

# 인프라 라이다 실시간 배경제거 알고리즘 개발

홍인준·박만복\*

한국교통대학교 전자공학과

## Development of Infrastructure Lidar Real-time Background Removal Algorithm

Injun Hong · Manbok Park\*

Electronic Engineering, Korea National University of Transportation, Chungbuk 27469, Korea

(Received 18 July 2023 / Revised 11 October 2023 / Accepted 14 October 2023)

**Abstract** : To ensure safety in driving autonomous vehicles, the use of information tools, such as installing sensors like Lidar on the side of the roads or infrastructure, is being studied. In the case of Lidar installed in an infrastructure system, it can be calculated based on the characteristic that Lidar always looks in the same direction to separate it from the background. A Lidar sensor uses a large amount of Point Cloud Data(PCD), and it entails a large amount of computation because it must survey a large space. Also, a Lidar sensor is used for a long time, so the background might change. Thus, the algorithm must be updated. In this paper, algorithms that generate data to remove the background are placed in parallel to reduce the amount of computation. It also proposes a method of computing the background with data before a certain frame to note the change in the background. An infrastructure lidar was then installed and verified. It was shown that the amount of noise generated during background removal was reduced to about 35.3%, compared to the previous result.

**Key words** : Lidar(라이다), Point cloud data(포인트 클라우드 데이터), Background remove(배경제거), Support vector machine(서포트 벡터 머신), Real-Time(실시간성)

### Nomenclature

- $\vec{w}$  : normal vector to the hyperplane
- $\vec{u}$  : support vector
- $b$  : position of hyperplane
- $s$  : space of target object, m
- $v$  : velocity of target object, m/s
- $dt$  : time difference, sec

### Subscripts

- $x, y$  : x, y axis
- $k$  :  $k^{th}$  sampling time

### 1. 서론

Lidar는 대상체에 빛을 비추고 그 반사광을 이용하여 물체를 인식하는 센서이다. Lidar 센서는 매우 높은 정밀

도와 Point Cloud Data(PCD)의 밀도가 높으며, 저조도 상황에서 문제없이 객체를 감지할 수 있다. Lidar 센서는 최근 자율주행 차량에 많이 사용되고 있다. 차량에 직접 부착하여 객체를 인식하는 것뿐만이 아니라 자율주행 차량의 안전한 운행을 위해 교차로와 같이 장애물이 많아 인식 범위가 제대로 확보되지 못하는 곳의 인프라 등에 설치하여 차량이 미처 인식하지 못한 객체를 인식하는 데에도 사용된다.

Lidar 센서로 객체를 인식하기 위해서는 객체인 부분과 객체가 아닌 부분을 분리할 수 있어야 한다. 차량에 설치되는 Lidar의 경우 센서가 움직이고 차량이 객체와 충돌하지 않다면 되기 때문에 많은 상황에서 지면만을 제거하는 방법을 사용한다. 하지만 인프라에 설치된 Lidar의 경우 센서가 고정되어 있고, 객체에 대한 정보를 차량에 전송해야 되기 때문에 객체를 제외한 나머지 배경을 전부 제거하여 사용된다.

\*Corresponding author, E-mail: ohnnuri@ut.ac.kr

<sup>†</sup>This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

본 논문에서는 인프라 Lidar를 사용하여 객체를 인식하기 위해 배경제거를 수행한다. 시간의 경과에 따른 배경의 변화에도 문제없이 사용할 수 있으며 실시간으로 동작이 가능하도록 배경 데이터를 업데이트 하는 내용을 다룬다. 2장에서는 인프라 등에 설치된 Lidar의 배경제거에 관한 다른 연구에 대하여 다룬다. 3장에서는 본 논문에서 제안한 알고리즘에 대하여 서술한다. 4장에서는 알고리즘의 성능을 확인하기 위해 진행한 실험 및 그 결과를 설명하고, 5장 결론으로 마무리된다.

