

Important Facility Guard System Using Edge Computing for LiDAR

Eun-Kyung Jo[†] · Eun-Seok Lee^{††} · Byeong-Seok Shin^{†††}

ABSTRACT

Recent LiDAR(Light Detection And Ranging) sensor is used for scanning object around in real-time. This sensor can detect movement of the object and how it has changed. As the production cost of the sensors has been decreased, LiDAR begins to be used for various industries such as facility guard, smart city and self-driving car. However, LiDAR has a large input data size due to its real-time scanning process. So another way for processing a large amount of data are needed in LiDAR system because it can cause a bottleneck. This paper proposes edge computing to compress massive point cloud for processing quickly. Since laser's reflection range of LiDAR sensor is limited, multiple LiDAR should be used to scan a large area. In this reason multiple LiDAR sensor's data should be processed at once to detect or recognize object in real-time. Edge computer compress point cloud efficiently to accelerate data processing and decompress every data in the main cloud in real-time. In this way user can control LiDAR sensor in the main system without any bottleneck. The system we suggest solves the bottleneck which was problem on the cloud based method by applying edge computing service.

Keywords : Edge Computing, LiDAR, Guard System, Point Cloud, Data Transformation

LiDAR용 엣지 컴퓨팅을 활용한 중요시설 경계 시스템

조 은 경[†] · 이 은 석^{††} · 신 병 석^{†††}

요 약

최근의 LiDAR(Light Detection And Ranging) 센서는 실시간으로 주변에 있는 물체를 스캔하는 데 사용된다. LiDAR 센서를 이용하여 주변 환경을 스캔할 경우 감지되었던 사물들에 대한 변화를 감지하고 실시간으로 움직이는 물체를 인식할 수 있다. 센서들의 제작 비용이 낮아지면서 LiDAR는 중요시설의 경계, 스마트시티, 자율주행차 등 다양한 산업 분야에서 다양하게 활용되고 있다. 이러한 LiDAR 데이터는 실시간에 사물을 스캔하는 만큼 입력 데이터의 크기가 크다. 따라서 이러한 LiDAR를 활용하는 시스템에서는 이러한 대용량 데이터의 실시간 처리가 병목이 될 수 있어서 이러한 대용량 처리에 대한 대안이 필요하다. 본 논문에서는 엣지 컴퓨팅 서버를 이용하여 방대한 포인트 클라우드를 압축하여 빠르게 처리하는 엣지 컴퓨팅 기법을 제안한다. LiDAR 센서의 레이저의 반사 범위가 제한되어 있으므로 실시간으로 넓은 영역을 스캔하기 위해서는 여러 대의 라이더를 사용해야 한다. 따라서 실시간으로 물체를 감지하거나 인식하기 위해서는 여러 개의 LiDAR 센서에 대한 데이터를 한 번에 처리해야 한다. 엣지 컴퓨팅은 데이터 가속을 수행하기 위해 포인트 클라우드를 효율적으로 압축하고 모든 데이터를 메인 클라우드에서 실시간에 압축해제 하여 사용할 수 있도록 설계되었다. 이를 통해 사용자는 시스템을 중앙에서 병목 없이 실시간에 LiDAR 센서들을 제어할 수 있다. 실험에 사용된 시스템은 이러한 엣지 컴퓨팅 서비스를 적용함으로써 기존 클라우드 기반 방식에서 문제였던 데이터 병목 현상을 효과적으로 해결하였다.

키워드 : 엣지 컴퓨팅, LiDAR, 경계 시스템, 포인트 클라우드, 데이터 변환

1. 서 론

기술의 발전으로 인하여 중요시설들이 늘어남에 따라 군대만이 아니라 공장이나 스마트시티 등 다양한 산업분야에서 시설에 대한 경계 시스템이 필요하게 되었다. 이 시스템은 사

람이 직접 수행하던 중요시설에 대한 경계 업무를 대신하는 시스템으로, 시설 외부에서 내부로 허가받지 않은 침입이 발생하는 상황을 사전에 방지하기 위해 사용된다.

이러한 경계 시스템에서는 최근 보편화 되기 시작한 LiDAR 센서[1]를 이용하여 고해상도의 정확한 3D Depth 정보를 얻을 수 있고 시야각 설정이 기기의 형태에 따라 자유롭다는 장점이 있다. 따라서 경계 시스템에서 10~50미터 사이의 근거리 감지된 사물에 대해 파악하는데 유리하다. 하지만 이러한 LiDAR 센서가 초당 스캔하는 데이터 량이 많기 때문에 센서에서 서버로 전송되어야 하는 데이터량이 많다. 이러한 문제는 다수의 라이더를 사용할수록 더욱 큰 문제가 생길 수 있다.

본 논문에서는 분산 처리에 적합한 엣지 컴퓨팅을 활용하여 이러한 LiDAR를 효율적으로 활용할 수 있는 클라우드형

※ 이 성과는 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2022R1A2B5B01001553).

※ 이 논문은 2022년 한국정보처리학회 ASK 2022에서 "스캔형 라이더 센서를 활용한 이동 물체 감지 방법"의 제목으로 발표된 논문을 확장한 것이다.

† 준 회 원 : 인하대학교 전기컴퓨터공학과 석사과정

†† 정 회 원 : 유한대학교 VR게임·앱학과 조교수

††† 중신회원 : 인하대학교 컴퓨터공학과 교수

Manuscript Received : August 1, 2022

Accepted : August 11, 2022

* Corresponding Author : Byeong-Seok Shin(bsshin@inha.ac.kr)

경계 시스템을 설계하였다. LiDAR 센서는 일반적으로 3D 포인트 클라우드를 제공한다. 이러한 3D 데이터를 초당 20~30세트를 제공하고 있다. 이것은 비디오의 픽셀 데이터와 유사하지만 LiDAR는 압축으로 유명한 비디오 분야에 비해서 압축 알고리즘에 대한 연구[2,3]들의 압축률이 효율적이지 않다. 또한 실시간성을 살리기 위해서도 이 많은 데이터를 서버 컴퓨터로 전부 전송하는 것은 비효율적이다. 제안하는 방법은 이러한 데이터를 효율적으로 줄일 수 있는 알고리즘과, 이것을 엷지 컴퓨팅 모델을 제공한다. 주요시설에서 침입에 대한 감지를 위해서는 정적인 포인트 데이터와 유동적인 포인트 데이터를 분리해내야 한다[4]. 이러한 데이터 분리를 엷지에서 수행할 수 있도록 엷지 device와 컴퓨팅 서비스를 제작하였다. 이 서비스는 최종적으로 감지된 물체에 대한 트래킹 정보를 메인 서버로 전송한다. 이러한 엷지 컴퓨팅을 활용한 시스템은 다수의 포인트 클라우드 데이터를 줄일 수 있을 뿐만 아니라 다수의 포인트 클라우드로 인한 사물 감지 연산 또한 효율적으로 줄일 수 있다.

