

# Complex urban dataset with multi-level sensors from highly diverse urban environments

The International Journal of  
Robotics Research  
2019, Vol. 38(6) 642–657  
© The Author(s) 2019  
Article reuse guidelines:  
[sagepub.com/journals-permissions](http://sagepub.com/journals-permissions)  
DOI: 10.1177/0278364919843996  
[journals.sagepub.com/home/ijr](http://journals.sagepub.com/home/ijr)



Jinyong Jeong<sup>1</sup> , Younggun Cho<sup>1</sup> , Young-Sik Shin<sup>1</sup> ,  
Hyunchul Roh<sup>2</sup> and Ayoung Kim<sup>1</sup>

## Abstract

*The high diversity of urban environments, at both the inter and intra levels, poses challenges for robotics research. Such challenges include discrepancies in urban features between cities and the deterioration of sensor measurements within a city. With such diversity in consideration, this paper aims to provide Light Detection and Ranging (LiDAR) and image data acquired in complex urban environments. In contrast to existing datasets, the presented dataset encapsulates various complex urban features and addresses the major issues of complex urban areas, such as unreliable and sporadic Global Positioning System (GPS) data, multi-lane roads, complex building structures, and the abundance of highly dynamic objects. This paper provides two types of LiDAR sensor data (2D and 3D) as well as navigation sensor data with commercial-level accuracy and high-level accuracy. In addition, two levels of sensor data are provided for the purpose of assisting in the complete validation of algorithms using consumer-grade sensors. A forward-facing stereo camera was utilized to capture visual images of the environment and the position information of the vehicle that was estimated through simultaneous localization mapping (SLAM) are offered as a baseline. This paper presents 3D map data generated by the SLAM algorithm in the LASer (LAS) format for a wide array of research purposes, and a file player and a data viewer have been made available via the Github webpage to allow researchers to conveniently utilize the data in a Robot Operating System (ROS) environment. The provided file player is capable of sequentially publishing large quantities of data, similar to the rosbag player. The dataset in its entirety can be found at <http://irap.kaist.ac.kr/dataset>.*

## Keywords

Dataset, urban, LiDARs, cameras, SLAM

## 1. Introduction

Recent research on autonomous vehicles relies on navigational and perceptual sensors for the localization, mapping, planning, and maneuvering of the vehicle. It is notable that multi-aspect datasets impose significant meaning for real-world targeted implementation and validation. Recently reported datasets focus on a variety of longer and larger environments obtained from various sensor modalities. This paper proposes to capture and present metropolitan data in a multi-modal fashion, specifically including cameras and light detection and ranging (LiDAR) sensors, two perceptual sensors that are widely utilized in autonomous vehicle research. Cameras mainly acquire visual sensor information of the environment to enhance semantic and metric localization and mapping performance. Despite the flourishing popularity and cost-effectiveness of cameras, visual images

captured by cameras are vulnerable to environmental and illumination variance. Researchers are actively overcoming such limitations through recent advancements in deep-learning-based approaches, widening the visual application in varying environments. However, even a single instance of perceptual aliasing could be problematic for autonomous vehicles. Unlike cameras, LiDARs measure structural information from range measurements and reflectance intensity.

<sup>1</sup>Department of Civil and Environmental Engineering, KAIST, Daejeon, South Korea

<sup>2</sup>Dyphi Co., Daejeon, South Korea

### Corresponding author:

Ayoung Kim, Department of Civil and Environmental Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, 34141 Daejeon, South Korea.  
Email: ayoungk@kaist.ac.kr

Using LiDARs alleviates issues related to changes in weather and illumination, though subtle distortion may occur due to fog and rain.

In order to complementarily exploit the characteristics of sensors, many researchers favor the combination of multiple sensors. Examples of datasets that involve the usage of the aforementioned sensors include the following. Camera image datasets (Cordts et al., 2016; Leung et al., 2011) provide images and semantic labels tagged with accurate camera positions for utilization in visual odometry, vehicle detection, and semantic segmentation. Event camera datasets involve recently introduced event cameras (Mueggler et al., 2017; Zhu et al., 2018) for the purpose of overcoming illumination and motion blur issues caused by conventional optical images. Datasets based on LiDAR sensors (Pomerleau et al., 2012) were widely adopted in three dimensional (3D) point cloud object detection, LiDAR odometry, and 3D mapping algorithms. In recent years, researchers have reported datasets for visual inertial research (Burri et al., 2016; Miller et al., 2018) that accurately estimates the position of a camera by employing Inertial Measurement Unit (IMU) sensors. Whereas the aforementioned datasets focus on a single sensor modality, multi-sensor datasets (Huang et al., 2010, 2018; Maddern et al., 2017) provide data related to cameras, LiDAR, and navigation sensors to enable users to select data according to the purpose in hand.

In the aforementioned datasets, however, the data were mostly collected from a single city and its suburbs whereas the captured diversity is rather intracity. Compared with the existing urban datasets, our dataset is most distinguishable by the level of complexity representing the diverse aspect of the urban area. Kilometers of urban canyon, multi-lane wide road, high-rise buildings, densely cluttered residential area collected from four different cities are the unique features of the provided dataset. As the complexity increases to metropolitan level, new critical challenges appear (e.g., sudden sensor data loss from a severe Global Positioning System (GPS) blackout, occlusion, and large traffic). We provide data from multiple cities including a complex metropolitan area, which was not covered by existing datasets.

In this dataset, typical LiDAR sensor data, 2D and 3D LiDAR, are provided. Two 3D LiDARs were mounted on the left- and right-hand sides of the vehicle at an angle of approximately 45° to maximize data acquisition coverage. Two 2D LiDARs were mounted on the rear and front of the vehicle facing downwards and upwards, respectively, to effectively acquire information on buildings and roads. The data of the Fiber Optic Gyro (FOG), IMU, GPS, and encoder sensors, which are navigation sensors for the purpose of estimating the pose of the vehicle, are provided in the CSV file format. The vehicle poses were estimated using the Simultaneous Localization and Mapping (SLAM) algorithm with the LiDAR data and data from navigation sensors such as GPS, FOG, and wheel encoders. The set of pose data was stored at a rate of approximately 100 Hz and was provided as a baseline. Using the estimated baseline, a

reconstructed 3D map can be viewed through WebGL on the website, and 3D map data are also provided in LASer (LAS) format.

Extending from our previous publication (Jeong et al., 2018), this study incorporates the addition of stereo camera data to support vision based robotics research. A summary of the key contributions of this paper are as follows:

- provides data for a variety of environments such as downtown area, apartment complexes, and underground parking lot;
- provides a baseline via a SLAM algorithm using highly accurate navigational sensors and semi-automatic loop-closure process; 到底能不能当作 Ground Truth 呢？
- provides sensor data with two levels of accuracy (economic sensors with consumer-grade accuracy and costly high-precision sensors);
- provides 3D point cloud data in LAS format reconstructed using SLAM algorithm and associated data viewer via WebGL;
- provides development tools for publishing and viewing data in the Robot Operating System (ROS) environment to support the general robotics community.

The structure of this paper is as follows. Section 2 summarizes the currently open datasets and compares them with the dataset presented in this paper. Section 3 describes the configuration of the sensor system for obtaining data and the calibration process for each sensor. Section 4 explains the structure of the dataset presented in this paper. Finally, Section 6 discusses the conclusions and future direction of this study.

## 2. Related works

Diverse capture of visual and structural scenes were released via various datasets and utilized in many computer vision studies and implementations. Such datasets performed as benchmark baselines and enabled the advancement of research. The datasets provided the ground truth of robot position and image labels, enabling many researchers around the world to research without system and data limitations. Table 1 summarizes recent datasets for autonomous navigation and 3D mapping.

The New College Dataset (Smith et al., 2009) provides data from a campus and a park that were obtained using a Segway robotic platform. A LiDAR was mounted on the side of the Segway to obtain distance measurements, providing both stereo and omnidirectional images. In response to the increasing interest in autonomous vehicles, the DARPA challenge has focused on a variety of topics since the early 2000s to test the development of autonomous driving technology. The DARPA dataset (Huang et al., 2010) presents the data acquired during the DARPA Urban Challenge in 2007 for research purposes. The data were acquired using a Massachusetts Institute of Technology

**Table 1.** Comparison with previous datasets on data-acquisition environment, sensor type, and ground-truth estimation method. The dataset name, target environment, ground-truth pose (GT pose) availability, and sensors are summarized.

