

# 실내 자율주행 로봇을 위한 3차원 다층 정밀 지도 구축 및 위치 추정 알고리즘

## 3D Multi-floor Precision Mapping and Localization for Indoor Autonomous Robots

강규리<sup>1</sup>·이대규<sup>2</sup>·심현철<sup>†</sup>  
Gyuree Kang<sup>1</sup>, Daegy Lee<sup>2</sup>, Hyunchul Shim<sup>†</sup>

**Abstract:** Moving among multiple floors is one of the most challenging tasks for indoor autonomous robots. Most of the previous researches for indoor mapping and localization have focused on singular floor environment. In this paper, we present an algorithm that creates a multi-floor map using 3D point cloud. We implement localization within the multi-floor map using a LiDAR and an IMU. Our algorithm builds a multi-floor map by constructing a single-floor map using a LOAM-based algorithm, and stacking them through global registration that aligns the common sections in the map of each floor. The localization in the multi-floor map was performed by adding the height information to the NDT (Normal Distribution Transform)-based registration method. The mean error of the multi-floor map showed 0.29 m and 0.43 m errors in the x, and y-axis, respectively. In addition, the mean error of yaw was 1.00°, and the error rate of height was 0.063. The real-world test for localization was performed on the third floor. It showed the mean square error of 0.116 m, and the average differential time of 0.01 sec. This study will be able to help indoor autonomous robots to operate on multiple floors.

**Keywords:** Autonomous Robot, Multi-floor Mapping, Localization

### 1. 서 론

최근 자율주행 기술발전과 더불어 급격한 비대면 서비스의 증가로 자율로봇 분야의 연구 및 개발이 활발하게 진행되고 있다. 특히 로봇 산업 시장 확대와 실내배달, 서빙, 경비, 방역 등의 서비스를 제공하는 로봇의 수요가 높아지면서, 실내 자율주행 로봇의 필요성 또한 증가하고 있다. 이러한 실내자율주행을 수행하기 위하여 필요한 핵심기술 중 하나는 실내환경 지도 구축(맵핑)과 로봇의 위치 추정이다. 건물 내부의 경우, GPS (global positioning system) 위성 신호를 수신할 수 없고, 라이다(Light Detecting and Ranging) 센서를 이용한 맵핑과 위

치 추정 알고리즘인 SLAM(simultaneous localization and mapping)이 활발하게 이용되고 있다.

특히 실내서비스 로봇의 경우에는 건물 내 층간 이동을 위해서 다층 정밀 지도 구축과 해당 정밀 지도 내에서의 로봇의 위치 추정 기술이 필요하다. 지상로봇이 온 승강기를 이용하여 층간 이동을 하므로 로봇이 건물의 층마다 이동하며 3차원 정밀 지도를 연속적으로 구축하는 방법은 층별 생성된 정밀 지도의 높이 정보를 건물 전체 지도에 적용하는 데 있어 어려움이 있다. 이에, 층별로 단일 축 지도를 따로 구축하여 다수의 층별 지도를 개별적으로 사용하는 방법도 있지만, 이는 층마다 다른 원점으로 인하여 층간 이동 시 새로운 층에서의 위치 추정에 초기화 작업이 매번 수행되어야 한다는 단점이 있다.

다층 정밀 지도 구축과 관련해서 기존에 진행된 연구에서는 여러 센서를 결합하는 센서 융합 기술을 활용하여 다층 맵핑을 진행하였다. 레이저 센서와 기압계를 활용한 연구<sup>1,2</sup>에서는, 레이저 센서를 활용하여 메트릭 맵핑을 진행하고, 기압계를 이용하여 층간 이동을 감지하여 다층 지도를 구축하였

Received : Oct. 20. 2021; Revised : Dec. 7. 2021; Accepted : Dec. 17. 2021

1. M.D Student, Robotics Program, KAIST, Daejeon, Korea  
(fingb20@kaist.ac.kr)

2. Ph.D Student, Electrical Engineering, KAIST, Daejeon, Korea  
(lee.dk@kaist.ac.kr)

† Professor, Corresponding author: Electrical Engineering, KAIST, Daejeon, Korea (hcshim@kaist.ac.kr)

다. 라이다 SLAM의 odometry 결과를 이용하여 연속된 다층 지도를 구축한 연구<sup>3)</sup>에서는 로봇 주행 환경의 바닥 정보를 추출하여 이를 제약조건으로 활용하여 지도 작성 과정에서 수직 방향의 LiDAR Odometry 정확도를 높였다. 하지만, 승강기를 이용하는 환경에서 맵핑을 할 수 없다는 한계점이 있다.

다층 환경에서 위치 추정과 관련된 연구에서는 와이파이 정보를 활용하여 로봇의 대략적인 위치를 추정한 연구<sup>4)</sup>, 자이로센서, 가속도계, 자력계, 기압계, 와이파이 및 GPS 수신기를 가진 스마트폰을 이용하여 각 센서의 신호 수신 세기를 통하여 층 정보를 얻은 연구<sup>5)</sup>가 있다.