제안하는 시스템은 중요시설 경계뿐만 아니라 다수의 라이더를 활용해야 하는 다른 분야에도 효율적으로 사용할 수 있다. 2절에서는 관련된 연구에 대해 설명하고, 3절에서는 본론에 대해 설명한다. 4절에서는 제안하는 방법에 대한 실험 결과를, 5절에서는 논문의 결론을 맺는다.

2. 관련 연구

기존 중요 시설물들에 대한 경비 업무는 CCTV와 그것을 지켜보는 사람에 의존하였다. 이러한 중요 시설물들을 효율적으로 하기 위해서 다양한 IOT 센서들을 이용하여 경계 시스템이 개발된다.

최근 이러한 경계시스템은 적외선 CCTV와 같은 영상기기를 활용한 Video Management System (VMS)을 이용하여 경비업무를 수행하고 있다. 이것은 사람보다 효율적으로 외부로부터 중요시설에 대한 침입을 감지할 수 있다. 보다 먼 거리에 대한 사물의 감지에 대한 요구사항이 생기면서 VMS를 활용한 경계 시스템은 RAdio Detection And Ranging (RADAR) 센서[3]로 시스템을 보완하였다. RADAR는 강한 전자파를 발사하고 그것이 물체에 맞고 반사되어 되돌아 오는 전자파를 분석하여 대상물과의 거리를 측정하는 장비로 Radio Direction Finder (RDF) 혹은 High Frequency Direction Finder (HFDF)으로도 알려져 있다. 레이더에 파장이 긴 저주파를 사용하면 전파의 감쇄가 작고 먼 곳까지 탐지할 수가 있지만 정밀한 측정이 되지 않아 해상도는 나빠진다. 반대로 파장이 짧은 고주파는 공기중에 포함되는 수증기, 눈, 비 등에 흡수 또는 반사되기 쉽기 때문에 감쇄가 커서 먼 곳까지 탐지하지 못하지만 높은 해상도를 얻을 수가 있다.

RADAR 센서는 연속적으로 송신기에서 생성된 전자파가 물체에 반사되어 수신기에 도착할 때까지의 시간을 통해서 물체가 위치한 각도와 거리를 계산할 수 있다. 이것은 펄스 방식과 도플러 방식[5], Frequency Modulated CW Radar[6] 등 다양한 방식으로 나뉘는데 사용하는 신호에 따

라 도플러방식처럼 짧은 거리용으로 사용할 수도 있고, 펄스 방식처럼 장거리용으로 사용할 수도 있다. 측정 주기에 따라서 물체의 움직임 또한 감지해낼 수 있으며, 물체가 움직임이 빠를수록 더 쉽게 감지할 수 있다. 하지만 이러한 RADAR 센서는 멀리 측정하기 위해서는 측정 주기가 길어질 수 밖에 없다. (주기가 짧을수록 멀리서 오브젝트에 반사된 신호가 다음 신호 송출까지 도달하지 못할 수 있다.)

Light Detection And Ranging (LiDAR) 센서[1]는 고출력의 펄스 레이저를 발사하여 레이저가 물체에 반사되어 돌아오기까지 걸린 시간으로 현재 위치와 물체와의 거리를 측정한다. 이러한 레이저를 이용한 거리 측정을 다양한 각도로 연속해서 수행할 경우 거리 정보와 조사 각도를 이용하여 반사된 물체의 위치 정보 변환할 수 있다. 이러한 데이터들의 군집을 포인트 클라우드라 하며 물체의 표면 정보를 유추하기 위해 좌표를 계산하는데 사용된다.

라이더의 경우에는 905nm에서 1,550nm 레이저를 활용하기 때문에 저주파의 전자파를 사용하는 RADAR[7]보다 더 정밀하게 공간을 측정할 수 있다. 이러한 라이더 센서를 사용한 감지 시스템은 RADAR 센서를 이용할 때보다 정확한 물체의 형상을 감지해 낼 수 있다는 장점이 있다. 따라서 라이더 센서는 항공기에서 지면을 스캔하여 시각화 할 수 있도록 대용량 포인트 클라우드를 삼각화 하는 용도로 사용되었다 [1]. 이 방법은 항공기에서 장시간 지면을 스캔하면서 포인트 클라우드를 생성한다. 이렇게 생성된 포인트 클라우드들은 하나의 PC에서는 처리할 수 없는 대용량 데이터였기 때문에 클라우드 컴퓨팅을 활용하여 분산 처리를 하거나[8] GPGPU를 활용[9]하는 등 다양한 off-line 방식으로 삼각화 작업을 가속화하는 기법이 연구되었다.

LiDAR 센서는 레이저를 이용하여 거리를 측정할 수 있는 센서로 다양한 종류가 존재한다. 대표적으로 Mechanical Scanning lidar[10]의 경우 모터를 이용하여 물리적으로 LiDAR 센서를 회전시켜 주변을 스캔하는 방식으로 수평 Field of View (FOV)가 넓은 장점이 있다. Solid State Lidar [11]의 경우 회전 장치 없이 고정된 형태로 사용되는 라이더이다. 주로 solid state lidar에서 사용되는 Micro Electro Mechanical Systems (MEMS) 방식[12], Optical Phase Array(OPA)[13] 등이 있다.

LiDAR가 대중적으로 보급이 되면서 다양한 산업분야에서는 LiDAR 데이터를 실시간에 활용할 수 있도록 다양한 연구를 진행하였다. 주로 사물을 스캔해 목표로 했던 solid state lidar와는 달리 달리 모터와 레이저 송수신기를 결합하여 Mechanical Scanning lidar[10]가 활용되기 시작하였다. 자율주행 자동차는 회전형 LiDAR를 이용하여 주변의 사물을 스캔한다[14]. 이것은 모터의 초당 회전수와 레이저의 송신 속도, 빛의 주파수 등을 활용하여 멀티 채널로 분리하는 작업을 통하여 보다 더 넓은 영역을 빠르게 스캔할 수 있도록 하였다. 최근 모바일 기기에도 LiDAR 센서가 탑재되어 사람과 자동차 등을 구분할 수 있는 알고리즘 [15]이나 실시간에 스캔한 실내를 3D 오브젝트로 모델링하는 알고리즘[16]들이 연구되고 있다.

