

비정형의 건설환경 매핑을 위한 레이저 반사광 강도와 주변광을 활용한 향상된 라이다-관성 슬램

Intensity and Ambient Enhanced Lidar-Inertial SLAM for Unstructured Construction Environment

정민우¹·정상우¹·장혜수²·김아영[†]

Minwoo Jung¹, Sangwoo Jung¹, Hyesu Jang², Ayoung Kim[†]

Abstract: Construction monitoring is one of the key modules in smart construction. Unlike structured urban environment, construction site mapping is challenging due to the characteristics of an unstructured environment. For example, irregular feature points and matching prohibit creating a map for management. To tackle this issue, we propose a system for data acquisition in unstructured environment and a framework for Intensity and Ambient Enhanced Lidar Inertial Odometry via Smoothing and Mapping, IA-LIO-SAM, that achieves highly accurate robot trajectories and mapping. IA-LIO-SAM utilizes a factor graph same as Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping (LIO-SAM). Enhancing the existing LIO-SAM, IA-LIO-SAM leverages point's intensity and ambient value to remove unnecessary feature points. These additional values also perform as a new factor of the K-Nearest Neighbor algorithm (KNN), allowing accurate comparisons between stored points and scanned points. The performance was verified in three different environments and compared with LIO-SAM.

Keywords: Intensity, Ambient, IA-LIO-SAM, LiDAR Odometry, SLAM

1. 서론

시공 및 준공 단계에서 현장에서의 관리를 통해 작업이 적절하게 진행되고 있는지 파악하는 것은 무척 중요한 일이다. 특히 대규모의 시공 및 준공에서 초기 발생한 오차는 후에 더욱 큰 오차를 유발할 수 있으며, 심각한 경우 재시공까지 고려해야 하기 때문에 현장에서 지속적인 관리는 필수적이다. 하지만 일부 비정형 환경은 사람뿐만 아니라 일반적인 하드웨어로 접근하기도 힘들며, 대규모 시공 및 준공에서 정밀한 관리는 오랜 시간을 소요하게 되어 진행도를 늦출 수 있다는 문제

가 존재한다. 따라서 사람 대신 현장에 접근할 수 있는 시스템을 고안하고, 해당 시스템으로 자료를 취득하여 시공 및 준공 과정을 관리하는 것이 효율적일 것이다. 취득된 자료는 동시적 위치추적 및 지도작성(SLAM, Simultaneous Localization And Mapping)에 사용되어 현장의 지도를 만들 수 있으며, 시공 현장 맵핑을 위해 라이다 SLAM이 사용될 수 있다. 라이다 SLAM 중 Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping (LIO-SAM)^[1]은 정확도가 높으며, 다른 라이다 SLAM에서 맵핑이 불가능한 지역도 맵핑을 성공적으로 한다는 점에서 매우 효과적인 방법이다. 하지만 비정형 환경에서는 관찰 대상만이 아닌 덩불이나 고르지 못한 지면에서도 다수의 특징점이 검출되기 쉽고, 특징점의 위치만을 활용하여 매칭하는 방식은 왜곡된 결과를 만들어 낼 수 있다는 단점이 있다. 더욱이 특징점이 많이 존재하지 않는 비정형 환경에서 잘못된 매칭은 결과에 치명적인 영향을 미칠 수 있다.

본 논문에서는 이러한 하드웨어와 소프트웨어 문제를 극복하고 시공 현장 모니터링에 적합한 시스템을 제안한다. 첫째, 비정형 환경에 적합한 시스템으로 [Fig. 1]에서 볼 수 있는 모바일

Received : Apr. 27. 2021; Revised : Jun. 2. 2021; Accepted : Jun. 6. 2021

※ This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (National Research for Smart Construction Technology: Grant 20SMIP-A158708-01)

1. Master Student, Dept. of Civil and Environmental Engineering, KAIST, Daejeon, Korea (jungminwo, dan0130@kaist.ac.kr)

2. Ph.D. Student, Dept. of Civil and Environmental Engineering, KAIST, Daejeon, Korea (iriter@kaist.ac.kr)

† Associate Professor, Corresponding author: Dept. of Civil and Environmental Engineering, KAIST, Daejeon, Korea (ayoungk@kaist.ac.kr)



[Fig. 1] Husky Unmanned Ground Vehicle. Using Robot Operating System (ROS), it can move at a speed up to 1 m/s and climb a hill with an angular degree of 45°

플랫폼을 선택하였으며, 추가로 라이다 센서, 관성 측정 장치(IMU, Inertial Measurement Unit), GPS (Global Position System)를 설치하여 시스템 주변의 자료를 취득할 수 있게 만들었다. 해당 시스템을 통해 취득된 자료는 SLAM을 통해 관찰 대상의 점군 지도를 만들어 현장의 관리 및 관찰을 용이하게 해줄 수 있다.

알고리즘으로는 Intensity and Ambient Enhanced Lidar Inertial Odometry via Smoothing and Mapping (IA-LIO-SAM)을 제안하여 비정형 환경에서 맵핑 정확도를 개선한다. 본 연구에서 활용한 라이다 센서, OS1-128에서 스캔된 각 점은 위치뿐만 아니라 intensity와 ambient 값 등을 가지고 있는데, 각 프레임에서의 특징점을 매칭할 때 해당 값을 추가로 활용할 수 있다. 또한 덤불이나 수풀 등의 경우 ambient 값이 고르게 퍼져 있는 특징이 있는데, 임계값을 설정하여 특정 점들을 무시함으로써 맵핑의 성능을 올릴 수 있다.

최종적으로, 아래의 항목들은 본 논문에서 제시하고자 하는 논점이다.

- 비정형 환경에서 적합한 시스템을 제안하여, 무인으로 데이터를 취득할 수 있다.
- Ambient 임계값을 설정하여 부정확한 매칭을 무시할 수 있다.
- Intensity와 Ambient 값을 추가적으로 활용하여 비슷한 위치에 있는 특징점을 구별함으로써 비정형 환경에서의 정확도를 개선한다.
- 제안한 시스템은 실제 공사현장을 포함한 다양한 환경에서 검증되었다.