Dataset	Environment	GT pose	Sensor					Reference
			Image	LiDAR	GPS	IMU	Encoder	
New college	Campus		✓	✓	✓	✓	✓	Smith et al. (2009)
DARPA	Urban	INS	✓	✓				Huang et al. (2010)
Marulan	Open area	INS	✓	✓	✓	✓	✓	Peynot et al. (2010)
Ford campus	Urban	INS	✓	✓	✓	✓	✓	Pandey et al. (2011)
KITTI	Urban	INS	✓	✓	✓	✓	✓	Geiger et al. (2013)
MIT stata	Building	SLAM	✓	✓	✓	✓	✓	Fallon et al. (2013)
Malaga	Urban		✓	✓	✓	✓		Blanco-Claraco et al. (2014)
MPO	Urban		✓	✓	✓			Jung et al. (2016)
NCLT	Campus	SLAM	✓	✓	✓	✓	✓	Carlevaris-Bianco et al. (2016)
Oxford	Urban	INS	✓	✓	✓	✓	✓	Maddern et al. (2017)
ApolloScape	Urban	INS	✓	✓	✓	✓	✓	Huang et al. (2018)
Complex urban (proposed dataset)	Urban	SLAM	✓	✓	✓	✓	✓	

这个表格把本文计算出的baseline作为 ground truth

(MIT) autonomous car, Talos, which traveled for approximately 90 km with a precise navigation system, 5 cameras, 12 2D LiDARs, and a single 3D LiDAR. With the development of autonomous navigation technology, the ways in which sensors are influenced by various environments have become a major issue, and a multitude of sensors have been developed to overcome such limitations. The Marulan dataset (Peynot et al., 2010) artificially created environments such as dust, fog, and rain, and data were obtained in various environments using LiDARs, a camera, a Radio Detection and Ranging (RADAR) sensor, and an infrared (IR) camera. With further progress being made regarding the localization of autonomous vehicles, studies have utilized High Definition (HD) maps composed of 3D point cloud for pose estimations of greater accuracy. The Ford dataset (Pandey et al., 2011) contains 3D LiDAR and omnidirectional camera data in addition to data from push-bloom type 2D LiDARs suitable for 3D mapping. This dataset incorporates both consumer-level IMU data and high-precision Inertial Navigation System (INS) data, and consists of large and small loop closures for SLAM algorithm research.

With an increasing number of studies focused on localization and SLAM research, a method of comparing the performance of developed algorithms became a necessity. The KITTI dataset (Geiger et al., 2013) provides such a method using common data. Online benchmarks for visual odometry, image tracking, and semantic segmentation are available on the website. This dataset is the most widely used dataset in autonomous vehicle and computer vision research. One of the most important purposes of using open datasets is for the ground-truth information. The MIT Stata dataset (Fallon et al., 2013) and the NCLT dataset (Carlevaris-Bianco et al., 2016) provide the ground-truth position of a robot estimated using SLAM algorithms. The NCLT dataset (Carlevaris-Bianco et al., 2016) and the Oxford dataset (Maddern et al., 2017) acquired data over

an extensive period of time at similar locations, including changes in data over time. Such data can be used to evaluate the way in which an algorithm is affected by changes in data over time. The Malaga dataset (Blanco-Claraco et al., 2014), which contains data for 36.8 km of running, can be used to evaluate the performance of algorithms in various environments. The ApolloScape dataset (Huang et al., 2018), which was released recently, provides an RGB video and the corresponding dense 3D point cloud. In addition, the ApolloScape dataset presents labeled images using an efficient labeling pipeline with high-precision 3D point clouds and provides lane information that was extracted based on color and shape.

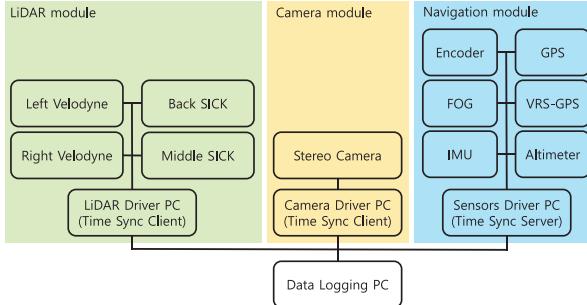
### 3. System overview

This section describes the configuration of the hardware platform and the calibration process of the LiDARs, cameras, and odometry.

#### 3.1. System configuration

Figure 1 shows the block diagram of the sensor hardware. The navigation sensor data were small in size and the acquisition speed was higher compared with cameras and LiDARs. Three PCs were used to collect the data without delays. Each PC received data from the navigation sensors, camera images, and LiDAR data, respectively. The system clocks of the three PCs were periodically synchronized using the Chrony library to ensure sub-millisecond clock calibration (Curnow, 2014). Each PC used an i7 processor, 512 GB solid state drive (SSD), and 32 GB DDR4 memory. The sensor drivers and logger were developed on the Ubuntu operating system.

The objectives of this sensor system, illustrated in Figure 2, were to acquire navigation sensor data with different accuracy level, diverse types of LiDAR sensor data,



**Fig. 1.** Block diagram of the sensor system. Three PCs were used to prevent data acquisition delay, and another PC saved the data in a local drive. All PC times were carefully synchronized using Chrony (Curnow, 2014), a NTP library .

and stereo images that can be used for various purposes. The presented dataset simultaneously provides relatively less-accurate data obtained from inexpensive sensors and highly accurate data obtained from expensive sensors to estimate the attitude of the vehicle. The dataset also provides two types of LiDAR data: 3D LiDAR, which collects data in 3D at low rates, and 2D LiDAR, which collects data in 2D at high rates. In addition, stereo images that are usable for visual odometry, object detection, and 3D reconstruction are provided. The details of sensor configuration are summarized in Figure 3 and Table 2.

A total of four LiDAR sensors (two 3D and two 2D) were mounted on the vehicle and were used to acquire distance measurements. 3D LiDAR sensors (**VLP-16** by Velodyne) simultaneously measure 16 range observations through 16 distance measuring sensors (16 channels) in the vertical direction while rotating horizontally, and are typically capable of obtaining data at rotation frequencies of 10–20 Hz. In this dataset, data was collected at a rate of 10 Hz. The 3D LiDAR sensors were mounted on the rear of the vehicle, tilted approximately 45° to the left and right. This configuration was intended to maximize data collection coverage as the vehicle moves in the forward direction. In addition, the VLP-16 can use Pulse Per Second (PPS) signals and GPS data to synchronize its internal counter.



**Fig. 2.** The mapping vehicle equipped with navigation sensors (red boxes) for estimating the position of the vehicle, as well as LiDAR and a front stereo camera (yellow boxes) for obtaining data from the surrounding environment .

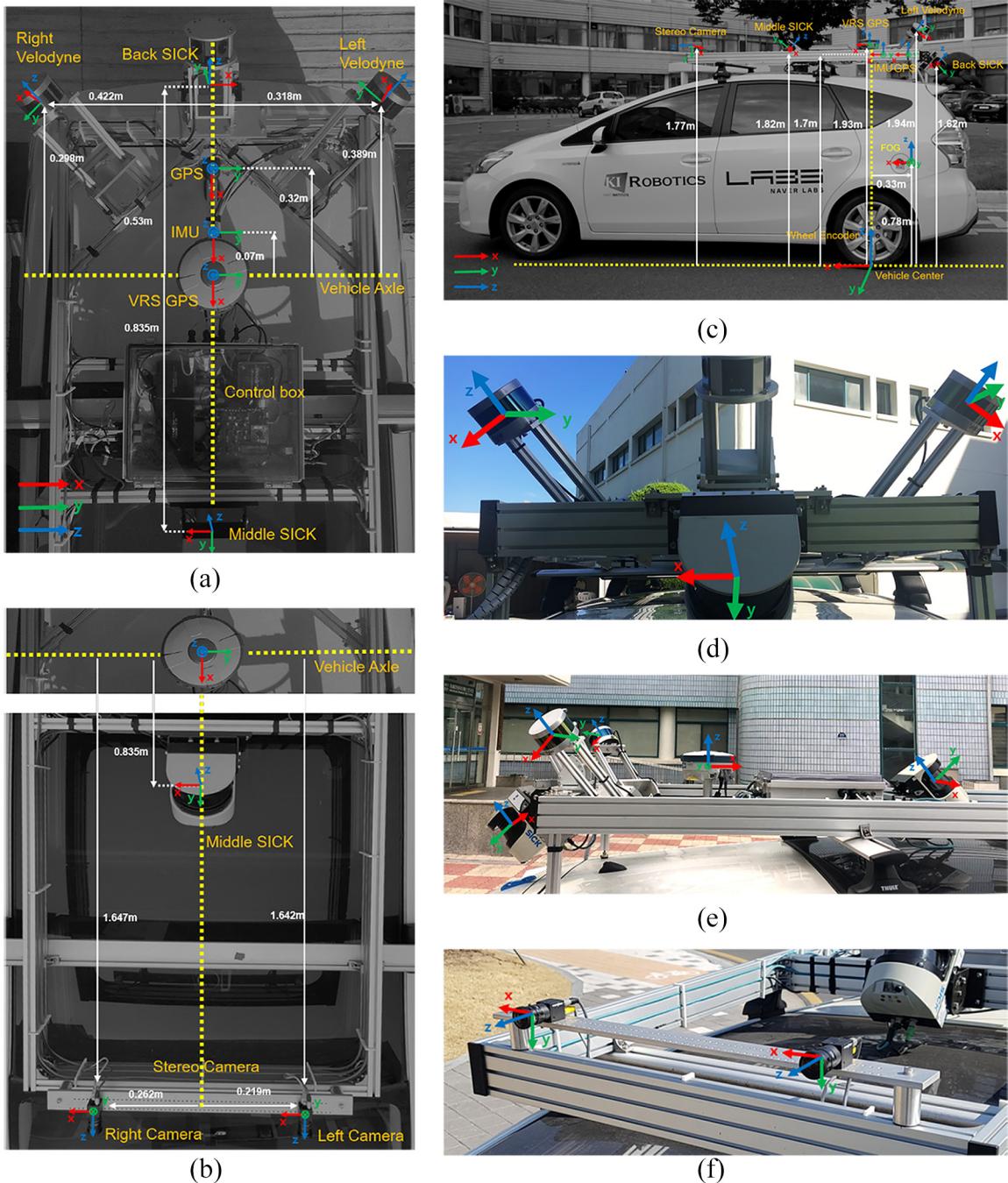
This tight synchronization might be helpful in case when only two-sensor synchronization is sufficient. However, when more complicated synchronization is needed for multiple sensors and logging PCs, we found that synchronizing the system clocks of the multiple logging PCs provides a more reliable solution to integrate the time stamps of the sensor data.