최근에 진행된 연구<sup>6)</sup>에서는 지도를 확장하는 방법으로 의미론적 지도 합성(semantic map fusion)을 사용하였다. 해당 연구에서는 두 개의 서로 다른 지도의 공통되는 부분을 정합 알고리즘을 활용해 대응시키는 방법을 활용하였지만, 높이가 일정한 환경에서만 실험을 진행했다는 점에서 본 논문과 차이점이 있다.

본 논문에서는 라이다 센서와 IMU 센서만을 활용하여 다층 실내 환경에서 정밀 지도를 구축하고, 위치를 추정하는 연구를 진행하였다. 이 논문에서 제시한 방법은 아래 세 가지의 이점을 가진다.

- 로봇의 층간 이동이 있는 실내환경에서의 자율주행을 위한 다층 정밀 지도를 구축한다.
- GPS 값이 부정확한 실내 환경에서 라이다 센서와 IMU 센서만을 활용하여 로봇의 위치를 추정한다.
- 층마다 동일한 원점을 제공함으로써, 층 이동 시 위치추정 초기화 작업을 필요로 하지 않는다.

단일 층 지도 구축은 논문의 2.1.1 부분에서, 다층 지도 구축은 2.1.2 부분에서 다룰 것이다. 구축된 다층 지도에서의 위치 추정은 2.2에서 살펴볼 것이다. 또한 논문의 3 부분은 카이스트 본원에 위치한 KI빌딩[Fig. 1]에서 진행된 실험 방법을 서술할 것이다. 마지막으로 4 부분에서 다층 정밀 지도 구축 및 위치 추정 실험 결과에 관하여 설명한다.

## 2. 알고리즘

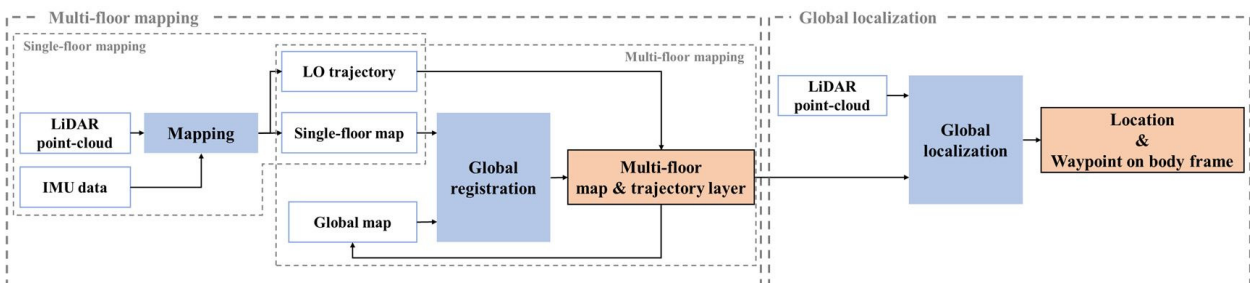
### 2.1 다층 정밀 지도 구축

다층 지도 구축은 [Fig. 2]와 같이 크게 단일 층 지도 구축, 층 누적을 통한 다층 지도 구축 두 단계로 나뉘어 진다. 단일 정밀 지도 구축 단계에서는 일반적인 LOAM<sup>8)</sup> 계열의 지도 구축 방식과 유사하게, 라이다 센서와 IMU 센서 정보를 활용하여 LiDAR Odometry (LO)를 얻어 지도를 구축한다.

구축된 단일 층 지도들을 수직으로 쌓아 올리는 층 누적 단계에서는 각 층이 이전 층과 동일하게 가지는 특징점(feature)들을 추출하여 전역 정합(global registration) 알고리즘을 활용하여 추가될 층의 변환 행렬(transform matrix)을 구한다. 이때 주어진 초기 위치에서 시작하여 매칭이 이루어질 경우 해당 행렬을 이용하여 층 전체 지도를 이동시켜 건물 전체 지도 위로 쌓아 올린다.



[Fig. 1] Indoor (a) and outdoor (b) pictures of KI building which is placed inside KAIST (Korean Advanced Institution of Science and Technology)



[Fig. 2] 3D multi-floor map building and global localization algorithm for multi-floor mapping. Left part of the graph demonstrates the multi-floor mapping algorithm in two steps: single-floor mapping using mapping algorithm, and multi-floor mapping system using global registration algorithm. The global localization algorithm on the right side of the graph uses real-time LiDAR point-cloud input and finds the location of the robot using 3D multi-floor map built in the previous step. The final goal of the system is getting both location and waypoints of the robot on body frame

