LIDAR-BASED MULTI-SENSOR MODULE DEVELOPMENT AND ASSESSMENT OF APPLICABILITY FOR DAMAGE INVESTIGATION OF INDOOR DISASTER

D. Y. Shin¹, D. H. Cho², J. W. Hong², S. H. Han², S. B. Cho¹, E. T. Lim¹, Y. H. Jung¹, S. S. Kim^{1*}

¹ Disaster Scientific Investigation Division, National Disaster Management Research Institute, Ulsan, Republic of Korea - (sdy718, tlqja85, charmante93, jyh1315, sskim73)@korea.kr

² Dept. of Convergence IT Engineering, Pohang University of Science and Technology, Pohang, Republic of Korea – (dohoon.cho, junwooh, soohee.han@postech.ac.kr)

Commission Ⅲ, ICWG Ⅲ/IVa

KEY WORDS: Indoor Mapping, 3D point cloud, Disaster Investigation, LiDAR, SLAM, Robot.

ABSTRACT:

As the risk of disaster scenes increases, the number of cases of acquiring disaster scenes information using unmanned robots is increasing. Because unmanned robots can be remotely controlled, sensors such as LiDAR and optical cameras are installed so that investigators can safely observe disaster scenes and acquire information. In particular, the information that can be acquired is different depending on the sensor characteristics, and a sensor module suitable for the purpose is being developed. Accordingly, the National Disaster Management research Institute(NDMI) also developed a investigation robot capable of acquiring information on disaster sites independently. Based on 3D point cloud data, we developed a multi-sensor module and SLAM algorithm customized to the investigation robot to collect quantitative information on the damage situation. To test the performance of the independently developed multi-sensor module, SLAM mapping was performed in a disaster building reproduced like a disaster scene, and various SLAM algorithms and distance comparison were performed. As a result, PackSLAM developed in this study showed the lowest error. In the future, to increase applicability at disaster sites, more precise experiments will be conducted by conducted by establishing a rough terrain environment.

1. INTRODUCTION

It is difficult to receive GNSS information from the outside in an indoor space. Therefore, the indoor space is mapped using the SLAM technique that can measure the location of the mobile platform while generating a high precision map of the surrounding environment using the multiple observation sensors mounted on the mobile platform (Thrun et al., 2005; Kaess et al., 2012; Zlot et al., 2012).

In addition, it is hard for investigators to directly perform damage investigations at indoor disaster sites of collapsed accidents due to the complex structure and dangerous environment. For this reason, unmanned robots equipped with advanced observation sensors such as optical, infrared cameras, or LiDAR are being used to acquire recent disaster damage information about indoor spaces and identify those who need to rescue.

Since the unmanned robot can be controlled remotely in an area that is difficult for investigators to access, it is possible to easily obtain the safety of the investigator and necessary information.

According to a recent previous study, in order to develop a continuum robot system with various information collection functions, a study was conducted to install several small sensors in a distributed manner(Yamauchi et al., 2022). Also, various robots are being developed according to the purpose of fire response and fire suppression, underwater inspection of bridges, and damage recovery(Park et al., 2017; Paravisi et al., 2019; Gan et al., 2021)..

As unmanned robots for disaster safety, Shark Robotics Colossus of France, iRobot of FLIR of USA, Spot of Boston Dynamics of USA, etc. are being developed, and the NDMI has Packbot of iRobot acquired in 2017.

Therefore, this study suggests a multi-sensor module that can be mounted on an investigation robot (iRobot Packbot) to acquire damage information when performing an indoor disaster site damage investigation considering the characteristics of SLAM. Also, LiDAR-based SLAM algorithm customized for the multisensor module of the investigation robot for indoor spatial mapping is suggested and its applicability for indoor damage investigation is evaluated in this paper.

2. METHOD

2.1 Development of Multiple Sensor Module

The multi-sensor module is composed of LiDAR-IMU-NUC. LiDAR is Ouster OS1-32, IMU is VectorNav VN100, and NUC is a model equipped with an Intel i5 CPU. The most important part of a system that utilizes data acquired from a sensor module composed of multiple sensors is that the sensor data should be synchronized with one system clock.

In particular, in the SLAM algorithm, which uses LiDAR (perceptive sensor) and IMU (proprioceptive sensor) as a strongly coupled system, it is important to synchronize as the same time zone the data measured in real-time and the data is transmitted.

^{*} Corresponding author

The multi-sensor module is a Linux-based operating system using an Intel NUC 8th generation development board equipped with an Intel Core i5 CPU, and is a synchronized system that can receive sensor data in real time through real-time kernel patching.

Therefore, To utilize the synchronized time in the SLAM system that performs indoor positioning and mapping, this study used a timestamp method that displays the reception time of the data received through the sensor communication protocol as the time of the computing unit.

Synchronized sensor data can be used through an interface using the ROS (Robot Operating System) platform, and it is possible to check the reception frequency whether the sensor data acquired from Ouster OS1-32 LiDAR and VectorNav VN100 IMU are being received in real time(Figure 1 ~ 2).