The 2D LiDAR sensors installed in the system are SICK LMS511 model and measure single-channel data. The LMS511 has a data-acquisition rate of up to 100 Hz and has greater accuracy than the VLP-16. On the other hand, the amount of data are relatively small as the LMS511 measures a single distance at a time. The 2D LiDAR sensors were installed facing forward and backward. The middle, forward-facing 2D LiDAR is suitable for obtaining data on the structures of buildings and infrastructure of the road, whereas the rear-facing 2D LiDAR is appropriate for acquiring information on the roads.

A stereo camera was utilized to obtain the front-view images. An external trigger was used for precise synchronization of the cameras, and the images were acquired at a speed of 10 Hz. The images were stored as loss-less PNG format as unrectified 8-bit Bayer pattern images.

**Table 2.** Sensors used in the sensor system (H: horizontal, V: vertical, CH: channel).

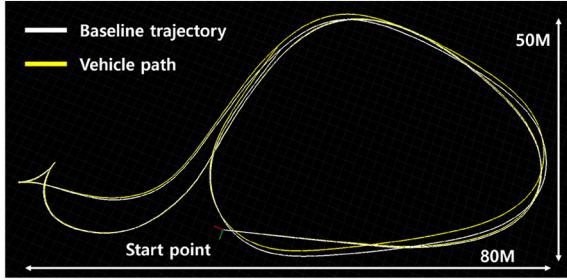
Type	Manufacturer	Model	Description	N	Frequency (Hz)	Specification
<b>3D LiDAR</b>	Velodyne	<b>VLP-16</b>	16 CH LiDAR (360° FOV)	2	10	100 m (range)
<b>2D LiDAR</b>	SICK	LMS-511	1 CH LiDAR (190° FOV)	2	100	80 m (range)
<b>Camera</b>	FLIR	FL3-U3-20E4C-C	Global shutter color camera	2	10	1,280 × 560 (resolution)
<b>GPS</b>	U-Blox	EVK-7P	Consumer-level GPS	1	10	2.5 m (accuracy)
<b>VRS-GPS</b>	SOKKIA	GRX 2	VRS-RTK GPS	1	1	H: 10 mm, V: 15 mm (accuracy)
<b>Three-axis FOG</b>	KVH	DSP-1760	Fiber optics gyro (3 axis)	1	1,000	0.05°/h (bias)
<b>IMU</b>	Xsens	MTi-300	Consumer-level AHRS	1	200	10°/h (bias)
<b>Encoder</b>	RLS	LM13	Magnetic rotary encoder	2	100	4,096 (resolution)
<b>Altimeter</b>	Withrobot	myPressure	Altimeter sensor	1	10	0.01hPa (resolution)



**Fig. 3.** Hardware sensor configuration. Top views of the (a) the rear section and (b) the front section, and (c) side view of entire sensor system with coordinate information. Sensor coordinates of each sensor are depicted on each subfigure, whereas the coordinate frame is color-coded with red ( $x$ ), green ( $y$ ), and blue ( $z$ ) axes, respectively. (d) Two 3D LiDARs are tilted by approximately 45°. (e) The middle 2D LiDAR faces upward to obtain building information and the rear 2D LiDAR faces downward towards the road to acquire road marking information. (f) The stereo camera faces the front of the vehicle.

The navigation sensors in the system consist of sensors with two levels of accuracy. The sensors for estimating the attitude of the vehicle are composed of IMU and three-axis FOG. The three-axis FOG provides highly precise rotation measurements and is used to calculate the baseline of the vehicle in this paper. On the other hand, IMU measures diverse data such as geomagnetism data, acceleration, and

angular velocity, but is less accurate than FOG. GPS is used to estimate the global position of the vehicle. The sensor system of this study uses two levels of GPS sensors. The Virtual Reference Station (VRS)-GPS obtains carrier data by communicating with the reference station and corrects the GPS data to calculate the global position with an accuracy within 10 cm. The commercial-grade GPS does not



**Fig. 4.** Encoder parameter calibration. The white line indicates the baseline trajectory used in the calibration and the yellow line represents the vehicle path calculated using optimized encoder parameters. The encoder parameters are optimized against this baseline trajectory.

have such correction processes, resulting in errors of several meters. The accuracy of GPS measurements depends on the number of satellites and the surrounding environment. In the case of complex cities with several high-rise buildings, the VRS-GPS also has difficulty in estimating position. The dataset of this paper presents data of a wheel encoder to accurately determine the travel distance of the vehicle and altimeter sensors to measure relative altitude changes.

Figure 3 shows the configuration of the sensors installed in the vehicle. The center of the sensor system is the ground surface with zero height at the center position of the rear wheel axle of the vehicle. The extrinsic parameters of all sensors in the vehicle coordinate system were calculated through a calibration process. Figure 3(a) and (b) show the top view of the rear and front sections of the sensor system, respectively, and Figure 3(c) shows the side view of the sensor system. Each sensor has individual local coordinate systems, and the red, green, and blue arrows represent the  $x$ ,  $y$ , and  $z$  reference axes for each local coordinate system. Most sensors were mounted outside the vehicle, with the exception of the three-axis FOG and the encoder. The three-axis FOG was mounted inside the vehicle, and a magnetic wheel encoder used for travel distance estimation was installed in the rear wheel of the vehicle. Additional details are listed in Table 2.

### 3.2. Odometry calibration

For accurate odometry calculation, calibration of the encoder parameters was performed using high-precision sensors: VRS-GPS and FOG. To ensure the accuracy of the VRS-GPS, a sensor used to estimate precise global position, calibration was performed in a wide and flat open space. Both wheel encoders that were attached to the rear wheels of the vehicle have a resolution of 4,096 pulses per round, and the forward kinematics of the vehicle is calculated using three parameters  $\mathbf{w} = (d_l, d_r, w_b)$ : the left and right rear wheel diameters, and the wheel base between the two rear wheels (Roh et al., 2017). In order to obtain the relative measurement of the vehicle using the VRS-GPS, the 2D pose graph

**Table 3.** Coordinate frame subscript.

Subscript	Description
$v$	Vehicle frame
$l$	Left 3D LiDAR (LiDAR reference frame)
$r$	Right 3D LiDAR
$m$	Forward looking 2D LiDAR in the middle
$b$	Backward looking 2D LiDAR in the rear
$L$	Left stereo camera
$R$	Right stereo camera

( $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ ,  $\mathbf{x}_i = (x_i, y_i, \theta_i)$ ) was constructed using data from the two sensors when accurate measurement values of the VRS-GPS were obtained. The parameter of the encoder was estimated through optimization using the relative transformation  $\mathbf{z}_i = \mathbf{x}_{i+1} \ominus \mathbf{x}_i$  on the vehicle coordinate system and forward motion from the kinematics  $\mathbf{k}_i(\mathbf{w})$ :

$$\mathbf{k}_i(\mathbf{w}) = \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta \theta_i \end{bmatrix} = \begin{bmatrix} l_{avr} \cos(\Delta \theta_i) \\ l_{avr} \sin(\Delta \theta_i) \\ l_{diff}/w_b \end{bmatrix} \quad (1)$$

