

Complex Urban LiDAR Data Set

Jinyong Jeong¹, Younggun Cho¹, Young-Sik Shin¹, Hyunchul Roh¹ and Ayoung Kim¹

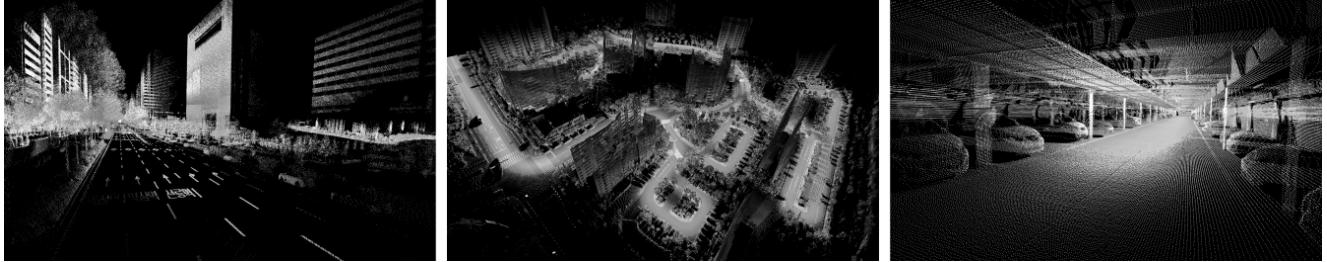


Fig. 1: This paper provides the complex urban data set including metropolitan area, apartment building complex and underground parking lot. Sample scenes from the data set can be found in <https://youtu.be/IguZjmLf5V0>.

Abstract—This paper presents a Light Detection and Ranging (LiDAR) data set that targets complex urban environments. Urban environments with high-rise buildings and congested traffic pose a significant challenge for many robotics applications. The presented data set is unique in the sense it is able to capture the genuine features of an urban environment (e.g. metropolitan areas, large building complexes and underground parking lots). Data of two-dimensional (2D) and three-dimensional (3D) LiDAR, which are typical types of LiDAR sensors, are provided in the data set. The two 16-ray 3D LiDARs are tilted on both sides for maximal coverage. One 2D LiDAR faces backward while the other faces forwards to collect data of roads and buildings, respectively. Raw sensor data from Fiber Optic Gyro (FOG), Inertial Measurement Unit (IMU), and the Global Positioning System (GPS) are presented in a file format for vehicle pose estimation. The pose information of the vehicle estimated at 100 Hz is also presented after applying the graph simultaneous localization and mapping (SLAM) algorithm. For the convenience of development, the file player and data viewer in Robot Operating System (ROS) environment were also released via the web page. The full data sets are available at: <http://irap.kaist.ac.kr/dataset>. In this website, 3D preview of each data set is provided using WebGL.

I. INTRODUCTION

Autonomous vehicles have been studied by many researchers in recent years, and algorithms for autonomous driving have been developed using diverse sensors. As it is important that algorithms for autonomous driving use data obtained from the actual environment, many groups have disclosed data sets. Data sets based on camera vision data such as [1], [2], [3] and [4] are used to develop various applications such as visual odometry, semantic segmentation, and vehicle detection. Data sets based on LiDAR data such as [4], [5], [6], [7] and [8] are used in applications such as object detection, LiDAR odometry, and 3D mapping. However, most data sets do not focus on highly complex urban environments (significantly wide roads, lots of dynamic

objects, GPS blackout regions and high-rise buildings) where actual autonomous vehicles operate.

A complex urban environment such as a downtown area poses a significant challenge for many robotics applications. Validation and implementation in a complex urban environment is not straightforward. Unreliable GPS, complex building structure, and limited ground truth are the main challenges for robotics applications in urban environments. In addition, urban environments have high population densities and heavy foot traffic, resulting in many dynamic objects that obstruct robot operations, and cause sudden environmental changes. This paper presents a LiDAR sensor data set that specifically targets the urban canyon environment (e.g. metropolitan area and confined building complexes). The data set is not only extensive in terms of time and space, but also includes features of large-scale environments such as skyscrapers and wide roads. The presented data set was collected using two types of LiDARs and various navigation sensors that possess both commercial-level accuracy and high-level accuracy.



Fig. 2: LiDAR sensor system for the complex urban data set. The yellow boxes indicate LiDAR sensors (2D and 3D LiDAR sensors) and the red boxes indicate navigation sensors (VRS-GPS, IMU, FOG, and GPS).