### 2.1.1 단일 층 지도 구축

실내환경의 경우, 일정한 바닥 높이를 가지기 때문에, 바닥 전체를 개별의 특징점으로 활용하는 LEGO-LOAM<sup>[9]</sup>을 백본(backbone) 알고리즘으로 활용하였다. LEGO-LOAM은, 라이다 센서로 시간  $\tau$ 에 들어온 점 구름 중, 평면 분할(plane segmentation)을 통하여 분류된 바닥에 해당하는 점들을 별개의 특징점( $F_\tau^f$ ) 활용한다. 이후, 나머지 점들을 모서리( $F_\tau^e$ )와 면( $F_\tau^p$ ) 특징점으로 분류한다. 이러한 분류방식은 특징점 추출 단계에서 비교적 적은 연산량과 빠른 속도를 가져 지상공돌로봇의 정밀 지도 구축에 적합하다<sup>[7]</sup>.  $F_\tau^e$ 는 이전 프레임에서 추출된 바닥 특징점( $F_{\tau-1}^f$ )과 이전까지의 LO( $Q_{\tau-1}$ )와의 비교를 통하여 현재 상태의 LO( $O_\tau$ )의 변환 행렬( $T_\tau$ )에서 z축의 이동( $t_\tau^z$ )과 pitch( $\beta_\tau$ ), roll( $\gamma_\tau$ )에 해당하는 변화량을 계산하는 데 활용되었다.  $F_\tau^e$ 는 이전 프레임의 모서리 특징점( $F_{\tau-1}^e$ ),  $Q_{\tau-1}$ 와 비교를 통해서 x축 이동( $t_\tau^x$ ), y축 이동( $t_\tau^y$ )과 yaw( $\alpha_\tau$ ) 변화량을 계산하는 데 활용되었다. 이때, IMU 센서 정보를 활용하여  $T_\tau$ 를 보정하여 변환의 정확도를 한 번 더 높였다.

단일 층 지도( $M_{1:\tau}$ )는 이전 프레임 지도( $M_{1:\tau-1}$ )위에  $T_\tau$ 를 통하여 변환된 특징점  $F_\tau^f$ ,  $F_\tau^e$ 와  $F_\tau^p$ 를 추가하여 만들어졌다. 이때, LO를 누적하여 만든 로봇의 궤도( $S_{1:\tau}$ )는 이후 위치 추정 파트(2.2.2)에서 위치에 대한 가이드를 주는 데 활용되었다.

### 2.1.2 다층 지도 구축

구축된 단일 k층 지도( $M_k$ )를 적층하여 통합된 다층 지도( $M^{total}$ )를 구축하기 위한 방법으로 NDT (Normal Distribution Transform) 기반의 정합(registration) 알고리즘[Algorithm 1]을

[Algorithm 1] 3D multi-floor mapping algorithm

#### Algorithm 1: Multi-floor mapping

```

1  Input:  $M_{1:n}, z_{bias}$ 
2  For  $M_i$  in  $M_{1:n}$  do:
3      Extract  $P_i^M = \{p_{i,1}^M, p_{i,2}^M, \dots, p_{i,j}^M\}$  from  $M_i$ 
      and  $P_i^T = \{p_{1:i-1,1}^T, p_{1:i-1,2}^T, \dots, p_{1:i-1,j}^T\}$ 
4      While  $d < \delta$  do:
5          minimize  $d = \sum_j ((R_k p_j^M + t_k) - p_{1:i-1,j}^T)$ 
6          Determine  $R_k$  and  $t_k$ 
7          Add  $z_{bias}$  to  $t_k$  ( $t_k \rightarrow t_k + z_{bias}$ )
8          Transform  $M_i$  by  $R_k$  and  $t_k$ 
9          Merge  $M_i$  with  $U_{1:i-1}$  ( $U_{1:i} = \sum_{j=1}^i R_k M_j + t_k$ )
10         Transform  $S_i$  by  $R_k$  and  $t_k$ 
11         Merge  $S_i$  with  $S_{1:i-1}$  ( $S_{1:i} = \sum_{j=1}^i (R_k S_j + t_k)$ )
12     If  $i = n$  do:
13          $M^{total} = U_{1:n}$  and  $S^{total} = S_{1:n}$ 
14     End for

```

사용하였다.  $M_k$ 를 앞서 만들어진 층간 누적 지도( $U_{1:k-1}$ )에 대응시키기 위한 회전 변환 행렬( $R_k$ )과 평행 이동 벡터( $t_k$ )를 찾는 것을 문제의 목표로 한다. 이때,  $U_{1:k-1}$ 은 이전 층까지의 지도들을 누적하여 만든 다층 지도이다.

$$U_{1:k-1} = \sum_{l=1}^{k-1} M_l^{trans} \quad (1)$$

$M_k$ 의  $U_{1:k-1}$ 로의 변환은 두 지도에서 동일하게 존재하는 구역을  $M_k$ 로부터 추출한 점군( $P_k^M = \{p_{k,1}^M, p_{k,2}^M, \dots, p_{k,j}^M\}$ )와,  $U_{1:k-1}$ 로부터 추출한 점군( $P_{1:k-1}^T = \{p_{1:k-1,1}^T, p_{1:k-1,2}^T, \dots, p_{1:k-1,j}^T\}$ )간의 변환으로 재정의될 수 있고, 이는 아래 식 (2)와 같이 표현될 수 있다.

