

# 라이다 임베디드 프로세서를 위한 동적 객체인식 아키텍처 구현

<sup>1</sup>정민우, <sup>2</sup>이상훈, <sup>3\*</sup>김대영

## Dynamic Object Detection Architecture for LiDAR Embedded Processors

<sup>1</sup>Minwoo Jung, <sup>2</sup>Sanghoon Lee, <sup>3\*</sup>Dae-Young Kim

### 요약

자율주행 환경은 실시간으로 상황이 급변하기 때문에 동적 객체인식 알고리즘이 반드시 필요하다. 또한, 자율주행자동차에 내장된 센서와 제어모듈이 증가하면서 중앙제어장치의 부하가 급격히 증가하고 있다. 중앙제어장치의 부하를 줄이기 위해서 단일 센서에서 출력되는 데이터의 최적화가 필요하다. 본 연구는 라이다에 탑재된 임베디드 프로세서를 기반으로 한 동적 객체인식 알고리즘을 제안한다. 라이다에서 출력되는 포인트클라우드 기반 객체인식을 위한 오픈소스들이 존재하지만, 대부분 고성능 프로세서를 요구한다. 라이다에 탑재된 임베디드 프로세서는 리소스 제약 때문에 기능 구현을 위한 최적화 된 아키텍처가 반드시 필요하다. 본 연구에서는 자율주행자동차를 위한 라이다 임베디드 프로세서 기반 동적 객체인식 아키텍처를 설계하고, 포인트클라우드 크기와 객체인식 처리 지연시간의 상관관계를 분석하였다. 제안하는 객체인식 아키텍처는 포인트클라우드 크기가 증가함에 따라 객체인식 처리 지연시간이 증가하였고, 특정한 지점에서 프로세서의 과부하가 발생하여 포인트를 처리하지 못하는 현상이 발생하였다.

### Abstract

*In an autonomous driving environment, dynamic recognition of objects is essential as the situation changes in real time. In addition, as the number of sensors and control modules built into an autonomous vehicle increases, the amount of data the central control unit has to process also rapidly increases. By minimizing the output data from the sensor, the load on the central control unit can be reduced. This study proposes a dynamic object recognition algorithm solely using the embedded processor on a LiDAR sensor. While there are open source algorithms to process the point cloud output from LiDAR sensors, most require a separate high-performance processor. Since the embedded processors installed in LiDAR sensors often have resource constraints, it is essential to optimize the algorithm for efficiency. In this study, an embedded processor based object recognition algorithm was developed for autonomous vehicles, and the correlation between the size of the point clouds and processing time was analyzed. The proposed object recognition algorithm evaluated that the processing time directly increased with the size of the point cloud, with the processor stalling at a specific point if the point cloud size is beyond the threshold*

**Keywords:** Object Detection, Embedded Processors, LiDAR, Architecture, Autonomous Vehicle

<sup>1</sup> 카네비컴 기술연구소, 책임연구원 ([mwjung@carnavi.com](mailto:mwjung@carnavi.com))

<sup>2</sup> 카네비컴 기술연구소, 선임연구원 ([shlee@carnavi.com](mailto:shlee@carnavi.com))

<sup>3\*</sup> 교신저자 대구가톨릭대학교 컴퓨터소프트웨어학부 조교수 ([kimdy@cu.ac.kr](mailto:kimdy@cu.ac.kr))

## I. 서론

자율주행자동차는 스스로 인지, 판단, 제어하는 기능을 가지고 있다. 자율주행자동차의 상황인지 기술은 주변 상황을 인식하여 판단하는 기술이다. 주변 상황 인식을 위해서 카메라, 레이더, 라이다가 가장 많이 활용되고 있다. 카메라를 활용한 인식 기술은 가장 오래된 방식이며, 개발의 편의성과 가격 측면에서 큰 장점을 가지고 있다 [1,2].