## 2. 관련 연구

인프라 등 도로변에서 Lidar를 이용하여 객체를 인식하는 내용에 관해서 많이 연구되고 있다. 대부분의 연구에서 배경을 제거하는 작업을 수행한다. 배경제거의 방법으로는 점들의 밀도를 사용하는 방법이 있다.<sup>1)</sup> 이 방법은 초기에 배경 제거를 위한 Voxel을 만들어 놓은 후 사용한다. Voxel을 만드는 방법은 공간을 동일한 크기의 3D Voxel로 나누고 여러 프레임을 누적시켜 점의 개수를 기준으로 임계값을 설정하여 배경을 제거하는 방법이다. 이 경우 2~3천장을 배경 데이터를 위해 누적시켜 사용하며 Voxel의 크기는 5 cm, 임계값은 거리에 따라 3개로 나누어서 사용한다. 해당 방법을 수정 없이 사용한 연구도 있고,<sup>2,3)</sup> 해당 방법을 사용한 후 객체에 지면과 접지한 부분을 제거하기 위해 인접한 채널 사이의 두 점을 뽑아 거리와 각도를 이용하여 제거한 연구도 있다.<sup>4)</sup> 또한 레이저의 수평 및 수직각을 고려하여 군집화를 통해 배경제거를 수행하였다.<sup>5)</sup> 임계값의 경우 Lidar의 오차를 고려하여 선택하였으며 이 연구에서 사용한 Lidar의 경우 오차 3 cm으로 임계값을 6 cm으로 선정하여 사용하였다. 또 다른 방법으로는 밀도의 변화를 이용하는 방법이 있다.<sup>6)</sup> 공간을 3D Voxel로 나누어 PCD의 개수를 기록하고 이전 프레임과 비교하여 변화를 살펴본다. 배경의 경우 변화가 없고, 객체의 경우 변화를 보이게 된다. 이 연구에서는 물체에 가려진 부분이 밀도의 변화를 줄 것이라고 예상했지만 테스트 데이터에서 그러한 현상은 발생하지 않았다고 한다. 공간을 차량이 들어갈 정도의 크기의 부분공간으로 나누고 혼합 모델을 이용하여 객체를 찾은 후 그 공간을 다시 부분공간으로 나눠서 배경을 찾는 방법도 있다.<sup>7)</sup> 충분히 많은 프레임을 누적시키면 정확하게 구할 수 있다고 한다. 약 1,500개의 프레임을 누적시킨 데이터를 KD Tree를 이용하여 군집화를 수행한 후 임계값 이상의 군집화 된 점을 배경으로 선택하는 방법도 있다.<sup>8)</sup> 또한 객체가 존재한다면 배경이 반사될 때 보다 거리가 짧게 측정된다는 점을 이용하는 방법

도 있다.<sup>9)</sup> 그 방법은 레이저의 수직, 수평각을 기준으로 데이터를 모아서 가장 먼 거리를 배경이라고 판단하는 방법이다. 거리가 0을 반환하는 경우 오류가 발생하는데 최대거리에 적절한 임계값을 미리 설정해 두어 이를 해결함을 보여주었다. 이와 비슷한, 수평 수직 분해능으로 나눈 셀에 거리를 저장하여 배경과 비교하는 방법을 사용하는 연구도 있다.<sup>10)</sup> 레이저의 거리 기반의 방법과<sup>11)</sup> 수평, 수직 분해능을 기준으로 하여 매트릭스를 만들고 셀에 점의 높이를 저장하여 배경일 경우의 높이와 비교하여 배경인지 아닌지를 판단하는 방법도 있다.<sup>12)</sup> 여러 프레임을 누적시켜 배경을 선택한 후 제거 할 프레임의 데이터를 겹쳐서 제거하는 방법을 사용하는 연구도 있었다.<sup>13)</sup> 또한 딥러닝을 사용한 방법도 활발히 연구되고 있다.<sup>14,16)</sup> 학습된 딥러닝 모델과 이의 정확도를 개선하기 위해 정밀지도를 이용하여 관심영역을 줄이는 방법을 사용하였고,<sup>14)</sup> 이미지 투영, 복셀 기반, 포인트 기반 방식 등 여러 가지 방식의 모델을 사용하여 객체 인식을 수행하였다.<sup>15)</sup> 3개의 딥러닝 모델을 학습시켜 선행차량 검출 성능을 포함한 성능을 비교한 연구도 있다.<sup>16)</sup>