제안하는 방법은 경계 시스템에서 위험요소로 판단되는 사물을 감지하고 사용자에게 위치를 즉시 보고해주는 실시간 경계 업무의 자동화를 목표로 한다. 경계 분야는 매우 정교하게 침입을 감지해 내야하기 때문에 RADAR만으로는 침입 감지측면에서 정확도가 부족하며, 무게 센서나 온도센서[17]를 활용한 방법으로는 커버할 수 있는 영역이 제한적이며 범위가 매우 적다. Mechanical scanning LiDAR와 RADAR를 활용한 본 시스템은 보다 넓은 지역을 hybrid하게 체크할 수 있으며, 각 센서의 특성을 잘 활용하여 침입을 감지할 수 있다.

3. LiDAR 데이터의 압축을 위한 엣지컴퓨팅 기법

본 논문은 라이다 센서에서 실시간으로 전송되는 대용량 포인트 클라우드 데이터를 효과적으로 줄일 수 있는 압축 알고리즘과 동시에 다수의 라이다 센서를 동기화하고 처리할 수 있는 중앙 제어 기술 개발을 목표로 한다. 이 목표를 달성하기 위해서 LiDAR 기반의 실시간 경계 시스템을 위해 엣지 컴퓨팅 방법을 제안한다.

주요 시설의 경우 군부대나 스마트 시티처럼 광범위한 지역을 모니터링 해야 하는 경우에는 상황에 맞는 다양한 센서들간의 연계가 매우 중요하다. 일반적으로 최근의 경계 시스템에서는 다양한 형태의 LiDAR센서를 함께 사용하고 있다. 이것은 경계할 지역의 특성에 맞추어 LiDAR의 모터를 설계하거나, 때로는 solid state LiDAR와 같은 형태가 필요하기 때문이다. 이 장비들은 서로 다른 회사에서 만들어지며 대부분이 자체적으로 프로토콜을 제공한다. 장비들은 서로 다른 형태의 데이터를 출력값으로 제공한다. 경계 시스템은 이러한 다양한 타입의 데이터를 사용자가 활용할 수 있도록 중앙에서 하나로 통일하고 관리할 수 있도록 데이터의 표준 형태가 설계되어야 한다.

제안하는 방법에서는 원거리에서 회전형 4채널 LiDAR를 사용하고, 그리고 사용자가 물체를 정확하게 식별하기 위해 카메라와 혼합된 MEMs 방식의 LiDAR는 근거리에서 사용한다. (Intel Realsense LiDAR · I5150모델을 사용하였다.) 회전형 LiDAR는 초당 25회를 회전하며 약 10만개의 포인트를 수집한다. MEMs방식의 경우는 카메라의 CCD와 같이 이차원 이미지 형태의 배열의 형태로 데이터가 수집된다.

Fig. 1은 제안하는 방법의 엣지 컴퓨팅 시스템의 동작 순서도에 대해 설명한다. 경계 시스템은 LiDAR 센서의 출력 데이터를 활용하여 움직이는 물체를 감지할 수 있다. 감지가 되었을 경우엔 감지된 물체의 위치를 파악해서 MEMs LiDAR가 그곳을 바라보게 하고 고해상도의 MEMs LiDAR에도 감지되었을 경우 사용자에게 침입을 알려주는 것을 목표로 하고 있다. 따라서 중앙 클라우드 서버의 management 서비스에서는 LiDAR데이터를 입력 받아서 움직이는 사물을 감지하였을 때 카메라에 사물을 향하도록 카메라가 설치된 브라켓을 회전하는 명령을 내려야한다. 따라서 앞서 설명한 다른 타입의 센서에 대한 중앙 서버의 효율적인 관리를 위해서 하나의 좌표계로 입출력 데이터를 변환할 수 있어야 한다. 센서

들은 연속적으로 꾸준히 데이터를 보내야 한다. 클라우드와 같은 환경에서는 TCP의 경우 혼잡제어 알고리즘[18]을 기기에서 제공하지 않는 한 네트워크의 대역폭이 작을 경우 패킷이 누적되는 문제가 생길 수 있다. 하지만 이러한 혼잡제어 모듈이 내장된 하드웨어는 흔치 않으며 대부분 UDP 프로토콜을 사용하고 있다.

UDP는 데이터그램이 유실되거나 순서가 바뀔 수 있다. 이러한 문제는 보통 장치와 서버 컴퓨터와의 네트워크 상태에 따라서 발생한다. 이러한 장치들을 사용할 경우에는 클라우드 컴퓨팅 환경보다는 로컬 서버에서 더욱 정확하게 처리할 수 있다. 제안하는 방법은 이러한 단점을 엣지 컴퓨팅을 활용하여 해결하였고 이것을 같이 LiDAR센서의 데이터를 받아 처리할 수 있는 엣지 컴퓨팅 서버를 사용하여 해결하였다. 엣지 컴퓨팅은 연산과 데이터 스토리지를 데이터 소스와 가까운 곳으로 가져와 응답 시간을 개선하고 대역폭을 절약하는 것을 목표로 한다. 따라서 제안하는 방법에서는 이러한 엣지 컴퓨팅을 활용하여 빠르게 데이터를 수집하고 클라우드의 중앙 매니징 서비스로 전송할 데이터 압축과 변환을 엣지 서버에서 수행한다.