¹J. Jeong, Y. Cho, Y. Shin, H. Roh and A. Kim are with the Department of Civil and Environmental Engineering, KAIST, Daejeon, S. Korea [jjy0923, yg.cho, youngsik, rohs-, ayoungk]@kaist.ac.kr

The structure of the paper is as follows. Section II describes the process of surveying existing publicly open data sets and comparing the characteristics. Section III provides an overview of the configuration of the sensor system. The details and specificity of the proposed data set are explained in Section IV. Finally, the conclusion of the study and suggestions for further works are provided in Section V.

II. RELATED WORKS

There are several data sets in the robotics field that offer 3D point cloud data sets of indoor/outdoor environments. The Ford Campus Vision and LiDAR Data Set [9] offers 3D scan data of roads and low-rise buildings. The data set was captured in a part of a campus using horizontally scanning 3D LiDAR mounted on the top of a vehicle. The KITTI data set [4] provides LiDAR data of less complex urban areas and highways, and is the most commonly used data set for various robotic applications including motion estimation, object tracking, and semantic classification. The North Campus Long-Term (NCLT) data set [7] consists of both 3D and 2D LiDAR data collected in the University of Michigan campus. The segway platform explored both indoor and outdoor environments over a period of 15 months to capture long-term data. However, these data sets do not address highly complex urban environments that include various moving objects, high-rise buildings, and unreliable positioning sensor data.

The Malaga data set [6] provides 3D point cloud data using two planar 2D LiDAR mounted on the side of the vehicle. The sensors were equipped in a push-broom configuration, and 3D point data was acquired as the vehicle moving forward. The Multi-modal Panoramic 3D Outdoor (MPO) data set [10] offers two types of 3D outdoor data sets: dense and sparse MPO. This data set mainly focuses on data for semantic place recognition. To obtain dense panoramic point cloud data, the authors utilized a static 3D LiDAR mounted on a moving platform. The Oxford RobotCar (Oxford) Dataset [11] collected large variations of scene appearance. Similar to the Malaga data set, this data set also used push-broom 2D LiDARs mounted on the front and rear of the vehicle. While the data sets mentioned above attempt to offer various 3D urban information, the data sets are not complex enough to cover the sophisticated environment of complex urban city scenes.

Compared to these existing data sets, the data set presented in this paper possess the following unique characteristics:

- Provides data from diverse environments such as complex metropolitan areas, residential areas and apartment building complexes.
- Provides sensor data with two levels of accuracy (economic sensors with consumer-level accuracy and expensive high-accuracy sensors).
- Provides baseline via SLAM algorithm using highly accurate navigational sensors and manual Iterated Closest Point (ICP).
- Provides development tools for the general robotics community via ROS.

- Provides raw data and 3D preview using WebGL targeting diverse robot application.

III. SYSTEM OVERVIEW

This section describes the sensor configuration of the hardware platform and the sensor calibration method.

A. Sensor Configuration

The main objective of the sensor system in Fig. 2 is to provide sensor measurements that possess different sensor accuracy levels. For the attitude and position of the vehicle, data from both relatively low-cost sensors and highly accurate expensive sensors were provided simultaneously. The sensor configuration is summarized in Fig. 3 and Table. I.

The system included both 2D and 3D LiDARs that provide a total of four LiDAR sensor measurements. Two 3D LiDARs were installed in parallel facing the rear direction and tilted 45° from the longitudinal and lateral planes. The structure of the tilted 3D LiDARs allow for maximal coverage as data on the plane perpendicular to the travel direction of the vehicle can be obtained. Two 2D LiDARs were each installed facing forward and backward, respectively. The rear 2D LiDAR faces downwards towards the road, while the frontal LiDAR installed in middle portion faces upwards toward the buildings.

For inertial navigational sensors, two types of attitude sensor data, a 3-axis FOG and an IMU, were provided. The 3-axis FOG provides highly accurate attitude measurements that are used to estimate a baseline, while the IMU provides general sensor measurements. The system also has two levels of GPS sensors, a VRS GPS and a single GPS. The VRS GPS provides up to cm-level accuracy when a sufficient number of satellites are secured, while the single GPS provides conventional-level position measurement. However, note that the availability of GPS is limited in urban environments due to the complex environment and the presence of high-rise buildings.