카메라를 활용한 영상처리 기술은 정적 객체를 쉽게 추출할 수 있지만, 복잡한 연산과 거리정보를 포함한 3 차원 정보를 획득하기 어려운 단점을 가지고 있다. 영상을 이용한 거리정보를 탐지하기 위해서 스테레오 카메라에 대한 많은 연구들이 진행되고 있다. 스테레오 카메라는 사람의 시각 시스템을 모방하여 2 차원의 좌우 영상을 입력받아 좌우 영상의 시차를 계산하여 3 차원 거리 정보를 획득한다. 스테레오 카메라는 최소 2 대 이상의 카메라를 필요하기 때문에 물리적 공간을 많이 차지하게 되고, 알고리즘 복잡도가 증가하는 단점이 있다.

레이더는 전파의 위상차이를 기반으로 객체에 대한 거리정보를 출력하는 방식으로 객체의 속도와 위상정보를 쉽게 추출할 수 있다. 레이더의 탐지거리가 다른 센서보다 길고 야간이나 구름 많은 날씨에서도 작동한다는 장점이 있지만, 해상도가 낮아서 밀집된 지역에서 정확한 객체 위상을 측정하거나 작은 물체를 감지하기 어려운 단점이 있다.

라이다는 레이저를 광원으로 사용하여 송수광 시간차를 기반으로 거리정보를 출력하는 방식이다. 라이다는 높은 분해능과 넓은 시야각을 가지고 있어 정밀한 객체인식이 가능하다. 하지만, 안개나 강우와 같은 악천후 조건에서 취약한 단점을 가지고 있다.

자율주행자동차 구현에서 주변환경 인식에 대한 중요성이 증가하면서 고분해능 라이다에 대한 관심이 높아지고 있지만, 실시간 주변환경 인식기술을 위한 알고리즘 구현에서는 신뢰성이 가장 중요한 요소이다. 자율주행자동차에 장착되는 센서와 제어모듈이 증가하면서 차량내부네트워크 복잡도와 데이터 양이 증가하여 중앙처리장치의 부하로 인한 처리 지연시간이 증가하는 현상이 발생한다. 이러한 문제점을 해결하기 위해서 센서에서 출력되는 정보를 최소화해야 한다. 라이다 기반 객체인식 알고리즘은 대부분 오픈소스를 활용하여 리소스가 풍부한 고성능 프로세서 플랫폼 환경에서 구현하고 있다. 제한된 리소스를 가지는 임베디드 시스템 상에 오픈소스를 탑재하면, 메모리 점유율과 소프트웨어 복잡도가 증가하여 실시간성을 보장할 수 없다. 임베디드 프로세서를 기반으로 실시간 객체인식 아키텍처를 구현하기 위해서 기능 및 성능 최적화가 반드시 필요하다. 본 연구에서는 라이다에 탑재된 임베디드 프로세서를 기반으로 실시간으로 동적 객체를 인식할 수 있는 아키텍처를 제안하고자 한다. 본 연구의 목적은 리소스 사용량과 데이터 처리시간을 최적화하여 객체인식의 실시간성을 확보하는 것이다. 본 논문의 구성은 2 장에서 객체인식과 관련된 알고리즘에 대한 연구를 소개하고, 3 장에서 제안하는 동적 객체인식 아키텍처 설계 방법론을 기술한다. 4 장에서는 동적 객체인식 아키텍처의 성능검증 환경과 결과에 대해서 분석하고 5 장에서 결론을 맺는다.

## II. 관련연구

라이다 기반 객체인식 아키텍처를 임베디드 프로세서에 탑재하는 것은 기존에 존재하는 객체인식 알고리즘을 최적화 해야한다는 의미이다. 라이다 기반 객체인식 알고리즘에 대한 필터링, 세분화, 군집화 등 다양한 연구가 진행되어 왔다.

라이다는 내부 오류나 외부 영향으로 인한 포인트클라우드에 이상치가 포함될 수 있다. 포인트클라우드 기반 객체인식의 정확도를 향상시키기 위해서 이상치 제거 알고리즘은 반드시 필요하다. 효율적인 이상치 제거 알고리즘은 포인트클라우드에서 정확한 특징점을 추출하여 객체의 경계를 명확히 정의할 수 있다. 최근에는 데이터 클러스터링 [3], 밀도기반함수 [4], 주성분분석[5-7] 등 3 차원 포인트클라우드를 기반 이상치 제거 알고리즘들이 연구되고 있다. 이러한 이상치 제거 알고리즘은 전체 포인트클라우드의 군집개수와 각 군집의 중심점에 대한 정보를 추출하기 위해서 군집 탐색 알고리즘이 필요하다. 군집 탐색 알고리즘은 가장 가까운

