산을 수행한다. k-1번째 시점의 STOM과 k 번째 시점의 정지장애물 지도의 차이를 비교하면 차량의 이동에 따른 odometry를 반대로 추정할 수 있다.

3.3. 정지 장애물 기준 지도 작성

생성된 STOM을 점유격자지도의 형태에서 point cloud 형태로 변환한 후 테스트베드 내의 코스를 수동으로 주행

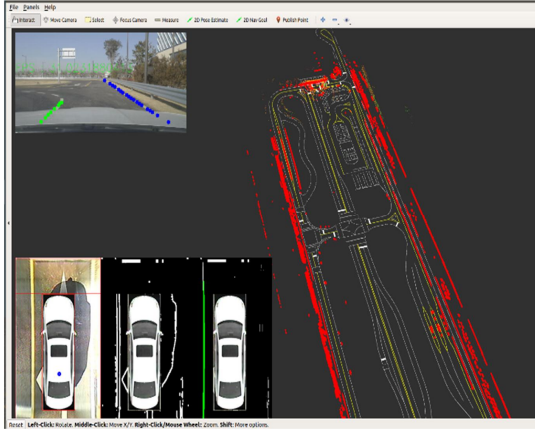


Fig. 2 STORM of the FMTC automated driving testbed

하며 RTK GPS에 나오는 좌표를 기준으로 point cloud를 누적하여 STORM을 작성 하였다. Point cloud를 누적한 후 voxelization을 통하여 point cloud의 개수를 줄여 계산 부하를 감소시켰다. Fig. 2는 테스트베드를 주행하며 얻어진 정지 장애물 기준 지도이다.

4. Vision 기반 차선 인지

Vision 기반 차선 인지 알고리즘은 크게 2가지를 사용 하였다. 첫번째는 전방카메라에 적용하였으며 Point Instance Network(PINet)으로 불리는 network를 사용하였다.⁽⁴⁾ 두 번째는 AVIM에 적용하였으며 AVIM으로 정지선과 옆차선을 인지한 뒤 HD map과 매칭하는 알고리즘을 사용하였다.⁽⁵⁾

5. 맵 매칭 및 위치 보정

5.1. 맵 매칭

맵의 종류는 HD map과 STOM을 이용하여 제작한 STORM가 있다. Reference map 전체를 불러오게 되면 용량이 커져 계산량이 커지기 때문에 계산 부하를 줄이기 위하여 추정 위치를 기준으로 일정 크기 만큼 떼어내어 불러오게 하였다. HD map의 경우 추정 위치 기준 전방 20m, 후방 8m, 좌우 방향으로 각각 5m만큼의 맵을 불러오게 하였다. STORM의 경우 더 넓은 범위의 주변의 정지 장애물과의 매칭을 위하여 전방 50m, 후방 20m, 좌우 방향으로 각각 20m 만큼을 불러오게 하였다. 맵 매칭 알고리즘은 두 개의 point cloud를 비교하는 NDT 알고리즘을 사용하였다.

5.2. 확장칼만필터 기반 위치 보정

맵 매칭 과정 및 odometry 추정의 과정에서 noise가 존재할 가능성이 존재한다. 따라서 차량의 위치 및 자세 정보에 대해 확장칼만필터(EKF)를 사용하여 Process update 및 measurement update를 수행하였다. 먼저 process update는 식 (1)과 같다.

$$x_k = \begin{pmatrix} p_{x,k} \\ p_{y,k} \\ \psi_k \end{pmatrix}, u_k = \begin{pmatrix} v_k \\ \gamma_k \end{pmatrix}, Q = \begin{pmatrix} 0.2^2 & 0 & 0 \\ 0 & 0.1^2 & 0 \\ 0 & 0 & \left(\frac{0.8}{\pi} * 180\right)^2 \end{pmatrix}$$

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, 0) = \begin{pmatrix} \hat{x}_{1,k-1} + \Delta t \cdot u_{1,k-1} \cdot \cos(\hat{x}_{3,k-1}) \\ \hat{x}_{2,k-1} + \Delta t \cdot u_{1,k-1} \cdot \sin(\hat{x}_{3,k-1}) \\ \hat{x}_{3,k-1} + \Delta t \cdot u_{2,k-1} \end{pmatrix}$$

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

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q$$

(1)

다음으로 measurement update는 식 (2)와 같다. 차선, 정지선, STORM에 대한 noise를 각각 다르게 설정하였다. Low cost gps 값을 HD map 상의 경로에 매칭하여 초기 위치를 정하고 LiDAR 및 Vision 센서 데이터를 measurement로 하여 확장칼만필터를 통해 추정 위치 및 자세, covariance matrix를 얻을 수 있었다.

