◎ 논 문

http://dx.doi.org/10.22680/kasa2021.13.4.014

도심자율주행을 위한 라이다 정지 장애물 지도 기반 차량 동적 상태 추정 알고리즘

김종호* · 이호준* · 이경수*,†

LiDAR Static Obstacle Map based Vehicle Dynamic State Estimation Algorithm for Urban Autonomous Driving

Jongho Kim*, Hojoon Lee*, Kyongsu Yi*,†

Key Words: Urban Autonomous Driving(도심 자율 주행), Vehicle Dynamic State(차량 동적 상태), LiDAR point cloud (라이다 포인트 클라우드), Normal Distribution Transformation(정규 분포 변환), Occupancy Grid Map (점유 격자 지도), Static Obstacle(정지 장애물), Robot Operating System(로봇 운영 체제)

ABSTRACT

This paper presents LiDAR static obstacle map based vehicle dynamic state estimation algorithm for urban autonomous driving. In an autonomous driving, state estimation of host vehicle is important for accurate prediction of ego motion and perceived object. Therefore, in a situation in which noise exists in the control input of the vehicle, state estimation using sensor such as LiDAR and vision is required. However, it is difficult to obtain a measurement for the vehicle state because the recognition sensor of autonomous vehicle perceives including a dynamic object. The proposed algorithm consists of two parts. First, a Bayesian rule-based static obstacle map is constructed using continuous LiDAR point cloud input. Second, vehicle odometry during the time interval is calculated by matching the static obstacle map using Normal Distribution Transformation (NDT) method. And the velocity and yaw rate of vehicle are estimated based on the Extended Kalman Filter (EKF) using vehicle odometry as measurement. The proposed algorithm is implemented in the Linux Robot Operating System (ROS) environment, and is verified with data obtained from actual driving on urban roads. The test results show a more robust and accurate dynamic state estimation result when there is a bias in the chassis IMU sensor.

1. 서 론

다양한 자율주행 연구가 진행되고 있는 현재, 인지와 판단 및 제어를 비롯한 자율주행의 모든 분야가 고정밀의 차량위치 정보를 필요로 하고 있다. 그러므로 지금까지 진행되어온 Level3이하의 자율주행 기술 연구에서는 이를 고정밀 RTK GPS의 도움을 통해 수행해왔다.(1) 하지만 주행 지역이 차량과 건물이 밀집한 도심지역으로 확대되는 Level4 이상의 자율주행에서 위성 신호에 기반하는 GPS에 의존하기에 많은 위험이 따른다는 우려가 있다.

LiDAR(Light Detection And Ranging)은 빛을 이용한 거리측정 센서로 거리정보에대한 높은 정확도를 갖는다. 이러한 LiDAR센서의 특징은 Vision sensor와 더불어서

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

자동차안전학회지: 제13권, 제4호, pp. 14~19, 2021
논문접수일: 2020.12.15, 논문수정일: 2021.9.9, 게재확정일: 2021.10.4

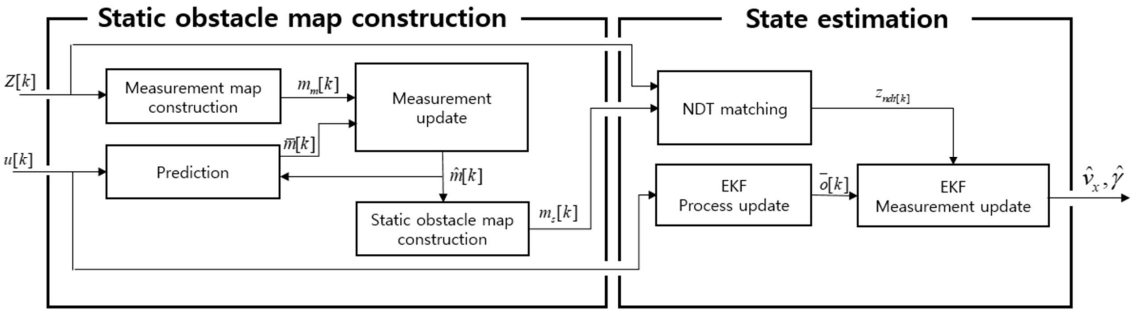

Fig. 1 Overall architecture of proposed system

차량의 인지정보형성을 위해 가장 많이 연구되고 있다.[2]