$$\text{minimize } d = \sum_{\forall j} ((R_k p_{k,j}^M + t_k) - p_{1:k-1,j}^T) \quad (2)$$

<sup>[10],[11]</sup>에 따라 식 (2)를 반복적으로 해를 찾는 방법(iteration)으로 해를 구하기 위해서  $P_k^M$ 과  $P_{1:k-1}^T$ 의 각 점에 대하여 정규 분포 노이즈를 갖도록 분산( $\sigma$ )을 부여하였다. 반복적으로 해를 찾는 과정 중  $d < \delta$ 인  $R_k$ 와  $t_k$ 를 찾은 경우, 반복을 멈춘다. 이후  $R_k$ 와  $t_k$ 에 층간 간격( $z_{bias}$ )을 더해 새로운  $t_\tau$ 를 정의하고, 변환행렬로  $M_k$ 를 이동시켜 얻은  $M_k^{trans} = R_k M_k + t_k$ 를  $U_{1:k-1}$ 에 누적하여 새로운 다층 지도  $U_{1:k}$ 를 정의하였다. 이 과정을 전체 층( $n$ )까지 반복하여 식 (3)과 같이 전체 층에 대한 지도인  $M^{total}$ 을 구축한다.

$$M^{total} = U_{1:n} \quad (3)$$

이때, 해당 k층의 누적 궤도( $S_k$ )를 동일한  $R_k$ 와  $t_k^{z_{bias}} = t_k + z_{bias}$ 를 이용하여 이동시켜  $M^{total}$ 에 대한 누적 궤도( $S^{total}$ )를 식 (4)와 같이 생성하였다.

$$S^{total} = \sum_{k=1}^n R_k S_k + t_k^{z_{bias}} \quad (4)$$

### 2.2 위치 추정 알고리즘

다층 정밀 지도에서의 라이다 센서 기반 위치 추정은 현재 k층의 높이( $z_{bias}$ )와 라이다 센서 정보를 이용하여 [Algorithm 2]와 같이 수행할 수 있다. 우선, 현재 층에 해당하는 지도 점군( $P_k^T$ )을 현재 층의 높이( $z_k$ )와 하임계점( $l_1$ ), 상임계점( $l_2$ )을 이용하여 아래 식 (5)와 같이 선별한다.

$$P_k^T = \{p_k | p_k \in M^{Total}, l_1 < p_{k,z} - z_k < l_2\} \quad (5)$$

라이다 센서 데이터( $\mathbf{P}^I = \{p_1^I, p_2^I, \dots, p_j^I\}$ )를  $\mathbf{P}_k^T$ 과 대응시키는 NDT 스캔 매칭 기반의 알고리즘을 통하여 현재 로봇의 위치를 추정한다. 이때, 스캔 매칭의 결과는 식 (1)과 동일한 방식으로 구해질 수 있다.

$$\text{minimize } \sum_{\forall j} ((R_{\tau}^{Robot} p_j^I + t_{\tau}^{Robot}) - p_{k,j}^T) \quad (6)$$

여기서  $R_{\tau}^{Robot}$  과  $t_{\tau}^{Robot}$  은 각각 지도 내에서의 로봇의 회전 변환 행렬과 평행 이동 벡터를 나타낸다. 한편 NDT 알고리즘 내부에서는 식 (6)과 같이 반복적으로 해를 찾는 과정에서  $R_{\tau}^{Robot}$  과  $t_{\tau}^{Robot}$  의 초기값을 부여하여 해당 위치를 초기값으로 사용하여 위치 추정을 위한 반복 계산이 빠르게 수렴하게 할 수 있다. 또한 스캔 매칭 기반의 위치 추정 알고리즘을 좀더 강인하게 수행하기 위해 본 연구진이 다른 논문에서 발표한 위치 추정 알고리즘을 활용하였다<sup>[12]</sup>. 따라서 본 연구에서는 정밀지도 구축 과정에서 생성한 층별 로봇의 궤도 정보  $S^T$ 을 활용하여 가장 가까운 위치에 해당하는 점의 좌표를 초기값으로 설정하여 해당 층에서 로봇이 위치 추정을 안정적으로 할 수 있도록 하였다.

층을 이동할 경우, 새로운 층( $j$ )에 대해서 새롭게  $\mathbf{P}_j^T$ 을 정의하고, 식 (6)의 NDT 스캔 매칭 기법을 이용하여 새로운 초기 위치를 설정하는 것을 반복한다. 이때, 빠른 초기 위치 설정을 위하여 이전 층에서의  $R_{\tau,k}^{Robot}$ ,  $t_{\tau,k}^{Robot}$  과 현재 층에 해당하는 궤도 정보( $S_j^T$ )을 활용한다.