$$z_{ndt,k} = \begin{pmatrix} p_{x,k} \\ p_{y,k} \\ \psi_k \end{pmatrix}, v_k \sim N(0, R), R_{lane} = \begin{pmatrix} 10^2 & 0 & 0 \\ 0 & 0.3^2 & 0 \\ 0 & 0 & \left(\frac{2.0}{\pi} * 180\right)^2 \end{pmatrix}$$

$$z_{ndt,k} = H_k \cdot x_k + v_k, H_k = I_{3 \times 3}$$

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

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

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (z_{ndt,k} - H_k \hat{x}_k^-) = (I - K_k) \hat{x}_k^- + K_k z_{ndt,k}$$

$$R_{stop} = \begin{pmatrix} 0.5^2 & 0 & 0 \\ 0 & 10^2 & 0 \\ 0 & 0 & \left(\frac{5.0}{\pi} * 180\right)^2 \end{pmatrix}, R_{lane} = \begin{pmatrix} 0.5^2 & 0 & 0 \\ 0 & 0.5^2 & 0 \\ 0 & 0 & \left(\frac{5.0}{\pi} * 180\right)^2 \end{pmatrix}$$

(2)

6. 실험 결과

같은 코스에 대하여 Vision기반 알고리즘만 사용한 결과와 LiDAR을 추가적으로 활용한 알고리즘의 결과를 비교하였다. Fig. 3은 종방향 위치 추정 오차를 나타내며 정지 장애물을 추가적으로 이용한 알고리즘이 더 좋은 위치 추정 성능을 보임을 확인할 수 있다. Fig. 4의 횡방향 위치 보다는 추정 오차가 크지만 주행에는 문제가 없음을 확인하였다. 횡방향의 경우, 차선 인지 알고리즘만 사용한 경우, 통과하지 못한 교차로가 있었으며 이는 Fig. 7과 Fig. 8에 나타내었다. 흰색 차량은 추정 위치를 나타내며 빨간

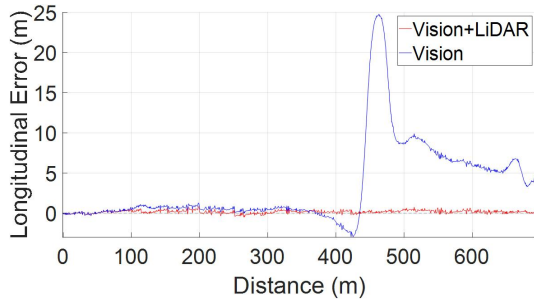


Fig. 3 Longitudinal estimation error

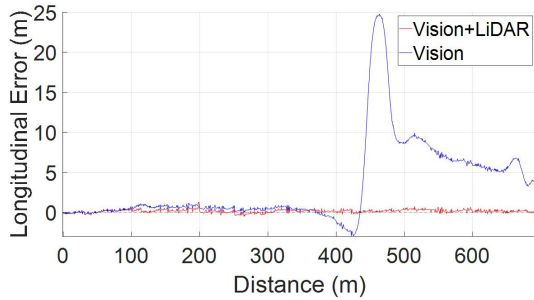


Fig. 4 Lateral estimation error

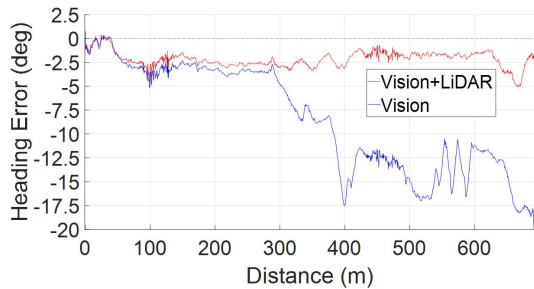


Fig. 5 Heading estimation error

색 차량은 RTK GPS에서 나오는 값을 의미한다. Fig. 6에 맵 매칭이 이루어졌는지의 여부를 도시하였는데 정지 장애물을 활용한 알고리즘의 경우는 매 순간 맵 매칭이 이루어졌지만 차선 인지 기반 알고리즘의 경우 교차로에서 차선인지가 되지 않은 것을 볼 수 있다. Fig. 5의 Heading 추정 오차를 보면 교차로 진입 전 약 2.5° 정도의 오차가 발생했는데 교차로 진입 후 measurement가 없어 오차를