본 논문에서는 알고리즘의 처리시간 확보를 위해 배경을 만들어놓고 간단한 비교만 수행한 방법<sup>1)</sup>이 가장 적합하다고 판단하였다. 하지만 초기에 배경 데이터를 생성하고 계속 사용하기 때문에 기상변화 등으로 노면의 높이가 바뀌거나 하는 등의 경우 대응하기 힘들다는 문제가 있다. 따라서 본 논문에서는 일정 주기마다 공간을 3D Voxel로 나누어 PCD를 여러 프레임 동안 누적하여 배경 데이터 만들어 업데이트하여 환경 변화에 강건하게 대처할 수 있도록 하였다. 또한 배경 데이터 생성을 병렬로 배치하여 실시간성을 확보하였다. 배경 제거를 위한 임계값의 경우에는 Support Vector Machine(SVM)을 이용하여 선정하였다.

## 3. 본 론

### 3.1 System Architecture

#### 3.1.1 Hardware Architecture

본 연구에서는 Photo 1과 같이 32ch Lidar와 산업용 PC로 환경을 구성하였으며, 상세 사양은 다음 Table 1과 같다.



Photo 1 Hardware architecture

Table 1 Hardware specification

	Specifications
32ch LiDAR	- Measurement range : 200 m - Range accuracy : Up to ± 3 cm - Angular resolution : Horizontal : up to 0.125 Deg, Vertical : 0.8 Deg - Distance resolution : 4 cm - Scanning frequency : 5/20 Hz - Wavelength : 905 nm - Eye-safety : Class 1 - Scanning layer : 4 vertical 이상 (0.8°) - Vertical field of view : 40° - Horizontal field of view : 360° - Data refresh time : 40/80 ms - Temp. range : -40 ~ +85 °C
Industrial PC	- Intel® 8th-Gen Core™ ihexa-core 35 W/ 65 W LGA1151 CPU - Low-profile chassis with hot-swappable 2.5" HDD/ SSD tray - MezzIO™ interface for easy function expansion - Rugged, -25 °C to 70 °C fanless operation - Up to 6x GigE ports, supporting 9.5 KB jumbo frame - M.2 2280 M key socket (Gen3 x 4) supporting NVMeSSD or Intel® Optane™ memory - 4x USB 3.1 Gen2 ports and 4x USB 3.1 Gen1 ports

3.1.2 Software Architecture

밀도기반 배경제거 알고리즘<sup>1)</sup>은 동작 초기에 배경 데이터 생성 이후 그 데이터를 이용하여 배경제거, 군집화, 트래킹 등을 수행한다. 이 방법은 초기에 입력된 PCD로 배경을 판단하기 때문에 시간의 흐름에 따라 배경이 변하는 경우에 효과적으로 대응하지 못한다. 장시간 알고리즘을 사용해야 하는 상황에서는 조금씩 변화하는 데이터가 누적되어 초기 배경데이터와 이후 데이터가 사

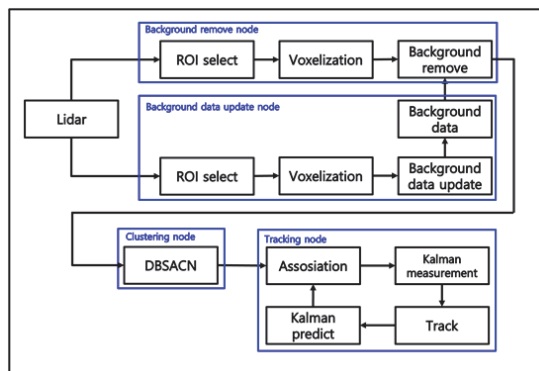


Photo 2 Software architecture

용에 문제가 생길 정도로 다르기 때문에 문제가 발생한다. 이를 해결하기 위해서는 알고리즘이 동작할 때 배경 데이터를 계속 업데이트할 수 있도록 설계를 해야 한다. Lidar의 경우 PCD의 수가 많기 때문에 배경 데이터를 업데이트 이후 다음 알고리즘을 실행하게 되면 많은 비용이 들게 되어 실시간으로 동작하기 힘들어진다. 그렇기에 배경 데이터 생성 부를 병렬로 배치하여 Photo 2와 같이 기존 알고리즘의 처리시간에 큰 영향을 끼치지 않는 방법으로 시스템을 설계하였다.