데이터의 압축과 변환을 수행하기 위해서 LiDAR 센서의 데이터를 Fig. 2와 같이 간소화하는 것이 가장 중요하다. 제안하는 방법은 시스템이 동작하기 전에 정적인 물체들을 스캔하여 클라우드에 있는 서버로 미리 전송해야 한다. 미리 전송된 정적인 물체에 대한 데이터는 움직이지 않는 물체만으로 구성된 포인트 클라우드 데이터이기 때문에 매번 스캔할 때마다

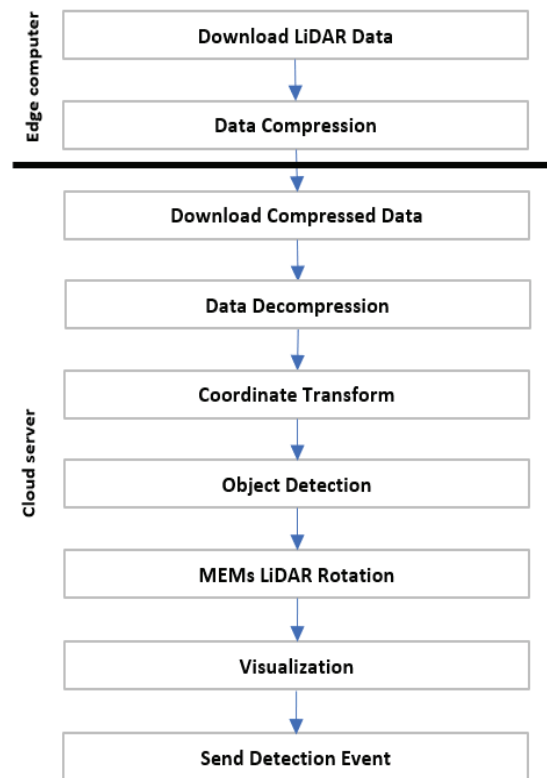


Fig. 1. Overall Procedure of Our Method

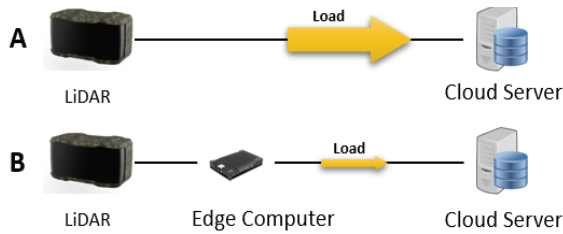


Fig. 2. Network Load Reduction According to the Proposed Compression Method Using Edge Computing

동일한 위치에서 검출된다. 따라서 이 데이터들을 정렬하여 저장해두면 스캔되는 데이터로부터 정적인 부분들을 제거하여 초당 전송해야 하는 데이터량을 효과적으로 줄일 수 있다. 이렇게 줄여진 데이터들은 클라우드 서버로 전송되어 통일된 3차원 공간좌표에 적용할 수 있도록 변환되어야 한다. 이것은 서로 다른 센서들의 데이터 값을 통일된 3D 월드 공간 좌표에서 확인할 수 있게 해준다. 제안하는 방법은 x, y, z 좌표 중에서 x 와 z 는 지도에서 널리 사용되는 UTM 좌표를 사용하였고 높이를 나타내는 y 는 geoid 높이 모델을 사용하였다.

LiDAR센서는 센서가 설치된 위치와 방향을 원점으로 물체가 감지된 각도와 거리를 출력값으로 한다. 2차원 좌표 (x, y) 에 대해 원점 O 와 물체와의 거리 d , 각도 θ 에 대해서 다음과 같은 특성을 갖는다.

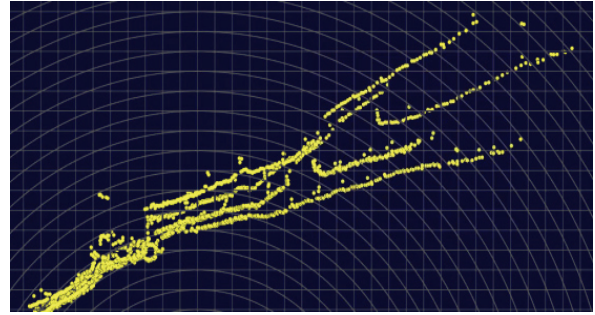
$$d = \sqrt{x^2 + y^2}, \quad y = d \cos \theta \quad (1)$$

거리 d 와 각도 θ 값이 주어질 경우 위의 식은 회전행렬 [19]을 통해서 다음과 같이 계산할 수 있다.

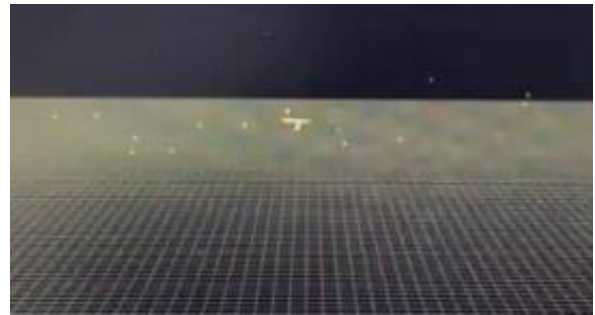
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Equation (2)에서 x 는 0 그리고 y 에 d 를 대입한 후 Equation (2)를 통하여 원하는 좌표값을 얻을 수 있다. Top View에서 지면을 xy 평면이라 하였을 때 결과는 Fig. 3의 (a)에서 볼 수 있다. 이러한 데이터를 보다 사실적으로 모니터링 하기 위해서 제안하는 방법은 3차원 좌표계를 기준으로 좌표를 통일해야 한다. 따라서 여기에 z 축이 추가가 되며, LiDAR 설치시에 x, y, z 세가지 축에 대해 기울어진 각도를 계산하여 회전 축으로 정한다. 이 축에 대해 회전행렬을 적용하여 x, y, z 값을 구하였다. 해당 좌표로 화면에 그려진 3차원 데이터는 Fig. 3의 (b)에서 찾아볼 수 있다.

주요시설에 일반적으로 경계해야하는 구역의 넓이가 LiDAR 보다 넓기 때문에 제안하는 시스템에서는 Fig. 4와 같이 다수의 LiDAR 센서를 활용한다. LiDAR 센서의 개수가 많아질수록 하나의 LiDAR 엣지 서버에서 처리해야하는 포인트 데이터 량이 점점 늘어 병목 현상을 야기할 수 있다. 제안하는 방법은 이러한 문제점을 데이터의 압축기법으로 해결할 수 있다.