The hardware configuration for the sensor installation are depicted in Fig. 3. Fig. 3(a) and Fig. 3(b) show the top and the side views of the sensor system, respectively. Each sensor possesses its own coordinate system, and the red, green and blue arrows in the figure indicate the x, y and z axes of each coordinate system. The figures also portray the relative coordinate values of each sensor relative to the reference coordinate system of the vehicle. The center of the reference coordinate system is located at the center of the vehicle rear axle with a height of zero.

Most sensors were mounted externally on the vehicle with the exception of the 3-axis FOG, which was installed inside the vehicle as shown. Magnetic rotary encoders were used to gauge wheel rotation, and were installed inside each wheel. The vehicle was equipped with 18-inch tires. All sensor data was logged using a personal computer (PC) with an i7 processor, a 512GB SSD, and 64GB DDR4 memory. The sensor drivers and logger were developed on the Ubuntu OS. Additional details are listed in Table. I.

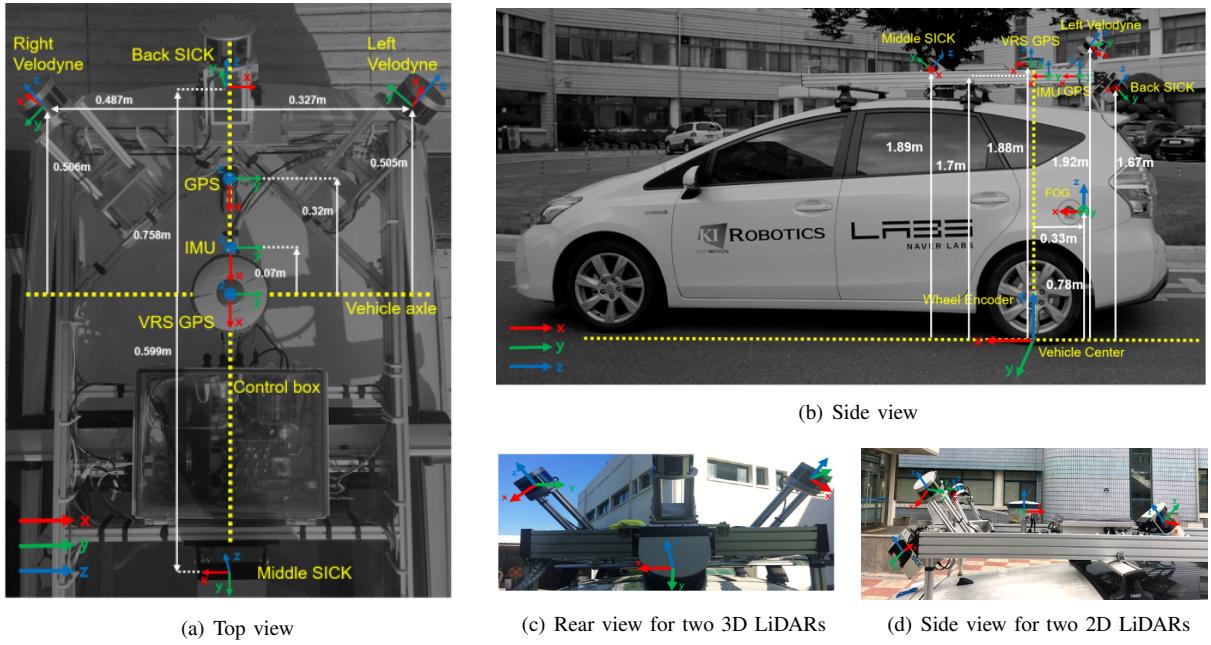


Fig. 3: Hardware sensor configuration. (a) Top view and (b) side view of the entire sensor system with coordinate frame. Each sensor is mounted on the vehicle, and the red, green, and blue arrows indicate the x, y, and z coordinates of the sensors, respectively. (c) Two 3D LiDARs are tilted 45° for maximal coverage. (d) The rear 2D LiDAR face downwards towards the road and the middle 2D LiDAR faces upwards to detect the building structures. Sensor coordinates are displayed on each sensor figure.