2. 선행 연구 조사

2.1 점군 스캔 일치

SLAM은 자신의 위치추적(Tracking)과 주변의 지도작성(Mapping)을 동시에 수행하며 진행된다. 라이다 SLAM의 경우, Iterative Closet Point (ICP)^[2], Generalized Iterative Closet

Point (G-ICP)^[3], Trimmed Iterative Closet Point (TrICP)^[4] 등을 이용하여 인접한 프레임에 점군의 상대적 위치를 비교하며 자신의 위치를 추적한다. 전체 점군을 비교하여 높은 정확도를 취득할 수 있지만, 시간 복잡도가 크다는 단점이 있기에 실시간으로 SLAM을 진행하는 것은 불가능하다. 따라서 Visual SLAM에서 Rectangle Feature^[5], 3-D straight lines^[6], Scale Invariant Feature Transform (SIFT)^[7], Speed Up Robust Features (SURF)^[8], Oriented FAST and Rotated BRIEF (ORB)^[9] 등의 특징을 검출하여 이용한 것과 같이, 주변 점군에서 점, 선, 면을 검출하여 이용하게 된다. 하지만 점군의 경우, 외부의 진동이나 회전 등으로 왜곡이 쉽게 생길 수 있기 때문에 검출된 특징들도 왜곡이 존재하게 된다. 오차가 지속적으로 쌓일 경우 결과에 큰 영향을 미칠 수 있기 때문에 왜곡을 보정할 수 있는 다른 센서의 존재가 필요하다.

2.2 라이다 SLAM

라이다 SLAM에서는 보통 IMU, GPS를 사용하여 왜곡을 완화한다. Lidar Odometry And Mapping (LOAM)^[10]에서는 면과 가장자리를 활용한 점을 특징으로 활용하며, 특징을 검출하기 전 IMU를 이용하여 점군을 보정해준다. 하지만, 특징을 검출하는 데 오랜 시간이 걸린다는 단점이 존재했고, 이를 개선한 것이 Lightweight and Ground-Optimized Lidar Odometry and Mapping (LeGO-LOAM)^[11]이다. 전체적인 방법은 LOAM과 동일하나 Levenberg-Marquardt algorithm를 활용해 계산 시간을 줄이고, Loop Closing을 추가하여 SLAM의 안정성을 높였다. 두 방법 모두 IMU를 활용하여 왜곡을 없애고 높은 정확도로 움직임을 추정할 수 있게 해주었지만, 최적화 단계에서는 IMU가 쓰이지 않고 있다는 특징이 존재한다.

이에 반해 Robocentric Lidar-Inertial State Estimator (R-LINS)^[12], Tightly Coupled Lidar Inertial Odometry and Mapping (LIOM)^[13], Tightly Coupled Lidar-Inertial-Ranging Odometry (LIRO)^[14], LIO-SAM 등에서는 IMU를 더욱 적극적으로 사용하여 정확도를 크게 향상한다. IMU Preintegration^[15]을 사용하여 두 프레임 사이의 IMU 값들을 하나의 상대적 위치 변화로 계산해내며, 계산의 반복 및 다수의 점군과 같은 불필요한 변수 등을 줄이면서 계산 복잡도를 해결하는 데 도움을 주었다. 이 중, LIO-SAM은 그래프를 이용한 라이다 SLAM으로 키프레임이 노드를 형성하며, 노드는 IMU preintegration factors, LiDAR odometry factors, GPS factors, loop closure factors로 연결되어 있다. 해당 요소들을 통해 노드 사이의 상대적 위치 변화를 추정하며, GPS 값을 통해 누적되는 오차를 막아주어 정확한 결과를 만든다. LIO-SAM의 LiDAR odometry factor는 크게 세가지 과정으로 나눌 수 있는데, 스캔된 점군을 해당 위치

에 배치한 후 특징점을 뽑고 최적화 과정을 거친다. 하지만 특징점은 매번 각 프레임의 같은 위치에서 검출되는 것이 아니기 때문에, 단순히 위치만을 이용하여 매칭한다면 주변 사물에서 검출된 점과 잘못된 일치가 발생하여 오차를 발생시킬 수 있다. 특히, 비정형 환경에서의 땅이나 덩불 등에서는 불규칙한 특징점이 검출될 수 있으므로 이러한 특징점이 매칭되는 것을 사전에 방지하는 것이 필요하다.

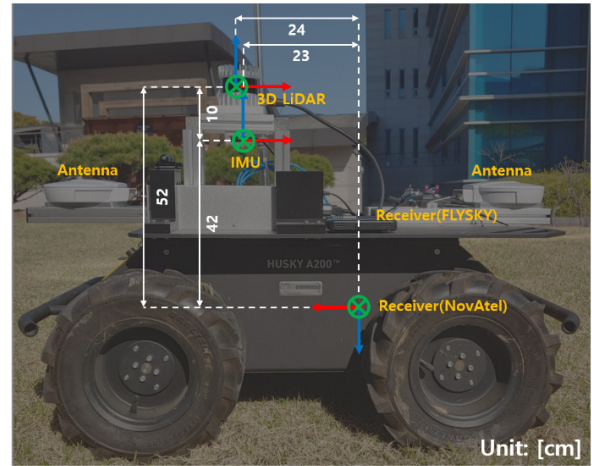
2.3 Intensity를 활용한 예시

라이다 점군의 intensity는 다양한 방면에서 사용되었다. Intensity Scan Context (ISC)^[16]는 라이다 점군의 intensity를 활용하여 디스크립터를 생성하고, 이를 바탕으로 방문했던 장소를 인식한다. [17]에서는 라이다로 스캔한 점을 이미지에 투영하고, 점의 intensity 값을 대응하는 픽셀의 값으로 넣는다. 그 후, 이미지에서 ORB 특징점을 검출한 뒤, 이를 이용해 기존에 방문했던 장소를 인식한다.