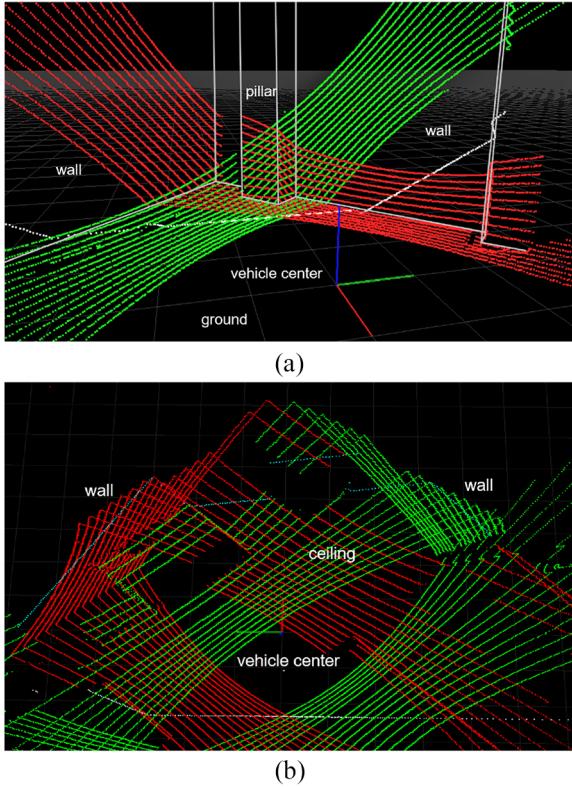
The kinematics of the vehicle is a function of average distance ( $l_{avr} = (\frac{c_l}{4,096} \pi d_l + \frac{c_r}{4,096} \pi d_r)/2$ ) and the differential distance ( $l_{diff} = \frac{c_l}{4,096} \pi d_l - \frac{c_r}{4,096} \pi d_r$ ) between the left and right rear wheels. Here  $c_l$  and  $c_r$  are the wheel encoder counts for each wheel,  $d_l$  and  $d_r$  are the diameters of each wheel, and  $w_b$  is the wheel base. The expression of the objective function is as follows:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i \| \mathbf{z}_i \ominus \mathbf{k}_i(\mathbf{w}) \|_{\Omega_i} \quad (2)$$

where  $\ominus$  is the inverse motion operator (Smith et al., 1990), and  $\Omega_i$  represents the uncertainty of the VRS-GPS and the FOG. The parameters calculated using the FOG and the VRS-GPS are used as initial parameter values. Using the calculated initial parameters, the encoder parameters were optimized further through comparison with a baseline trajectory estimated via SLAM. Figure 4 illustrates the calibration process of the encoder parameters using the baseline. The calculated parameters are provided in the EncoderParameter.txt file in the calibration folder.

### 3.3. LiDAR calibration

LiDAR calibration is also required for relative coordinate transformations of each LiDAR sensor. Table 3 summarizes the subscript of each coordinate frame used in this paper. This study mainly relies on the 3D LiDAR point cloud in the calibration phase as there is no guarantee of direct overlap between two 2D LiDARs. In order to estimate the extrinsic parameter of LiDAR sensors, the following three processes were performed.



**Fig. 5.** The point clouds obtained from the calibration process. The corners of the building are used to distinguish point clouds on planes that are perpendicular to each other. The red and green point clouds are data obtained from the two 3D LiDARs. The white and blue point clouds are data from the two 2D LiDARs. The red, green, and blue lines that are perpendicular to each other represent the reference coordinate system of the vehicle. (a) Front view of LiDAR data. (b) Top view of LiDAR data.

**3.3.1. 3D LiDAR to 3D LiDAR.** The left 3D LiDAR was used as the sensor associated to the reference frame among the four LiDARs installed in the vehicle throughout the entire LiDAR calibration process. First, estimation of the relative position of the 3D LiDARs is required. In contrast to line-by-line acquisition of 2D LiDAR sensors, 3D LiDARs acquire 3D data of space. Figure 5 shows the LiDAR calibration process. The green and red points are the left and right 3D LiDAR data, respectively. As shown in the figure, for 3D LiDAR calibration, the overlapping data of the two 3D sensors was utilized and relative rotation ( $\mathbf{R}_{lr}$ ) and translation ( $\mathbf{t}_{lr}$ ) were calculated by minimizing the error between the projected points using the GICP algorithm (Segal et al., 2009):

$$\mathbf{R}_{lr}^*, \mathbf{t}_{lr}^* = \underset{\mathbf{R}_{lr}, \mathbf{t}_{lr}}{\operatorname{argmin}} \sum_i \{ \mathbf{p}_{l,i} - (\mathbf{R}_{lr} \cdot \mathbf{p}_{r,i} + \mathbf{t}_{lr}) \} \quad (3)$$

Computed transformation maps the  $i^{th}$  right LiDAR point cloud data ( $\mathbf{p}_{r,i}$ ) to the corresponding left LiDAR point cloud data ( $\mathbf{p}_{l,i}$ ):

**Table 4.** Example of sensor transformation. Positional data are in meters and rotational data are in degrees.

Type	Description [ $x, y, z$ , roll, pitch, yaw]
$\mathbf{P}_{vl}$	Vehicle with respect to left 3D LiDAR [−0.31, 0.38, 1.94, 1.45°, 44.75°, 137.39°]
$\mathbf{P}_{vr}$	Vehicle with respect to right 3D LiDAR [−0.29, −0.42, 1.95, 179.63°, 135.43°, 45.53°]
$\mathbf{P}_{vb}$	Vehicle with respect to rear 2D LiDAR [−0.53, −0.009, 1.62, 44.72°, −179.08°, 90.44°]
$\mathbf{P}_{vm}$	Vehicle with respect to middle 2D LiDAR [0.83, −0.01, 1.82, 143.93°, 0.29°, 90.90°]
$\mathbf{P}_{vL}$	Vehicle with respect to left stereo camera [1.64, 0.21, 1.78, −94.09°, 0.6°, −89.94°]
$\mathbf{P}_{vR}$	Vehicle with respect to right stereo camera [1.64, −0.26, 1.77, −94.36°, 0.96°, −89.47°]

**3.3.2. 3D LiDAR to the vehicle.** Using the computed transformation between the 3D LiDAR sensors, the data of the two sensors were aligned to generate the combined point cloud. As the calibration process is performed on a planar ground surface, the height of the point cloud on the ground should be zero given the center of the vehicle is on the ground surface. Thus, by adjusting the height of the merged point cloud above the ground to zero, it is possible to compute the roll, pitch, and  $z$  values of the two 3D LiDAR sensors in the vehicle reference coordinate system. First, the points on the ground were extracted using the random sample consensus (RANSAC) algorithm by fitting the plane. It is possible to estimate the roll, pitch, and  $z$  values of the 3D LiDAR sensors by adjusting  $z$  value to zero as the least square problem using singular value decomposition (SVD). This procedure alleviates the relative height difference between the vehicle center and the two 3D LiDARs.

However, adjustments of the relative position ( $x, y$ ) and the angular (yaw) difference are required. The calibration was performed by operating the vehicle in a round trip and examining common objects during the trip, comparing scenes from two reversed viewpoints. For this process, VRS-GPS and FOG, which are high-precision sensors, were employed. This part of the calibration process was performed in an open space to ensure reliable GPS reception. During the round trip, the precise position of the vehicle was obtained by mainly relying on the VRS-GPS. 3D point clouds generated from in/outbound trips were examined by focusing on the common objects along the trip and securing alignment.

**3.3.3. 3D LiDAR to 2D LiDAR.** In the previous two steps, the transformation of the two 3D LiDARs was estimated in the vehicle coordinate system. The final step involves the calculation of the transformation of the 2D LiDARs ( $\mathbf{P}_{vb}, \mathbf{P}_{vm}$ ) with respect to the vehicle center. A direct point-based comparison is challenging as there is no overlap between the two 2D LiDARs. Therefore, the transformation

of the 2D LiDARs in the vehicle coordinate system was estimated by identifying the relationship between the 3D LiDARs and the 2D LiDARs. The rear 2D LiDAR data overlaps with the 3D point cloud on the backside, and the data of the 2D LiDAR in the middle section overlaps with the 3D point cloud above the vehicle. The relationship between the 2D LiDAR and 3D LiDAR data are estimated by performing point-to-plane alignment using complex structures around the vehicle:

$$\begin{aligned}\mathbf{P}_{vb} &= \mathbf{P}_{vl} \times \mathbf{P}_{bl}^{-1} \\ \mathbf{P}_{vm} &= \mathbf{P}_{vl} \times \mathbf{P}_{ml}^{-1}\end{aligned}\quad (4)$$

The calibration result is provided with the datasets in both Euler format and  $SE(3)$  format. Table 4 lists an example of the calculated transformation values of each LiDAR sensor.