3.2 배경제거 및 배경 데이터 생성·업데이트

인프라 혹은 도로변에 설치되어 있는 Lidar는 항상 동일한 위치를 바라보고 있기 때문에 배경을 식별하여 제거할 수 있다. 그렇기에 배경제거는 차량 등에 설치되어 있는 Lidar와는 다른 고정된 위치에 설치되어 있는 Lidar의 특성에 의해 처리되는 프로세스이다. 위에서 설명한 것과 같이 설치하는 Lidar가 처리하는 데이터는 보행자, 차량 등과 같은 객체들의 위치 및 속도의 데이터이다. 그렇기 때문에 지면, 초목, 건물 등의 배경은 필요가 없고, 필요없는 부분이 다음 프로세스로 넘어가게 되면 필요적으로 연산량이 증가하게 된다. 따라서 본 논문의 배경제거의 목적은 필요한 PCD를 최대한 유지하면서 필요없는 PCD를 제거하여 연산량을 줄이는 것이다.

3.2.1 배경 데이터 생성·업데이트 방법

밀도 기반의 배경 데이터를 생성하기 위해서 관심영역을 정하고 그 영역을 동일한 크기를 갖는 Voxel로 나눈다. Voxel의 크기가 작으면 프레임이 많이 필요하고 크기가 크면 객체의 PCD가 제거 될 수 있기 때문에 적당한 값을 찾아 맞춰주는 것이 필요하다. 원하는 개수의 프레임 동안 PCD가 존재하는 위치에 있는 Voxel에 값을 더해 주어 데이터를 생성한다. 배경 PCD의 위치에 있는 Voxel은 매 프레임마다 계속 값이 더해질 것이기 때문에 배경이 아니거나 객체가 지나간 Voxel보다 높은 값을 갖게 된다. 임계값을 정해서 임계값 이상인 Voxel을 배경이라고 판단하여 배경제거에 사용될 데이터를 얻는다.

시간이 지남에 따라 배경은 변화할 수 있고 그에 맞게 배경 데이터가 변화해야하기 때문에 배경 데이터 생성을 초기에 한 번만 하는 것이 아니라 계속 업데이트를 해줘야 한다. 프레임을 계속 누적하여 쌓는 등의 방법을 사용하면 변화하는 경우 그에 맞게 데이터가 생성되지 않을 수 있다. 그래서 본 논문에서는 현 시점을 기준으로 이전 프레임을 이용하여 배경 데이터를 생성하기로 결정하였다. 여러 번의 실험 결과 Photo 3과 같이 100프레임씩 분할한 후 현 시점에서 300프레임 이전까지의 값들

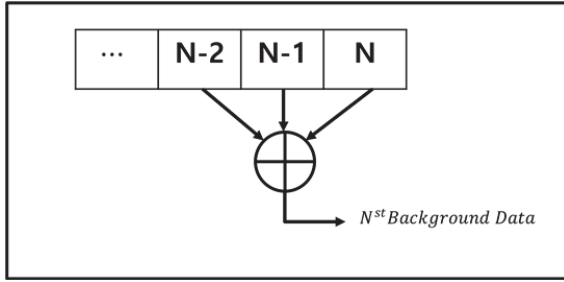


Photo 3 To create background data

을 합하여 배경 데이터를 생성하는 것이 가장 적합하다고 판단하였다.

### 3.2.2 파라미터 선정

배경을 구분하는 임계값을 실험적인 방법으로 구해도 되지만 크게 되면 노이즈가 많아져 후에 동작하는 알고리즘의 비용을 증가시키고 너무 작게 되면 객체의 PCD가 손실될 수 있기 때문에 적절한 값을 선정하는 작업이 중요하다. 본 논문에서는 배경에 대한 임계값을 선정하기 위해 기계학습 방법 중 하나인 Support Vector Machine (SVM)을 사용하였다.

SVM은 학습에 사용된 Sample을 구분하기 위해 Margin이 최대가 되는 경계를 구하는 알고리즘이다.

경계  $d$ 는 다음과 같이 가정할 수 있으며

$$d = \vec{w} \cdot \vec{u} + b \tag{1}$$

Margin은 다음과 같이 나타난다.

$$margin = \frac{2}{\|\vec{w}\|} \tag{2}$$

Margin을 최대로 하는 경계를 찾는 것이므로 식 (2)를 최대로 하는  $\vec{w}$ 를 찾는 것이 SVM의 해가 된다.