이웃 검색 시간을 줄이기 위해서 이진 탐색 구조를 확장한 KD-Tree[8-10]와 Octree 를 많이 활용하고 있다. KD-Tree 는 동적 탐색 알고리즘으로 빠른 탐색 속도를 가지지만, 최악의 경우 모든 하위 노드를 검색하는 단점을 가지고 있다 [11]. Octree 는 단순한 구조와 메모리 효율성을 가지지만, KD-Tree 보다 동작속도가 느리다는 단점을 가진다 [12]. 이상치가 제거된 포인트클라우드를 분리하기 위한 세분화 알고리즘도 필요하다. 세분화에 대한 연구는 Hough 변환 및 RANSAC 과 같은 알고리즘을 이용하여 경계치를 기준으로 객체를 분류하는 방법들이 연구되고 있다 [13-17].

기존 연구들의 대부분은 리소스가 풍부한 고성능 프로세서에 탑재되거나 정적 객체인식 알고리즘들에 대해서 다루었다. 본 논문에서는 기존 연구에서 다룬 알고리즘을 임베디드 프로세서에 적용 가능하도록 처리시간과 CPU 점유율을 최적화하여 실시간으로 동적 객체 인식을 할 수 있는 동적 객체인식 아키텍처를 제안한다.

### III. 동적 객체인식 아키텍처 설계

본 연구는 라이다 센서에서 출력된 포인트클라우드에 대한 원시데이터를 기반으로 동적 객체인식을 위한 아키텍처를 설계하였다. 제안하는 동적 객체인식 아키텍처는 전처리부, 군집화, 객체처리부로 구분된다. 전처리부는 무의미한 값과 노이즈를 필터링하여 포인트클라우드를 재구성하고 프로세서에서 발생하는 연산량을 줄여준다. 군집화는 재구성된 포인트클라우드를 포인트의 연속성을 고려함으로써 객체별 포인트클라우드를 세분화하여 나타낸다. 객체처리부는 분류된 객체의 상태 정보를 연산하여 출력한다. 제안하는 동적 객체인식 아키텍처는 처리속도와 정확성을 측정하여 성능평가를 진행한다. 그림 1 은 동적 객체인식 아키텍처의 순서도를 보여주고 있으며, 점선은 본 연구의 개발범위를 표시하고 있다.

