

Point Cloud 정보를 이용한 도로 상황 검출 방법

안희준, 김민석, 김영준, 나세현, 홍인기

경희대학교, 전자공학과

hmk6160, gogosing611, donjomyo, 2015104016, ekhong@khu.ac.kr

Road Conditions Detection Method
using Point Cloud Information

Hee-Jun Ahn, Min-Seok Kim, Young-Jun Kim, Se-Hyeon Na, Een-Kee Hong

Dept. of Electronics Engineering, Kyung Hee Univ.

요약

본 논문에서는 자율주행차용 도로 환경의 상태 검출 방법을 제안한다. LiDAR 센서를 통해 수집된 3D point cloud에서 좌표 데이터를 활용하여 물체 및 도로의 각 point 별 3차원 공간좌표를 얻을 수 있다. 이 데이터를 활용해 도로의 균열 혹은 포트홀과 같은 결함을 검출하는 방법을 구현한다. 센서 기준 point로부터 각각의 point 지점까지의 거리를 구하고, 이를 활용해 현재 도로의 상태를 파악하는 방법을 제시한다. 실제 데이터가 아닌 이상적인 환경의 시뮬레이션 환경을 구현하여 시뮬레이션 상의 균열을 나타낸 데이터와 알고리즘을 통해 찾은 데이터를 비교하여 두 데이터 집합의 error rate 및 정확도 성능을 검출하여 hole의 위치를 파악한다.

1. 서론

최근 자율주행 자동차의 안전한 주행을 위해 여러 가지 센서가 사용되고 있고, 그중 LiDAR 센서는 물체의 거리와 형태를 3D point cloud로 확인할 수 있는 센서다. 3D point cloud는 Laser가 물체에 부딪힌 뒤 반사되어 돌아오는 물리적 속성을 이용하여 물체의 3차원 좌표 데이터를 수집할 수 있다. 또한, 돌아오는 광선의 세기를 통해 intensity 정보를 기록한다. 좌표와 intensity 정보를 갖는 point들을 물체로 인식하기 위해 segmentation 작업을 거치고, 이를 통해 자율주행자동차 주행에 필요한 정보를 획득하게 된다.[1] 본 논문은 이 중에서 3차원 좌표 데이터만을 활용한다. 센서를 기준 point로 하여 물체 및 도로에 대해 수집된 point들과의 거리를 구하고 point별 거리 차이를 통해 도로의 균열을 찾아낼 수 있는 방식을 제안한다. 데이터는 실제 데이터를 사용하지 않고 임의로 시뮬레이션 환경을 조성하여 알고리즘을 제안했다. 제안된 알고리즘을 통해 시뮬레이션 상에서 설정한 hole의 위치를 파악한다.

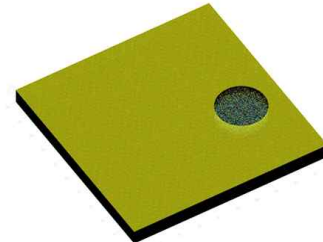


그림 1. 시뮬레이션 환경 (x, y = 500 x 500)

그림 1 시뮬레이션 환경을 확대한 모습을 그림 2에 나타내었다. 평면에 해당하는 point는 검은색, hole에 해당하는 부분은 하늘색으로 표현했다. 각 point 별 인덱스를 p_n 로 지정한다.

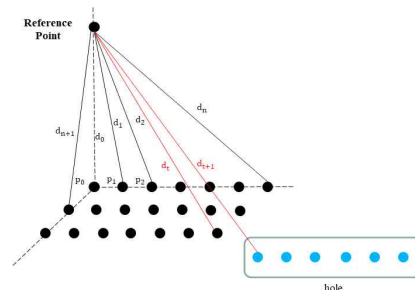


그림 2. 평면과 hole 부분 확대

II. 알고리즘 설명

제안하는 도로 상태 검출 기법은 다음과 같다. 임의의 시뮬레이션 환경을 조성하여 그림 1에 나타내었다. 3차원 (x, y, z) 좌표를 설정한다. x, y 좌표는 500 x 500의 point 크기를 가지고 z = 0인 부분은 평면, z ≠ 0인 부분은 균열 or 포트홀이라고 가정하였다. (x, y, z) 좌표는 (0, 0, 0)부터 시작하고 다음 point와의 간격은 1로 지정하였다. LiDAR 센서 기준 point의 좌표는 (0, 0, 500)으로 정했다.