본 논문에서는 상단에 기술한 것과 같이 경계를 구하는 방식의 분류 방법 때문에 Voxel의 데이터로 배경을 분류해야 되는 이 문제의 해결에 적합하다고 판단하였다.

SVM의 학습에 사용될 데이터는 고정된 위치에서 항상 동일한 방향을 바라보는 라이더 PCD로 제작된 Voxel이다. 그렇기에 셀의 값, 위치 등을 입력으로 넣을 수 있다. 하지만 위치를 입력으로 하면 긴 시간동안의 동작을 감안하여 설계를 한 본 알고리즘의 경우 기상환경의 변화 등의 이유로 배경의 위치가 틀어질 수 있고, 많은 양의 계산을 처리해야 하는 특성상 매 Voxel마다 SVM을 연산한다고 가정하면 많은 비용이 소모된다.

그렇기에 본 논문에서는 Voxel의 값만을 SVM의 입력으로 사용하고, 학습시켜 임계값을 계산 한 뒤 간부등호 연산을 통해 배경을 분류하는 방법을 사용하였다.

### 3.2.3 배경제거

배경제거 알고리즘의 경우 이후에 군집화 등의 다른 알고리즘이 동작해야하기 때문에 간단한 연산이 필요하다. 생성되었던 배경 데이터를 PCD와 비교하기 위해서 일정량의 연산이 필요한데 이를 배경 데이터 생성부에서 수행하여 배경제거 자체의 연산량을 줄였다.

배경제거에서는 바로 사용이 가능한 배경 데이터를 받아 Lidar에서 입력된 PCD를 배경데이터와 동일한 크기의 Voxel 형태로 변경 후 배경데이터와 비교하여 배경인 부분의 제거한다.

### 3.3 군집화

배경이 제거된 PCD를 입력으로 하여 Density-Based Spatial Clustering of Applications with Noise(DBSCAN)알고리즘을 수행하여 군집화를 진행한다.

DBSCAN은 파라미터로 거리  $eps$ 와 개수  $min\_pnt$ 를 받아서 동작한다. 모든 Point에 대해서 하나의 Point를 선택한 후 나머지 Point와의 거리를 구해서 거리가  $eps$  이내인  $min\_pnt$  이상인 경우 하나의 군집으로 묶는 동작을 반복하며 동작의 결과는 Photo 4와 같다. 이때 군집이 되었을 때 중심이 되는 점과 군집에는 속해 있지만 중심이 되지 않은 점, 그리고 아무 곳에도 속하지 않는 노이즈를 구분하여 출력한다.

DBSCAN의 경우 앞서 설명한 것처럼 점 사이의 거리와 그 안에 포함되는 점의 개수를 파라미터로 갖는데 파라미터 선정은 배경제거 이후 남아있는 노이즈가 제거될 수 있도록 실험적인 방법으로 선정하였다.

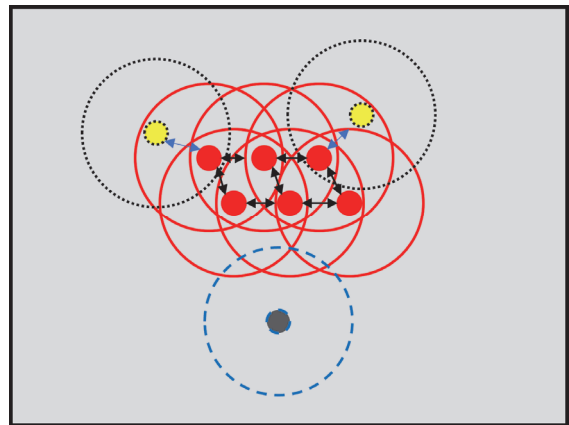


Photo 4 DBSCAN



### 3.4 트래킹

군집화 된 PCD를 객체 단위로 받아 Kalman filter를 사용하여 트래킹을 진행한다.

Kalman filter는 다음 수식과 같은 방법을 사용하여 노이즈가 가우시안 분포를 따른다고 가정하여 다음 상태를 추정하는 필터이다.

$$X_k = F_k X_{k-1} + B_k U_k + W_k \quad (3)$$

$$z_k = H_k X_k + V_k \quad (4)$$

$$W_k = N(0, Q_k) \quad (5)$$

$$V_k = N(0, R_k) \quad (6)$$

여기서 X는 State, F는 Prediction matrix, B는 Control-input model, U는 Control vector, H는 Measurement matrix, z는 Measurement, W는 Process noise이며, V는 Measurement noise이다.