<pre>Pie Ed View Sarch Temmini Tabs Heip //hom/dolloon/citim_un/tr(/pits</pre>			do	hoon@d	ohoon-zenboo	ok: ~/Proje	ts/PackSLAM/	packl	log/bags/2021-09-13	0 0 0
<pre>mt: 0.035 max: 0.549 std dev: 0.0345s window: 170 mt: 0.035 max: 0.549 std dev: 0.03745s window: 180 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03745s window: 190 wwrage rate: 10.002 wwrage rate: 10.003 mt: 0.035 std dev: 0.03255s window: 220 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03205s window: 220 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 220 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 220 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 220 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 220 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 220 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 220 mt: 0.035 max: 0.549 std dev: 0.03255 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 200 mt: 0.035 max: 0.549 std dev: 0.03255 window: 200 mt: 0.035 max: 0.549 std dev: 0.03255 window: 200 mt: 0.035 max: 0.549 std dev: 0.03255 window: 200 mt: 0.035 max: 0.549 std dev: 0.03255 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03055 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02855 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02855 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02855 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02855 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02855 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02855 window: 200 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02855 window: 301 wwrage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02885 window: 301 wwrage rate: 10.051 max: 0.549 std dev: 0.02885 window: 301 wwrage rate: 0.959 wwrage rate: 0.959 wwrage rate: 0.959 wwrage rate: 0.959 wwrage rate:</pre>	File Ed									
awrrage rate: 10.001 mt: 0.013 max: 0.549 std dev: 0.03742 window: 100 wwrrage rate: 10.001 mt: 0.033 max: 0.549 std dev: 0.03555 window: 200 mt: 0.033 max: 0.549 std dev: 0.03458 window: 210 wwrrage rate: 10.001 mt: 0.013 max: 0.549 std dev: 0.03468 window: 210 wwrrage rate: 0.957 mt: 0.013 max: 0.549 std dev: 0.03468 window: 220 wwrrage rate: 0.031 max: 0.549 std dev: 0.03358 window: 220 wwrrage rate: 0.031 max: 0.549 std dev: 0.03258 window: 220 wwrrage rate: 0.031 max: 0.549 std dev: 0.03258 window: 220 wwrrage rate: 0.031 max: 0.549 std dev: 0.03258 window: 220 wwrrage rate: 0.031 max: 0.549 std dev: 0.03258 window: 220 wwrrage rate: 0.031 max: 0.549 std dev: 0.03258 window: 220 wwrrage rate: 0.031 max: 0.549 std dev: 0.03258 window: 220 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.03258 window: 220 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.03058 window: 220 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.03058 window: 220 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.03058 window: 220 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.03058 window: 200 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 200 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.041 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.041 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.051 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.051 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.990 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate: 0.990 mt: 0.031 max: 0.549 std dev: 0.02058 window: 301 wwrrage rate:	/home	/dohoon/catkin_ws/sr	c/PackS		dohoon@doh	oon-zenboo	k: ~/Project		dohoon@dohoon-zenbook: ~/Project ×	<u>n</u> -
Mil: 0.033 max 0.5495 std dev: 0.03745 window: 100 warrage rate: 10.001 0.5495 std dev: 0.03555 window: 200 warrage rate: 10.002 0.5495 std dev: 0.03555 window: 200 warrage rate: 0.033 max 0.5495 std dev: 0.03555 window: 200 warrage rate: 0.037 0.5495 std dev: 0.03555 window: 200 warrage rate: 0.037 0.5495 std dev: 0.03555 window: 200 warrage rate: 0.038 max 0.5495 std dev: 0.03355 window: 200 warrage rate: 0.0315 max 0.5495 std dev: 0.03355 window: 220 warrage rate: 0.051 max 0.5495 std dev: 0.03255 window: 220 warrage rate: 0.051 max 0.5495 std dev: 0.03255 window: 220 warrage rate: 0.051 max 0.5495 std dev: 0.03255 window: 220 warrage rate: 0.051 max 0.5495 std dev: 0.03255 window: 200 warrage rate: 0.051 max 0.5495 std dev: 0.03255 window: 200 warrage rate: 0.051 max 0.5495 std dev: 0.03055 window: 200 warrage rate: 0.051 max 0.5495 std dev: 0.03055 window: 200 warrage rate: 0.051 max 0.5495 std dev: 0.03055 window: 200 warrage rate: 0.051 max 0.5495 std dev: 0.02055 window: 201 warrage rate: 0.051 max 0.5495 std dev: 0.020555 window: 301 warrage ra		min: 0.013s max:		std de	v: 0.03845s	window:				
awerage rate: 10.001 mt: 0.003 max: 0.549 std dev: 0.03644 window: 100 wwerage rate: 10.001 mt: 0.033 max: 0.549 std dev: 0.03458 window: 200 mt: 0.033 max: 0.549 std dev: 0.03478 window: 210 mt: 0.0318 max: 0.549 std dev: 0.03468 window: 220 wwerage rate: 0.997 mt: 0.0318 max: 0.549 std dev: 0.03306 window: 220 wwerage rate: 0.040 mt: 0.0318 max: 0.549 std dev: 0.03268 window: 220 wwerage rate: 0.040 mt: 0.0318 max: 0.549 std dev: 0.03258 window: 230 werage rate: 0.041 mt: 0.0318 max: 0.549 std dev: 0.03258 window: 241 mt: 0.0318 max: 0.549 std dev: 0.03258 window: 260 merage rate: 10.001 mt: 0.0318 max: 0.549 std dev: 0.03258 window: 260 merage rate: 10.031 mt: 0.0318 max: 0.549 std dev: 0.03258 window: 260 merage rate: 10.031 mt: 0.0318 max: 0.549 std dev: 0.03258 window: 270 mt: 0.0318 max: 0.549 std dev: 0.03258 window: 280 merage rate: 10.031 mt: 0.0318 max: 0.549 std dev: 0.02958 window: 280 merage rate: 0.041 mt: 0.0318 max: 0.549 std dev: 0.02958 window: 280 merage rate: 0.041 mt: 0.0318 max: 0.549 std dev: 0.02958 window: 280 merage rate: 0.041 mt: 0.0318 max: 0.549 std dev: 0.02958 window: 280 merage rate: 0.041 mt: 0.0318 max: 0.549 std dev: 0.02958 window: 280 merage rate: 0.048 mt: 0.0318 max: 0.549 std dev: 0.02968 window: 301 merage rate: 10.051 mt: 0.0318 max: 0.549 std dev: 0.02968 window: 301 merage rate: 0.051 max: 0.549 std dev: 0.02968 window: 301 merage rate: 0.051 max: 0.549 std dev: 0.02868 window: 301 merage rate: 0.549	average									