### 3.4. Stereo camera calibration

The calibration process of the stereo camera is divided into two parts. First is the stereo calibration for the calculation of the internal parameters of the cameras and the relationship between the two cameras. The second involves extrinsic calibration for the calculation of the transformation of the two cameras in the vehicle center coordinate system.

**3.4.1. Stereo intrinsic calibration.** The stereo calibration includes the calculation of internal parameters for both cameras and the estimation of the transformation between the cameras. Calibration of the cameras was performed using a checkboard (Zhang, 2000), whereas the internal parameters and the transformation between the two cameras were simultaneously calculated. The average reprojection error calculated using the estimated parameters was approximately 0.2 pixels. The computed information can be used for the rectification of the stereo image (Figure 6). The calculated parameters are provided in the calibration folder in yaml format (`left.yaml`, `right.yaml`) for ROS and mat format (`stereoParmas.mat`) for MATLAB.

**3.4.2. Stereo extrinsic calibration.** Stereo extrinsic calibration yields the position of the stereo camera in the vehicle coordinate system. The stereo cameras, which face forward at the front of the vehicle, possess few overlaps with the LiDAR data, and thus general calibration methods are infeasible. Hence, the SLAM results are used to solve calibration problems of non-overlapping sensor systems (Scott et al., 2015). In an environment where GPS accuracy is guaranteed, the path of the vehicle obtained through the SLAM algorithm using LiDAR data has a high level of accuracy and is close to the ground truth. Extrinsic parameters ( $\mathbf{P}_{vL}$ ) are calculated by projecting the global point clouds that were reconstructed through the vehicle path onto each image:



**Fig. 6.** A superimposed image of rectified stereo images. The red and blue road signs are the image data of the left camera and right camera, respectively.

$$\mathbf{p}_i^{(k)} = \mathbf{K} \mathbf{P}_{vL}^{-1} \mathbf{P}_g^{(k)-1} \mathbf{P}_i \quad (5)$$

In (5),  $\mathbf{P}_i$  is the  $i^{th}$  global point cloud,  $\mathbf{P}_g^{(k)}$  is the global position of the vehicle corresponding to  $k^{th}$  image,  $\mathbf{P}_{vL}$  is the extrinsic parameter of the left stereo camera, and  $\mathbf{K}$  is the intrinsic parameter of the camera. Here  $\mathbf{p}_i^{(k)}$  is the projected  $i^{th}$  point on the  $k^{th}$  image. The normalized information distance (NID) between the gray image value of the projected point and the intensity values of the LiDAR points for the multiple images is used as the cost function:

$$NID(X, Y) = \frac{2H(X, Y) - H(X) - H(Y)}{H(X, Y)} \quad (6)$$

where

$$\begin{aligned}H(X) &= - \sum_{x \in X} P_x \log(P_x) \\ H(Y) &= - \sum_{y \in Y} P_y \log(P_y) \\ H(X, Y) &= - \sum_{\substack{x \in X \\ y \in Y}} P_{xy} \log(P_{xy})\end{aligned}$$

The extrinsic parameter value can be estimated by calculating the parameter that minimizes the NID value through optimization:

$$\hat{\mathbf{P}}_{vL} = \underset{\mathbf{P}_{vL}}{\operatorname{argmin}} NID(X, Y; \mathbf{P}_{vL}) \quad (7)$$

For more information on the process, please refer to Jeong et al. (2019).

## 4. Dataset description

This section describes the format and structure of the data, the baseline trajectory, and the development tools.

**Table 5.** Sequence lists of dataset (ASN: average satellite number, N: navigation sensor, L: 2D and 3D LiDAR sensors, S: stereo camera). Navigation sensors includes GPS, VRS-GPS, FOG, IMU, an encoder, and an altimeter sensor.

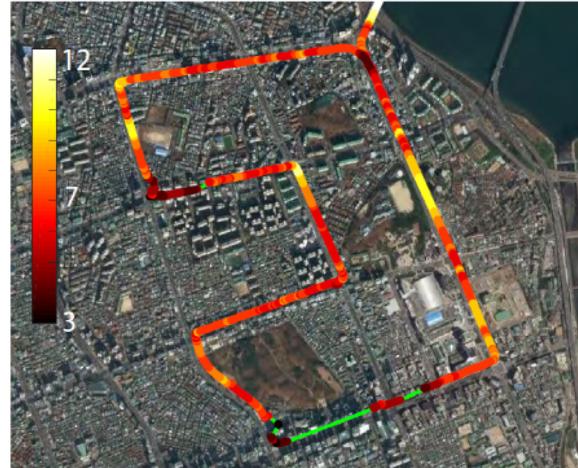
Sequence Name	Location	Description	GPS ASN	Complexity	Sensor	Path length
Urban00	Gangnam, Seoul	Metropolitan area	7.7	★★★	N, L	10.54 km
Urban01	Gangnam, Seoul	Metropolitan area	5.6	★★★	N, L	11.72 km
Urban02	Gangnam, Seoul	Residential area	4.0	★★★★★	N, L	4.19 km
Urban03	Gangnam, Seoul	Residential area	5.0	★★★★★	N, L	3.06 km
Urban04	Pangyo	Metropolitan area	7.5	★★★	N, L	16.35 km
Urban05	Daejeon	Apartment complex	7.5	★★	N, L	2.72 km
Urban06	Daejeon	Highway	11.0	★★	N, L, S	20.82 km
Urban07	Daejeon	Apartment complex	8.6	★★	N, L, S	2.549 km
Urban08	Daejeon	Residential area	8.1	★★	N, L, S	1.56 km
Urban09	Pangyo	Metropolitan area	8.2	★★★	N, L, S	15.7 km
Urban10	Dongtan	Metropolitan area	7.2	★★★	N, L, S	14.67 km
Urban11	Suwon to Seoul	Highway	9.4	★★	N, L, S	17.33 km
Urban12	Namsan	Tunnel	7.8	★★	N, L, S	12.07 km
Urban13	Kwanghwa-moon	Metropolitan area	10.3	★★★	N, L, S	2.36 km
Urban14	Gangnam	Metropolitan area	7.4	★★★	N, L, S	8.2 km
Urban15	Gangnam	Residential area	6.2	★★★	N, L, S	5.43 km
Urban16	Yeouido	Metropolitan area	8.5	★★★★★	N, L, S	21.84 km
Urban17	Mapo-bridge	Bridge	10.7	★★	N, L, S	10.32 km
Campus00	Daejeon	KAIST campus	9.8	★★	N, L	9.56 km

#### 4.1. Experimental environment

This dataset encompasses the characteristics of diverse urban environments from wide roads with multiple lanes (10 lanes or greater) to narrow roads with high-rise buildings. Table 5 lists the provided sequences, showing the average number of GPS satellites, the type of sensor, and the length of the vehicle path for each sequence. Figure 7 shows the path of the urban00 data among the provided sequences. The color of the circle that overlaps with the path indicates the number of satellites received at each location: the brighter the color of the circle, the greater the number of received satellite data. In general, GNSS-based commercial mobile mapping systems require more than 10 satellites for accurate localization. However, such satellite reception is hardly achievable in the urban environments, as indicated by the average satellite numbers in Table 5. Sample scenes captured by cameras and LiDARs are provided in Figures 12, 13, and 14.

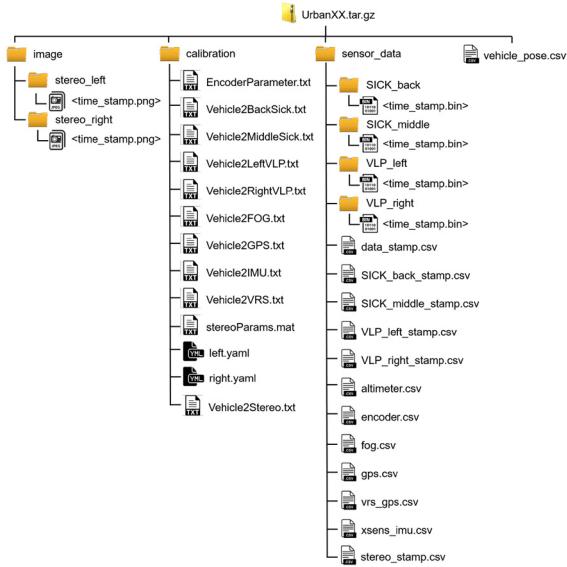
#### 4.2. Data format

Figure 8 depicts the structure of the provided data. Data were collected in the ROS environment using the ROS timestamp. All data are provided in a single compressed file with the tar.gz extension. Calibration is performed upon the collection of data for accurate calibration values. The calibration values for each sequence are stored in the calibration folder whereas all sensor data are located in the sensor\_data folder. The image data are provided separately from the sensor data and the image data should be posed as shown in Figure 8 to use the provided file player.