### 3. 실험 방법

#### 3.1 하드웨어 세팅

지상로봇은 [Fig. 3]과 같이 2개의 구동바퀴와 4개의 보조바퀴로 구성된 differential wheel구성의 로봇을 활용하였다. 라이다 센서는 16개의 채널과 수직 30°, 수평 360°의 시야각, 100 m의 센서 범위를 가지는 3차원 라이다 센서를 사용하였다. IMU 센서는 정밀 지도 구축과 위치 추정에는 Intel® Core™ i7-11700B Processor (24M Cache, up to 4.80 GHz)가 포함된 탑

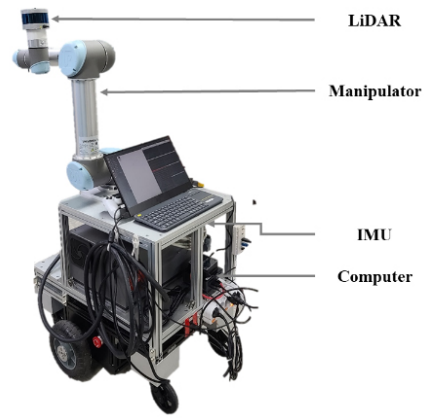
재형 PC가 활용되었다.

#### 3.2 실험환경

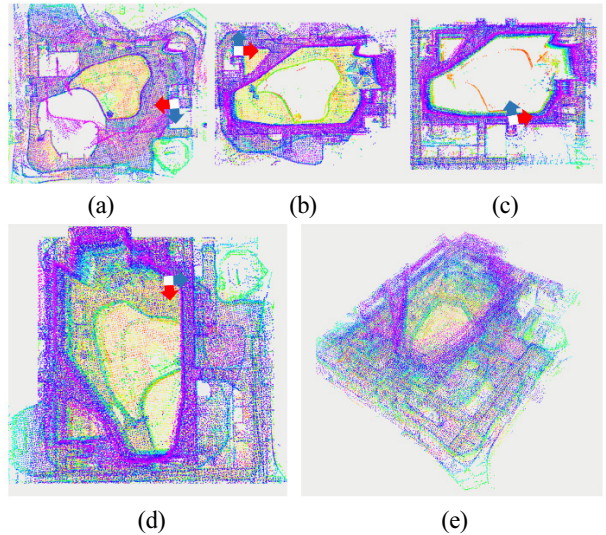
다층 정밀 지도 구축 알고리즘은 KAIST 본교 캠퍼스에 위치한 KI빌딩 1, 2, 3층을 대상으로 테스트되었다.

##### 3.2.1 다층 정밀 지도 구축

각층의 정밀 지도를 구축하는 단계에서는 Loop closure를 통하여 지도 제작 시 생겨난 슬립 현상을 보정하였다. 각층의 지도 제작 시작 위치는 층마다 상이하게 하여, 단일 층 정밀 지



[Fig. 3] The mobile manipulator robot used for multi-floor mapping and localization

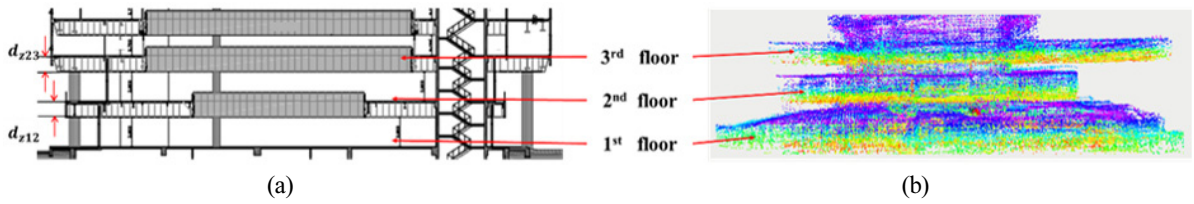


[Fig. 4] 3D point cloud map of singular floor (a-c), and multiple floor (d-e) of the KI building. (a-c): Map of the first, second, and third floor. (d), and (e) is the map of the map that combined map (a-c) after each transformation. (d) is the top-down view, and (e) is the diagonal view of the map. The white square, red arrow, and blue arrow each represents the center point, x-axis, and y-axis of the map

[Algorithm 2] 3D multi-floor map localization

Algorithm 2: Multi-floor localization	
1	<b>Input:</b> $M^{total}, z_{bias}, P_i^I$
2	<b>For</b> $p \in M^{total}$ <b>do:</b>
3	<b>If</b> $l_1 < p_z < l_2$ <b>do:</b>
4	Add $p$ to $P_i^T$
5	<b>While</b> $d < \delta$ <b>do:</b>
6	minimize $d = \sum_{\forall j} ((R_{\tau}^{Robot} p_j^I + t_{\tau}^{Robot}) - p_{k,j}^T)$
7	Determine $R_{\tau}^{Robot}$ and $t_{\tau}^{Robot}$





[Fig. 5] The drawing of the building side-view (a) and side-view of the map (b). The experiment was done on the first, second, and the third floor.  $d_{z12}$ , is defined as the gap between the ceiling of the first floor and the floor of the second floor.  $d_{z23}$  represents the same between second and the third floor

도의 x, y 원점은 각각 다르게 설정되었다. 다층 정밀 지도는 단일 층 지도를 낮은 층수부터 쌓아 올리는 방식으로 제작하였다. 단일 층 지도와 다층 지도 사이의 공통 특징점은 중앙 복도를 활용하였다.