Odometry란 로봇공학에서 위치 속성에 관한 변화 값을 의미한다. 따라서 2차원 평면상에서 거동하는 자율주행차량의 odometry는 두 시점 사이의 위치와 방향의 차이로 나타내어진다. 자율주행 차량의 odmetry정보는 GPS를 이용한 측정값의 변화 또는 LiDAR sensor를 통한 SLAM (Simultaneous Localization And Mapping)의 방법으로 획득할 수 있으며, 주로 자차량을 원점으로 하는 자율주행 차량의 좌표계에서는 GPS의 미 수신상태에서 odometry로부터 획득한 차량의 동적상태(속도 및 각속도)에 기반한 Dead reckoning을 수행한다. 또한 odometry정보를 이용하여 인지정보의 예측에 이용되기도 한다.[3]

따라서 이 논문에서는 LiDAR의 측정결과로부터 차량 주변의 정지장애물에 대한 정보를 획득하여 점유격자지도 형태의 정지장애물지도를 작성한다. 그리고 연속된 정지장애물지도의 형상이동을 NDT(Normal Distribution Transformation) 알고리즘을 이용하여 매칭하여 차량의 odometry를 계산한다. 마지막으로 차량 odometry를 측정치로 하는 확장칼만필터(Extended Kalman Filter)에 기반하여 차량의 속도 및 각속도를 추정한다.

상기한 통합 자율주행차량 동적상태추정 알고리즘은 ROS에서 구현되어, LiDAR 센서가 장착된 자율주행 차량을 통해 도심도로에서 실험한다. 실험 결과로 5.1에서는 odometry 계산결과를 다른 방법들과 비교한다. 5.2에서는 추정된 차량의 속도 및 각속도의 정확도에 대해서 검증한다.

2. 실험 차량 센서 구성 및 데이터 획득

2.1. 실험 차량 센서 구성

본 연구에서 제시하는 차량 동적상태 추정 알고리즘에 대한 성능검증을 위한 실험차량의 센서구성은 Table 1에 나타내었다. 실험 차량으로 전기차 Hyundai IONIQ을 이용하였으며, LiDAR 센서로 4개의 Velodyne 사의 VLP-16을 음영지역이 없도록 장착하였다. 또한 실험결과의 검증을 위한 참조데이터로 이용하기 위해 Inertial Labs사의 INS-D RTK GPS를 사용하였다.

Table 1 Sensor specifications for a test car

Components/Model	Specification
Test Car / Hyundai IONIQ	- Type: Electric 4-door sedan - Length: 4470 mm - Width: 1820 mm
3D-LiDAR / Velodyne VLP-16	- Channels: 16 - Update rate: 5-20 Hz - Position accuracy: 3 cm - Angular resolution (horizontal): 0.2 degree
Reference GPS/ Inertial Labs INS.P	- Update rate: 200Hz - Position accuracy: 0.4 m (DGPS)/ 0.01 m (RTK) - Velocity accuracy: 0.03 m/s

2.2. 실험 데이터 획득

알고리즘에 검증에 대한 데이터 획득을 위한 주행은 상암 자율주행 테스트베드(Fig. 2)에서 수행하였다. 상암자율주행 테스트베드는 대한민국 국토지리정보원에서 고정밀 지도(HD map)이 작성 및 제공되어 자율주행차량의 실험 및 결과 검증이 용이하다. 또한 본문에서 제안하는 동적상태 추정은 건물과 차량이 밀집한 도심도로에서의 성능검증이 목표이기 때문에, 서울 도심지역에 존재하고 일반차량의 통행이 많은 상암 자율주행 테스트베드는 적합한 실험지역으로써 선정되었다.

Fig. 2 Sangam automated driving testbed & HD map

3. 정지장애물 지도 작성

3.1. 정지장애물 지도

정지장애물 지도란 차량 좌표계를 원점으로 하는 사전에 지정된 높이와 너비, 해상도를 갖는 점유격자지도 형태로 나타내어진다. 이 점유격자지도의 각 cell이 갖는 값은 해당 cell의 위치에 정지물체가 존재할 확률이며, 따라서 0과 1사이 값을 갖는다. 정지장애물지도는 특정 경계(Threshold) 값 이상을 갖는 cell에 대하여 재정의되어 자율주행차량의 판단 및 거동 계획에 이용 될 수 있다.[4]

3.2. Process update

정지장애물지도는 두가지 단계에 따라 업데이트된다. 첫번째는 Process update로 정지장애물지도의 원점인 차량이 이동함에 따라 발생하는 좌표계의 이동을 반영한다. 따라서 Fig. 3과 같이 [k−1] 시점의 정지장애물지도의 각 cell에 대하여 위치를 구하고, 차량이동에 의해 발생하는 위치변화를 반영하여 새로운 cell의 값으로 대입한다.