**Fig. 7.** Data collection route (Urban00) in a metropolitan area overlaid with the number of satellites for computing global position. The color of the circle represents the number of satellites; a brighter color indicates a larger number of satellites received the data. The sections without circles are areas where no satellite data were obtained owing to the complexity of the urban environment.

1. **3D LiDAR data:** The Velodyne VLP-16, which is used as the 3D LiDAR sensor, is a sensor with 16 channels and provides measurement data in a per-packet basis. For sensor stability, data were acquired at a rotational speed of 10 Hz. The timestamp at the end of each rotation was used as the timestamps of the packets. The sensor measurements were stored in the folders VLP\_left and VLP\_right under the name of <time\_stamp.bin>. For intuitive use



**Fig. 8.** Structure of a single sequence. The image data are provided separately from other sensor data due to the size of the image data. After downloading the image data, the image data must be manually relocated to be properly loaded by the file player, which is one of the provided development tools.

of the data, the packets were converted into Cartesian coordinate values and reflectance values and were saved in floating-point binary format. Each point consists of four items ( $x, y, z, R$ ). Here,  $x, y$ , and  $z$  are the local Cartesian coordinate values of the sensor, and  $R$  is the reflectance of the point. The timestamps of all 3D LiDAR sensor data were stored sequentially in `VLP_left_stamp.csv` and `VLP_right_stamp.csv`.

2. **2D LiDAR data:** The SICK LMS-511, which is used as the 2D LiDAR sensor, acquired data at a

maximum speed of 100 Hz. 2D LiDAR data were stored in the `SICK_back` and the `SICK_middle` folders in a floating-point binary format. The file name of each datum is the timestamp of each scan data. In the case of 2D LiDAR, raw data ( $r, R$ ) that were not converted into Cartesian coordinates was provided to reduce the size of the data. Here  $r$  is the range value of each measurement, and  $R$  is the reflectance value. The Field of View (FOV) of the LMS-511 sensor is  $190^\circ$ , ranging from  $-5^\circ$  to  $185^\circ$ . When the data are acquired at 100 Hz, the angle between two measurements is  $0.666^\circ$ . The  $i$ th range measurement can be converted into Cartesian coordinates using

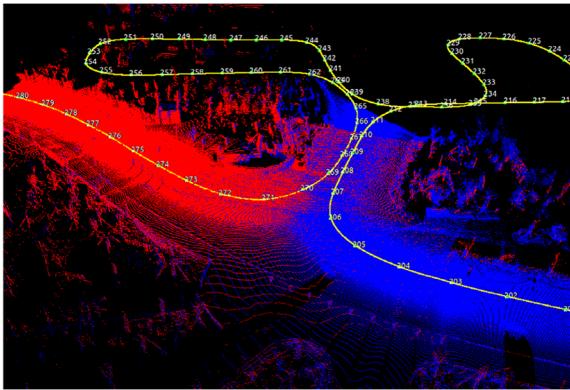
$$\begin{aligned}x_i &= r \cos(-5 + i \times 0.666) \\y_i &= r \sin(-5 + i \times 0.666)\end{aligned}\quad (8)$$

In addition, the timestamps of all 2D LiDAR data were saved sequentially in `SICK_back_stamp.csv` and `SICK_middle_stamp.csv`.

3. **Stereo image:** The stereo images were acquired at 10 Hz and stored in the loss-less PNG format in unrectified 8-bit Bayer pattern images. The Bayer pattern of the images is RGGB. An external trigger was used for accurate synchronization of the stereo camera. Images are stored in the `stereo_left` and `stereo_right` folders in the `image` folder, and each image is named using timestamps. The timestamps of all stereo images are stored in order in `stereo_stamp.csv` located in the `sensor_data` folder. The image data and the sensor data are provided separately for convenient downloading. The provided raw Bayer images are able to be converted into RGB images using the `demosaic` function of MATLAB or the `cvtColor` function of OpenCV (Figure 9). Rectification of the stereo images is easily



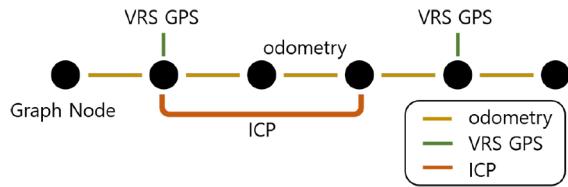
**Fig. 9.** The dataset provides unrectified 8-bit Bayer pattern images (a). This image can be converted into rectified RGB images (b) using calibration data.



**Fig. 10.** The baseline generation process using the ICP algorithm. The yellow line represents the path of the vehicle center, and the numbers beside the green dots represent the node numbers. The red and blue point clouds represent the local point clouds belonging to two adjacent nodes. The relative pose between the two adjacent nodes is calculated by the ICP algorithm that uses the point clouds.

processed using the provided MATLAB file (`stereoParams.mat`) or ROS parameter data (`left.yaml`, `right.yaml`).

4. **Data sequence:** The names and timestamps of all sensor data were sequentially stored in the `data_stamp.csv` file in the form of (timestamp, sensor name).
5. **Altimeter data:** The altitude data measured by the altimeter sensor was stored in the `altitude.csv` file in the form of (timestamp, altitude).
6. **Encoder data:** The incremental pulse count values of the wheel encoder were saved in the `encoder.csv` file in the form of (timestamp, left count, right count).
7. **FOG data:** The relative rotational motion between consecutive sensor data was saved in the `fog.csv` file in the form of (timestamp, delta roll, delta pitch, delta yaw).
8. **GPS data:** The global position measured using the commercial-grade GPS sensor was stored in the `gps.csv` file in the form of (timestamp, latitude, longitude, altitude, 9-tuple vector (position covariance)).
9. **VRS-GPS data:** The accurate global position measured by the VRS-GPS sensor was stored in the `vrs_gps.csv` file in the form of (timestamp, latitude, longitude, x coordinate, y coordinate, altitude, fix state, number of satellite, horizontal precision, latitude std, longitude std, altitude std, heading validate flag, magnetic global heading, speed in knots, speed in km, GNVTG mode, orthometric altitude). The  $x$  and  $y$  values were calculated in the universal transverse mercator (UTM) coordinate system. The fix state is a number representing the state of the VRS-GPS (1: normal; 2: DGPS; 4: fix; 5: float). The



**Fig. 11.** Constraint between graph nodes. All nodes are configured with relative constraints using odometry information. The odometry information is calculated using data obtained from the wheel encoder and FOG. If the VRS-GPS data are accurate, a global constraint is added to each node, and a relative constraint is added between nodes through the ICP algorithm.

specifications of the VRS-GPS listed in the sensor list (Table 2) is the accuracy of fixed state.

10. **IMU data:** The incremental rotational pose, gyroscope data, linear acceleration, and magnetic field data were stored in the `imu.csv` file in the form of (timestamp, quaternion x, quaternion y, quaternion z, quaternion w, Euler x, Euler y, Euler z, Gyro x, Gyro y, Gyro z, Acceleration x, Acceleration y, Acceleration z, MagnetField x, MagnetField y, MagnetField z).

[这里给出了IMU的原始数据](#)

#### 4.3. Baseline trajectory using SLAM

One of the primary purposes of using a public dataset is to exploit the baseline provided with the dataset. However, obtaining a precise baseline position of a vehicle is challenging in complex urban areas where GPS data are highly sporadic. Both general GPS and high-precision VRS-GPS cannot calculate accurate global position owing to the environmental complexity of such environments.