mit: 8.033 max. 0.409 std dev: 0.00454 window: 190 warrage rate: 10.001 warrage rate: 10.002 warrage rate: 10.002 warrage rate: 10.002 warrage rate: 0.003 warrage rate: 0.003 warrage rate: 0.004 warrage rate: 0.007 warrage rate: 0.007 warrage rate: 0.007 warrage rate: 0.007 warrage rate: 0.008 warrage rate: 0.008 warrage rate: 0.008 warrage rate: 10.001 warrage rate: 0.013 max warrage rate: 0.013 max warrage rate: 0.010 warrage rate: 0.011 warrage			0.549s	std de	v: 0.03742s					
werage rate: 10.001 mt: 0.003 max: 0.549 std dev: 0.03555 window: 200 werage rate: 10.001 mt: 0.033 max: 0.549 std dev: 0.03476 window: 210 werage rate: 0.907 mt: 0.033 max: 0.549 std dev: 0.03408 window: 220 werage rate: 0.907 mt: 0.035 max: 0.549 std dev: 0.03308 window: 220 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03255 window: 220 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03265 window: 241 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03265 window: 220 werage rate: 0.907 mt: 0.035 max: 0.549 std dev: 0.03265 window: 280 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03055 window: 270 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.03055 window: 280 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02055 window: 280 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02055 window: 280 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02055 window: 201 mt: 0.035 max: 0.549 std dev: 0.02055 window: 201 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02055 window: 201 werage rate: 10.001 mt: 0.035 max: 0.549 std dev: 0.02055 window: 201 werage rate: 10.001 werage rate: 0.001 werage rate: 0.00	average		0 540+	and day		ud adams	100			
<pre>min: 0.013s max: 0.450s std dev: 0.0355s window: 200 werrage fate: 10.02 min: 0.013s max: 0.450s std dev: 0.03476s window: 210 werrage fate: 0.937 min: 0.013s max: 0.450s std dev: 0.03130s window: 220 werrage fate: 10.031 min: 0.013s max: 0.450s std dev: 0.03130s window: 220 werrage fate: 10.031 min: 0.013s max: 0.450s std dev: 0.03250s window: 220 werrage fate: 10.031 min: 0.013s max: 0.450s std dev: 0.03250s window: 220 werrage fate: 10.031 min: 0.013s max: 0.450s std dev: 0.03250s window: 220 werrage fate: 10.031 min: 0.013s max: 0.450s std dev: 0.03250s window: 220 werrage fate: 10.031 min: 0.013s max: 0.450s std dev: 0.03250s window: 220 werrage fate: 10.031 min: 0.013s max: 0.450s std dev: 0.03250s window: 220 werrage fate: 0.031 min: 0.013s max: 0.450s std dev: 0.03055 window: 220 werrage fate: 0.031 min: 0.013s max: 0.450s std dev: 0.03055s window: 220 werrage fate: 0.031 min: 0.013s max: 0.450s std dev: 0.03055s window: 220 werrage fate: 0.013 min: 0.013s max: 0.450s std dev: 0.03055s window: 201 werrage fate: 0.013s max: 0.450s std dev: 0.02305s window: 321 werrage fate: 0.013s max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.051 min: 0.013s max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.013s max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.013s max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.013s max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.013 max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.013 max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.501 max: 0.450s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.501 max: 0.540s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.501 max: 0.540s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.501 max: 0.540s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.501 max: 0.540s std dev: 0.02305s window: 311 werrage fate: 0.501 min: 0.501 max: 0.540s std dev: 0.02305s window: 311 werrage fate:</pre>	average			stu de						
<pre>mi:: 0.013 max: 0.449 std dev: 0.03476 window: 210 wrrage fat:: 0.907 mi:: 0.013 max: 0.449 std dev: 0.03468 window: 220 wrrage fat:: 10.011 mi:: 0.013 max: 0.449 std dev: 0.03468 window: 230 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.0325 window: 241 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03252 window: 240 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03265 window: 250 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03265 window: 250 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03265 window: 250 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03165 window: 260 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03055 window: 270 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03055 window: 270 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.03055 window: 210 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 211 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 211 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 10.01 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 0.051 mi:: 0.013 max: 0.449 std dev: 0.02055 window: 201 wrrage fat:: 0.051 mi:: 0.051 max: 0.549 std dev: 0.02066 window: 301 wrrage fat:: 0.990 mi:: 0.9</pre>				std de	v: 0.03555s	window:				