TABLE I: Specifications of sensors used in sensor system (H: Horizontal, V: Vertical)

Type	Manufacturer	Model	Description	No.	Hz	Accuracy	Range
3D LiDAR	Velodyne	VLP-16	16 channel 3D LiDAR with 360° FOV	2	10		100 m
2D LiDAR	SICK	LMS-511	1 channel 2D LiDAR with 190° FOV	2	100		80 m
GPS	U-Blox	EVK-7P	Consumer level GPS	1	10	2.5 m	
VRS GPS	SOKKIA	GRX 2	VRS-RTK GPS	1	1	H: 10 mm , V: 15 mm	
3-axis FOG	KVH	DSP-1760	Fiber optics gyro (3 axis)	1	1000	0.05°/h	
IMU	Xsens	MTi-300	Consumer level gyro enhanced AHRS	1	100	10°/h	
Wheel encoder	RLS	LM13	Magnetic rotary encoder	2	100	4096 (resolution)	
Altimeter	Withrobot	myPressure	Altimeter sensor	1	10	0.01hPa (resolution)	

B. Odometry Calibration

For accurate odometry measurements, odometry calibration was performed using high-precision sensors: VRS GPS and FOG. The calibration was conducted in a wide and flat open space that guaranteed precision of the reference sensors, VRS GPS and FOG. As two-wheel encoders were mounted on the vehicle, the forward kinematics of the platform can be calculated using three parameters $\mathbf{w} = (d_l, d_r, w_b)$: the left and right wheel diameters, and the wheel base between the two rear wheels. To obtain relative measurements from global motion sensors, a 2D pose graph $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ was constructed whenever accurate VRS GPS measurements were received. The coordinates of the VRS GPS and FOG are globally synchronized, and a node is added from hard-coupled measurements $\mathbf{x}_i = (x_i, y_i, \theta_i)$ written in vehicle center coordinate. Least square optimization was used to obtain optimized kinematic parameters \mathbf{w}^* using relative motion from the graph $\mathbf{z}_i = \mathbf{x}_{i+1} \ominus \mathbf{x}_i$ and forward motion from the kinematics $\mathbf{k}_i(\mathbf{w})$. The mathematical expression of

the objective function is

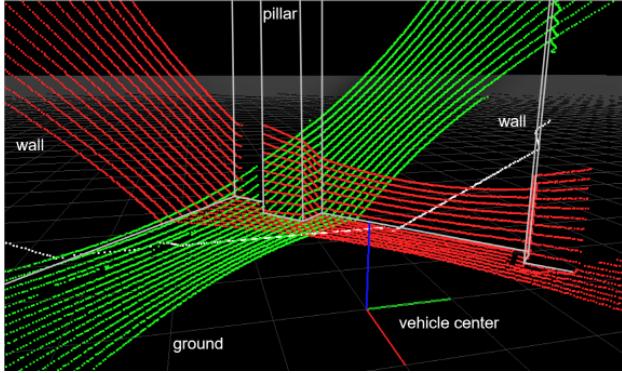
$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i \|\mathbf{z}_i \ominus \mathbf{k}_i(\mathbf{w})\|_{\Omega_i} \quad (1)$$

where \ominus is the inverse motion operator [12] and Ω_i represents the measurement uncertainty of VRS GPS and FOG. The calibrated parameters are provided in EncoderParameter.txt file in calibration folder.

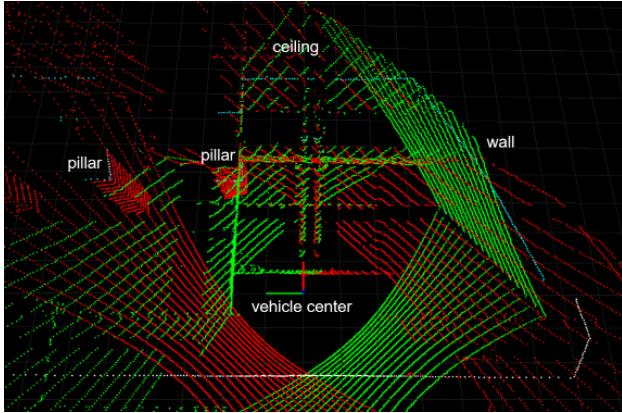
C. LiDAR Extrinsic Calibration

The purpose of this process is to calculate accurate transformation between the reference vehicle coordinates and the coordinates of each sensor. Three types of extrinsic calibration are required to achieve this purpose. Extrinsic LiDAR calibration between the four LiDAR sensors was performed via optimization. The Table. II represents each coordinate frame.