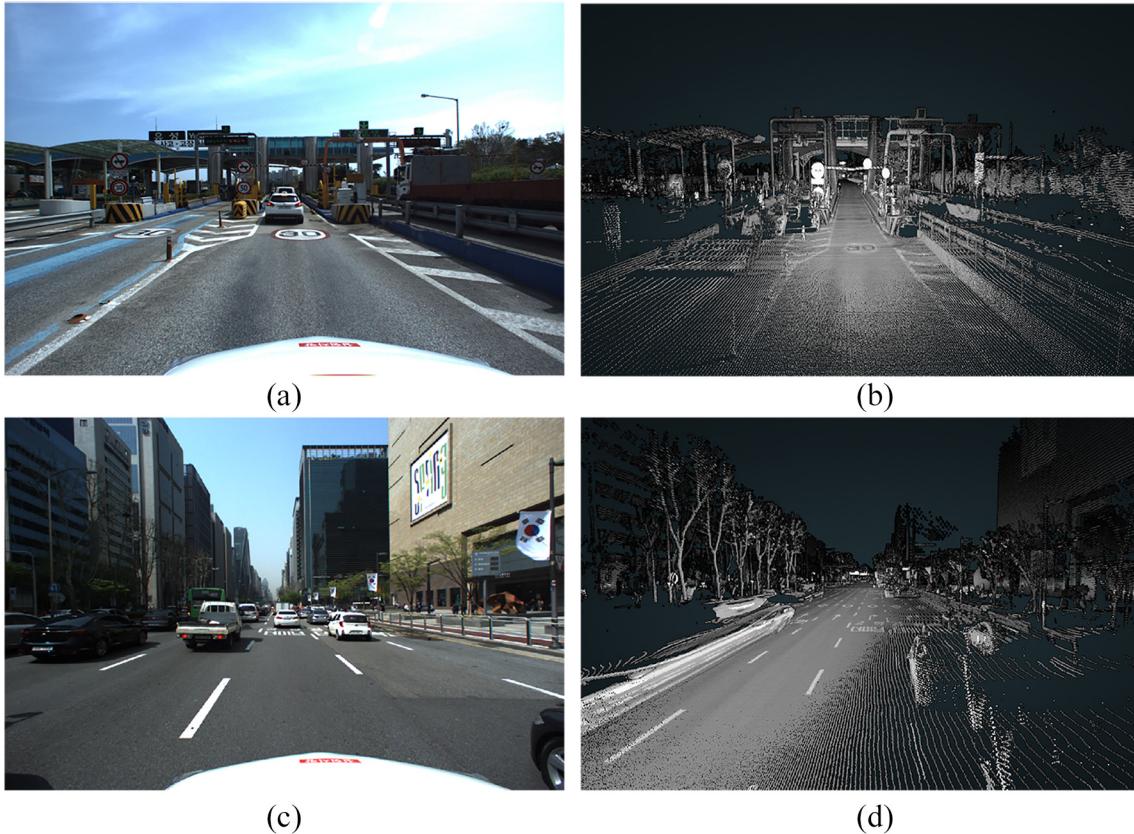
In this paper, the baseline trajectory was estimated using an incremental smoothing and mapping (iSAM) (Kaess et al., 2008) pose-graph SLAM framework (Figure 11). Rotational measurements from FOG and distance measurements from the encoder sensor were utilized to calculate the sequential relative constraint of the SLAM framework. In the case of VRS-GPS measurements, a measurement with low uncertainty was adopted as the partial constraint. The data obtained from the sensors and the optimization process can be used to calculate the pose graph with high accuracy. However, in areas with a multitude of high-rise buildings, the accuracy of the graph decreased owing to the unreliable GPS data, which can improve global accuracy. In order to overcome this problem, loops were created in the paths for data acquisition, and the accumulated error was compensated through the loop-closing process. Upon the occurrence of a loop in the path of a vehicle, the relative poses for loop closing were calculated using the Iterative Closest Point (ICP) (Segal et al., 2009) algorithm with a local point cloud at each node (Figure 10). Local point clouds were first



**Fig. 12.** 3D reconstruction result using 2D LiDAR data. The provided dataset involves diverse environments such as wide and narrow roads, complex structures, and underground environments. Reconstructed 3D data are provided through WebGL on our webpages to intuitively assist users in understanding the dataset. (a) Multi-lane road. (b) Apartment complex. (c) and (d) Intersection of a narrow road. (e) Urban tunnel. (f) Bridge and intersection. (g) Entrance to an underground parking lot. (h) Complex metropolitan area.

created based on the initial pose graph in the vicinity of the two nodes where loops occurred. The relative pose

between the generated local point clouds were calculated using the ICP algorithm. The calculated relative pose was



**Fig. 13.** Sample stereo images and associated 3D point cloud in various environments. The dataset provides stereo data and sensor data with precisely synchronized timestamps. In addition, reconstructed 3D point clouds are also provided in LAS format. (a) Stereo image of a tollgate on a highway. (b) 3D point cloud of a tollgate on a highway. (c) Stereo image of a metropolitan area. (d) 3D point cloud of a metropolitan area.

added as a constraint of the graph, and the position of the vehicle (baseline) was estimated using the node of the optimized graph.

The generated baseline is stored in the `vehicle_pose.csv` file at a rate of 100 Hz. However, it is not recommended to use the provided baseline as the ground truth for mapping or localization as the complexity of the environment differs greatly depending on the sequence.

还是不建议

#### 4.4. WebGL 3D demonstration

The dataset mainly provides raw data acquired by the mobile mapping system (MMS). In order to enhance the preview capability, reconstructed 3D point clouds based on the baseline obtained through SLAM are provided to the user through Potree (Schütz, 2016) WebGL on our website. The WebGL allows all users to visualize the environment in which the data were collected and the route traveled by the MMS vehicle.

#### 4.5. LAS data

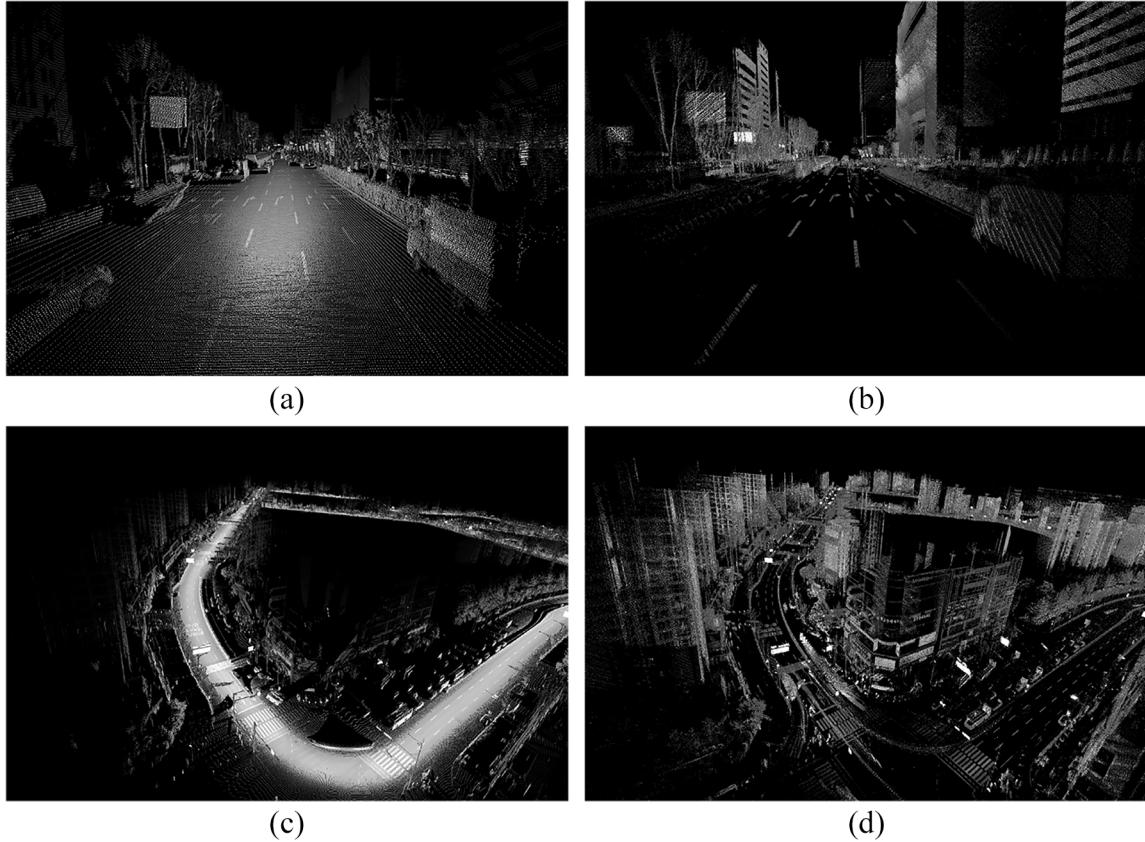
The LAS format is a public file format for 3D point clouds. The LAS data of the reconstructed 3D point clouds for all

sequences are provided in this dataset. The provided LAS data can be used as prior information for research such as localization and place recognition using LiDAR or images. In addition, the LAS data can be used as source data for segmentation and classification of point clouds. The LAS data contains 3D information in the UTM coordinate system and the intensity values of the point cloud.