(a)



(b)

Fig. 3. The 3D View of the Proposed Method. (a) is When the Important Area is Visualized in the Top View. (b) is When the Important Area is Visualized with a Quarter View

제안하는 압축기법은 이전 데이터와의 유사도 검사를 통해서 수행한다. 일반적으로 경계 시스템에서 주로 사용되는 라이다는 넓은 field of view (FOV)가 넓고 거리가 멀며, 초당 레이저의 송수신 횟수가 많을수록 좋다. 이러한 조건을 만족하는 회전형 다채널 LiDAR이다. 이 LiDAR는 모터가 회전하며 최대 360도의 FOV를 지원하며 멀티채널을 지원할 경우 vertical 축으로도 감지가 가능하다는 장점이 있다. 하지만 단일 모터를 사용하기 때문에 회전이 하나라 채널당 2차원 평면에 대한 감지만 가능하다. 이 LiDAR센서의 모터는 회전하기 때문에 감지 범위가 채널당 2차원 원으로 표현되기 때문에 제안하는 방법은 Hausdorff distance[20]를 이용하는 감지 방법에 대해 제안한다. Hausdorff distance 는 위상 공간상에서 두 물체 사이의 거리를 측정하는 데 사용한다. 이 방법은 위상 차이를 이용하는 방법으로 LiDAR 센서에서 보내는 데이터그램의 순서대로 정렬만 잘하면 특정 집단의 거릿값만으로 손쉽게 위상 차이를 구할 수 있다. Fig. 4는 해당 방법의 예시에 관해서 설명한다.

Fig. 4의 blue 선의 경우 초기에 감지한 지형지물에 대한 거리 데이터를 의미한다. 붉은 물체가 감지되었을 때는 정적인 포인트들에서 측정된 원래의 거리 값과는 다르게 빨간 화살표에 해당하는 detected distance 값이 데이터그램을 통해 전달된다. LiDAR 센서의 모터의 경우 일정한 레이저 측정 주기로 360도를 회전하고 있어서 항상 동일한 각도에 대해 거리 값을 측정한다. 따라서 original 포인트 o 들의 집합을 O 라고 할 때, 새로 감지된 포인트 p 의 주변 인덱스들에 해당하는 o 집합 O 과 붉은 물체가 감지된 포인트인 p 에서

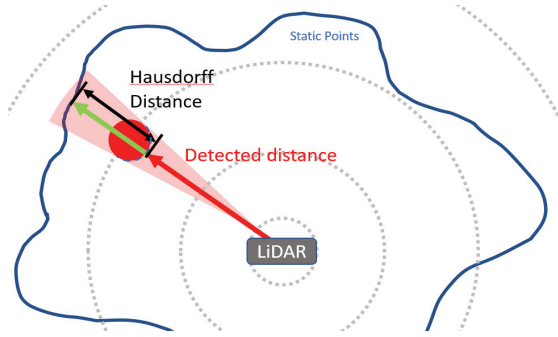


Fig. 4. Proposed Method to Select only the Detected Points Using the Hausdorff Distance

가장 멀리 떨어져 있는 o 에 대한 Hausdorff distance인 H 를 다음 수식과 같이 구할 수 있다.

$$\forall o \in O : H(o, P_a) = \max : \sqrt{((o_a \sin \theta)^2 + (d_n \cos \theta - p_a^2)}$$

$$O' = \{o \mid o \in O, o \text{ is within } \tau \text{ of } p\} \quad (3)$$

식에서 p 와 o 의 아래 첨자로 사용한 a 는 해당 포인트의 거리를 뜻한다. τ 는 각도를 의미하며 이것은 o 를 읽어들 때 p 의 인덱스로부터의 앞뒤로 어느 범위의 인덱스값을 사용할 것인지를 선택한다. 이렇게 구한 H 값이 인식하고자 하는 사물의 크기 이상일 경우, 우리는 새로운 물체를 감지해냈다고 할 수 있다.

하지만 LiDAR 센서의 경우 비나 눈이 오거나 반사판을 사용할 경우와 같이 레이저에 간섭이 생길 수 있는 경우에는 노이즈가 발생하거나 물체를 감지하지 못할 수 있다. 이러한 문제점을 해결하기 위해서 제안하는 방법은 두가지 해결방법을 동시에 사용하였다.

첫번째는 감지된 p 에 대한 군집화를 하여 객체를 인식하는 방법이다. 이것은 객체의 크기에 따라 최소 감지할 수 있는 영역과 포인트의 개수를 정해 놓고, 가장 먼저 감지되는 p 에 대해서 군집을 형성한다. 새로운 p 가 감지될 때마다 군집의 중앙에서 물체의 크기에 해당하는 범위 안에 있는지 체크하며, 군집과 떨어져 있을 경우 새로운 군집을 만든다. 군집화는 K-Means 알고리즘을 통해 수행하였다. 이때 K의 개수가 감지된 사물의 숫자는 아니다. 탐지하려는 물체에 대해 라이다가 감지할 수 있는 최대 포인트의 개수를 탐지하지 못했을 경우에 제안하는 방법에서는 잘못 감지된 데이터로 판단하고 파기한다. 이것으로 노이즈에 대한 문제점을 쉽게 해결할 수 있으며, 이 연산은 포인트마다 병렬로 연산 할 수 있기 때문에 GPGPU를 썼을 경우 클라우드에서 CPU만 사용하였을 경우보다 훨씬 빠른 속도로 물체를 검출해낼 수 있다.

두번째 방법은 LiDAR 이외의 센서에서 물체를 감지하여 LiDAR 센서에서 감지한 물체와 유사지점에서 발견되는가를 비교해보는 방법이다. 이 방법은 장치간의 객체 인식 타이밍, 허용 오차범위 등 다양한 차이점이 존재하기 때문에 가장 성능이 나쁘고 느린 장치에 맞추어 알고리즘을 수행한다. 제안하는 방법에서는 MEMs LiDAR의 카메라를 이용한 객체 인식 알고리즘[21] 오차율을 줄였다.

각 장비들은 엣지 컴퓨터에서 데이터를 추리는 작업을 수행한다. 우선 첫번째 방법인 군집화 방법은 이 엣지 컴퓨터에서 수행된다. 이 때 노이즈에 해당되는 데이터가 필터링이 되면 사물로 판단되는 포인트들만 군집화하여 사물별로 구분할 수 있다. MEMs LiDAR는 원거리에 쓰이는 회전형 LiDAR에서 물체를 감지한 후 그곳을 향하도록 회전을 하게 된다. 따라서 MEMs LiDAR의 경우에는 가장 늦게 사물을 인식하게 된다.

이러한 시간차의 발생은 빠르게 움직이는 물체가 있을 경우에 서로 다른 위치에서 사물을 감지하여 제대로 탐지했음에도 불구하고 오탐으로 인지할 수 있는 위험성을 가지고 있다. 따라서 제안하는 방법은 시간축에 한하여 unification을 통해서 감지된 물체의 위치를 동일한 시간대에 체크해야 한다. 동일한 시간대에 물체를 측정하기 위해서 제안하는 방법은 우선순위를 사용한다. 가까운 곳에서 물체가 감지가 된 상황이 가장 우선순위가 높은 경우에 속한다. 반면 감시 구역에서 가장 멀리 있는 데이터를 인지하는 우선순위가 가장 낮은 상황이다. 따라서 감지된 데이터의 거리 기준으로 처리 순서를 정할 경우 가장 위험한 지역부터 차례대로 감지 이벤트가 처리 될 것이다.