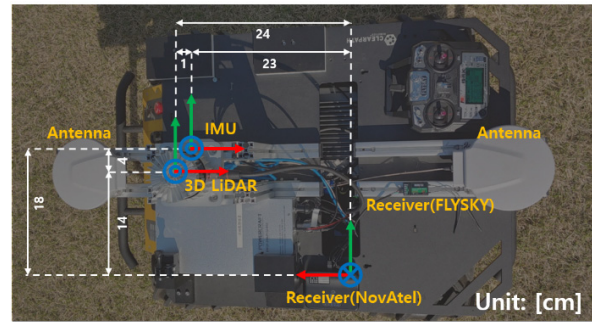
Intensity는 장소 인식뿐만 아니라 두 스캔 사이의 위치 변환 행렬을 찾는 데도 이용된다. [17]에서와 마찬가지로, [18]에서는 이미지의 특징점을 지속적으로 추적하여 visual odometry를 얻는 방법을 소개한다. [19]은 본 논문의 방법과 유사한 것으로, Intensity Enhanced LiDAR Odometry and Mapping (I-LOAM)을 소개한다. 하지만, [19]에서는 intensity에 가중치를 두고 특징점 매칭이 일어난 반면, 본 논문에서는 ambient를 추가적으로 활용하였다. 또한, 모든 특징점을 매칭하는 I-LOAM과 다르게, 매칭에 불필요한 점들은 사전에 제외된다는 점에서 차이가 있다.

3. 시스템 소개

채택한 모바일 플랫폼은 [Fig. 2]에 보이는 것과 같이 라이다, IMU, 2개의 안테나와 해당 신호를 수신 받는 GNSS+INS 수신기, 조종기 수신기로 구성되어 있다. 라이다는 OS1-128을 사용했으며 스캔된 각 점들의 위치뿐만 아니라 intensity, ring, range, ambient, time 등의 정보를 제공한다. 또한 OS1의 경우 OS2에 비해 넓은 화각을 가지고 있어서, 경사가 급한 부분에서 유리하게 작용할 수 있다. 플랫폼 작동을 위한 송신기는



(a) Side view of hardware



(b) Top view of hardware

[Fig. 2] Hardware sensor configuration. (a) side view, (b) top view of entire sensor system with coordinate information. The coordinate system of each sensor is marked with red (x), green (y), and blue (z), and the distance of each sensor is also displayed. A unit in the images is centimeters

FlySky FS-i6 2.4G 6CH PPM RC Transmitter를 사용하였다. 센서들의 정보는 [Fig. 2]와 [Table 1]에 요약되어 있다.

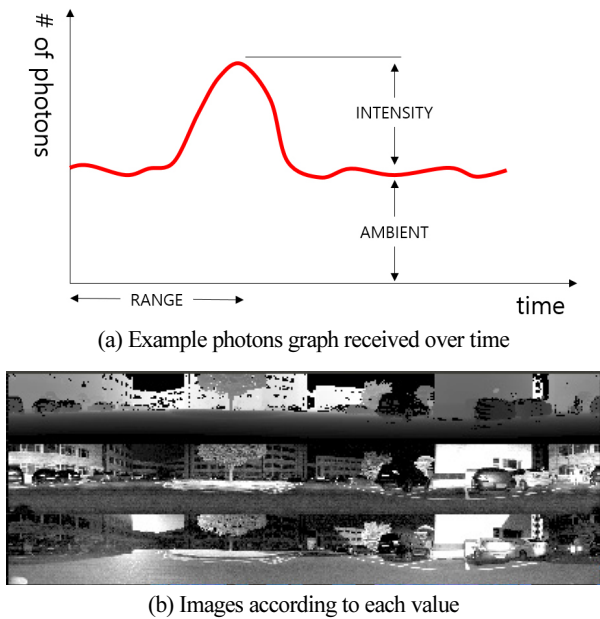
4. 연구 방법

공사현장과 같은 비정형 환경에서는 검출되는 특징점이 적고, 불규칙하다는 특징이 있다. 따라서 기존 라이다 SLAM을 활용한다면 LiDAR odometry의 값이 부정확하게 나오고, 이는 지도 생성에 큰 어려움을 준다. 즉, 추가적인 정보를 사용하여

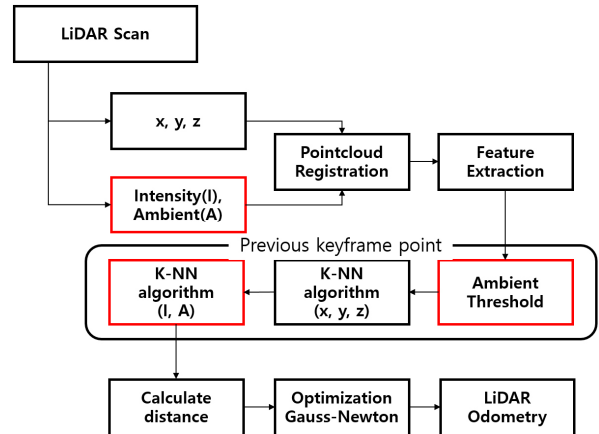
[Table 1] Sensor used in the sensor system

Type	Manufacture	Model	Description	N	Frequency (Hz)	Specification
3D LiDAR	OUSTER	OS1-128	128 CH LiDAR (45° FOV)	1	10	120 m (range)
IMU	LORD	3DM-GX5-45	Consumer-level AHRS	1	1000	±2° degree for dynamic
Antenna	NovAtel	GNSS-501	Dual antenna	2	-	Multi signal received
Receiver	NovAtel	SPAN CPT7	GNSS+INS receiver	1	20 (INS : 200)	1cm (RTK Accuracy)
	FLYSKY	FS-IA6B	Joystick Receiver	1	-	600 m (range)

개선해야 할 필요가 있다. 이를 위해 앞서 언급한 OS1-128을 사용하였으며, 해당 라이다는 각 점마다 intensity, ambient, range 등의 값을 추가적으로 제공한다. Intensity의 경우, 라이다에서 송신된 입자가 점에 반사되어 다시 수신된 입자의 수를 활용하여 결정되며, 가장 많은 입자가 수신된 시간을 이용해 range를 구한다. 즉, 반사도가 높은 물체에서는 높은 intensity 값을 가지게 된다. 하지만 라이다가 수신하는 입자 중에서는 햇빛과 같이 주변 환경으로부터 수신된 입자도 존재한다. 이때 라이다가 주변 환경으로부터 수신한 입자의 수를 활용하여 결정되는 것이 ambient이다. 즉, 라이다가 밝은 환경에 위치한다면 ambient의 값은 높게 측정된다. 정리하면, 라이다는 주변 환경으로부터 입자를 수신하는데, 수신된 입자 수를 통해 ambient 값이 계산되며 해당 값은 시간에 따라 일정한 편이다. 하지만 라이다에서 송신한 입자가 점에 반사되어 다시 라이다에 수신되었을 때, 라이다에서 수신한 총 입자 수는 증가할 것이다. 이때 증가한 입자 수가 intensity가 되며, 라이다에서 입자를 송신한 시각과 입자 수가 최대가 되는 시각의 차이를 이용해 측정된 거리가 range가 된다. Intensity와 ambient 값의 개요는 [Fig. 3(a)]에 나와있다. 또한, [Fig. 3(b)]는 점의 값들을 활용하여 만든 이미지를 보여주는데, intensity 차이가 크게 없는 곳에서 ambient 차이가 나며 그 반대도 있는 것을 확인할 수 있다. 즉, 특징점 매칭을 수행할 때 intensity와 ambient 값도 포함시킨다면 상호보완적으로 도움이 됨을 알 수 있다.