awerage rate: 9.997 nt: 8.0815 max: 0.5495 std dev: 0.034085 window: 220 wwerage rate: 9.997 nt: 8.0815 max: 0.5495 std dev: 0.033355 window: 220 mt: 8.0315 max: 0.5495 std dev: 0.032555 window: 241 awerage rate: 10.031 nt: 9.0315 max: 0.5495 std dev: 0.032555 window: 250 mt: 9.0315 max: 0.5495 std dev: 0.032655 window: 250 awerage rate: 9.997 nt: 9.0315 max: 0.5495 std dev: 0.032655 window: 200 awerage rate: 9.997 nt: 9.0315 max: 0.5495 std dev: 0.030955 window: 270 awerage rate: 10.031 mt: 9.0315 max: 0.5495 std dev: 0.030955 window: 270 awerage rate: 9.998 nt: 9.0315 max: 0.5495 std dev: 0.020955 window: 201 mt: 9.0315 max: 0.5495 std dev: 0.020955 window: 201 awerage rate: 10.031 nt: 9.0315 max: 0.5495 std dev: 0.020955 window: 201 awerage rate: 10.031 nt: 0.0315 max: 0.5495 std dev: 0.020955 window: 201 awerage rate: 10.031 nt: 0.0315 max: 0.5495 std dev: 0.020955 window: 201 awerage rate: 10.031 nt: 0.0315 max: 0.5495 std dev: 0.020956 window: 301 awerage rate: 10.031 mt: 0.0315 max: 0.5495 std dev: 0.020956 window: 301 awerage rate: 10.031 mt: 0.0315 max: 0.5495 std dev: 0.020956 window: 301 awerage rate: 0.0315 max: 0.5495 std dev: 0.020956 window: 301 awerage rate: 0.938	average									
m(r): 0.033 max: 0.409 std dev: 0.03005 window: 220 warrage rate: 0.907 0.549s std dev: 0.03305 window: 230 warrage rate: 0.001 0.549s std dev: 0.03255 window: 241 warrage rate: 0.001 0.549s std dev: 0.03255 window: 241 warrage rate: 0.010 0.549s std dev: 0.03255 window: 241 warrage rate: 0.0316 max: 0.549s std dev: 0.03255 window: 240 warrage rate: 0.0316 max: 0.549s std dev: 0.03255 window: 250 warrage rate: 0.0316 max: 0.549s std dev: 0.03055 window: 270 warrage rate: 0.0518 max: 0.549s std dev: 0.03055 window: 280 warrage rate: 0.051 warrage rate: 0.0518 max: 0.549s std dev: 0.020355 window: 280 warrage rate: 0.051 m(r): 0.0318 max: 0.549s std dev: 0.020355 window: 280 warrage rate: 0.051 warrage ratet: 0.051 warrage rate: 0.051<			0.549s	std de	v: 0.03476s	window:				
average rate: 9.997 nt: 0.013 nax: 0.549 std dev: 0.03138 window: 230 wwerage rate: 10.001 nt: 0.013 nax: 0.549 std dev: 0.03259 window: 230 werage rate: 10.001 nt: 0.013 nax: 0.549 std dev: 0.03252 window: 250 nt: 0.013 nax: 0.549 std dev: 0.03052 window: 260 werage rate: 10.001 nt: 0.013 nax: 0.549 std dev: 0.03053 window: 270 werage rate: 10.001 metric 0.013 nax: 0.549 std dev: 0.03055 window: 270 werage rate: 0.013 nax: 0.549 std dev: 0.03055 window: 280 metric 0.013 nax: 0.549 std dev: 0.02055 window: 280 metric 0.013 nax: 0.549 std dev: 0.02055 window: 280 metric 0.013 nax: 0.549 std dev: 0.02055 window: 201 werage rate: 10.001 mi: 0.013 nax: 0.549 std dev: 0.02055 window: 301 werage rate: 10.001 mi: 0.013 nax: 0.549 std dev: 0.02385 window: 301 werage rate: 10.001 metric 0.015 nax: 0.549 std dev: 0.02885 window: 301 werage rate: 10.001 metric 0.015 nax: 0.549 std dev: 0.02886 window: 301 werage rate: 0.549 std dev: 0.02886 window: 301 werage rate: 0.549 std dev: 0.02886 window: 301 werage rate: 0.549 std dev: 0.02866 w	average									
<pre>min: 0.013 max: 0.409 std dev: 0.03358 window: 230 werage rate: 10.001 min: 0.013 max: 0.409 std dev: 0.03255 window: 241 werage rate: 10.001 min: 0.013 max: 0.409 std dev: 0.03205 window: 230 werage rate: 10.001 min: 0.013 max: 0.549 std dev: 0.03025 window: 230 werage rate: 0.013 max: 0.549 std dev: 0.03025 window: 270 werage rate: 0.013 max: 0.549 std dev: 0.03095 window: 270 werage rate: 0.013 max: 0.549 std dev: 0.03095 window: 280 werage rate: 0.013 max: 0.549 std dev: 0.02095 window: 280 werage rate: 0.959 std dev: 0.02095 window: 280 werage rate: 0.958 std dev: 0.02095 window: 281 werage rate: 0.958 std dev: 0.02095 window: 281 werage rate: 0.958 std dev: 0.02095 window: 281 werage rate: 0.549 std dev: 0.02095 window: 381 werage rate: 0.549 std dev: 0.02095 window: 381 werage rate: 0.549 std dev: 0.02095 window: 381 werage rate: 0.549 std dev: 0.02086 window: 331 werage rate: 0.998</pre>	average		0.5495	sta de	V: 0.034005	window:				
average rate: 10.001 mt: 20.013 max: 0.549s std dev: 0.03259s window: 241 average rate: 10.031 mt: 0.013 max: 0.549s std dev: 0.03202s window: 250 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.03103s window: 260 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.03905s window: 270 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.03935s window: 280 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.02985s window: 201 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.02985s window: 201 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.02985s window: 301 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.02985s window: 301 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.02886s window: 301 average rate: 10.001 mt: 0.013 max: 0.549s std dev: 0.02846s window: 301 average rate: 0.998 mt: 0.549s std dev: 0.02846s window: 301 average rate: 0.998 mt: 0.549s std dev: 0.02846s window: 301 average rate: 0.998	uverage		0.5495	std de	v: 0.03330s	window:	230			
average rate: 10.001 werge ML: 20.013 pass: 0.549 std dev: 0.03202s window: 220 werge rate: 0.549 werge rate: 0.549 werge rate: 0.549 mt: 0.0213 pass: 0.549 std dev: 0.03092s window: 270 mt: 0.0213 pass: 0.549 std dev: 0.03092s window: 280 werge pass: 0.549 std dev: 0.02982s window: 281 werge pass: 0.549 std dev: 0.02985s window: 281 mt: 0.0313 mass: 0.549 std dev: 0.02985s window: 281 werge rate: 10.001 mt: 0.0315 mass: 0.549 std dev: 0.02868s window: 381 werge rate: 9.998 werge rate: 9.998	average									
<pre>Mil: 0.013s max: 0.450s std dev: 0.03202s window: 256 Wercape fait: 0.013s max: 0.450s std dev: 0.03140s window: 266 Wercape fait: 0.013s max: 0.450s std dev: 0.0305s window: 270 wercape fait: 10.001 Wercape fait: 0.013s max: 0.450s std dev: 0.03035s window: 280 Wercape fait: 0.013s max: 0.450s std dev: 0.02035s window: 201 wercape fait: 0.013s max: 0.450s std dev: 0.02035s window: 301 mil: 0.013s max: 0.450s std dev: 0.02035s window: 301 wercape fait: 0.043s max: 0.450s std dev: 0.02035s window: 301 wercape fait: 0.043s max: 0.450s std dev: 0.02035s window: 301 wercape fait: 0.043s max: 0.450s std dev: 0.02035s window: 301 wercape fait: 0.043s max: 0.450s std dev: 0.02036s window: 311 wercape fait: 0.051s max: 0.540s std dev: 0.02086s window: 321 wercape fait: 0.959 wercape fait: 0.959 wercape fait: 0.959 wercape fait: 0.959</pre>				std de	v: 0.03259s	window:				