1) *3D LiDAR to 3D LiDAR*: Among the four LiDAR sensors installed in the vehicle, the left 3D LiDAR sensor was used as a reference frame for calibration. By calculating



(a) Front view of LiDAR sensor data



(b) Top view of LiDAR sensor data

Fig. 4: Point cloud captured during the LiDAR calibration. A corner of the building was used for the calibration to provide multiple planes orthogonal to each other. Red and green point cloud are left and right 3D LiDAR point cloud respectively. The white and azure point cloud are rear and middle 2D LiDAR point cloud respectively. The red, green, and blue lines perpendicular to each other refer to the reference coordinate system of the vehicle.

TABLE II: Coordinate frame subscript

Subscript	Description
<i>v</i>	Vehicle frame
<i>l</i>	Left 3D LiDAR (LiDAR reference frame)
<i>r</i>	Right 3D LiDAR
<i>m</i>	Forward looking 2D LiDAR in the middle
<i>b</i>	backward looking 2D LiDAR in the rear

the relative transformation of other LiDAR sensors with respect to the left 3D LiDAR sensor, a relative coordinate transform was defined among all the LiDAR sensors. The first relative coordinate transform that should be computed is the transform between the left and right 3D LiDAR sensors. Generalized Iterated Closest Point (GICP) [13] was applied to calculate the required transformation that maps the i^{th} right LiDAR point cloud data ($\mathbf{p}_{r,i}$) to the corresponding left LiDAR point cloud data ($\mathbf{p}_{l,i}$). Fig. 4 shows the LiDAR sensor data during the calibration process. As shown in the figure, the relative rotation (\mathbf{R}_{rl}) and translation (\mathbf{t}_{rl}) of

the two 3D LiDAR sensors can be calculated using data from the overlap region between the two 3D LiDAR data by minimizing the error between the projected points (2).

$$\mathbf{R}_{rl}^*, \mathbf{t}_{rl}^* = \underset{\mathbf{R}_{rl}, \mathbf{t}_{rl}}{\operatorname{argmin}} \sum_i \{ \mathbf{p}_{l,i} - (\mathbf{R}_{rl} \cdot \mathbf{p}_{r,i} + \mathbf{t}_{rl}) \} \quad (2)$$

2) *3D LiDARs to the Vehicle*: Using the previously computed coordinate transformation, the two 3D LiDAR points are aligned to generate merged 3D LiDAR points. The next step is to find the transformation to match the ground points in the merged 3D LiDAR points to zero. The ground points are first detected using the Random Sample Consensus (RANSAC) algorithm by fitting a plane. The height value of all plane points should be zero. Formulating as above, the least square problem was solved using Singular Value Decomposition (SVD).

3) *3D LiDAR to 2D LiDAR*: Completing the previous two steps calculates the transformation between the vehicle and the two 3D LiDAR coordinates and the resulting point cloud is properly grounded. In the following step, 3D LiDAR data that overlap with 2D LiDAR data are used to estimate the transformation between the 2D LiDAR sensor and the vehicle. Structural information was used to consider plane-to-point alignment. Planes are extracted from 3D LiDAR data and points from the 2D scan lines are examined in this optimization process. Through this process, it is possible to calculate the transformation from the vehicle to each 2D LiDAR sensor ($\mathbf{P}_{vb}, \mathbf{P}_{vm}$).

$$\begin{aligned} \mathbf{P}_{vb} &= \mathbf{P}_{vl} \times \mathbf{P}_{bl}^{-1} \\ \mathbf{P}_{vm} &= \mathbf{P}_{vl} \times \mathbf{P}_{ml}^{-1} \end{aligned} \quad (3)$$

For accurate calibration values, the transformation was provided in both $SE(3)$ and Euler formats with the data set. Table. IV shows calibrated sample coordinate transforms.

IV. COMPLEX URBAN DATA SET

This section describes the urban LiDAR data set regarding formats, sensor types and development tools. The data provides a diverse level of complexity captured in a real urban environment.

A. Data Description

The data set of this paper covers various features in large urban areas from wide roads with ten lanes or greater, to substantially narrow roads with high-rise buildings. Table. III describes the overview of the data set. As the data set covers highly complex urban environments where GPS is sporadic, the depicted GPS availability map was overlaid on the mapping route as in Fig. 5. Table. III shows the GPS reception rate, which represents the average number of satellites of VRS GPS data, for each data set. Ten satellites are required to calculate the accurate location regularly. The complexity and wide road rate were evaluated for each data set and are shown in Table. III.