Association의 경우 할당 문제에서 비용이 최소화 되도록 하는 방법으로 수행하였다.

Kalman filter는 다음과 같이 위치 x와 y를 입력으로 받는 등속운동을 가정하여 설계하였다.

$$X = [s_x, v_x, s_y, v_y]^T \quad (7)$$

$$F = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (9)$$

이후 Kalman filter에서 Predict된 위치와 3.3에서 설명한 방법을 사용하여 군집화된 PCD데이터를 비용이 최소화 되도록 할당하여 Association을 수행한다.

## 4. 실험 및 결과

### 4.1 실험 데이터

#### 4.1.1 실험 데이터 구성

실험에 사용된 시스템은 Photo 5와 같이 한국교통대학교 회전교차로에 약 3 m높이에 설치되어있는 32 ch 라이다를 이용하였고 차량, 사람 등의 객체가 지나가는 상황의 데이터를 사용하였다.

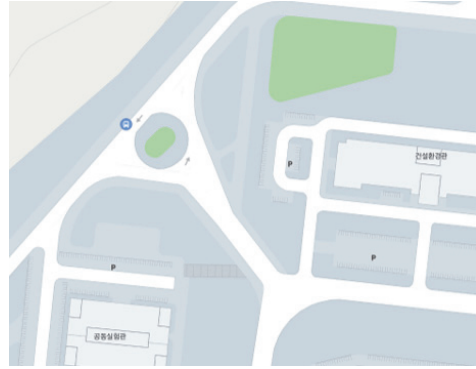


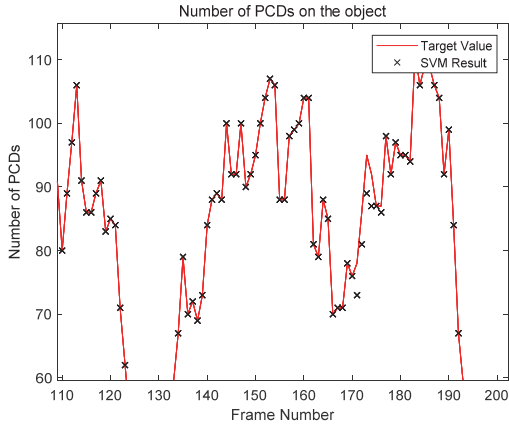
Photo 5 System installation location

#### 4.1.2 SVM 학습 데이터 제작

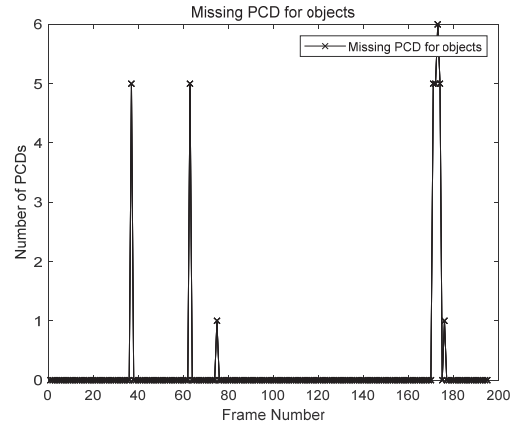
SVM에 학습을 시키기 위해서는 라벨링 된 데이터가 필요하다. 하지만 PCD의 경우 포인트로 이루어져 있고 개수가 많기 때문에 이미지보다 라벨링이 난해하다는 문제가 있다. 그렇기 때문에 배경제거의 임계값을 노이즈가 군집화 단계에서 객체로 인식되지 않는 선까지 최대한 증가시킨 후 노이즈를 제거하여 분류된 객체의 PCD를 객체의 정답값 혹은 목표치로 가정하여 사용하였다.