[Fig. 3] (a) Example for channels of OS1-128. Based on the amount of photons received, the intensity, ambient, and range are determined. (b) Image created based on the values of points. From above, the range, intensity, and ambient images are shown



[Fig. 4] Diagram of proposed method. The black box is proposed by LIO-SAM, and the red box is newly included. Ambient was used to set thresholds and for K-Nearest Neighbor (K-NN) algorithm with intensity

본 연구에서는 앞서 설명된 intensity와 ambient 값을 활용한 IA-LIO-SAM을 제안한다. Intensity 값의 경우 각도나 거리에 따라 값이 달라져, 전체 경로에서 보았을 때는 일관성을 유지하기 힘들다. 일관성이 유지되지 않으면, 특징점 매칭 시 false-positive matching이 생길 수 있으며 이는 큰 오차를 유발할 수 있다. 하지만 LIO-SAM의 경우 인접한 시간대와 프레임의 점들만을 비교하기 때문에 intensity 값의 일관성이 유지된다고 생각할 수 있으며, 이는 ambient 값에도 적용할 수 있다. 따라서 intensity, ambient 값을 추가로 사용하면 위치만을 사용하여 매칭한 것보다 더 효과적인 결과를 낼 수 있을 것이다. IA-LIO-SAM의 전체적인 과정은 [Fig. 4]에서 볼 수 있다.

4.1 키프레임에서의 특징점 검출

먼저, 새로운 라이다 스캔이 들어왔을 때 각 점의 x, y, z , intensity, ambient를 저장하고 특징점을 검출하게 된다. 특징점은 해당 점 주변 지역의 roughness를 계산해서 가장자리 특징점과 면 특징점으로 구분할 수 있는데, 큰 값을 가질 경우 가장자리 특징점, 반대로 작은 값을 가질 경우 면 특징점으로 구분한다. 자세한 과정은 [8]에서 찾을 수 있다. i 번째 스캔에서의 각 특징점 그룹을 F_i^e, F_i^p 해당 프레임을 $F_i = \{F_i^e, F_i^p\}$ 로 표현할 것이며, F_i 는 라이다 좌표계인 L 에 위치해 있다. 하지만 모든 프레임에 대해 매칭을 진행한다면 많은 계산 시간을 소모하므로, 사용자 조건에 만족하는 프레임을 키프레임으로 뽑고, 키프레임 사이의 프레임은 무시할 것이다. 만약 j 번째 프레임이 i 번째 키프레임으로 선택되었다면 $K_i = F_j$ 이며, K_i 는 현재 로봇의 상태 x_i 를 대표한다. IA-LIO-SAM에서는 LIO-SAM과 같이 선택 조건을 1 m와 10° 로 설정하였다.

4.2 이전 키프레임에서의 특징점 매칭 후보 형성

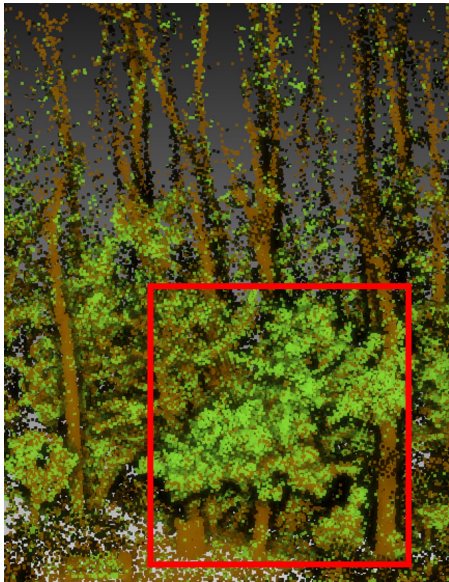
정확한 매칭을 하기 위해, 두개의 연속된 키프레임만을 비교하지 않고 정해진, n 개만큼의 인접한 키프레임을 추가로 비교하게 된다. n 개의 키프레임은 모두 L 좌표계에 위치해 있으므로, $\{K_{i-n}, \dots, K_i\}$ 에 변환 행렬 $\{T_{i-n}, \dots, T_i\}$ 을 적용시켜 W , 월드 좌표계로 바꿔주어야 한다. 이후 키프레임을 합하면 새로운 맵인 M_i 이 형성되며, M_i 는 다음과 같은 두개의 개별적인 맵을 가진다.

$$\begin{aligned} M_i &= \{M_i^e, M_i^p\} \\ M_i^e &= K_i'^e \cup K_{i-1}'^e \cup \dots \cup K_{i-n}'^e, \quad K_i'^e = T_i K_i^e \\ M_i^p &= K_i'^p \cup K_{i-1}'^p \cup \dots \cup K_{i-n}'^p, \quad K_i'^p = T_i K_i^p \end{aligned} \quad (1)$$

최종적으로 M_i^e, M_i^p 에 다운 샘플링을 진행하여 중복되는 특징점을 제거한다. IA-LIO-SAM에서는 LIO-SAM과 같이 $n=25$ 를 선택했으며, M_i^e, M_i^p 의 다운 샘플링 해상도는 0.2 m, 0.4 m로 선택하였다.