TABLE III: Dataset lists

Data number	No. Subset	Location	Description	GPS reception rate	Complexity	Wide road rate	Path length
Urban00	2	Gangnam, Seoul	Metropolitan area	7.49	★★★	★★★★★	12.02 km
Urban01	2	Gangnam, Seoul	Metropolitan area	5.3	★★★	★★★★	11.83 km
Urban02	2	Gangnam, Seoul	Residential area	4.58	★★★★★	★	3.02 km
Urban03	1	Gangnam, Seoul	Residential area	4.57	★★★★★	★	2.08 km
Urban04	3	Pangyo	Metropolitan area	7.31	★★★	★★★★	13.86 km
Urban05	1	Daejeon	Apartment complex	7.56	★★	★★	2.00 km



(a) Urban00 (Gangnam, Seoul, Metropolitan area) (b) Urban01 (Gangnam, Seoul, Metropolitan area) (c) Urban02 (Gangnam, Seoul, Residential area)



(d) Urban03 (Gangnam, Seoul, Residential area)

(e) Urban04 (Pangyo, Metropolitan area)

(f) Urban05 (Daejeon, Apartment complex)

Fig. 5: Data collection route illustrating VRS GPS data. The green line represents the VRS GPS based vehicle path. The color of the circles drawn in the route represents the number of satellites used in the GPS calculation result; a brighter circle indicates more satellites were used. As complexity increases, fewer satellites are seen. The sections without circles are the areas where no satellites are seen, and no solution of position is available.

TABLE IV: Summary of LiDAR sensor transformation. Positional data is in meter and rotational data is in degree.

Type	Description [x, y, z, roll, pitch, yaw]
P_{vl}	Vehicle w.r.t left 3D LiDAR [-0.505, 0.327, 1.926, 1.618°, 44.84°, 137.0°]
P_{vr}	Vehicle w.r.t right 3D LiDAR [-0.506, -0.488, 1.939, -179.5°, 135.7°, 46.02°]
P_{vb}	Vehicle w.r.t rear 2D LiDAR [-0.758, -0.107, 1.673, 46.08°, -179.8°, 90.30°]
P_{vm}	Vehicle w.r.t middle 2D LiDAR [0.600, -0.111, 1.890, 145.5°, 1.371°, 90.11°]

B. Data Format

For convenience in downloading the data, the entire data was split into subsets of approximately 6 GB in size. Both the whole data set and the subsets are provided. The path of each data set can be checked through the `map.html` file in each folder. The file structure of each data set is depicted in Fig. 6. All data was logged using ROS timestamps. The data set is in a compressed `.tar` format. For accurate sensor transformation values, calibration was performed prior to each data acquisition. The corresponding calibration data can

be found in the `calibration` folder along with the data. All sensor data is stored in the `sensor_data` folder.

1) 3D LiDAR data

The 3D LiDAR sensor, Velodyne (VLP-16), provides data on a per-packet basis. Velodynes rotation rate is 10 Hz, and the timestamp of the last packet is used as the timestamp of the data at the end of one rotation. 3D LiDAR data is stored in the `VLP_left` and `VLP_right` folders in the `sensor_data` folder in a floating-point binary format, and the timestamp of each rotation data is used as the name of the file (`<time_stamp.bin>`). Each point consists of four items (x , y , z , R). x , y , and z denote the local 3D Cartesian coordinate values of each LiDAR sensor, and R is the reflectance value. The timestamps of all 3D LiDAR data are stored sequentially in `VLP_left_stamp.csv` and `VLP_right_stamp.csv`.

2) 2D LiDAR data

In the system, the 2D LiDAR sensors were operated at 100 Hz. The 2D LiDAR data is stored in



Fig. 6: File directory layout for a single data set.

the SICK_back and the SICK_middle folder in the sensor_data folder in a floating-point binary format. Similar to 3D LiDAR data, the timestamp of each scan data is used as the name of the file. To reduce the file size, the data of 2D LiDAR consists of two items (r, R). r is the range value of each point, and R is the reflectance value. The sensors Field of View (FOV) is 190° , where the start angle of the first data is -5° , and the end angle is 185° . The angle difference between each sequential data is 0.666° . Each point can be converted from i^{th} range measurement to a Cartesian coordinate using this information (4). The timestamps of all 2D LiDAR data are stored sequentially in SICK.back_stamp.csv and SICK.middle_stamp.csv.

$$\begin{aligned} x_i &= r \cos(-5 + i \times 0.666) \\ y_i &= r \sin(-5 + i \times 0.666) \end{aligned} \quad (4)$$

3) Data sequence

The sensor_data/data_stamp.csv file stores the names and timestamps of all sensor data in order in the form of (timestamp, sensor name).