#### 4.6. Development tool

The following tools are provided with the dataset for user convenience. All tools were created in the ROS environment for the robotics community.

1. **File player:** The file player reads data from the provided sequence and publishes the data as an ROS message. Specific types of messages were defined and released via the GitHub webpage for efficient data delivery. As urban environments are spaces with abundant dynamic objects and traffic lights, a substantial number of stationary sections occurred during the data-collection phase. As the data associated with such stationary periods may be of less importance for researchers, the provided file player is able to skip



**Fig. 14.** Sample reconstructed 3D point cloud comparison between 2D (left) and 3D (right) from the same location. Point clouds created with 2D LiDAR data possess relatively uniform intensity values and high positional accuracy. Point clouds built with 3D LiDAR data have a wide range of mapping.

- such sections and adjust the publishing speed. The file player also publishes stereo images in the `image` folder of the root directory. The file player is unable to find images if the name of the image folder is not exactly identical to `image`. Check the folder name of stereo images when copying the image data into the file directory. In addition, the player automatically reads stereo calibration files (`left.yaml`, `right.yaml`) from the calibration folder and publishes camera info messages for the generation of rectified images and disparity images through `stereo_image_proc` package. Ensure the calibration data are located in the correct folder, as shown in Figure 8.
2. **Data viewer:** The data viewer is provided to visually verify data by subscribing the data that is published by the file player. The data viewer displays most navigation sensor data as well as LiDAR sensor data that has been converted into the vehicle reference coordinate system. Stereo images that are converted into RGB images are displayed in the viewer. All provided players and viewers were built by using the libraries provided in ROS without additional dependency.

## 5. Potential application

This dataset provides raw sensor data, reconstructed 3D points, and the position of the vehicle estimated via SLAM, hence providing a broad utility to related fields in robotics, computer vision, and urban engineering. Direct application to use the dataset includes various perceptual navigation researches such as SLAM and odometry. The dataset serves multi-modal registration and localization research. By utilizing the two-level accuracy in the sensor data, map building and localization algorithms can be tested and compared with the baseline and 3D maps as prior information.

Apart from robotics research, the dataset is suited to civil and urban engineering. Notably, the presented 3D point cloud data. In this field of study, a 3D map obtained by the airborne LiDAR has been widely adopted to obtain information on a wide area in civil engineering research. Unlike airborne-based data, this dataset utilizes a ground-level viewpoint for the urban 3D point cloud. Many model-based approaches in urban planning and analysis can be validated using real urban data.

## 6. Conclusion

The purpose of this paper was to provide LiDAR and image data captured from diverse complex urban environments where GPS reception is unreliable and sporadic. The GPS data for estimating global position and the data from navigation sensors such as IMU and FOG are provided with two levels of accuracy. Although commercial-grade sensors (GPS, IMU) are less expensive and less accurate, sensors such as FOG and VRS-GPS are expensive but provide highly accurate sensor measurements. Therefore, researchers are able to selectively use data for estimating the position of the vehicle according to the purpose of the study. In addition, this dataset provides the baseline of the vehicle position, which was estimated using the high-precision sensors, LiDAR sensors, and the SLAM algorithm. The dataset will be continually updated, and the accuracy of the baseline will be improved using the enhanced SLAM algorithm.

## Acknowledgement

A portion of the paper is accepted and will be presented in part at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Brisbane, Australia, 2018 (Jeong et al., 2018).

## Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Ministry of Trade, Industry and Energy (MOTIE), Korea under the Industrial Technology Innovation Program (number 10051867) and a project of Naver Labs Corporation (High-Definition Map Based Precise Vehicle Localization Using Cameras and LIDARs).

## ORCID iDs

- Jinyong Jeong  <https://orcid.org/0000-0003-2362-0512>
- Younggun Cho  <https://orcid.org/0000-0003-2025-7770>
- Young-Sik Shin  <https://orcid.org/0000-0002-9653-0633>
- Hyunchul Roh  <https://orcid.org/0000-0003-0729-8359>
- Ayoung Kim  <https://orcid.org/0000-0001-9829-2408>

## References

- Blanco-Claraco JL, Moreno-Dueñas FA and González-Jiménez J (2014) The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario. *The International Journal of Robotics Research* 33(2): 207–214.
- Burri M, Nikolic J, Gohl P, et al. (2016) The EUROC micro aerial vehicle datasets. *The International Journal of Robotics Research* 35(10): 1157–1163.
- Carlevaris-Bianco N, Ushani AK and Eustice RM (2016) University of Michigan North Campus long-term vision and LiDAR dataset. *The International Journal of Robotics Research* 35(9): 1023–1035.
- Cordts M, Omran M, Ramos S, et al. (2016) The Cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223.
- Curnow R (2014) Chrony. <https://chrony.tuxfamily.org>
- Fallon M, Johannsson H, Kaess M and Leonard JJ (2013) The MIT Stata Center dataset. *The International Journal of Robotics Research* 32(14): 1695–1699.
- Geiger A, Lenz P, Stiller C and Urtasun R (2013) Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* 32(11): 1231–1237.
- Huang AS, Antone M, Olson E, et al. (2010) A high-rate, heterogeneous data set from the DARPA Urban Challenge. *The International Journal of Robotics Research* 29(13): 1595–1601.
- Huang X, Cheng X, Geng Q, et al. (2018) The Apolloscape dataset for autonomous driving. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 954–960.
- Jeong J, Cho Y and Kim A (2019) Road is enough! Extrinsic calibration of non-overlapping stereo camera and LiDAR using road information. *arXiv preprint arXiv:1902.10586*.
- Jeong J, Cho Y, Shin YS, Roh H and Kim A (2018) Complex urban LiDAR data set. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Brisbane, pp. 6344–6351.
- Jung H, Oto Y, Mozos OM, Iwashita Y and Kurazume R (2016) Multi-modal panoramic 3D outdoor datasets for place categorization. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4545–4550.
- Kaess M, Ranganathan A and Dellaert F (2008) iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics* 24(6): 1365–1378.
- Leung KY, Halpern Y, Barfoot TD and Liu HH (2011) The UTIAS multi-robot cooperative localization and mapping dataset. *The International Journal of Robotics Research* 30(8): 969–974.
- Maddern W, Pascoe G, Linegar C and Newman P (2017) 1 year, 1000 km: The Oxford Robotcar dataset. *The International Journal of Robotics Research* 36(1): 3–15.
- Miller M, Chung SJ and Hutchinson S (2018) The visual–inertial canoe dataset. *The International Journal of Robotics Research* 37(1): 13–20.
- Mueggler E, Rebecq H, Gallego G, Delbrück T and Scaramuzza D (2017) The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research* 36(2): 142–149.
- Pandey G, McBride JR and Eustice RM (2011) Ford Campus vision and LiDAR data set. *The International Journal of Robotics Research* 30(13): 1543–1552.
- Peynot T, Scheding S and Terho S (2010) The Marulan data sets: Multi-sensor perception in a natural environment with challenging conditions. *The International Journal of Robotics Research* 29(13): 1602–1607.
- Pomerleau F, Liu M, Colas F and Siegwart R (2012) Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research* 31(14): 1705–1711.
- Roh H, Jeong J and Kim A (2017) Aerial image based heading correction for large scale SLAM in an urban canyon. *IEEE Robotics and Automation Letters* 2(4): 2232–2239.
- Schütz M (2016) Potree: Rendering large point clouds in web browsers. Master Thesis, Technische Universität Wien, Wien. Available at: <https://www.cg.tuwien.ac.at/research/publications/2016/SCHUETZ-2016-POT/> (accessed March 2019).
- Scott T, Morye AA, Piniés P, Paz LM, Posner I and Newman P (2015) Exploiting known unknowns: Scene induced cross-calibration of LiDAR–stereo systems. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3647–3653.

- Segal A, Haehnel D and Thrun S (2009) Generalized-ICP. In: *Robotics: Science and Systems*, Vol. 2.
- Smith M, Baldwin I, Churchill W, Paul R and Newman P (2009) The New College vision and laser data set. *The International Journal of Robotics Research* 28(5): 595–599.
- Smith R, Self M and Cheeseman P (1990) Estimating uncertain spatial relationships in robotics. In: *Autonomous Robot Vehicles*. New York: Springer, pp. 167–193.
- Zhang Z (2000) A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(11): 1330–1334.
- Zhu AZ, Thakur D, Ozaslan T, Pfommer B, Kumar V and Danilidis K (2018) The multi vehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters* 3(3): 2032–2039.