식 (1)에 알고리즘에 사용될 파라미터를 나타낸다. $p_0(0, 0, 0)$ 부터 시작하여 LiDAR 기준 point로부터 각 point까지의 거리를 d_n , D_n 은 현재 인덱스와 다음 인덱스와의 d 의 차이를 의미한다. D_n 의 값은 a 로 수렴하게 되고 a 의 값을 기준으로 구멍인지 평면인지를 판단한다. D_i 에서 a 가 갑자기 커지는 부분을 구멍으로 판단할 수 있다. D_n 을 구할 때 예를 들어 (500, 0, 0)의 point에서 다음 point는 (0, 1, 0)이기 때문에 그림 2에서 d_n 과 d_{n+1} 와의 차가 아닌 d_0 과 d_{n+1} 의 차이를 이용해야 임계값 a 를 활용할 수 있다.

$$\begin{aligned}
 D_0 &= |d_0 - d_1| = a \\
 D_1 &= |d_1 - d_2| \simeq a \\
 D_2 &= |d_2 - d_3| \simeq a \\
 &\vdots \\
 D_n &\neq |d_n - d_{n+1}| = |d_0 - d_{n+1}| \simeq a \\
 &\vdots \\
 D_i &\neq |d_i - d_{i+1}| \gg a \\
 &\vdots
 \end{aligned} \quad (1)$$

III. 시뮬레이션 결과

그림 3에서 알고리즘을 통해 찾은 D의 변화 그래프를 나타내었다. D의 값이 크게 차이가 나는 지점이 hole에 해당하는 point를 의미한다. 평면에 해당하는 부분은 그래프에서 확인할 수 있듯이 일정한 임계값 a 를 가지고 있다.

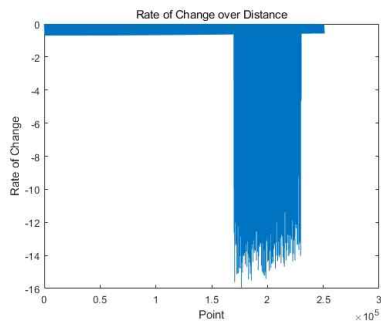


그림 3. 알고리즘을 통해 찾은 D의 변화 그래프

식(1)의 a 의 크기를 0부터 15까지 변화시켜 알고리즘을 통해 얻은 hole에 대한 데이터와 실제 설계 조건의 hole data의 값을 비교하여 얻은 error rate 그래프를 그림 4에 나타내었다. D의 값이 0.725일 때 0.4558%의 error rate로 가장 좋은 성능을 보였다. Error rate가 아주 큰 값이 아니라면 그래프에서 나타난 바와 같이 0.725 이후의 값들도 이전의 값들보다 상대적으로 작은 error이고, 도로 상에서 균열을 검출할 때 error가 어느 정도 있어도 변화율의 크기가 큰 지점들의 집합이 구성되어 있으면 그 지점을 hole이라고 실제 판단이 가능하다. 따라서 hole의 집합만 구성이 된다면 본 논문의 목적인 도로 위의 균열을 판단하는 데 있어 정확도는 큰 고려사항이 되지는 않을 것이다.

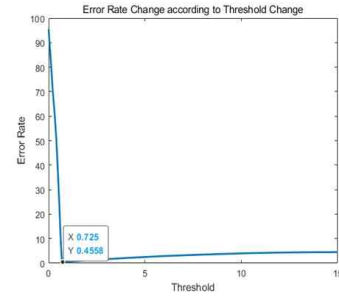


그림 4. 알고리즘과 실제 데이터 간 error rate 그래프

iv. 결론

시뮬레이션 환경이 이상적인 상황이었고 실제 데이터가 아니었기 때문에 좋은 성능이 나와 hole의 위치를 파악할 수 있었지만, 실제 환경에서는 point의 위치도 일정하지 않고 평면에 해당하는 point 또한 높이차로 인해 z의 값이 전부 다르므로 해당 알고리즘을 완전히 적용하기에는 무리가 있다. 해당 알고리즘은 각 point를 인덱스에 할당하여 이전 인덱스와 다음 인덱스 간의 차이만을 통해 알고리즘을 구현했지만, 실제 환경에서는 두 개 이상의 거리 차이를 활용하여 그 평균을 이용하여 정확성을 더 높여 균열을 찾는 확률을 높일 수 있을 것으로 생각된다. 혹은 공간적인 정상성 자료를 이용하여 기계학습을 사용할 수 있다.[2]

후에 5G에서 시스템 간 통합이 가능해지면 자율주행 자동차가 수집한 point cloud 도로 데이터를 실시간으로 전송하여 도로의 균열을 바로 검출하여 조치가 가능해질 것이다. 혹은 고정되어있는 RSU(Road Side Unit)에서 지속적으로 도로를 추적하고 데이터화하여 균열이 생겼을 때 발생하는 데이터의 변화를 통해서도 도로 상태를 검출할 수 있을 것이다.

ACKNOWLEDGMENT

Put sponsor acknowledgments.

참 고 문 헌

- [1] 김정우, 서승우. "자율주행자동차를 위한 도로환경에서의 3D Point Cloud Segmentation Algorithm 구현", 2016년도 대한전자공학회 정기총회 및 추계학술대회, p646-647, 2016.11
- [2] N.Gilardi, Samy Bengio. "Local Machine Learning Models for Spatial Data Analysis", IDIAP-RR 00-34 Reference EPFL-ARTICLE-82651, 2000