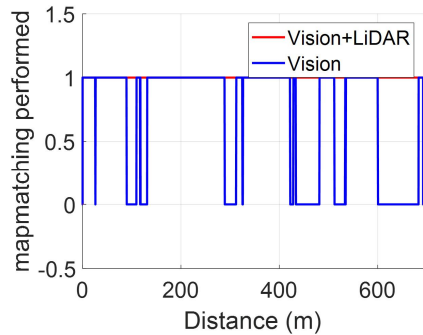


Fig. 6 Whether map matching performed

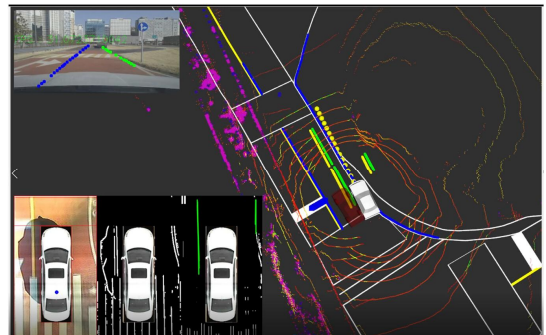


Fig. 7 Estimation failure case of Vision algorithm

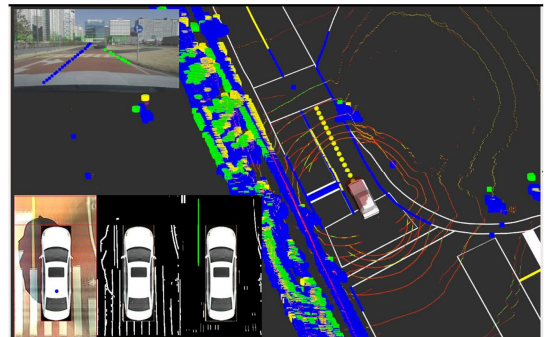


Fig. 8 Estimation success case of Vision + LiDAR algorithm

Table 2 Result comparison of the Algorithms

	Longitudinal MSE/Var (m,m ²)	Lateral MSE/Var (m,m ²)	Heading MSE/Var (deg,deg ²)
Vision	3.629/ 31.909	-3.294/ 100.487	-8.4782/ 36.051
Vision+LiDAR	0.149/ 0.037	-0.038/ 0.024	-2.077/ 0.799

보정하지 못하고 오차가 누적됨을 볼 수 있다. 반면 정지 장애물 알고리즘을 추가한 경우 맵 매칭이 교차로에서도 꾸준히 일어났으며 정상적으로 코스를 주행하였으며 오차에 대한 결과를 Table 2에 나타내었다.

7. 결 론

Vision기반 차선인지 알고리즘의 경우 차선이 인지되는 상황에서는 위치 추정 성능이 우수하지만 교차로에서 차선인지가 불가능한 경우 위치 추정 오차가 누적되며 다음 교차로에 도착했을 때 잘못된 차선으로 진입하는 현상이 발생하였다. 반면, 제안된 알고리즘을 추가한 경우 교차로에서도 지속적으로 맵 매칭이 일어났고 그 결과 코스 내의 모든 교차로를 차선 오인 없이 통과할 수 있었다. 하지만 제안된 베이지안 룰 기반의 STORM의 경우 정지해 있는 모든 물체를 STORM으로 작성하기 때문에 실제 주행 시 STORM을 작성할 때와 비교하여 주변환경이 달라질 수 있다. 따라서 복잡한 도심환경에서 측위를 할 경우, 모든 정지 장애물에 대한 지도가 아닌 불변의 정지 장애물만을 이용하여 STORM을 작성하는 알고리즘이 추가로 필요하다.

후 기

본 연구는 국토교통부 도심도로 자율협력주행 안전·인프라 연구 사업의 연구비지원(과제번호 19PQOW-B2473-01)에 의해 수행되었습니다.

참고문헌

- (1) 김민규, 박준규, 2013, “위성 수신환경 변화에 따른 인터넷 RTK GPS 측량의 정확도 평가”, 한국측량학회지, 31(4), pp. 277~283.
- (2) Dominguez, Raúl, et al., 2011, LIDAR based perception solution for autonomous vehicles., 2011 11th International Conference on Intelligent Systems Design and Applications, IEEE, pp. 790~795.
- (3) 김종호, 이호준, 이경수, 2020, “도심자율주행을 위한 라이다 정지 장애물 기반 차량 동적 상태 추정 알고리즘”, 한국자동차안전학회 추계 학술대회, Vol. 9, No.3, pp.24-30.
- (4) Ko, Y., Jun, J., Ko, D., Jeon, M., 2021, “Key points estimation and point instance segmentation approach for lane detection”, IEEE Transactions on Image Processing, Vol. 30, pp. 2977~2988.
- (5) 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.