4.3 현재 키프레임에서의 특징점 매칭 후보 선정

새롭게 얻어진 K_{i+1} 과 4.2에서 얻어진 M_i 의 특징점을 매칭하기 위해서는 K_{i+1}' 로의 변환이 필요하다. 이를 위해 $i+1$



[Fig. 5] Pointcloud scanned from IA-LIO-SAM. It was scanned in the forest and set to $\lambda=750$. When the ambient of the point is greater than λ , it is displayed in green, and when it is smaller than λ , it is displayed in brown. The unnecessary spots disappear and most of the trees remain

시간에서 IMU 값을 얻고, 예측되는 변환 행렬 \hat{T}_{i+1} 을 획득한다. 이후, M_i, K_{i+1}' 의 특징점을 매칭하게 된다. 하지만 \hat{T}_{i+1} 은 IMU만을 통해 예측된 변환 행렬이기 때문에 W 로의 변환이 정확하지 않을 수 있다. 더욱이 덤불이나 수풀 등에서는 특징점이 일관되게 생성되지 않는데, 이 위치의 차이가 더 커져서 잘못된 매칭을 유발시킬 수 있다.

덤불이나 수풀 등의 환경에 위치한 점들은 투과되는 빛의 양이 많기 때문에 주변의 크고 불투명한 사물에 비해 ambient 값이 크다는 특징이 있다. 따라서 ambient 임계값으로 λ 를 설정하고, λ 를 넘는 점들은 매칭 대상에서 제외시킨다. 특정 점들이 제외된 프레임은 $K_{i+1}'(\lambda)$ 라고 표현한다. [Fig. 5]에서는 비정형 환경에서 ambient 값에 따른 점들의 모습을 보여준다. IA-LIO-SAM에서는 λ 를 설정하기 위해, 대상 환경에 주로 분포하는 불투명 물체의 ambient 값을 라이다 스캔에서 먼저 취득하였다. 그 후, 해당 값보다 조금 높게 λ 값을 설정하였다.

4.4 점군 매칭 및 LiDAR odometry 계산

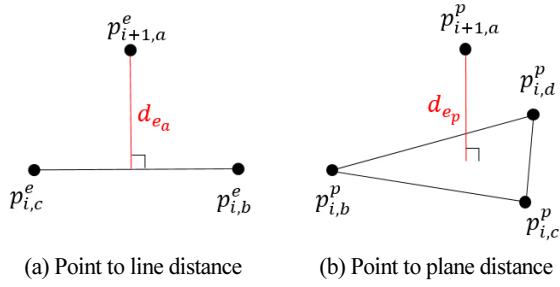
$K_{i+1}'^e(\lambda), K_{i+1}'^p(\lambda)$ 에 속한 각 특징점의 위치를 활용하여, M_i^e, M_i^p 에서 K-Nearest Neighbor (K-NN) 알고리즘을 통해 각 특징점과 대응하는 N_i 개의 점을 찾는다. 그 후에, 찾으려고 했던 특징점의 intensity, ambient 값을 활용하여, N_i 개의 특징점에서 K-NN 알고리즘을 통해 대응하는 N_i 개의 특징점을 찾는다.

M_i^e, M_i^p 에 속한, 각 N_i 개의 특징점은 선과 평면으로 만들어질 수 있고 $K_{i+1}'^e(\lambda), K_{i+1}'^p(\lambda)$ 에 있는 특징점과의 거리는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} d_{e_a} &= \frac{|(p_{i+1,a}^e - p_{i,b}^e) \times (p_{i+1,a}^e - p_{i,c}^e)|}{|p_{i,b}^e - p_{i,c}^e|} \\ d_{p_a} &= \frac{|(p_{i+1,a}^p - p_{i,b}^p) \times (p_{i+1,a}^p - p_{i,d}^p)|}{|(p_{i,b}^p - p_{i,c}^p) \times (p_{i,b}^p - p_{i,d}^p)|} \end{aligned} \quad (2)$$

공통적으로 a, b, c, d 는 점들을 나타내기 위한 표지이다. d_{e_a} 에서 $p_{i+1,a}^e$ 는 $K_{i+1}'^e(\lambda)$ 에 속하는 특징점이며, $p_{i,b}^e$ 와 $p_{i,c}^e$ 는 M_i^e 에서 만들어진 선을 형성하는 특징점이다. 또한, d_{e_p} 에서 $p_{i+1,a}^p$ 는 $K_{i+1}'^p(\lambda)$ 에 속하는 특징점이며, $p_{i,b}^p, p_{i,c}^p, p_{i,d}^p$ 는 M_i^p 에서 만들어진 면을 형성하는 특징점이다. [Fig. 6]는 앞서 설명된 특징점의 위치를 보여준다.

그 후, 가우스-뉴턴 방법은 다음을 최소화하며 최적의 변환 행렬을 찾아낸다.



[Fig. 6] The feature points located at M_i form a line or plane, and calculate the distance from the feature points at $K'_{i+1}(\lambda)$. (a) $p_{i+1,a}^e$ belongs to $K'_{i+1}(\lambda)$. $p_{i,b}^e$ and $p_{i,c}^e$ are feature points belonging to M_i^e . (b) $p_{i+1,a}^p$ belongs to $K'_{i+1}(\lambda)$. $p_{i,b}^p$, $p_{i,c}^p$ and $p_{i,d}^p$ are feature points belonging to M_i^p

$$\min_{T_{i+1}} \left\{ \sum_{p_{i+1,a}^e \in K'_{i+1}(\lambda)} d_{e_a} + \sum_{p_{i+1,a}^p \in K'_{i+1}(\lambda)} d_{p_a} \right\} \quad (3)$$

끝으로 x_i 와 x_{i+1} 사이의 변환, $\Delta T_{i,i+1} = T_i^T T_{i+1}$ 을 찾아낼 수 있고, 이것이 LiDAR odometry factor로 작용한다.

5. 연구 결과

5.1 실험 환경

실험은 정형 환경(Seq 1~3), 비정형 환경(Seq 4~6), 공사 현장(Seq 7), 터널 환경(Seq 8)에서 진행되었다. 비정형 환경과 정형 환경에서는 각각 3가지 데이터를 취득하였다. 각 데이터 별로 LIO-SAM, IA-LIO-SAM을 10번씩 시도하여 결과값을 획득하였다.