Algorithm Map_PU($m[k-1]$, $\hat{o}$)

$$\mathrm{T}=\begin{bmatrix}\cos(\hat{o}_\theta) & -\sin(\hat{o}_\theta) & \hat{o}_x\\ \sin(\hat{o}_\theta) & \cos(\hat{o}_\theta) & \hat{o}_y\\ 0 & 0 & 1\end{bmatrix}$$

for all cells m_i of $m[k-1]$ do

$p^i = index2position(i)$

$i_{pu} = position2index(T^{-1}p^i)$

$\bar{m}_{i_{pu}}[k] = m_i[k-1]$

endfor

return $\bar{m}[k]$

Fig. 3 Static obstacle map process update algorithm

3.3. Measurement update

정지장애물지도의 두번째 단계는 Measurement update이다. Measurement update는 LiDAR센서세 의해 인지되는 Point cloud 입력으로 각 cell의 측정치(Measurement)를 정의하고, 베이지안(Bayesian) 룰에 기반하여 사후 확률(Posteriori)로 정지장애물지도를 업데이트하는 과정이다.[5]

Measurement update를 진행하기 위한 첫번째 과정으로 Measurement map을 도입한다. Measurement map이란 정지장애물지도와 같은 크기를 갖는 점유격자지도 형태로 나타내어지며 LiDAR point cloud입력과 이동물체정보에 따라 각 cell의 상태를 Occluded(1), Free(2), Moving(3)의 세가지 값으로 표현한다. 이때 이동물체정보는 LiDAR 센서를 이용하는 별도의 알고리즘을 포함하여 차량좌표계에서 정의되는 모든 알고리즘에 대해 동일하게 이용될 수 있다.

k시점에 스캔 된LiDAR point cloud는 Moving track과 함께 Measurement map의 각 cell의 값을 Occluded(Cell 내부 LiDAR point 검출), Free(Cell 내부 LiDAR point 미검출), Moving(셀 내부 이동물체검출) 으로 정의하도록 한다(Fig. 4).

이렇게 만들어진 Measurement map을 이용하여, 정지장애물지도는 재귀 베이지안(Bayesian) 룰에 따라 식 (1)에 의해 measurement update된다. 이 때 베이지안룰을 적용하는 것에 필요한 사전확률(Priori)는 3.2에서 서술한 process update된 정지장애물 지도를 이용하고, 유사도(Likelihood)은 경험에 따라 Fig. 5와 같이 정의하였다.

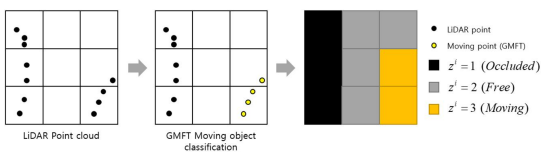

Fig. 4 Measurement classification using LiDAR point cloud

x^i_{static} \ z^i	1(Occluded)	2(Free)	3(Moving)
0(Unknown)	0.3	0.4	0.3
1(Static)	0.6	0.3	0.1

Fig. 5 Conditional Static probability of cell

$$P(x^i_{static}=1 \mid z^i[k],\cdots,z^i[0])$$
$$\overset{\text{Bayes rule}}{=} \frac{P(z^i[k] \mid x^i_{static}=1, z^i[k-1],\cdots,z^i[0]) \times P(x^i_{static}=1 \mid z^i[k-1],\cdots,z^i[0])}{P(z^i[k] \mid z^i[k-1],\cdots,z^i[0])}$$
$$\overset{\text{Markov}}{=} \frac{P(z^i[k] \mid x^i_{static}=1) \times P(x^i_{static}=1)}{P(z^i[k])}$$
$$\overset{\text{Bayes rule}}{=} \frac{\overbrace{P(z^i[k] \mid x^i_{static}=1)}^{\text{Likelihood}} \times \overbrace{P(x^i_{static}=1)}^{\text{Priori}}}{\sum_j P(z^i[k] \mid x^i_{static}=j) \times P(x^i_{static}=j)} \quad (1)$$

4. 차량 오도메트리 및 동적상태 추정

4.1. NDT 알고리즘

Normal Distribution Transformation(NDT)[6] 알고리즘은 Point cloud 간의 매칭을 통해 변환행렬을 계산하는 Scan matching 알고리즘 중의 하나이다. NDT는 Target point cloud를 Grid마다 분포에 따른 정규분포 함수로 표현하고 수치해석적 방법을 통해 Input point cloud를 수렴시키는 것이 특징이며, 그러므로 Point cloud의 크기에 상관없이 분포양상만을 이용하여, ICP(Iterative closest point)와 같은 Scan matching 알고리즘에 비해 수행속도가 빠르다.