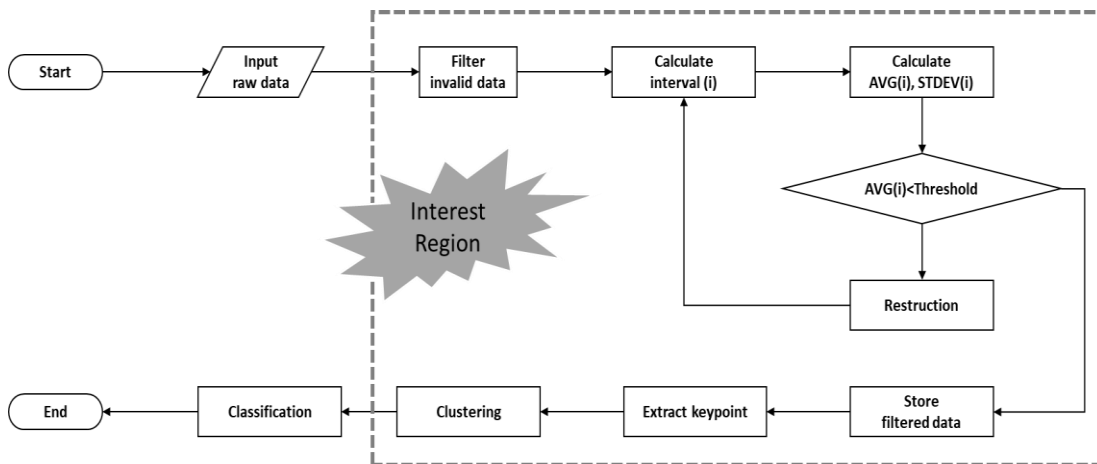


Figure 1. Flow chart of proposed architecture

#### 3.1 데이터 수집

본 연구에서 사용하는 라이다 센서는 BLDC 모터를 이용하여 수평으로 전방을 모니터링 할 수 있는 16 채널 스캐닝 라이다 센서이다. 기존 라이다 센서는 측량, 고정밀지도, 대기분석 등 다양한 분야에서 사용되었다. 본 연구에서 활용한 16 채널 스캐닝 라이다는 차량 전방 장애물을 감지하기 위한 목적으로 제작되었기 때문에 기존 고정밀 라이다와 비교하여 데이터 양이 확연히 적다. 16 채널 스캐닝 라이다의 최대감지거리는 150m 이고, 전방 시야각은  $145^{\circ} \times 9.6^{\circ}$  (H×V)의 성능을 가지고 있다. 단일 프레임의 데이터는 18560 개 포인트클라우드 구성되어 있으며, 데이터 출력속도는 초당 30 프레임이다. 원시데이터는 점의 위상과 거리 정보를 제공하고 있다. 수평 위상정보는 0 에서 1159 까지 정수형으로 제공되며, 거리정보는 실수형으로 제공된다. 데이터는 채널 단위로 입력되고, 채널이 모두

입력된 후 다음 라인 정보가 입력된다. 입력된 정보는 순차적으로 메모리에 저장되어 프로세서에서 호출할 수 있다.

### 3.2 전처리부

전처리부는 군집화 단계에서 객체인식에 대한 연산량을 줄이고, 정확성을 향상하기 위해 노이즈를 제거하는 과정을 포함하고 있다. 전처리부는 라이다에서 입력된 원시데이터의 불필요한 데이터를 제거하는 필터들을 포함한다.

필터부는 Pass Through 필터와 이상치 제거 필터로 구성되어 있다. Pass Through 필터를 이용하여 원시데이터에서 무의미한(Invalid) 데이터를 제거하여 필요한 데이터만 통과시키고, 군집화 된 포인트클라우드에서 이상치 제거 필터를 이용하여 객체인식에서 불필요한 노이즈 성분을 제거한다. Pass Through 필터는 거리정보(Distance)를 기준으로 오류나 객체가 존재하지 않는 정보인 라이다 측정범위를 벗어난 데이터를 필터링한다. 필터링 된 데이터는 좌표변환을 통하여 데이터를 재배열하여 동적 메모리에 저장한다. 라이다로부터 입력된 데이터는 라이다의 수직각도를 고려한 채널(Channel, C), 라이다와 이루는 상대각도(Relative angle,  $\theta$ ), 라이다로부터 떨어진 거리(Distance, D)를 포함하고 있다. 채널, 상대속도, 거리 정보를 가진 점들은 구면좌표계로 표현된다. 컴퓨터 연산의 편의성을 위해서 구면좌표계를 직교좌표계로 변환한다. 직교좌표계로 변환된 점은 3 차원(X, Y, Z) 값을 가지게 된다. 수직각은 채널별 해상도(Vertical resolution, Vr)를 기반으로 계산한다. 수식은 좌표변환에 대한 행렬 연산을 보여주고 있다.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} \sin\theta\cos\omega & \cos\theta\cos\omega & -\sin\omega \\ \sin\theta\sin\omega & \cos\theta\sin\omega & \cos\omega \\ \cos\theta & -\sin\theta & 0 \end{bmatrix} \begin{bmatrix} P_r \\ P_\theta \\ P_\omega \end{bmatrix}$$