RTK-GPS는 불연속적으로 측정된 값은 제외하여 SLAM 결과의 검증 시 기준 값으로 활용하였다. z축 방향의 실측 값은 정확도가 낮은 편이기 때문에 검증 시 x축과 y축 방향에 대한 Root-Mean-Square-Error (RMSE)만 고려하였다. 하지만 오르막이나 내리막이 크게 존재하는 Seq 4~6의 경우 이를 포함시켰다.

5.2 정형 환경

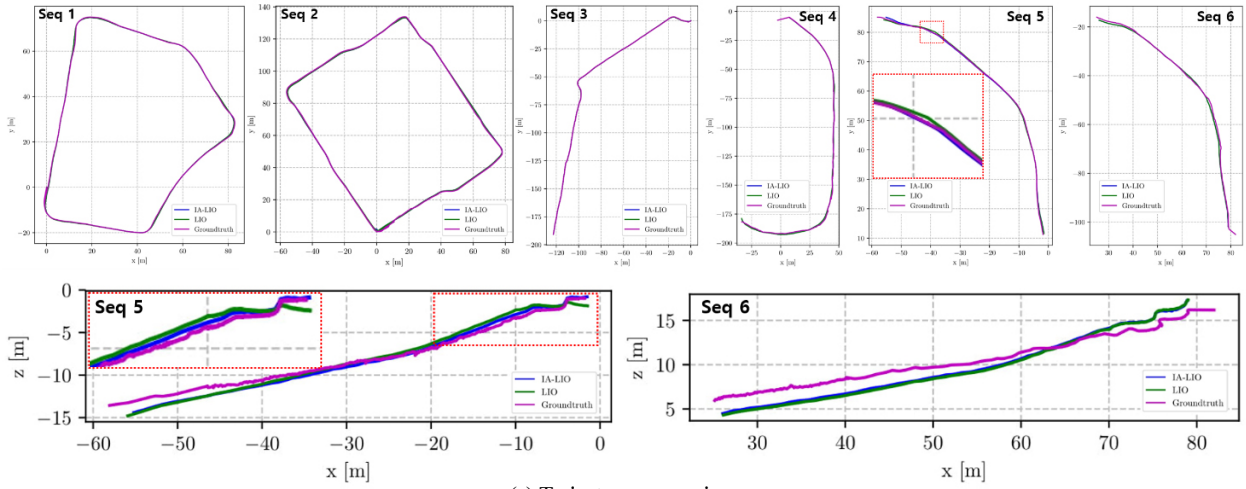
정형 환경은 Seq 1~3으로 구성되어 있으며 ambient 임계값 $\lambda=800$ 에서 검증되었다. 데이터는 아스팔트 위 또는 보도에서 취득하였다. [Fig. 7]과 [Table 2]를 통해 Seq 1를 제외한 Seq 2, 3의 RMSE 값이 더 큰 것을 확인할 수 있다. 실제로 정형 환경에서는 ambient 임계값을 통해 불필요한 특징점만 지우는 것이 아니다. 또한, 데이터 습득 시 비정형 환경에 비해 떨림이 적기 때문에 \hat{T}_{i+1} 를 통한 특징점의 W 좌표계로의 변환이 상대적으로 정확하다. 이런 점들을 고려할 때, 정형 환경에서 IA-LIO-SAM이 LIO-SAM에 비해 정확한 결과를 내지 못할 수 있다. 그럼에도 Seq 1에서는 오히려 좋은 결과를 얻을 수 있었고, 가장 경로가 긴 Seq 2의 경우에도 IA-LIO-SAM과 LIO-SAM의 차이가 크게 나지 않았다. 결과적으로 값의 차이가 크지 않기 때문에 이 정도의 수치는 확률적으로 무시할 만하고, IA-LIO-SAM이 정형에서도 무리 없이 사용할 수 있음을 알 수 있다.

5.3 비정형 환경

비정형 환경은 Seq 4~6로 구성되어 있으며 ambient 임계값 $\lambda=750$ 에서 검증되었다. Seq 4의 경우 돌길, Seq 5, 6의 경우 흙길에서 데이터를 취득하였으며 이는 [Fig. 7(b)]에서 확인 가능하다. Seq 5와 6은 같은 장소에서 취득하였지만 Seq 5는 내리막 환경, Seq 6은 오르막 환경이라는 점에 차이가 존재한다. [Fig. 7(a)], [Fig. 7(c)] 및 [Table 2]는 각 데이터에서 생성된 경로, RMSE 그래프 및 결과값을 보여준다. 모든 데이터에서 RMSE 값이 줄었다는 사실을 알 수 있으며, 비정형 환경에서 IA-LIO-SAM이 불필요한 특징점을 제거하고, intensity와 ambient를 이용한 매칭이 성공적으로 이루어졌음을 유추할 수 있다. 또한 정형 환경과 달리 추가 성분 고려에 따른 효과가 더 크게 나타남을 볼 수 있다.

[Table 2] Dataset translation error (meters)

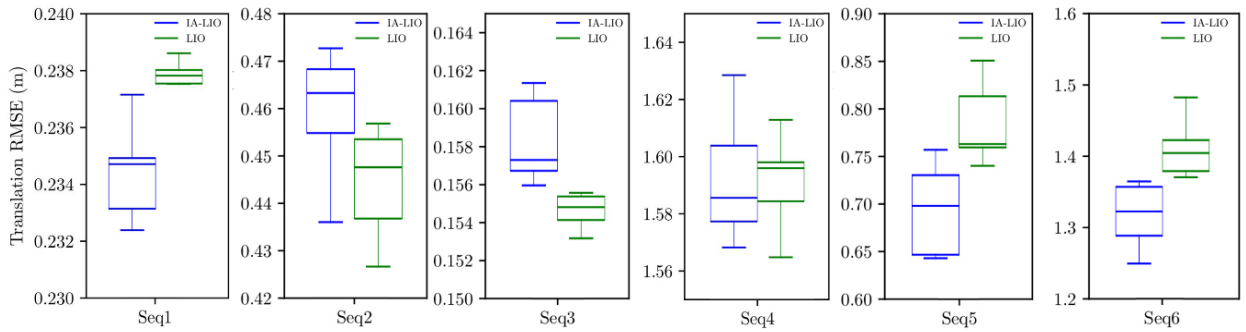
Structured Environment						Unstructured Environment					
Data	Algorithm	Mean	Median	Min	Max	Data	Algorithm	Mean	Median	Min	Max
Seq 1 (301 m)	IA-LIO	0.234	0.235	0.232	0.237	Seq 4 (247 m)	IA-LIO	1.585	1.586	1.496	1.628
	LIO	0.237	0.238	0.235	0.239		LIO	1.591	1.596	1.565	1.613
Seq 2 (408 m)	IA-LIO	0.459	0.463	0.434	0.473	Seq 5 (114 m)	IA-LIO	0.695	0.699	0.644	0.758
	LIO	0.444	0.448	0.427	0.457		LIO	0.791	0.764	0.741	0.915
Seq 3 (137 m)	IA-LIO	0.160	0.157	0.156	0.175	Seq 6 (137 m)	IA-LIO	1.337	1.323	1.249	1.545
	LIO	0.155	0.155	0.153	0.157		LIO	1.412	1.405	1.371	1.482