### 3.2.2 다층 지도에서의 위치 추정

위치 추정 알고리즘은 다층 지도에서 가장 고층이자 누적 오차가 가장 큰 3층에서 [Fig. 4]의 안쪽 복도에 해당하는 경로를 따라 진행하였다. 해당 층의 대략적인 고도는 건물의 설계도를 참고해 넣어주었다. 추정 결과는 평균제곱오차와 추정 속도를 통해 나타내었다.

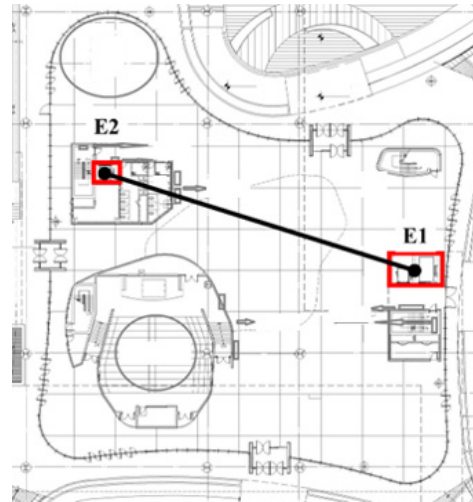
## 4. 실험 결과

### 4.1 다층 정밀 지도 구축

KI 빌딩의 1층부터 3층에서 구축한 단일 층 지도는 [Fig. 4(a-c)]에서 보는 것과 같다. 각 지도의 원점과 x축 y축 방향은 각각 흰색 점과 빨간색 화살표, 파란색 화살표로 표시하였다. 구축된 알고리즘은 [Algorithm 1]과 같은 방법으로 빌딩의 1층부터 3층까지 순서대로 적용한 결과, 구축된 다층 지도의 모습은 [Fig. 4(d-e)], [Fig. 5]와 같다. 각각의 단일 층 지도에서는 서로 다른 원점을 가졌지만, 구축된 다층 지도는 [Fig. 4(a)]와 같이 적용의 기준으로 삼은 1층의 원점으로 통일되었다.

다층 지도의 x, y 축과 yaw의 정확도는 [Fig. 6]에서 E1, E2로 표시된 두 개의 승강기의 x, y 좌표를 이용하여 측정하였다. [Fig. 4(d-e)]에서 구축된 3차원 다층 지도에서, 각 층의 E1, E2 승강기의 중심점을 구하고, 두 개의 중심점의 연결한 선분의 중점의 좌표를 비교하여 층간 x, y 축의 차이를 계산하였다.

결과는 [Table 1]에서와 같이 평균적으로 0.29 m, 0.43 m의 층간 차이를 보였다. 각 층의 E1과 E2를 연결한 직선의 기울기를 비교하여 yaw의 차이를 구하였는데, 1층과 2층 사이에는 0.43°, 2층과 3층 사이에는 1.58°로 평균적으로 1.00°의 차이를 보였다. z 축은, 구축한 3차원 다층 지도에서의 층간 높이 차이와 [Fig. 5(a)]의 도면상 수치( $d_{z12}$ ,  $d_{z23}$ )와 비교하여 오차와 오



[Fig. 6] A drawing of the first floor. E1 and E2 marked with red boxes are elevators. The center points of the elevators on each floor were calculate using feature points on the 3D multi-floor map

[Table 1] X-axis, y-axis and yaw error between floors

	X error [m]	Y error [m]	Yaw error [°]
1 <sup>st</sup> -2 <sup>nd</sup> floor	0.36	0.74	0.43
2 <sup>nd</sup> -3 <sup>rd</sup> floor	0.23	0.13	1.58
Average	0.29	0.43	1.00

[Table 2] Z-axis error between floors

	Z error [m]	Error rate
1 <sup>st</sup> -2 <sup>nd</sup> floor	0.06	0.03
2 <sup>nd</sup> -3 <sup>rd</sup> floor	0.20	0.09
Average	0.13	0.065

차를 계산하였다. 이때, 3차원 다층 지도에서의 층간 거리는 위층의 바닥과 아래층에 해당하는 점 중에서 xy 거리가 0.3 m 이내인 점들을 대응시켜 두 점의 z 값의 차이를 계산하여 구하였다. 1층과 2층 사이의 z 축 거리를 구할 때는 총 1454개의 점, 2층과 3층 사이는 총 1892개의 점들을 대응시켜 구한 값을 평균내었다. 그 결과, [Table 2]과같이 1층과 2층 간 z 축 거리의 오차는 0.06 m로 0.03 오차율을 보였고, 2층과 3층 사이에는 0.2 m의 오차와 0.09의 오차율을 보였다.