제안하는 방법은 warning과 alert 으로 총 2가지 단계로 위험도를 나누었다. Warning의 경우 원거리 LiDAR만 감지 가능한 거리의 경우에서 물체가 감지 되었을때 warning으로 처리한다. alert는 MEMs LiDAR를 해당 위치로 회전을 시키고 이때의 시간 정보를 기록 한다. MEMs LiDAR가 회전하여 사물의 방향을 보고 사물을 감지하였을 때 alert를 발생시키며 이것은 사용자에게 Fig. 5와 같이 형광색(하늘색) 팝업 윈도우 등의 방법으로 적극적으로 알려주는 방법을 취한다. 이 방법은 2가지 센서 모두를 활용하여 더 적극적으로 사용자에게 위험에 대해 알릴 수 있으며, 하나의 센서에서 발견을 못했을 경우 잡음이나 없는 사물에 대한 오감지라 판단한다. 이것은 기존의 시스템[11]에서 잘못된 탐지에 대한 불편함을 효과적으로 해결할 수 있다.

경계 시스템에서는 위험을 감지하고 사용자에게 위험 상황에 대한 것을 알려준다. 사용자는 위험 이벤트를 통해서 현장을 살펴볼 수 있는 관리 시스템을 필요로 한다. 이 시스템은 MEMs LiDAR를 제어하여 위험이 감지된 구역을 모니터링 수 있어야 하며, 센서들을 제어하거나 센서들이 출력하는 데이터를 모니터링 할 수 있어야 한다. 이러한 관리 시스템은 Fig. 5와 같이 통합된 월드 좌표계를 통한 3D Viewer, 카메라 정보를 확인할 수 있는 카메라 Viewer, 센서들의 세팅 값을 설정하고 조정할 수 있는 환경설정 페이지를 제공한다.

시각화하기 위해서 기존 시스템에서는 장치에 접근하여 센서로부터 받거나 중간에 bypass 서비스를 통하여 데이터를 클라우드 내부에서 전달받는 방법을 주로 사용하였다. 이것은 delay가 발생할 수는 있지만 필요할 때 안정적으로 내부에서 데이터를 Content Delivery Network(CDN)[22]을 이용하여 형태로 안전하게 받을 수 있다는 장점이 있다. 이것은 중요 시설과 거리가 멀리 떨어진 외부에서 모니터링을 하던 데이터 전송에 효과적이다.

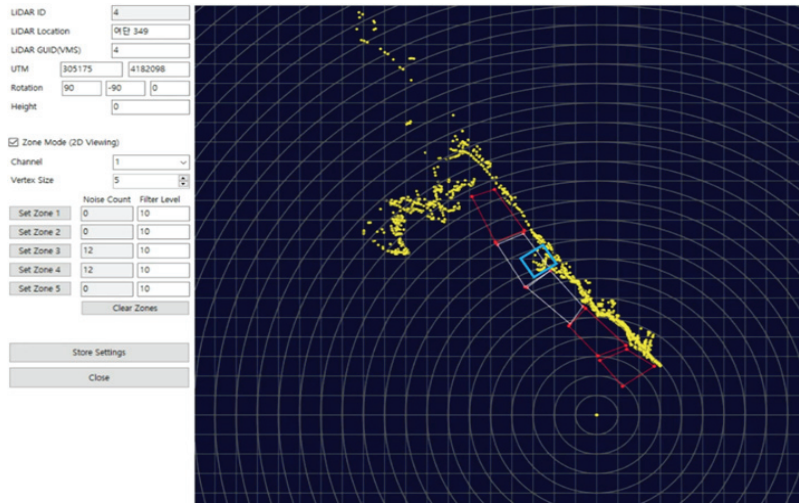


Fig. 5. User Interface and 3D View of Our Method

제안하는 방법은 클라우드 환경을 이용하는 것뿐만이 아니라 웹 클라이언트 어플리케이션에서 데이터를 재사용하여 기존에 병목이었던 실시간 렌더링에 관련된 부분들을 해결할 수 있다. LiDAR, MEMs LiDAR와 같이 실시간에 수집하는 데이터들이 많은 장치는 중앙 서버에서 정적인 한 데이터를 가지고 있을 경우에 전송할 데이터량을 효과적으로 줄일 수 있다. 이것은 데이터를 시각화해야 하는 client 응용프로그램에도 해당된다.

본 시스템의 client 응용프로그램은 웹서비스 형태로 제공된다. 이것은 클라우드 컴퓨팅에서 SaaS(Software as a Service)의 형태로 모니터링 서비스를 제공하여 사용자들이 간단하게 모니터링을 할 수 있도록 하였다. 이 시스템은 클라우드 서비스들을 통하여 엣지 서버와 소통을 하면서 장치들에 대한 수치들을 직접 컨트롤 하거나 제어할 수 있고 클라우드 서비스를 통하여 실시간으로 엣지 서버에서 처리하고 있는 데이터를 모니터링 할 수 있다. Fig. 5의 오른쪽 영역은 3D view로 LiDAR에서 입력으로 받는 데이터에 대해 하나의 공간에 시각화를 한다. 이곳에 사물이 감출되면 popup으로 경고를 하며, MEMs LiDAR 카메라에 대한 view는 창에서 모니터링 할 수 있다.

4. 실험 결과

실험은 사람의 왕래가 적은 공터에서 사람의 접근에 대해 감지하는지를 측정하도록 설계하였다. 25m의 감지 range를 갖는 4 ch mechanical scanning LiDAR를 사용하였다. 두 센서 모두 UDP Protocol로 데이터를 전송한다. MEMs LiDAR는 360° 확대할 수 있는 원격 회전형 모터와 MEMs 방식인 Intel Realsense I515모델을 사용하였다. 원거리 LiDAR 센서는 5대, Realsense 는 1대 사용하였다. 엣지 컴퓨터의 경우 하나의 장비에 전부 연결하여 사용하였다.