Pass through 필터를 통과한 데이터는 이상치(Outlier) 정보를 포함한다. 이상치 정보는 포인트클라우드 군집화의 성능에 큰 영향을 미치기 때문에 반드시 제거되어야 한다. 이상치를 제거하기 위해서 사분범위(Interquartile range)를 활용하였다. 사분범위의 표본은 점 사이의 간격을 활용하고, 점 사이 간격은 유클리드 거리 계산법을 적용한다. 이상치 정보는 점 사이 간격이 일정 범위 이상의 정보로써 순간적으로 값이 크게 변화하는 것을 의미한다. 사분범위를 계산하기 위해서 구간별 사분위수를 구한다. 이상치 추출을 위해서 사분위수에 경계계수를 연산하여 허용오차를 정의한다. 허용오차를 벗어난 지점의 데이터는 이상치로 간주하여 제거한다. 일반적인 사분범위 기반 이상치 제거는 가우시안 분포를 가정하여 상위경계와 하위경계에 대해서 필터링을 수행하지만, 라이다 신호는 밀접한 점에 대해서는 큰 영향이 없으므로 상위경계만 고려한다. 그림 2 는 사분범위를 이용한 이상치 제거 결과를 보여주고 있다. 이상치는 모두 노이즈로 고려하기 어렵지만, 포인트가 특정치 이하의 밀도를 가지는 경우 객체 존재유무에 대한 판단이 어려워서 밀도에 대한 특정치를 정의하여 이상치 제거를 수행하였다.

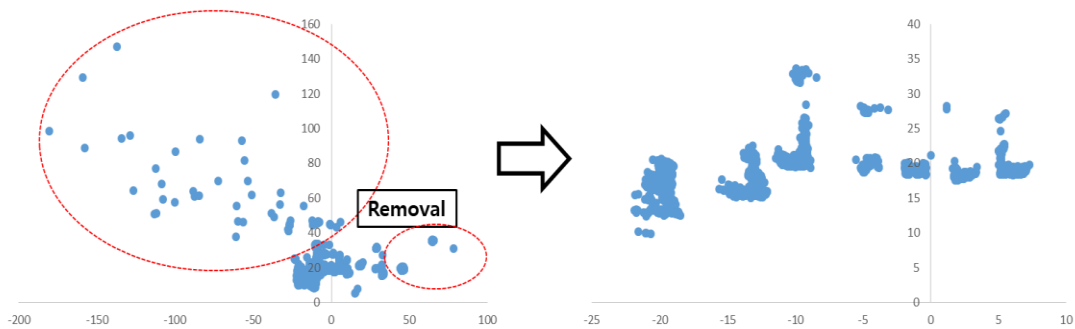


Figure 2. Outlier removal

### 3.3 군집화

이상치를 제거한 포인트 클라우드는 마할라노비스 거리를 기반으로 중앙객체 군집화 알고리즘을 이용하였다. 점 사이 거리를 측정하는 방법은 일반적으로 유클리드 거리계산법을 사용하지만, 포인트클라우드와 같이 군집데이터의 밀집도를 파악하기 위해서 마할라노비스 거리계산법을 많이 사용하고 있다. 마할라노비스 거리계산법은 유클리드 거리계산법에 공분산 행렬 개념을 도입하여 두 데이터 집합 사이의 방향성과 상관도를 분석한다. 수식은 마할라노비스 공식을 보여주고 있으며,  $C^{-1}$ 은 공분산 역행렬을 나타낸다.

$$d(P_n, P_{n-1}) = \sqrt{(P_n, P_{n-1})^T C^{-1} (P_n, P_{n-1})}$$

중앙객체 군집화(K-medoids)는 객체 개수  $k$  개의 군집을 만든다. 동적 객체인식에서 객체의 개수가 얼마나 존재하는지 알 수 없다. 최적의 클러스터 개수를 정하는 방법은 실루엣 기법을 사용하였다. 실루엣 기법은 데이터 클러스터의 일관성을 해석하고 검증하는 방법이다. 효율적인 클러스터 분리는 다른 군집과 충분한 거리 이상 떨어져 있고, 동일 군집 내 데이터 간격은 가까운 것을 의미한다. 실루엣 계수는 클러스터 분리의 효율성을 수치적으로 나타낼 수 있는 지표이다. 수식은 실루엣 계수를 표현하고 있으며  $S(i)$ 는 실루엣 계수이고, 1에 가까울수록 최적화 되었다는 것을 의미한다.  $a(i)$ 는 개별 군집의 데이터 응집도를 나타내고,  $b(i)$ 는 군집 간 분리도를 나타낸다.