#### 4.2 SVM 및 배경제거 결과

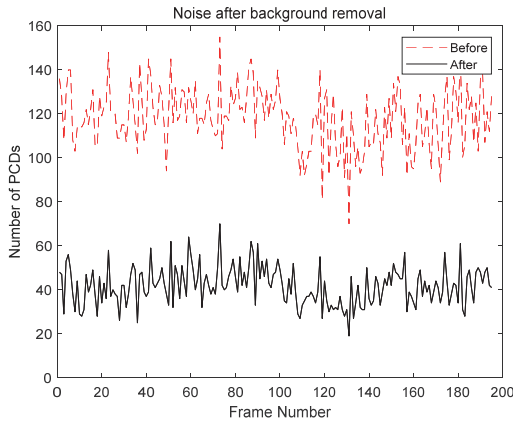
배경제거의 파라미터에 대한 성능으로는 배경이 얼마나 잘 제거되었는지와 객체의 PCD가 어느정도 손실이 생겼는지로 판단할 수 있다. 배경이 얼마나 잘 제거되었는지는 배경제거 이후 군집화 단계에서 DBSCAN으로 필터링 된 노이즈의 개수를 사용해서 성능을 확인하고, 객체의 PCD의 손실은 SVM 학습에서 정답값으로 사용한 데이터를 기준으로 결과를 확인 해 보았다.



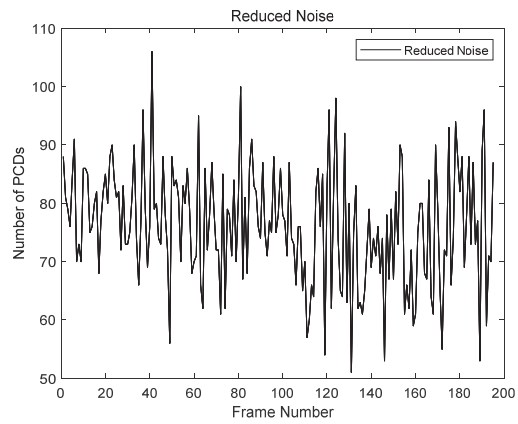
(a)



(b)



(c)



(d)

Photo 6 Result (a) Number of PCDs on the object, (b) Noise after background removal, (c) Noise after background removal, (d) Reduced noise

Photo 6(a)는 실험 데이터에서 추출된 객체의 PCD개수를 일부분 나타낸 그래프이다. 몇몇 프레임에서 객체의 PCD가 목표치(혹은 정답값)보다 누락된 것을 확인할 수 있는데 그 차이가 200프레임동안 측정했을 때 평균 0.16개로 크지 않다. 이는 Photo 6(b)에 나타나 있다.

Photo 6(c)는 객체의 PCD 손실이 최소화되도록 실험적으로 구한 파라미터를 적용 시 발생하는 배경제거 노이즈와 SVM으로 구한 파라미터를 적용시 발생하는 노이즈에 대한 그래프이다. Photo 6(d)에서 노이즈가 200프레임에서 평균 76.2개가 줄어들었음을 확인할 수 있다.

실험적으로 구했을 때와 SVM을 통해 구했을 때 객체의 PCD의 누락은 매우 적으면서 노이즈의 양은 크게 감소한 결과를 볼 수 있다.

### 4.3 알고리즘 처리시간 결과

사용하고 있는 Lidar센서의 경우 10 Hz의 주사율을 갖

기에 실상황에서 원활한 사용을 위해서는 알고리즘의 처리시간이 0.1초 안으로 들어와야 한다.

배경데이터 생성 및 업데이트 부분이 0.5초에서 배경 송신 시 약 0.9초로 나타나고, 배경제거부터 트래킹까지 0.7초로 나타난다.

## 5. 결론

본 논문에서는 인프라에 설치된 Lidar의 실시간성 확보 및 환경 변화에 적응할 수 있는 배경제거 알고리즘을 제안하였다. 일정 주기마다 배경 데이터를 업데이트하여 환경 변화에 강건하게 대처할 수 있도록 하였다. 또한 배경 데이터 생성을 병렬로 배치하여 실시간성을 확보하였다.

이를 검증하기 위해 실제 환경에서 테스트를 하였다. 그 결과 센서의 주기 0.1초에 맞는 처리시간을 보였고, 약 26,000개의 배경 PCD 중 발생한 노이즈의 양이 기존

117.9개에서 41.7개로 개선됨을 보였다. 또한 객체의 PCD 누락이 기존보다 1% 이내로 미미하게 증가한 결과를 보였다.

## 후 기

본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호 RS-2021-KA160501).