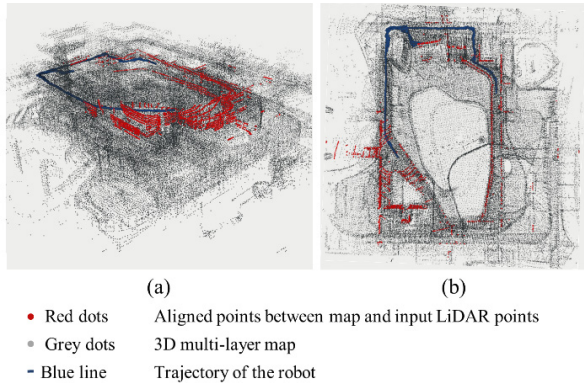
#### 4.2 다층 정밀 지도에서의 위치 추정

3층에서 진행된 다층 지도에서의 위치 추정 실험 결과는 [Fig. 7]과 같다. [Fig. 8]에 나타난 위치 추정 결과의 평균제곱오차의 평균은 0.116 m이다. 이때 위치 추정 평균미분시간은 0.01초이다.

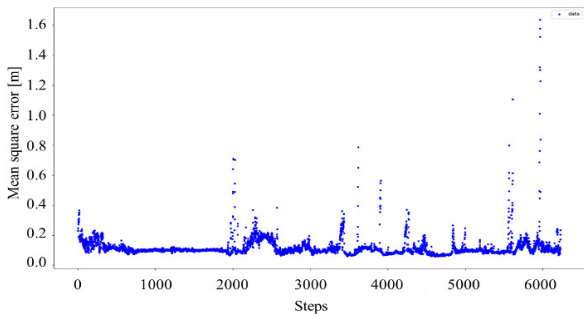
### 5. 결론

본 논문에서는 LOAM 기반의 3차원 다층 정밀 지도 구축과 위치 추정 알고리즘을 제안하였다. 점구름으로 이루어진 단일 층 지도를 전역 정합 알고리즘을 이용하여 적층하는 방식으로 구축된 다층 지도의 정확도는 층 간의 x, y 좌표 및 yaw의 차이와 설계도면과 비교한 z축의 오차로 평가하였다. 누적 오차가 가장 큰 3층에서 진행된 위치 추정 실험 결과 실시간이 가능한 빠른 속도와 정확성을 보였다.

다층 3층으로 진행된 다층 지도 구축은 층이 높아질수록 오차가 누적되어 정확성이 낮아진다는 단점을 가진다. 이를 해결하기 위한 층 누적 방향과 방법에 대해서 추가적인 연구를 통하여 개선되어야 할 것이다. 또한 현재까지는 층간 동일한 구조를



[Fig. 7] Side (a) and top-down (b) of 3D multi-floor map localization on the third floor. The grey dots represent the multi-floor map, and the red dots indicates the aligned points between the map and the LiDAR point cloud. The blue line is the trajectory of the robot



[Fig. 8] A graph of mean square error of the localization test on the third floor

활용해 각 층의 지도를 정합하기 때문에 층마다 전혀 다른 구조를 가지는 건물의 경우에는 다층 지도를 생성하는데 있어 한계점이 있다. 따라서 건물의 구조의 제약을 받지 않고 다층지도를 구축하는 방법에 대한 연구가 추가적으로 이루어져야 한다.

### Notation

Parameter	Description
$F_{\tau}^f$	Floor feature of point cloud 평면 분할을 통하여 분류된 바닥에 해당하는 점
$F_{\tau}^e$	Edge feature of point cloud 포인트 클라우드 중 모서리에 해당하는 특징점
$F_{\tau}^p$	Non-floor plane feature of point cloud 포인트 클라우드 중 면에 해당하는 특징점
$Q_{\tau}$	Lidar odometry (LO) 라이다 오도메트리 (LO)
$t_{\tau}$	Translation vector at time $\tau$ 시간 $\tau$ 에서의 평행이동 벡터
$R_{\tau}$	Rotational matrix at time $\tau$ ( $3 \times 3$ ) 시간 $\tau$ 에서의 회전 변환 행렬 ( $3 \times 3$ )
$M_{1:\tau}$	A single-layer map generated until time $\tau$ 지도가 생성되기 시작한 시점부터 시간 $\tau$ 까지 생성된 단일 층 지도
$S_{1:\tau}$	The trajectory of the robot until time $\tau$ 지도가 생성되기 시작한 시점부터 시간 $\tau$ 까지 저장된 로봇의 궤도
$M_k$	A single-floor map of the kth floor 구축된 단일 k-층에 대한 지도
$M^{total}$	Integrated multi-layer map 통합된 다층 지도
$U_{1:k}$	Multi-floor map stacked up to the kth floor k-층까지 적층된 다층 지도
$P_k^M$	Point cloud extracted from the kth floor map 구축된 단일 k-층 중 추출된 점군
$P_k^T$	The target point cloud extracted for the kth floor among the point clouds of the previously built multi-floor map 앞서 미리 구축된 다층 지도의 점군 중 k-층에 대해 추출된 타겟 점군
$R_k$	Rotation matrix that matches the kth floor map to the multi-floor map 단일 k-층을 다층 지도로 변환하는 회전 변환 행렬
$t_k$	Translation matrix that matches the kth floor map to the multi-floor map 단일 k-층을 다층 지도로 변환하는 평행 이동 벡터
$M_k^{trans}$	Point cloud transformed through the kth floor transformation matrix 단일 k-층을 변환 행렬을 통해 회전/평행 이동한 점군
$P^I$	Lidar sensor data 라이다 센서 데이터
$R_{\tau}^{Robot}$	Estimated rotation matrix of the robot within the map 지도 내에서 추정된 로봇의 회전 변환 행렬
$t_{\tau}^{Robot}$	Estimated translation vector of the robot within the map 지도 내에서 추정된 로봇의 평행 이동 벡터 (로봇의 지도 내 x,y,z 위치)