average rate: 10.001 nt::0.003 nax: 0.549s std dev: 0.03149s window: 200 average rate: 9.997 nt::0.031s nax: 0.549s std dev: 0.03092s window: 270 nt::0.031s nax: 0.549s std dev: 0.03093s window: 280 average rate: 10.001 nt::0.031s nax: 0.549s std dev: 0.02935s window: 291 nt::0.031s nax: 0.549s std dev: 0.02955s window: 301 average rate: 10.001 nt::0.031s nax: 0.549s std dev: 0.02965s window: 301 average rate: 10.001 nt::0.031s nax: 0.549s std dev: 0.02968s window: 311 average rate: 10.031 nt::0.031s nax: 0.549s std dev: 0.02868s window: 321 average rate: 10.031 average rate: 0.938 average rate: 0.938 average rate: 0.938	average									
ni: 0.033 max: 0.449 std dev: 0.03045 window: 200 werage rate: 0.933 max: 0.449 std dev: 0.03075 window: 270 metri: 0.033 max: 0.549 std dev: 0.03055 window: 270 metri: 0.035 max: 0.549 std dev: 0.02955 window: 280 werage rate: 10.001 metri: 0.031 max: 0.549 std dev: 0.02955 window: 301 werage rate: 10.001 werage rate: 10.001 metri: 0.031 max: 0.549 std dev: 0.02955 window: 301 werage rate: 10.001 metri: 0.031 max: 0.549 std dev: 0.02864 window: 311 werage rate: 10.091 metri: 0.549 std dev: 0.02864 window: 331 werage rate: 0.549			0.549s	std de	v: 0.03202s	window:	250			
average rate: 9.997 nt: 8.0813 nax: 0.5495 std dev: 0.03992s window: 270 average rate: 10.001 mt: 0.0315 nax: 0.5495 std dev: 0.030935 window: 280 nt: 0.0315 nax: 0.5495 std dev: 0.029355 window: 291 average rate: 0.5495 nt: 0.0315 nax: 0.5495 std dev: 0.02985 window: 381 average rate: 10.081 nt: 0.0315 nax: 0.5495 std dev: 0.02986 window: 331 average rate: 0.5495 nt: 0.0315 nax: 0.5495 std dev: 0.02886s window: 331 average rate: 0.5495 std dev: 0.02866 window: 331 average rate: 0.5495 std dev: 0.02866 window: 331 average rate: 0.5495 std dev: 0.02866s window: 331 average rate: 0.5495 std dev: 0.02865 std dev: 0.02865 std de	average		A 5404	and day		ud adams				
n(1: 0.033 max: 0.495 std dev: 0.03032s window: 270 werager ate: 10.001 m(1: 0.013s max: 0.495 std dev: 0.03035s window: 280 werager ate: 10.001 merager ate: 10.001 mic: 0.013s max: 0.549 std dev: 0.02935s window: 291 merager ate: 10.001 mic: 0.013 max: 0.549 std dev: 0.02937s window: 301 werager ate: 10.001 mic: 0.013 max: 0.549 std dev: 0.02895 window: 301 werager ate: 10.001 mic: 0.013 max: 0.549 std dev: 0.02895 window: 311 werager ate: 10.001 mic: 0.013 max: 0.549 std dev: 0.02865 window: 321 werager ate: 0.031 max: 0.549 std dev: 0.02865 window: 321 werager ate: 0.938 max: 0.549 std dev: 0.02865 window: 331 werager ate: 0.938 max: 0.549 std dev: 0.02865 window: 331 werager ate: 0.938 max: 0.549 std dev: 0.02865 window: 331	average			310 00		active.				
Thi: 0.033 max: 0.496 std dev: 0.03035 window: 200 min: 0.033 max: 0.496 std dev: 0.02035 window: 201 werage rate: 0.954 max: 0.496 std dev: 0.02035 window: 201 werage rate: 0.601 min: 0.0315 max: 0.549 std dev: 0.02035 window: 301 werage rate: 10.001 werage rate: 0.031 max: 0.549 std dev: 0.02086 window: 321 werage rate: 0.935 max: 0.549 std dev: 0.02086 window: 331 werage rate: 0.903	or croge		0.549s	std de	v: 0.03092s	window:				
average rate: 10.001 ntit: 0.031 max: 0.5495 std dev: 0.029855 window: 201 average rate: 9.998 ntit: 0.031 max: 0.5495 std dev: 0.029375 window: 301 average rate: 10.001 ntit: 0.031 max: 0.5495 std dev: 0.028965 window: 311 average rate: 10.538 ntit: 0.0315 max: 0.5495 std dev: 0.028465 window: 321 average rate: 0.5495 ntit: 0.0315 max: 0.5495 std dev: 0.028465 window: 331	average									
nin: 0.033 max. 0.496 std dev: 0.02985 window: 201 werzope fait: 0.633 max. 0.496 std dev: 0.02935 window: 381 werzope fait: 0.633 max. 0.549 std dev: 0.02986 window: 311 averzope fait: 10.601 merzope fait: 0.054 max. 0.549 std dev: 0.02886 window: 321 werzope fait: 0.959 merzope fait: 0.959 merzope fait: 0.959			0.549s	std de	v: 0.03039s	window:				
average rate: 9.998 n(t): 8.031 max: 0.5495 std dev: 0.629375 window: 301 average rate: 10.001 n(t): 0.6031 max: 0.5495 std dev: 0.628965 window: 311 average rate: 10.001 average rate: 10.303 n(t): 0.033 max: 0.5495 std dev: 0.628465 window: 321 m(t): 0.035 max: 0.5495 std dev: 0.628465 window: 331 average rate: 9.998	average									
n(i: 8,035 max 0,449 std dev: 0.02037s window: 301 wersopr fat: 1,043 wersopr fat: 1,043 m(i: 8,051 max 0,549 std dev: 0.02096 window: 311 m(i: 8,051 max 0,549 std dev: 0.02846 window: 321 wersopr fat: 9,998 wersopr fat: 9,998 wersopr fat: 9,998	average		0.5495	std de	V: 0.029855	window:				
nth: 0.013s max: 0.549s std dev: 0.02898s window: 311 mnth: 0.013s max: 0.549s std dev: 0.02848s window: 321 averager rate: 9.998 nth: 0.013s max: 0.549s std dev: 0.02848s window: 331 averager rate: 9.998	overage		0.549s	std de	v: 0.02937s	window:				
average rate: 10.001 mt: 8.0631 max: 0.5495 std dev: 0.628485 window: 321 average rate: 5,998 Werage rate: 5,998	average									
Mn: 0.013s max: 0.549s std dev: 0.02848s window: 321 averager cate: 0.908 Mn: 0.013s max: 0.549s std dev: 0.02806s window: 331 averager cate: 0.908				std de	v: 0.02898s	window:				
average rate: 9.998 min: 0.013s nax: 0.549s std dev: 0.02886s window: 331 average rate: 9.998	average									
mln: 0.013s max: 0.549s std dev: 0.02806s window: 331 average rate: 9.998			0.5495	std de	v: 0.02848s	window:				
average rate: 9.998	average		0.5495	std de	1 0.02806s	window:				
	average									
	_			std de	v: 0.02768s					