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

중앙객체 군집화 알고리즘은 이상치에 영향을 받는 평균값이 아닌 중앙값을 고려하기 때문에 평균 군집화 알고리즘 보다 이상치에 대한 영향이 크지 않다. 실루엣 기법을 통해 정해진  $k$  를 적용하고 점 사이 거리를 연산하여  $k$  군집들의 중심점을 정한다.  $k$  개의 중심점에서 각 클러스터의 거리 합이 최소가 되는 군집을 구한다. 그림 3은 실외 환경에서 군집화 된 모습을 보여주고 있다. 다수의 차량과 사람이 밀집되게 존재하는 상황에서도 잘 동작하는 것을 확인하였다.



Figure 3. Object clustering



#### IV. 아키텍처 성능검증

스캐닝 라이더에 제안하는 아키텍처를 탑재하여 성능을 검증하였다. 라이더 하드웨어에 장착된 프로세서는 ARM Cortex-A9 DUAL Core 이고, 운영체제는 PetaLinux 를 탑재하였다. 프로세서는 작업을 병렬로 처리하기 위해서 멀티프로세싱으로 동작한다 하나의 프로세서는 원시데이터를 저장하여 데이터를 패킷화하는 작업을 수행하고, 다른 프로세서는 객체인식을 위한 연산만 수행한다. 군집화 성능은 실내와 실외에서 다중 동적 객체에 대한 생성과 추적 기능을 검증하였다. 그림 4 는 실험에 사용된 라이더와 성능검증 환경을 보여주고 있다. 본 연구에서 사용된 라이더는 카네비컴에서 개발한 16 채널 전방 스캐닝 라이더이다. 실험환경은 실내와 실외에서 실험을 하였으며, 라이더 특성 상 악천후 환경은 피했다. 라이더의 장착 높이는 차량의 전방 그릴부분을 고려하여 약 70 cm 높이에 고정시켰다. 실외 환경은 밀집된 도로 환경과 넓은 공터로 나누어 실험을 하였다.

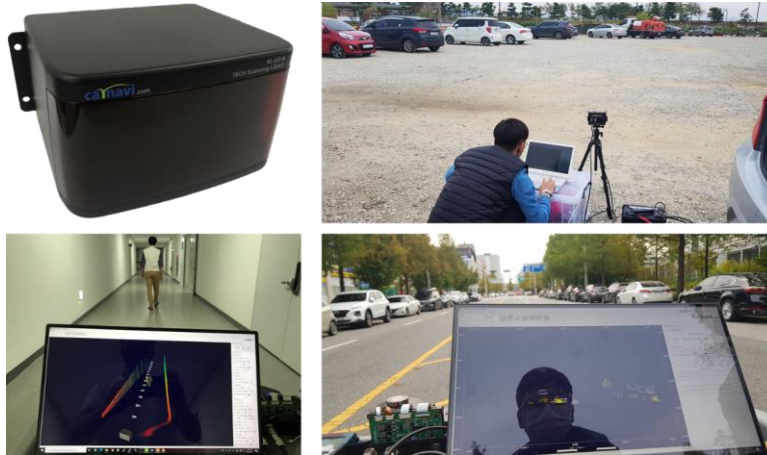


Figure 4. Test environment

그림 5 는 객체의 개수를 변화시켜 포인트클라우드 크기를 조절하고, 포인트클라우드 크기 변화에 따른 프로세서 점유율과 처리시간을 나타내고 있다. 프로세서 점유율과 처리시간은 포인트 개수가 유지되는 동안 누적된 값의 평균을 계산하였다. 포인트클라우드 크기가 작을 때 처리시간과 프로세서 점유율이 매우 낮은 것을 확인할 수 있었다. 실험에 사용된 라이더는 30 Hz 로 동작하기 때문에 단일 프레임의 한계 처리시간은 33 ms 이하로 가정하였다.