## References

- [1] R. Philippsen, L. Sentis, and O. Khatib, "An open source extensible software package to create whole-body compliant skills in personal mobile manipulators," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, 2011, DOI: 10.1109/IROS.2011.6095163.
- [2] Z. Fan, C. Li, Y. Wang, L. Zhao, L. Yao, G. Zhu, Z. Li, H. Xie, and Y. Xiao, "3D Mapping of Multi-floor Buildings Based on Sensor Fusion," *2017 International Conference on Industrial Informatics - Computing Technology*, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, China, 2017, DOI: 10.1109/ICIICII.2017.10.
- [3] X. Wei, J. Lv, J. Sun, and S. Pu, "Ground-SLAM: Ground Constrained LiDAR SLAM for Structured Multi-Floor Environments," arXiv:2103.03713, 2021, DOI: arXiv:2103.03713v1.
- [4] K. S. Kim, S. Lee, and K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting," *Big Data Analytics*, vol. 3, no. 4, 2018, DOI: 10.1186/s41044-018-0031-2.
- [5] Y. Li, Z. Gao, Z. He, P. Zhang, R. Chen, and N. El-Sheimy, "Multi-sensor multi-floor 3D localization with robust floor detection," *IEEE Access*, vol. 6, 2018, DOI: 10.1109/ACCESS.2018.2883869.
- [6] Q. Zhang, M. Wang, and Y. Yue, "Robust Semantic Map Matching Algorithm Based on Probabilistic Registration Model," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, DOI: 10.1109/ICRA48506.2021.9561176.
- [7] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang, and D. Wang, "Collaborative semantic understanding and mapping framework for autonomous systems," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, 2020, DOI: 10.1109/TMECH.2020.3015054.
- [8] J. Zhang and S. Sanjiv, "LOAM: Lidar Odometry and Mapping in Real-time," *Robotics: Science and Systems*, vol. 2, no. 9, 2014, [Online], [https://www.ri.cmu.edu/pub\\_files/2014/7/Ji\\_LidarMapping\\_RSS2014\\_v8.pdf](https://www.ri.cmu.edu/pub_files/2014/7/Ji_LidarMapping_RSS2014_v8.pdf).
- [9] 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)*, Madrid, Spain, 2018, DOI: 10.1109/IROS.2018.8594299.
- [10] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*(Cat. No. 03CH37453), Las Vegas, NV, USA, 2003, DOI: 10.1109/IROS.2003.1249285.
- [11] C. Ulaş and H. Temeltaş, "3D multi-layered normal distribution transform for fast and long range scan matching," *Journal of Intelligent & Robotic Systems*, vol. 71, 2013, DOI: 10.1007/s10846-012-9780-8.
- [12] D. Lee, G. Kang, B. Kim, and D. H. Shim, "Assistive Delivery Robot Application for Real-World Postal Services," *IEEE Access*, vol. 9, 2021, DOI: 10.1109/ACCESS.2021.3120618.



## 강규리

2020 성균관대학교 기계공학부(학사)

2020~현재 KAIST 로봇공학학제전공 석사과정

관심분야: 자율로봇, 자율 시스템, 로보틱스



## 이대규

2018 국민대학교 자동차공학(학사)

2020 KAIST 미래자동차학제전공(석사)

2020~현재 KAIST 전기 및 전자공학부 박사과정

관심분야: 로보틱스, 항법



## 심현철

1991 서울대학교 기계공학(학사)

1993 서울대학교 기계공학(석사)

2000 University of California, Berkeley Mechanical Engineering(박사)

관심분야: 로보틱스, 드론, 자율로봇, 자율주행차, 무인 항공기