Figure 1. Ouster OS1-32 LiDR data ROS reception speed

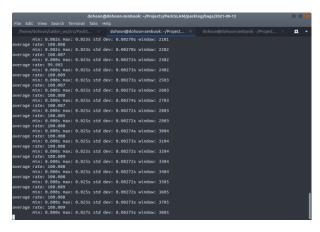


Figure 2. VectorNav VN100 IMU data ROS reception speed

When power is supplied from an external battery, power is distributed to each sensor, and then data acquisition starts from the internal board of each LiDAR and IMU, it can be checked through the ROS (Robot Operating System) platform. Finally, the multi-sensor module was combined with an external battery, a wireless network router, and a module mount, and was boarded next to the investigation robot's arm (Figure 3 and table 1).

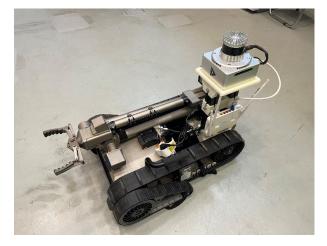


Figure 3. Investigation robot multi-sensor module

Module	Component	feature	
	LiDAR	Ouster OS1-32	
Multi-sensor	IMU	VctorNav VN100	
	NUC	Intel NUC8v-5PNB	
Mount	Mount	3D printing	
D I	Battery	4S, Li0ion 40800mAh	
Power supply	DC adapter	19V, 120W	
Communication device	Wireless network router	ipTime A604	

Table 1. Composition of final multi-sensor module

2.2 SLAM based Indoor Mapping

LOAM is a representative LiDAR-based SLAM algorithm widely used for real-time indoor mapping. To obtain real-time performance, LOAM estimates the posture of 6-DOF (Degree of Freedom) using point cloud data through a software system parallel configuration and simultaneously performs voxel-based mapping (Zhang et al., 2014).