## References

- 1) J. Q. Wu, H. Xu, Y. Sun, J. Y. Zheng and R. Yue, "Automatic Background Filtering Method for Roadside LiDAR Data," *Transportation Research Record*, Vol.2672, No.45, pp.106-114, 2018.
- 2) J. X. Zhao, H. Xu, H. C. Liu, J. Q. Wu, Y. C. Zheng and D. Y. Wu, "Detection and Tracking of Pedestrians and Vehicles using Roadside LiDAR Sensors," *Transportation Research Part C*, Vol.100, pp.68-87, 2019.
- 3) Y. P. Cui, Y. Sun, H. Xu, J. X. Zhao and J. Q. Wu, "Automatic Vehicle Tracking with Roadside LiDAR Data for the Connected-Vehicles System," *IEEE Intelligent Systems*, Vol.34, No.3, pp.44-51, 2019.
- 4) J. R. Chen, S. Tian, H. Xu, R. Yue, Y. Sun and Y. P. Cui, "Architecture of Vehicle Trajectories Extraction with Roadside LiDAR Serving Connected Vehicles," *IEEE Access*, Vol.7, pp.100406-100415, 2019.
- 5) Y. J. Song, H. Zhang, Y. Q. Liu, J. Z. Liu, H. B. Zhang and X. G. Song, "Background Filtering and Object Detection with a Stationary LiDAR Using a Layer-Based Method," *IEEE Access*, Vol.8, pp.184426-184436, 2020.
- 6) B. Lv, H. Xu, J. Q. Wu, Y. Tian and C. G. Yuan, "Raster-Based Background Filtering for Roadside LiDAR Data," *IEEE Access*, Vol.7, pp.76779-76788, 2019.
- 7) Y. S. Zhang, H. Xu and J. Q. Wu, "An Automatic Background Filtering Method for Detection of Road Users in Heavy Traffics Using Roadside 3-D LiDAR Sensors with Noises," *IEEE Sensors Journal*, Vol.20, No.12, pp.6596-6604, 2022.
- 8) Z. H. Liu, Q. Z. Li, S. M. Mei and M. H. Huang, "Background Filtering and Object Detection with Roadside LiDAR Data," 2021 4th International Conference on Electron Device and Mechanical Engineering(ICEDME), pp.296-299, 2021.
- 9) H. Liu, C. Y. Lin, B. W. Gong and D. Y. Wu, "Extending the Detection Range for Low-Channel Roadside LiDAR by Static Background Construction," *IEEE Transactions on Geoscience and Remote Sensing*, Vol.60, No.5702412, 2022.
- 10) J. X. Zhang, W. Xiao, B. Coifmanm and J. P. Mills, "Vehicle Tracking and Speed Estimation from Roadside Lidar," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol.13, pp.5597-5608, 2020.
- 11) W. Xiao, B. Vallet, K. Schindler and N. Paparodits, "Simultaneous Detection and Tracking of Pedestrian from Panoramic Laser Scanning Data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol.III-3, pp.295-302, 2016.
- 12) J. X. Zhao, H. Xu, X. T. Xia and H. C. Liu, "Azimuth-Height Background Filtering Method for Roadside LiDAR Data," *IEEE Intelligent Transportation Systems Conference*, pp.2421-2426, 2019.
- 13) Z. Y. Zhang, J. Y. Zheng, X. Wang and X. L. Fan, "Background Filtering and Vehicle Detection with Roadside Lidar Based on Point Association," 37th Chinese Control Conference, pp.7938-7943, 2018.
- 14) J. E. Jeong and S. C. Kee, "Real-Time Object Behavior Prediction Based on LIDAR Suitable for Urban Autonomous Driving," *Transactions of KSAE*, Vol.30, No.7, pp.589-598, 2022.
- 15) M. H. Sun, D. H. Paek and S. H. Kong, "A Study on Deep Learning Based Lidar Detection Neural Networks for Autonomous Driving," *Transactions of KSAE*, Vol.30, No.8, pp.635-647, 2022.
- 16) Y. B. Kim, T. H. Lee and B. S. Song, "Training and Performance Analysis of Vehicle Detection Neural Networks to Field Test and Simulation Datasets of Multi-channel Lidar," *Transactions of KSAE*, Vol.29, No.12, pp.1123-1132, 2021.