클라우드는 아마존 클라우드를 사용하였으며, 엣지 컴퓨터로 Latte Panda alpha 모델을 사용하였다. 엣지 서버는

LiDAR 센서에서 바로 데이터를 받아 데이터를 압축하고 변환하는 서비스를 제공하며, 클라우드 서버에 시스템에서 움직이는 사물이 감지되면 해당 물체의 포인트 데이터들과 이벤트를 제공한다.

이러한 시스템을 야외 1열로 된 철조망에 설치하였으며, 감지 영역에 사람이 움직이며 감지 여부를 파악하였다. 카메라 객체 인식은 YOLO[21]라고 불리는 공개 소스 프레임워크를 사용하여 움직이는 물체를 감지하였다.

우리는 2가지 실험을 통하여 제안하는 방법의 효율성을 입증할 것이다. 첫 번째 실험은 엣지 서버를 사용할 때와 사용하지 않으면 따른 전송 데이터량을 비교한다. 실험은 5명의 인원이 10~20m 지점에서 자유롭게 돌아다니는 환경에서 수행되었다. Fig. 6은 20초 동안 클라우드의 메인 서버로 전송된 데이터량을 엣지 컴퓨터를 사용한 경우와 사용하지 않은 기존의 클라우드 기반 방법을 사용한 경우를 비교하여 측정하였다. 압축을 수행하지 않은 case는 20초 동안 초당 평균 14,012,779 byte를 전송하였으며, 엣지 컴퓨터로 압축을 수행한 데이터는 초당 평균 122,749 byte를 전송한 것으로 측정되었다. 제안한 엣지 컴퓨팅 서비스를 통하여 약 99% 데이터가 절약되었다. 감지율은 압축을 수행하지 않은 경우에서 실제로 UDP로 인하여 유실되는 데이터가 발생하였다. 총 LiDAR 데이터에서 발생하였으며 총 100번의 스캔된 포인트 클라우드 중 11회 1명의 사람을 탐지를 하지 못한 경우 발생하였다. 엣지 컴퓨터에서 연산한 경우에는 모든 데이터가 정확하게 전부 도착하였으며, 오탐지는 발생하지 않았다. 이것은 압축하지 않은 경우에 누적된 데이터량으로 인하여 네트워크 부하가 늘어남에 따라 후반부에 데이터그램의 손실이 발생하여 생긴 문제였다.

두번째 실험은 제안하는 데이터 압축기법을 토대로 한 client PC에서의 화면을 3D로 시각화하는데 소요되는 속도를 비교하여 제안하는 엣지 컴퓨팅 방법의 효율성을 입증한다. Fig. 7은 압축을 하지 않고 LiDAR로부터 raw 데이터를 매번 전송받아 렌더링 한 결과와 제안하는 방법을 이용하여 압축하

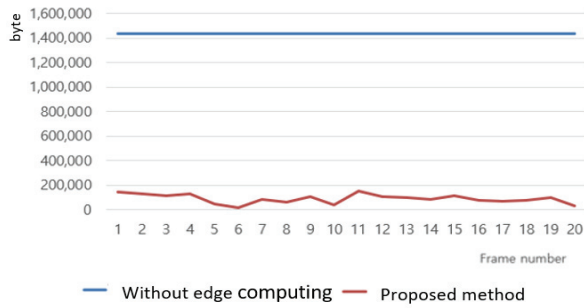


Fig. 6. The Red Line Shows the Transmitted Data of Guard System Without the Edge Computer, and the Blue Line Shows the Transmitted Data by Using the Proposed Edge Computing Method.

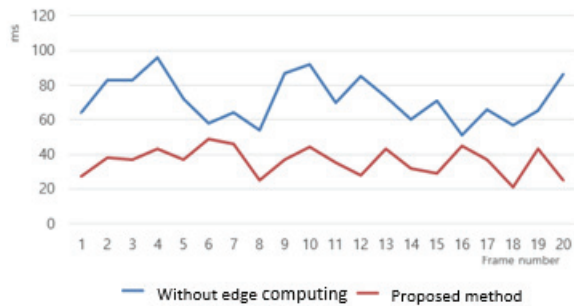


Fig. 7. Experimental Results of Rendering Speed of 3D View. The Red Line Shows the Result of the Method Without Using the Edge Computer, and the Blue Line Shows the Proposed Edge Computing Method

는 방법을 적용하여 렌더링 한 결과를 비교하였다. 가속화가 없는 경우는 실험 환경 특성상 private 클라우드로 구축 되어 있는 관계로 원격에서 직접 센서에 접속할 수 없었기 때문에 센서에서 client가 직접 데이터를 수집하도록 설계하였다. 그럼에도 불구하고 제안하는 방법은 화면 버퍼를 재사용하는 방법으로 평균 2배 이상으로 빠른 속도로 결과를 시각화 하였다.

이 두 실험으로 제안하는 방법은 엣지 컴퓨터를 활용하여 성능적인 측면에서도 데이터그램 유실률과 네트워크 부하를 줄여 더 정확하고 빠르게 시스템을 운영할 수 있었다. 또한 모니터링 시스템에서도 시각화에 필요한 데이터량을 효율적으로 줄여 client PC의 부담을 줄이고 네트워크 병목을 줄일 수 있었다.

5. 결 론

최근 중요시설의 경계 시스템이나 스마트시티의 감시 체계 등에서는 LiDAR 센서로 움직이는 사물을 실시간에 감지하여 무단 침입으로 인한 사고를 사전에 감지하는 것을 목표로 하고 있다. 제안하는 가속화 방법은 다수의 LiDAR 센서를 추가로 활용하여 이러한 사전 감지를 하는 방법이다. LiDAR 센서는 빠르고 정교하지만 스캔하는 포인트 클라우드 데이터 양이 많고 이러한 데이터는 클라우드 서버에 전부 전송이 되

어 처리되어야 했기 때문에 네트워크 과부하와 컴퓨팅 오버헤드가 주요 문제점이었다. 제안하는 방법은 엣지 컴퓨터를 활용하여 LiDAR 센서에서 스캔하는 데이터를 효율적으로 줄이고, 중앙 서버에서 한 번에 처리할 수 있는 좌표계로 변환하여 네트워크와 중앙 서버의 부하를 줄였다. 또한 client PC 또한 직접 LiDAR 센서에 접속해서 raw 데이터를 보는 것이 아닌 압축된 데이터를 실시간으로 복원하여 client PC에서 렌더링을 수행할 수 있도록 하여 서버의 과부하뿐만 아니라 client PC의 과부하를 막을 수 있었다.