The LIO-SAM (LiDAR Inertial Odometry via Smoothing and Mapping) algorithm, derived from LOAM, includes an IMU constructed using a coefficient graph of a rigidly coupled system of LiDAR and an inertial measurement device. And it has a positioning module, a mapping module that optimizes the point cloud map, and real-time performance, so it is composed of an optimization system suitable for a multi-sensor system(Shan et al., 2020, Figure 4).

In addition, it is a SLAM system that can calculate reliable results by using other sensors even in areas where the positioning reliability of LiDAR and IMU is low by configuring an integrated system between various sensors such as GNSS that provides real coordinates.

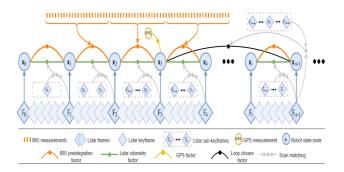


Figure 4. System structure of LIO-SAM factor graph

Based on the LIO-SAM algorithm, this study calculates the ring field(correspondence) and time field (motion distortion) of the 3D point cloud data, which were not available in the multisensor module, and expresses the points as a 5D vector with a total of 5 values including 3D spatial information(Figure 5). The ring field removes false associations between point clouds when searching for correspondence for ICP registration, and the time field is used to correct motion distortion included in the point cloud data.

The additionally calculated information was implemented to be utilized in the multi-sensor module, and a customized SLAM algorithm for the multiple sensor module of the investigation robot called PackSLAM was developed.

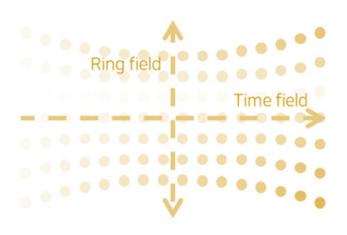


Figure 5. Concept of ring/time field

2.3 Study Area

In this study, multi-sensor module customized SLAM algorithm experiment was performed in the disaster environment building of the Disaster Robotics Centre of the Korea Institute of Robotics & Technology Convergence(KIRO) located in Buk-gu, Pohang-si, Republic of Korea.

The disaster environment building consists of several rooms and two floors, and is designed similarly to the disaster environment because various auxiliary materials are distributed.

For this reason, empirical experiments and research on various robots related to disaster safety are being conducted.

Figure 6 shows the front view of the disaster building, and Figure 7 shows the interior of the disaster building.

In this study, quantitative numerical analysis of indoor space was performed by acquiring 3D point cloud data through PackSLAM on the second floor of a disaster building.



Figure 6. Disaster environment building of disaster robotics centre



Figure 7. Interior view of disaster building

3. RESULTS AND DISCUSSIONS

To evaluate the accuracy of the 3D point cloud data acquired with PackSLAM, LiDAR data acquired from Leica BLK360 was used as reference data (ground truth).

BLK360 is a compact LiDAR that is specialized for portability, so it is widely used for indoor space mapping.

Although it is manufactured in a small size, it has good performance such as wavelength 830 nm, range accuracy 4 mm @ 10 m, and point density 5 mm @ 10 m (Figure. 8 and Table 2).



Figure 8. Leica BLK360

Division	Specification		
Class	Laser class 1		
Wavelength	830 nm		
Coverage accuracy	4 mm @ 10 m, 7 mm @ 20 m		
FOV	300° (V) × 360°(H)		
Precision	- 20 mm @ 10 m(fast) - 10 mm @ 10 m(normal) - 5 mm @ 10 m(high density)		

Table 2. BLK360 specifications

And to compare the accuracy of the PackSLAM developed in this study, data were also acquired with the A-LOAM, LeGO-LOAM, and LIO-SAM algorithms among the LOAM-type algorithms(Toschi et al., 2015).

Figure 9 to 11 show the 3D point clouds acquired by the LOAM series algorithm, and Figure 12 shows the 3D point clouds acquired by PackSLAM.



Figure 9. A-LOAM result

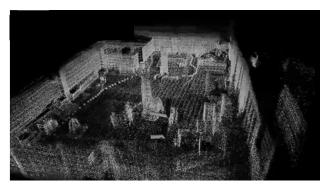


Figure 10. LeGO-LOAM



Figure 11. LIO-SAM

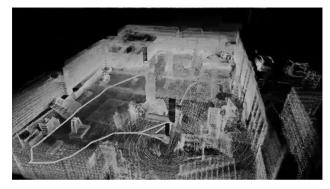


Figure 12. PackSLAM result

Accuracy evaluation was performed by dividing three sections within the second floor of the disaster building and calculating the distance for each section(Figure 13). And quantitative accuracy comparison was performed through BLK360.

This method divides some areas from 3D point cloud data to measure the distance between areas, selects an area with a uniform shape such as a wall, and calculates the distance observation error through the nearest method to evaluate the accuracy.

As a result, obtained by calculating the distance between point clouds, the average value and distribution of distance values can be obtained.

The average distance can be used as a measure to evaluate the accuracy of the geometric structure of the generated entire spatial map.

Figure 14 explains the concept of the accuracy evaluation technique as a figure.

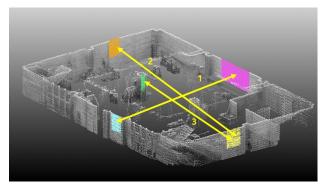


Figure 13. Distance measurement section for accuracy assessment

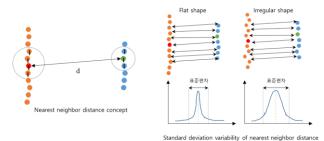


Figure 14. Schematic diagram of accuracy evaluation method

As a result of the above evaluation, PackSLAM showed the lowest error than other SLAM algorithms in all sections, as shown in table 3~5.