4) Altimeter data

The sensor_data/altitude.csv file stores the altitude values measured by the altimeter sensor in the form of (timestamp, altitude).

5) Encoder data

The sensor_data/encoder.csv file stores the incremental pulse count values of the wheel encoder in the form of (timestamp, left count, right count).

6) FOG data

The sensor_data/fog.csv file stores the relative rotational motion between consecutive sensor data in the form of (timestamp, delta roll, delta pitch, delta yaw).

7) GPS data

The sensor_data/gps.csv file stores the global position measured by commercial level GPS sensor. The data format is (timestamp, latitude, longitude, altitude, 9-tuple vector (position covariance)).

8) VRS GPS data

The sensor_data/vrs_gps.csv file stores the accurate global position measured by VRS GPS sensor. The data format is (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 knot, speed in km, GNVTG mode). The x and y coordinates use the UTM coordinate system in the meter unit. The fix state is a number indicating the state of the VRS GPS. For example, 4, 5, and 1 indicates the fix, float, and normal states, respectively. The accuracy of the VRS GPS in the sensor specification list (Table. I) is the value at the fix state.

9) IMU data

The sensor_data/imu.csv file stores the incremental rotational pose data measured by AHRS IMU sensor. The data format is (timestamp, quaternion x, quaternion y, quaternion z, quaternion w, Euler x, Euler y, Euler z).

C. Baseline Trajectory using SLAM

The most challenging issue regarding the validity of data sets is to obtain a reliable baseline trajectory under highly sporadic GPS measurements. Both consumer-level GPS and VRS GPS suffer from GPS blackouts due to building complexes.

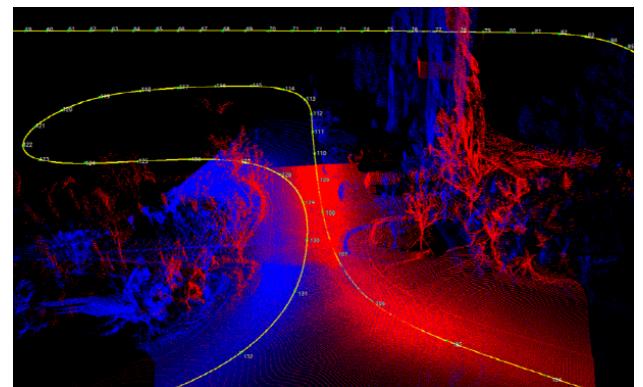
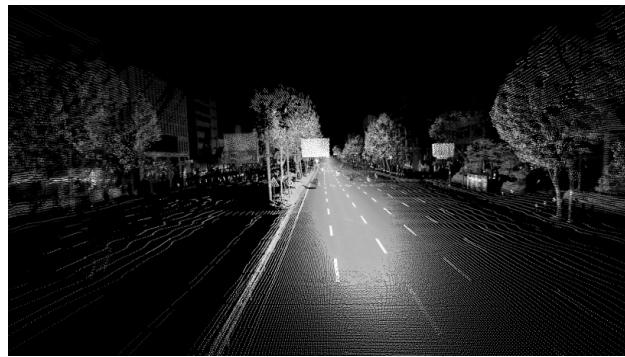


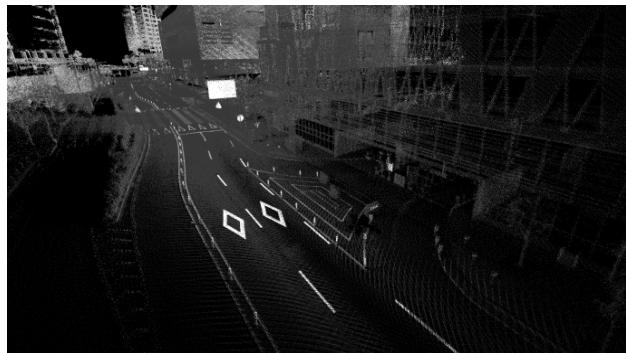
Fig. 7: Baseline generation process using ICP. The yellow line is the path of the vehicle and the number with the green point is the number of the graph node. The red and blue pointcloud are local sub-map of two nodes. The relative poses of the nodes are computed by ICP.