정지장애물 지도를 NDT를 이용하여 matching하기 위해, 경계치(Threshold)확률 이상을 갖는 cell을 위치에 따라 point로 반환하여, Point cloud형태를 만든다. 그리고 NDT를 수행하여 [k−1]과 [k]시점 간의 정지장애물 지도의 이동을 변환행렬로 획득한다. 정지장애물은 차량의 이동에 의해 발생하는 Odometry에 반대의 거동을 보이므로, 이에따라 식과 같이 변환 행렬의 값을 음수화하여 차량의 odometry를 계산할 수 있다(Fig. 6).

4.2. 확장칼만필터 기반 추정

4.1의 결과로 획득하게되는 odometry는 정지장애물지도의 변화가 크지 않은 저속상황에서 noise가 존재할 수 있다. 그러므로 차량샤시의 imu센서로부터 측정되는 차량속도, 각속도를 이용하여 확장칼만필터(EKF)에 기반하여 odometry를 필터링하는 과정이 필요하다.

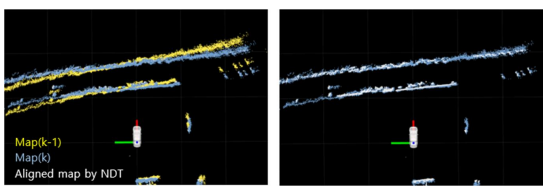

Fig. 6 Aligned map by NDT matching algorithm

확장칼만필터를 이용하기 위해, 추정하는 상태를 차량 odometry선정하고, 샤시의 imu센서로부터 측정되는 차량속도와 각속도를 제어입력으로 가정하여 식 (2)와 같이 시스템을 구성하였다.

$$x_k=\begin{pmatrix} o_x \\ o_y \\ o_\theta \end{pmatrix},\ u_k=\begin{pmatrix} v_x \\ \gamma \end{pmatrix},\ w_k \sim N(0,Q), Q=\begin{pmatrix} 0.3^2 & 0 & 0 \\ 0 & 0.1^2 & 0 \\ 0 & 0 & (\frac{5}{\pi}*180)^2 \end{pmatrix} \quad (2)$$

위치와 방향을 갖는 강체로 가정한 차량모델은 비선형 시스템이므로 이산화(Discretization)및 선형 근사(1st order linear approximation)를 수행한 시스템 행렬은 아래와 같이 표현된다.[7]

따라서 시스템에 따른 차량 odometry의 process update는 식 (3)과 같다.

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

다음으로 NDT matching을 통해 획득한 odometry를 measurement로 확장칼만필터의 measurement update를 수행한다. 4.1에서 서술한 것과 같이, NDT의 결과로 획득하는 변환행렬은 차량 odometry발생에 의한 좌표계 변환을 나타내므로 식과 같이 각 성분을 음수화하여 odometry 측정치로 사용되었다. 또한 측정치와 추정하고자하는 상태가 동일한 정보이므로 측정행렬 H를 단위행렬로 하여 Measurement update를 수행하였다(식 (4)).

$$z_{ndt,k}=\begin{pmatrix} o_x \\ o_y \\ o_\theta \end{pmatrix},\ v_k\sim N(0,R),\ R=\begin{pmatrix} 3.0^2 & 0 & 0 \\ 0 & 1.0^2 & 0 \\ 0 & 0 & \left(\frac{1.0}{\pi}*180\right)^2 \end{pmatrix} \quad (4)$$
$$z_{ndt,k}=H_k\cdot x_k+v_k,\ H_k=I_{3X3}$$
$$K_k=P^-_kH^T_k(H_kP^-_kH^T_k+R)^{-1}=P^-_k(P^-_k+R)^{-1}$$
$$P^+_k=(I-K_kH_k)P^-_k=(I-K_k)P^-_k$$
$$\hat{x}^+_k=\hat{x}^-_k+K_k(z_{ndt,k}-H_k\hat{x}^-_k)=(I-K_k)\hat{x}^-_k+K_kz_{ndt,k}$$

앞서 상기한 확장칼만필터를 통해 noise가 감소된 odometry를 획득할 수 있고, 이를 알고리즘 시간 간격에 따라 차량의 동적 상태로 계산하였다.