A-LOAM and LeGO-LOAM used only LiDAR information, whereas LIO-SAM and PackSLAM algorithms showed better excellent performance because they optimized the graph using the combined system of LiDAR and IMU(Shan et al., 2020).

SLAM type	Avg. distance(m)	St. dev.(m)	Error(m)
A-LOAM	16.351	0.203	-0.115
LeGO-LOAM	16.376	0.195	-0.140
LIO-SAM	16.298	0.043	-0.062
PackSLAM	16.225	0.066	0.011
BLK360	16.236	0.047	-

Table 3. Accuracy assessment at Sect. 1

SLAM type	Avg. distance(m)	St. dev.(m)	Error(m)
A-LOAM	24.589	0.103	0.214
LeGO-LOAM	24.630	0.094	0.173
LIO-SAM	24.821	0.083	-0.018
PackSLAM	24.810	0.083	-0.007
BLK360	24.803	0.053	-

Table 4. Accuracy assessment at Sect. 2

SLAM type	Avg. distance(m)	St. dev.(m)	Error(m)
A-LOAM	15.895	0.076	-0.037
LeGO-LOAM	15.724	0.032	0.134
LIO-SAM	15.874	0.050	-0.016
PackSLAM	15.873	0.051	-0.015
BLK360	15.858	0.009	-

 Table 5. Accuracy assessment at Sect. 3

Looking at the distance accuracy by section, the algorithm that showed the most similar results to BLK360 in section 1 is PackSLAM. PackSLAM showed the lowest error of 0.011 m, whereas LeGO-LOAM showed the highest error of 0.140 m. Similarly for section 2, PackSLAM showed the lowest error of 0.007 m. The algorithm showing the highest error is A-LOAM. It was 0.214 m, which showed a large error of over 20 cm. Finally, in section 3, PackSLAM also showed the lowest error at 0.015 m. The algorithm with the highest error is LeGO-LOAM.

As a result of the distance error analysis for each section, the error was generally high in the order of PackSLAM \rightarrow LIO-

SAM \rightarrow A-LOAM \rightarrow LeGO-LOAM.

In the case of PackSLAM, it was found that the method using the point cloud correlation of the ring field and the distortion correction of the time field can obtain improved results compared to the existing LIO-SAM.

4. CONCLUSION

In this study, the multi-sensor module that can be mounted on an investigation robot was developed. Also, a suggested PackSLAM algorithm customized for the investigation robot, was developed. In conclusion, the PackSLAM suggested in this paper showed observation performance with an error of less than 2 cm and demonstrated the applicability for investigating indoor disaster sites.

In the future, it is expected to quickly and accurately acquire disaster damage information by being deployed to indoor disaster sites that are difficult for investigators to access due to earthquakes or collapse accidents.

However, since this study only performed distance analysis for indoor space, it is necessary to expand field applicability by performing more various experiments based on the accuracy shown in the study results.

In order to be put into actual disaster scenes and used as immediate response equipment, we plan to conduct research by increasing the difficulty of the experiment by constructing a rough terrain rather than a flat area.

ACKNOWLEDGEMENTS

This research outputs are the part of the project "Disaster Field Investigation using Mobile Robot Technology(Π)", which is supported by the NDMI(National Disaster Management research Institute) under the project number NDMI-MA-2022-06-02. The authors would like to acknowledge the financial support of the NDMI.

REFERENCES

Gan, Z., Huang, G., Zhang, J., Liu, X., Shan, C., 2021. The Control System and Application Based on ROS Elevating Fire-Fighting Robot, *In Journal of Physics: Conference Series*, 2029(1), 012004. doi.org/10.1088/174-6596/2028/1/012004.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., Dellaert, F., 2012. iSAM2: Incremental smoothing and mapping using the bayes tree, *The International Journal of Robotics Research*, 31, 217-236. doi.org/10.1177/0278364911430419.

Paravisi, M., H Santos, D., Jorge, V., Heck, G., Gonçalves, L. M., Amory, A., 2019. Unmanned Surface Vehicle Simulator with Realistic Environmental Disturbances, Sensors, 19(5), 1068. doi.org/10.3390/s19051068.

Park, S., Oh, Y., Hong, D., 2017. Disaster Response and Recovery from the Perspective of Robotics, *International Journal of Precision Engineering and Manufacturing*, 18, 1475-1482. doi.org/10.1007/s12541-017-0175-4.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., 2020. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping, *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 5135-5142. doi.org/10.1109/IROS45743.2020.9341176.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics, Cambridge, MA, The MIT Press. doi.org/10.1145/504729.504754.

Toschi, I., Rodriguez-Gonzalvez, P., Remondino, F., Minto, S., Orlandini, S., Fuller, A., 2015. Accuracy Evaluation of a Mobile Mapping System with Advanced Statistical Methods, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5), 245-253. doi.org/10.5194/isprsarchives-XL-5-W4-245-2015.

Yamauchi, Y., Ambe, Y., Nagano, H., Konyo, M., Bando, Y., Ito, E., Arnold, S., Yamazaki, K., Itoyama, K., Okatani, T., Okuno, H. G., & Tadokoro, S., 2022. Development of a Continuum Robot Enhanced with Distributed Sensors for Search and Rescue, *ROBOMECH Journal*, 9(1), 1-13. doi.org/10.1186/s40648-022-00223-x.

Zhang, J., Singh, S., 2014. LOAM: Lidar Odometry and Mapping in Real-time, *Robotics: Science and System*.

Zlot, R., Bosse, M., 2012., Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine, in *The 7th international Conference on Field and Service Robots*, Matsushima, Japan, July. doi.org/10.1007/978-3-642-40686-7_32.