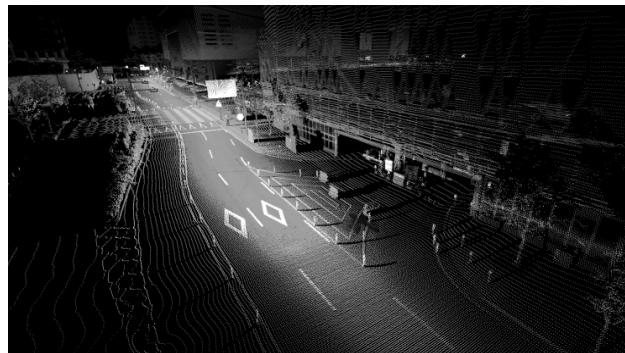
(a) Wide road (3D LiDAR)



(b) Wide road (2D LiDAR)



(c) Complex building entrance (3D LiDAR)



(d) Complex building entrance (2D LiDAR)



(e) Road markings (3D LiDAR)



(f) Road markings (2D LiDAR)



(g) High-rise buildings in complex urban environment (3D LiDAR)



(h) High-rise buildings in complex urban environment (2D LiDAR)

Fig. 8: Point cloud sample data from 3D LiDAR data and 2D LiDAR data. Two different LiDAR provide different aspect of the urban environment.

In this study, baselines were generated via pose-graph SLAM. Our strategy is to incorporate highly accurate sensors (VRS GPS, FOG and wheel encoder) in the initial baseline generation. Further refinement over this initial trajectory is performed using semi-automatic ICP for the revisited places (Fig. 7). Manual selection of the loop-closure proposal is piped into the ICP as the initial guess, and the baseline trajectory is refined using ICP results as the additional loop-closure constraint.

The generated baseline trajectory is stored in `vehicle_pose.csv` at the rate of 100 Hz. However, it is not desirable to use baseline trajectory as the ground truth for mapping or localization benchmarking as the SLAM results depend on the complexity of the urban environment.

D. Development Tool

The following tools were provided for the robotics community along with the data set.

1) File player

To support the ROS community, a File player that publishes sensor data as ROS messages is provided. New message types were redefined to convey more information, and were released via the GitHub webpage. In urban environments, there are many stop periods during data logging. As most of the algorithm does not require data in stop periods, the player can skip the stop period for convenience, and control data publishing speeds.

2) Data viewer

A data viewer is provided to check the data transmitted through the file player. The data viewer allows users to monitor the data that the player publishes in a visual manner. The data viewer shows all sensor data and the 2D and 3D LiDAR data converted to the vehicle coordinate system. The Provided player and viewer were built with libraries provided by ROS without additional dependencies.

V. CONCLUSION AND FUTURE WORK

This paper provided challenging data set targeting extremely complex urban environments where GPS signals are not reliable. The data set provided a baseline generated using SLAM algorithms with meter level accuracy. The data sets also offer two-levels of sensor pairs for attitude and position. Commercial-grade sensors are less expensive and less accurate, while sensors such as FOG and VRS GPS are more accurate and can be used for verification. The data sets captured various urban environments with different levels of complexity such as Fig. 8, from metropolitan areas to residential areas.

Our future data sets will be continually updated and the baseline accuracy will be improved. The future plan is to enrich the data set by adding a front stereo camera rig for visual odometry and a 3D LiDAR to detect surrounding obstacles.

ACKNOWLEDGMENT

This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE), Korea under Industrial Technology Innovation Program (No.10051867) and [High-Definition Map Based Precise Vehicle Localization Using Cameras and LiDARs] project funded by Naver Labs Corporation.

REFERENCES

- [1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
- [2] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *IEEE Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.
- [3] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 304–311.
- [4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [5] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.
- [6] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, "The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario," *International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.
- [7] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.
- [8] C. H. Tong, D. Gingras, K. Larose, T. D. Barfoot, and É. Dupuis, "The canadian planetary emulation terrain 3d mapping dataset," *International Journal of Robotics Research*, vol. 32, no. 4, pp. 389–395, 2013.
- [9] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.
- [10] H. Jung, Y. Oto, O. M. Mozos, Y. Iwashita, and R. Kurazume, "Multi-modal panoramic 3d outdoor datasets for place categorization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 4545–4550.
- [11] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
- [12] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous robot vehicles*. Springer, 1990, pp. 167–193.
- [13] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: science and systems*, vol. 2, 2009.