References

- [1] H. W. Yoo et al., "MEMS-based lidar for autonomous driving," *Elektrotechnik und Informationstechnik*, Vol.135, No.6, pp.408-415, 2018. doi: 10.1007/s00502-018-0635-2.
- [2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.13, No.7, pp.560-576, 2003.
- [3] J. Yan, W. Pu, S. Zhou, H. Liu, and Z. Bao, "Collaborative detection and power allocation framework for target tracking in multiple radar system," *Information Fusion*, Vol.55, pp.173-183, 2020.
- [4] F. Nardo, D. Peressoni, P. Testolina, M. Giordani, and A. Zanella, "Point cloud compression for efficient data broadcasting: A performance comparison," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp.2732-2737, 2022. doi: 10.1109/WCNC51071.2022.9771764.
- [5] E. Yavari, H. Jou, V. Lubecke, and O. Boric-Lubecke, "Doppler radar sensor for occupancy monitoring," in *2013 IEEE Topical Conference on Power Amplifiers for Wireless and Radio Applications*, pp.145-147, 2013. doi: 10.1109/PAWR.2013.6490217.
- [6] G. L. Charvat and L. C. Kempel, "Synthetic aperture radar imaging using a unique approach to frequency-modulated continuous-wave radar design," *IEEE Antennas and Propagation Magazine*, Vol.48, No.1, pp.171-177, 2006. doi: 10.1109/MAP.2006.1645606.
- [7] M. Zhao, A. Mammeri, and A. Boukerche, "Distance measurement system for smart vehicles," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pp.1-5, 2015. doi: 10.1109/NTMS.2015.7266486.
- [8] M. Hongchao and Z. Wang, "Distributed data organization and parallel data retrieval methods for huge laser scanner point clouds," *Computers & Geosciences*, Vol.37, No.2, pp.193-201, 2011. doi: 10.1016/j.cageo.2010.05.017.

[9] D. Oryspayev, R. Sugumaran, J. DeGroot, and P. Gray, "LiDAR data reduction using vertex decimation and processing with GPGPU and multicore CPU technology," *Computers & Geosciences*, Vol.43, pp.118-125, 2012. doi: 10.1016/j.cageo.2011.09.013.

[10] J. Huang, S. Ran, W. Wei, and Q. Yu, "Digital integration of LiDAR system implemented in a Low-Cost FPGA," *Symmetry*, Vol.14, No.6, pp.1-14, 2022. doi: 10.3390/sym14061256.

[11] E. S. Lee, E. K. Jo, and B. S. Shin, "A moving object detection method using scan-type LiDAR sensor," *Proceedings of the Annual Spring Conference of Korea Information Processing Society Conference (KIPS)*, Vol.29, No.1, pp.88-89, 2022.

[12] W. Dingkan, W. Connor, and X. Huikai, "MEMS mirrors for LiDAR: A review," *Micromachines*, Vol.11, No.5, pp.456, 2020.

[13] C. V. Poulton et al., "Coherent solid-state LIDAR with silicon photonic optical phased arrays," *Optics letters*, Vol.42, No.20, pp.4091-4094, 2017. doi: 10.1364/OL.42.004091.

[14] M. E. Warren, "Automotive LIDAR technology," in *2019 Symposium on VLSI Circuits*, pp.C254-C255, 2019. doi: 10.23919/VLSIC.2019.8777993.

[15] B. Lv et al., "LiDAR-Enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities," *IEEE Access*, Vol.7, pp.79895-79907, 2019. doi: 10.1109/ACCESS.2019.2923421.

[16] W. Zhang and D. Yang, "Lidar-based fast 3D stockpile modeling," in *2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, pp.703-707, 2019. doi: 10.1109/ICICAS48597.2019.00152.

[17] E. Felemban, "Advanced border intrusion detection and surveillance using wireless sensor network technology," Vol.2013. doi: 10.4236/ijcns.2013.65028.

[18] F. Chiariotti, A. A. Deshpande, M. Giordani, K. Antonakoglou, T. Mahmoodi, and A. Zanella, "QUIC-EST: A QUIC-Enabled scheduling and transmission scheme to maximize Vol with correlated data flows," *IEEE Communications Magazine*, Vol.59, No.4, pp.30-36, 2021. doi: 10.1109/MCOM.001.2000876.

[19] E. Kim and S.-Y. Park, "Extrinsic calibration between camera and LiDAR sensors by matching multiple 3D planes," *Sensors*, Vol.20, No.1, pp.1-17, 2020. doi: 10.3390/s20010052.

[20] A. Singh, A. Kamireddypalli, V. Gandhi, and K. M. Krishna, "LiDAR guided small obstacle segmentation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.8513-8520, 2020. doi: 10.1109/IROS45743.2020.9341465.

[21] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers." In *2018 IEEE International Conference on Big Data (Big Data)*, pp.2503-2510, 2018.

[22] T. Ojanperä, J. Mäkelä, O. Mämmelä, M. Majanen, and O. Martikainen, "Use cases and communications architecture for 5G-enabled road safety services," in *2018 European Conference on Networks and Communications (EuCNC)*, pp.335-340, 2018. doi: 10.1109/EuCNC.2018.8443193.

조 은 경



https://orcid.org/0000-0002-9551-6813
 e-mail : misty25678@naver.com
 2006년 상명대학교 법학과(학사)
 2021년~현 재 인하대학교
 전기컴퓨터공학과 석사과정
 관심분야 : Artificial Intelligence, 3D
 Visualization, User Interface,
 Computer Graphics

이 은 석



https://orcid.org/0000-0002-8837-8491
 e-mail : elflee77@yuhan.ac.kr
 2003년 인하대학교 컴퓨터공학부(학사)
 2011년 인하대학교 컴퓨터정보공학과(석사)
 2011년 인하대학교 컴퓨터정보공학과(박사)
 2020년~현 재 유한대학교 VR게임·앱학과 조교수
 관심분야 : Computer Graphics, Realtime Rendering, Game
 Engine, Terrain Rendering GPGPU

신 병 석



https://orcid.org/0000-0001-7742-4846
 e-mail : bsshin@inha.ac.kr
 2003년 서울대학교 컴퓨터공학부(학사)
 2011년 서울대학교 컴퓨터공학과(석사)
 2011년 서울대학교 컴퓨터공학과(박사)
 2010년~현 재 인하대학교 컴퓨터공학과
 교수
 관심분야 : Computer Graphics, Realtime Rendering,
 Volume Rendering, Artificial Intelligence