5. 알고리즘 구현 및 실험결과

상기한 정지장애물지도 기반 차량 동적상태 추정알고리즘은 Linux ROS환경에서 C++ 언어로 구현되었다.

5.1. EKF based odometry estimation result

첫번째로 제안한 알고리즘의 차량 odometry 추정에 대한 성능을 확인하기 위해 상암 자율주행 테스트베드 지역의 500m 코스를 U-turn 주행하여 획득한 데이터를 이용하였다. U-turn주행은 차량의 각속도가 크게 발생하게 되어 차량 IMU센서의 측정오차를 확인할 수 있다. 또한 결과데이터의 참값과의 비교를 위해 고정밀 RTK GPS의 IMU 측정치를 함께 획득하였다.

Fig. 7은 U-turn 시나리오 주행 중의 제안한 알고리즘

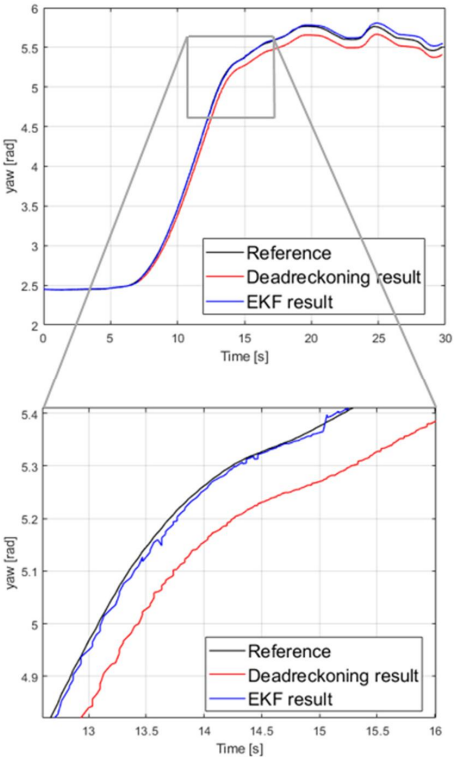

Fig. 7 Yaw angle estimation accumulation (U-turn)

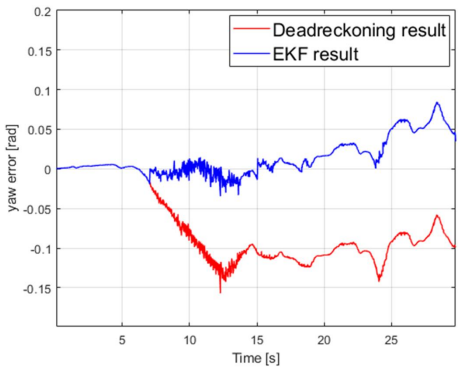

Fig. 8 Yaw angle estimation error (U-turn)

을 이용하여 추정된 Odometry의 yaw를 누적한 결과이다. Imu 센서로부터 측정된 yaw rate을 시간에 따라 누적한 결과와 비교하였을 때, LiDAR 정지장애물지도를 이용하여 획득한 측정치가 EKF를 통해 odometry를 추정하는 것에 반영되었음을 확인할 수 있었다. 또한 Fig. 8에서 U-turn 종료 시점에서 차량의 yaw 오차가 0.15라디안에서 0.05라디안으로 약 60% 감소하는 효과를 보였다.

5.2. Dynamic state estimation result

두번째로 제안한 알고리즘의 차량의 종방향속도에 대한 추정성능을 확인하기 위해 상암 자율주행 테스트베드 지역에서 약 2km의 복합주행 시나리오 데이터를 이용하였다. 실험에서 차량의 종방향 속도는 제안한 알고리즘의 확장칼만필터가 추정한 odometry로부터 이동량을 계산하고, 알고리즘의 시간간격으로 나누어 도출하였다.