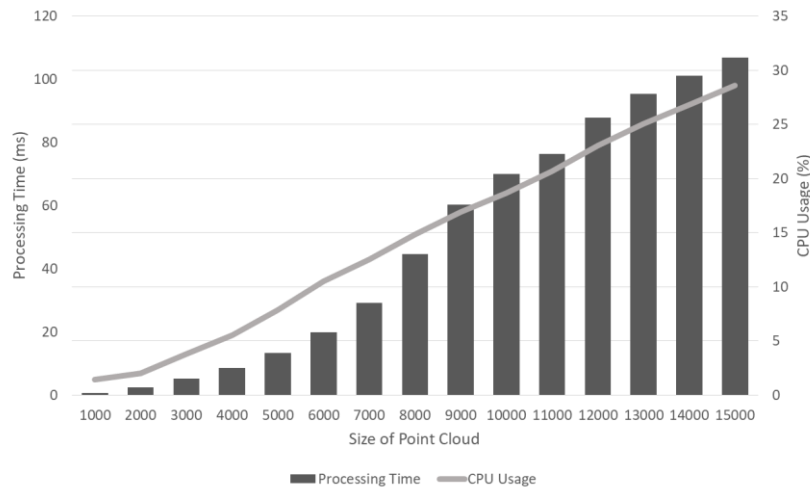


Figure 5. Performance of proposed architecture

제안하는 아키텍처는 출력 가능한 전체 포인트클라우드 크기의 약 85% 이하에서 한계 처리시간을 보장하는 것을 확인하였다. 존재하고 있는 객체의 개수에 따라 새로운 객체가 생성되었을 때 등록시간이 달랐지만, 15 ms 이하에서 객체를 인식하는 것을 확인하였다.

## V. 결론

본 논문에서는 제한적인 리소스를 가지는 임베디드 프로세서를 포함하는 라이다에서 생성된 포인트클라우드를 이용한 동적 객체인식 아키텍처를 제안하고, 실험을 통하여 아키텍처의 성능을 검증하였다. 제안하는 아키텍처는 입력된 데이터를 가공할 수 있는 필터들로 구성된 전처리부와 필터링 데이터의 분리와 객체인식 기능을 포함하고 있는 군집화부로 구성되어 있다. 기존 객체인식 아키텍처는 객체인식과 관련된 대부분의 알고리즘을 포함하고 있어 복잡하고 무겁다. 제안하는 아키텍처는 동적 객체인식에 필요한 부분만 설계하여 경량화를 실현하였다. 아키텍처 경량화는 리소스가 제한된 임베디드 프로세서 상에서 동적 객체인식을 가능하게 한다. 라이다 임베디드 프로세서를 통해 동적 객체인식을 수행함으로써 센서에서 출력되는 데이터를 획기적으로 줄일 수 있어 자율주행자동차에서 차량내부 네트워크 구성의 편의성을 제공할 수 있을 것이다. 본 논문에서는 동적 객체인식 아키텍처의 처리시간만 고려하여 최적화를 진행하였다. 향후 연구에서는 객체인식 정확도를 최적화하여 동적 객체인식 아키텍처의 신뢰성을 확보할 것이다.

## V. 감사의 글

이 연구는 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2018R1D1A1B07041296).

## VI. 참고문헌

- [1] Thakran, Y. Toshniwal, D. "Unsupervised outlier detection in streaming data using weighted clustering," In Proc. of the International Conference on Intelligent Systems Design & Applications(ISDA), Kochi, India, 2012, pp. 27–29.
- [2] Schall, O. Belyaev, A. Seidel, H.P. "Robust filtering of noisy scattered point data," In Proceedings of the Eurographics/IEEE Vgtc Conference on Point-Based Graphics(PBG), NY, USA, 21–22 June 2005; pp. 71–77.
- [3] Zaman, F. Wong, Y.P. Ng, B.Y. "Density-based denoising of point cloud," In Proc. of the 9th International Conference on Robotic, Vision, Signal, Processing and Power Applications(RVSPPA), Springer, Singapore, 2017.
- [4] Huang, D. Du, S. Li, G. Zhao, C. Deng, Y. "Detection and monitoring of defects on three-dimensional curved surfaces based on high-density point cloud data," Precision Engineering., Vol. 52, pp. 79–95, 2018.
- [5] Narváez, E.A.L. Narváez, N.E.L. "Point cloud denoising using robust principal component analysis," In Proceedings of the First International Conference on Computer Graphics Theory and Applications(GRAPP), Setúbal, Portugal, 2006, pp. 25–28.
- [6] Zhu, H. Zhang, J.;Wang, Z. "Arterial spin labeling perfusion MRI signal denoising using robust principal component analysis," J Neurosci Methods, 2017, 295, 10–19.
- [7] Mahmoud, S. M. Lotfi, A. Langensiepen, C. "User activities outlier detection system using principal component analysis and fuzzy rule-based system," In Proc of the International Conference on Pervasive Technologies Related to Assistive Environments(PTRAE), Crete, Greece, 2012, pp. 1–8.
- [8] Bentley, J.L. "Multidimensional Divide and Conquer," Communication. ACM, Vol. 23, No. 4, pp. 214–229, 1980.
- [9] Friedman, J.H. Bentley, J.L. Finkel, R.A. "An Algorithm for Finding Best Matches in