(a) Trajectory comparison



(b) Data Environment



(c) Box plot of RMSE

[Fig. 7] Result of dataset. (a) shows x-y trajectories of IA-LIO-SAM (blue), LIO-SAM (green), groundtruth (pink). Image below is the x-z trajectories. The real world image is shown in (b). Left image is top view of Seq 1 (red), Seq 2 (white), Seq 3 (blue). The top view of Seq 4 (blue region) and Seq 5,6 (red region) are in the middle image. In the right image, above is front view of Seq 4, and below is view of Seq 6 (also Seq 5). The result is shown in (c) through the box plot of RMSE

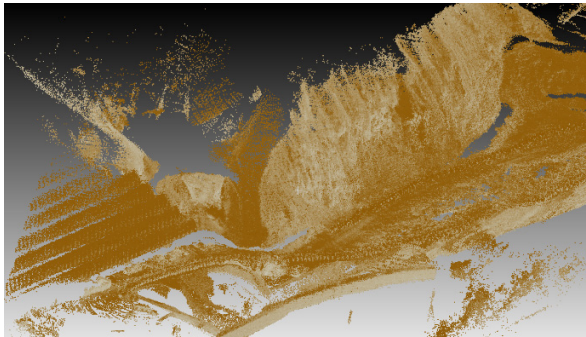
5.4 공사 현장

경기도 광주에 위치한 안성-성남 고속도로 공사 현장에서 Seq 7을 취득하였으며 ambient 임계값 $\lambda=1400$ 에서 검증되었다. [Fig. 8(a)]는 공사 현장 환경을 보여주며 [Fig. 8(b)]는 IA-LIO-SAM을 통해 제작된 점군 지도를 나타낸다. 제한된

UGV가 돌길이나 경사 등이 많이 위치한 공사 현장에서 정상적으로 작동함을 확인하였으며 비정형 환경과 마찬가지로 ambient 임계값 설정을 통해 불필요한 특징점을 제거할 수 있었다. 끝으로 [Table 3]를 통해 모든 지표에서 LIO-SAM에 비해 IA-LIO-SAM이 더욱 높은 정확도를 보여준 것을 확인할 수 있다.



(a) Data Environment



(b) Map from IA-LIO-SAM

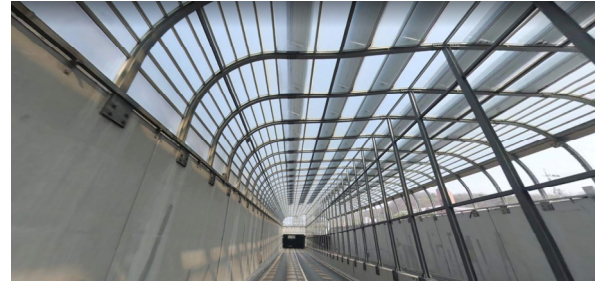
[Fig. 8] Construction site data. (a) shows the actual environment of the construction site. Not only the UGV being scanned, but also a running excavator are visible. (b) is a part of point cloud from IA-LIO-SAM. The color of points was expressed according to the ambient value

[Table 3] Construction site translation error (meters)

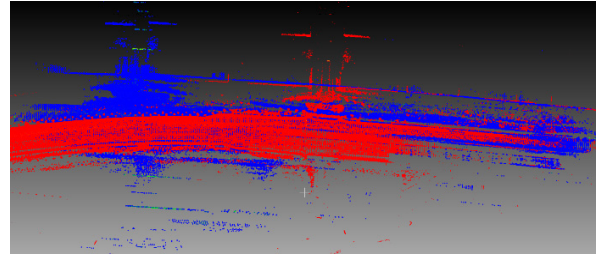
Data	Algorithm	Mean	Median	Min	Max
Seq 7 (635 m)	IA-LIO	0.427	0.428	0.414	0.441
	LIO	0.439	0.440	0.429	0.450

5.5 터널 환경

마지막 실험은 [Fig. 9(a)]에서 볼 수 있는 세종시 도심 터널에서 검증되었다. 터널은 좁고, 검출되는 특징점이 유사하다는 점에서 SLAM이 성공적으로 되기 힘들다. 규모상 제한된 UGV 시스템으로 데이터를 취득하기 힘들기 때문에, [20]에서 소개된 차량 및 OS1-64를 이용하여 Seq 8을 취득하였으며 ambient 임계값 λ 는 설정하지 않았다. LIO-SAM의 경우 초기 점군 지도를 형성하는데 실패하였으며, [Fig. 9(b)]에서 볼 수 있듯이 사거리나 건물의 위치가 왜곡되며 생성되었다. 이는 터널의 유사한 특징점으로 인한 매칭 실패가 원인이었다고 추측된다. 하지만 IA-LIO-SAM의 경우 성공적인 결과를 얻었다.



(a) Data Environment



(b) Map from LIO-SAM and IA-LIO-SAM



(c) IA-LIO-SAM map aligned with Kakao map

[Fig. 9] Sejong tunnel data. It was acquired by installing OS1-64 in a car. (a) is a Kakao map load view of the data acquisition location. (b) is a part of point cloud from LIO-SAM (blue) and IA-LIO-SAM (red). (c) shows a comparison with the real world

터널 환경에서의 RTK-GPS는 취득이 불가능하여 검증 시 기준 값을 활용할 수 없었다. 하지만 [Fig. 9(c)]에서 보여주듯이 완성된 지도를 통한 정성적인 평가가 가능하며, 왜곡이 발생하지 않은 채 실제 지도와 잘 일치함을 알 수 있다. 또한 본 실험은 유사한 특징점으로 인해 매칭이 어려운 환경으로 알려진 터널에서 비교 대상인 LIO-SAM이 점군 지도 제작에 실패하였음에도 불구하고 IA-LIO-SAM은 성공적인 결과를 얻었다는 점에 의의가 존재한다.