Fig. 9는 복합주행 시나리오의 주행결과 제안한 알고리

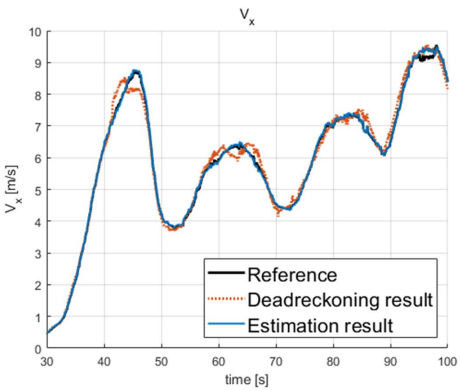

Fig. 9 Velocity estimation (Combined driving)

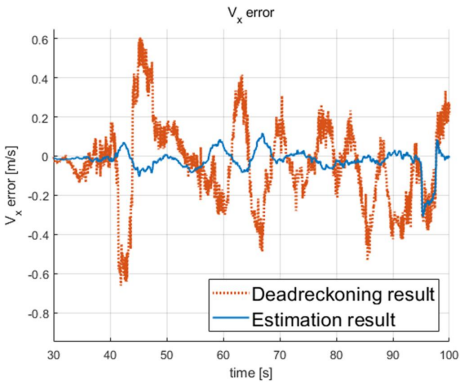

Fig. 10 Velocity estimation error (Combined driving)

즘이 추정한 종방향 속력과 샤시의 IMU센서의 측정결과를 도시하였으며, 비교를 위해 고정밀 RTK GPS의 측정결과를 도시하였다. IMU의 측정결과만으로 계산한 Dead reckoning 방법은 가속과 감속상황에서 제안한 알고리즘의 추정결과에 비해 노이즈가 발생하는 것을 확인하였다. 반면 제안한 알고리즘은 IMU센서의 측정결과에 정지장애물지도가 추가로 반영되어 노이즈가 감소하고 Reference data인 RTK GPS의 측정치에 근사한 것을 확인하였다. Fig. 10은 제안한 알고리즘의 추정결과와 Dead reckoning 결과를 Reference data와의 오차로 나타내었으며, 전체 주행 구간의 RMSE(Root Mean Square Error)가 0.32m/s에서 0.15m/s로 약 46% 감소함을 확인하였다.

6. 결 론

본 연구에서는 도심도로에서 자율주행차의 동적상태추정을 위한 알고리즘을 제안하였다. LiDAR Point cloud로부터 이동물체정보를 제거한 정지장애물지도를 작성하고, NDT를 이용한 정지장애물지도의 위치변화로 차량의 odometry를 계산하였다. 그리고 확장칼만필터에 기반하여 차량 제어 입력을 포함하여 odometry를 추정하여 그 결과로 차량의 동적 상태를 계산하였다.

제시된 알고리즘은 ROS환경에서 구현되었으며, LiDAR 및 검증을 위한 고정밀 GPS센서를 장착한 자율주행차량을 구성하여 실제 도심도로에서 획득한 데이터를 이용하여 실험하였다. 그 결과로 차량 IMU 센서 단독으로 측정한 결과에 비해 차량 속도 및 각속도의 추정 성능이 향상됨을 확인하였으며, 확장칼만필터를 통해 추정된 차량 odometry또한 IMU 센서 만을 이용하여 Dead reckoning을 수행한 결과보다 정확도가 증가했음을 확인하였다.

후 기

본 논문은 산업통상자원부 산업기술혁신사업(10079730, 자동차전용도로/도심로 자율주행 시스템 개발 및 성능평가)의 지원을 받아 수행하였습니다.

참고문헌

(1) Trimble, Tammy E., et al., 2014, Human factors evaluation of level 2 and level 3 automated driving concepts: Past research, state of automation technology, and emerging system concepts.

(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) Wen, W., Hsu, L. T. and Zhang, G., 2018, Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong, Sensors, 18(11), 3928.

(4) Ferri, G., Jakuba, M. V., Mondini, A., Mattoli, V., Mazzolai, B., Yoerger, D. R. and Dario, P., 2011, Mapping multiple gas/odor sources in an uncontrolled indoor environment using a Bayesian occupancy grid mapping based method, Robotics and Autonomous Systems, 59(11), 988~1000.

(5) 윤정식, 이호준, 이경수, 2019, "자율주행 인지성능 개선을 위한 점유 그리드 맵 응용", 한국자동차안전학회 춘계 학술대회 논문집.

(6) Ripperda, Nora and Claus Brenner, 2005, Marker-free registration of terrestrial laser scans using the normal distribution transform. Proceedings of the ISPRS working group 4.

(7) Dongwok, K., Beomjun, K., Taeyoung, C. and 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.