- Logarithmic Expected Time,” *ACM Trans. Math. Software*. Vol. 3, No. 3, pp. 209–226, 1977.
- [10] Moore, A. “Very Fast EM-Based Mixture Model Clustering Using Multiresolution Kd-trees,” In *Proc. Advances in Neural Information Processing Systems II (NIPS)*, MA, USA, 1999, pp. 543–549.
- [11] Han, S.; Kim, S.; Hoon Jung, J.; Kim, C.; Yu, K.; Heo, J. “Development of a hashing-based data structure for the fast retrieval of 3D terrestrial laser scanned data,” *Comput. Geosci*, Vol. 39, pp. 1–10, Feb, 2012.
- [12] Han, S.-H.; Lee, S.-J.; Kim, S.-P.; Kim, C.-J.; Heo, J.; Lee, H.-B. A Comparison of 3D R-tree and Octree to Index Large Point Clouds from a 3D Terrestrial Laser Scanner. *J. Korean Soc. Surv. Geod. Photogramm. Cartogr.* 2011, 29, 39–46.
- [13] Vosselman, G. Dijkman, S. “3D building model reconstruction from point clouds and ground plans,” *Int. Arch. Photogramm. Remote Sens*, Vol. 34, pp. 37-44, 2001.
- [14] Overby, J. Bodum, L. Kjems, E.B. Ilsøe, P.M. “Automatic 3D Building Reconstruction from Airborne Laser Scanning and Cadastral Data Using Hough Transform,” *Int. Arch. Photogramm. Remote Sens*. Vol. 35, pp. 296-301, 2004.
- [15] Borrmann, D. Elseberg, J. Lingemann, K. Nüchter, A. “The 3d hough transform for plane detection in point clouds: A review and a new accumulator design” *3D Res*. Vol. 2, No. 3, Jan, 2011.
- [16] Anagnostopoulos, I. Patrăucean, V. Brilakis, I. Vela, P. “Detection of walls, floors and ceilings in point cloud data,” In *Proc. Constr. Res. Congr, San Juan, Puerto Rico*, 2016, pp. 2302–2311.
- [17] Schnabel, R. Wahl, R. Klein, R. “Efficient RANSAC for point-cloud shape detection,” *Comput. Graph. Forum*, Vol. 26, No. 2, pp. 214-226, Jun, 2007.



## 저자 소개

---



정민우 (*Minwoo Jung*)

2015년 2월 경북대학교 전자공학과 박사  
2015년~2018년 경북 IT 융합산업기술원 선임연구원  
2018년~현재 (주)카네비컴 기술연구소 책임연구원

관심분야: 차량용 센서, 임베디드 소프트웨어, 레이저광학, 사물인터넷



이상훈 (*Sanghoon Lee*)

2013년 2월 경북대학교 전자공학과 박사수료  
2015년~2018년 경북 IT 융합산업기술원 연구원  
2018년~현재 (주)카네비컴 기술연구소 선임연구원

관심분야: 아날로그 회로설계, ASIC, LIDAR 센서



김대영 (*Dae-Young Kim*)

2010년 8월 경희대학교 대학원 컴퓨터공학과 박사  
2010년~2013년 LIG 넥스원 통신연구센터 선임연구원  
2013년~2015년 (주)에어플러그 선임연구원  
2015년~2017년 창신대학교 컴퓨터소프트웨어공학과 조교수  
2017년~현재 대구가톨릭대학교 컴퓨터소프트웨어학부 조교수

관심분야: 모바일 네트워킹 및 컴퓨팅, 머신러닝, 네트워크 시스템

---