6. 결 론

IA-LIO-SAM은 intensity와 ambient를 활용하여 LiDAR odometry factor의 성능을 향상시켰다. Ambient 값을 통해 물체를 구별할 수 있었으며, 이는 비정형 환경에서 불필요한 특징점을 제거하는데 효과적이다. 또한, 위치를 통한 특징점 매

칭에 intensity와 ambient를 통한 매칭을 추가시킴으로 LiDAR odometry factor를 강화할 수 있었다. IA-LIO-SAM은 총 네 가지 환경에서 실험이 이루어졌다. 비정형 환경과 공사 현장에서는 LIO-SAM에 비해 더 높은 정확도를 얻을 수 있었고, 정형 환경에서는 유사한 결과를 얻어 어느 환경에서나 적용 가능함을 보여주었다. 특히, 터널 환경에서는 LIO-SAM은 맵핑에 실패하지만 IA-LIO-SAM에서는 성공하는 것을 추가적으로 보여준다. 추후에 적응적인 필터링을 적용하여 불필요한 매칭 후보를 제외하고, 제안한 알고리즘을 사용한다면 정형 환경에서도 높은 정확도를 얻을 수 있을 것이다. 또한, 앞으로는 점군의 위치뿐만 아니라 intensity와 ambient 등의 값 역시도 다양한 분야에서 사용될 수 있을 것이다.

References

- [1] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, DOI: 10.1109/IROS45743.2020.9341176.
- [2] P. J. Besl and N. D. McKay, "A Method for Registration of 3D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992, DOI: 10.1109/34.121791.
- [3] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robotics: Science and Systems*, Seattle, USA, 2009, DOI: 10.15607/RSS.2009.
- [4] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The Trimmed Iterative Closest Point algorithm," *Object recognition supported by user interaction for service robots*, Quebec City, QC, Canada, 2002, DOI: 10.1109/ICPR.2002.1047997.
- [5] J.-M. Lee and G.-W. Kim, "A Camera Pose Estimation Method for Rectangle Feature based Visual SLAM," *Journal of Korea Robotics Society*, vol. 11, no. 1, Mar., 2016, DOI: 10.7746/jkros.2016.11.1.033.
- [6] J. H. Lee, G. Zhang, and I. H. Suh, "Motion Estimation Using 3-D Straight Lines," *Journal of Korea Robotics Society*, vol. 11, no. 4, Dec., 2016, DOI: 10.7746/jkros.2016.11.4.300.
- [7] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004, DOI: 10.1023/B:VISI.0000029664.99615.94.
- [8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008, DOI: 10.1016/j.cviu.2007.09.014.
- [9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision*, pp. 2564-2571, 2011, DOI: 10.1109/ICCV.2011.6126544.
- [10] J. Zhang and S. Singh, "Low-drift and Real-time Lidar Odometry and Mapping," *Autonomous Robots*, vol. 41, pp. 401-416, 2017, DOI: 10.1007/s10514-016-9548-2.
- [11] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758-4765, 2018, DOI: 10.1109/IROS.2018.8594299.
- [12] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8899-8906, 2020, DOI: 10.1109/ICRA40945.2020.9197567.
- [13] H. Ye, Y. Chen, and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3144-3150, 2019, DOI: 10.1109/ICRA.2019.8793511.
- [14] T.-M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen, and L. Xie, "LIRO: Tightly coupled lidar-inertia-ranging odometry," *IEEE Int. Conf. Robot. Automat.*, 2020, [Online], <https://arxiv.org/abs/2010.13072>.
- [15] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1-21, Feb., 2017, DOI: 10.1109/TRO.2016.2597321.
- [16] H. Wang, C. Wang, and L. Xie, "Intensity Scan Context: Coding Intensity and Geometry Relations for Loop Closure Detection," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2095-2101, 2020, DOI: 10.1109/ICRA40945.2020.9196764.
- [17] T. Shan, B. Englot, F. Duarte, C. Ratti, and R. Daniela, "Robust Place Recognition using an Imaging Lidar," *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, [Online], <https://arxiv.org/abs/2103.02111v2>.
- [18] X. Chen, T. Labe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss, "OverlapNet: Loop Closing for LiDAR-based SLAM," *Robotics: Science and Systems (RSS)*, 2020, [Online], <https://arxiv.org/abs/2105.11344v1>.
- [19] Y. S. Park, H. Jang, and A. Kim, "I-LOAM: Intensity Enhanced LiDAR Odometry and Mapping," *2020 17th International Conference on Ubiquitous Robots (UR)*, Kyoto, Japan, pp. 455-458, 2020, DOI: 10.1109/UR49135.2020.9144987.
- [20] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex Urban Dataset with Multi-level Sensors from Highly Diverse Urban Environments," *International Journal of Robotics Research*, vol. 38, no. 6, pp. 642-657, 2019, DOI: 10.1177/0278364919843996.



정 민 우

2021 KAIST 건설 및 환경공학과(공학사)
2021~현재 KAIST 건설 및 환경공학과
석사과정

관심분야: LiDAR SLAM, HD Map



장 혜 수

2018 KAIST 건설 및 환경공학과(공학사)
2020 KAIST 건설 및 환경공학과(석사)
2021~현재 KAIST 건설 및 환경공학과
박사과정

관심분야: 수중 로봇, 로봇 비전, SLAM



정 상 우

2021 KAIST 기계공학과(공학사)
2021 KAIST 전산학부(공학사)
2021~현재 KAIST 건설 및 환경공학과
석사과정

관심분야: SLAM



김 아 영

2005 서울대학교 기계항공공학부(공학사)
2007 서울대학교 기계항공공학전공(공학석사)
2007 서울대학교 기계항공공학전공(공학석사)
2014~현재 한국과학기술원 건설및환경공학과
부교수

관심분야: 영상기반 SLAM