LUMPI README

Steffen Busch

May 2022

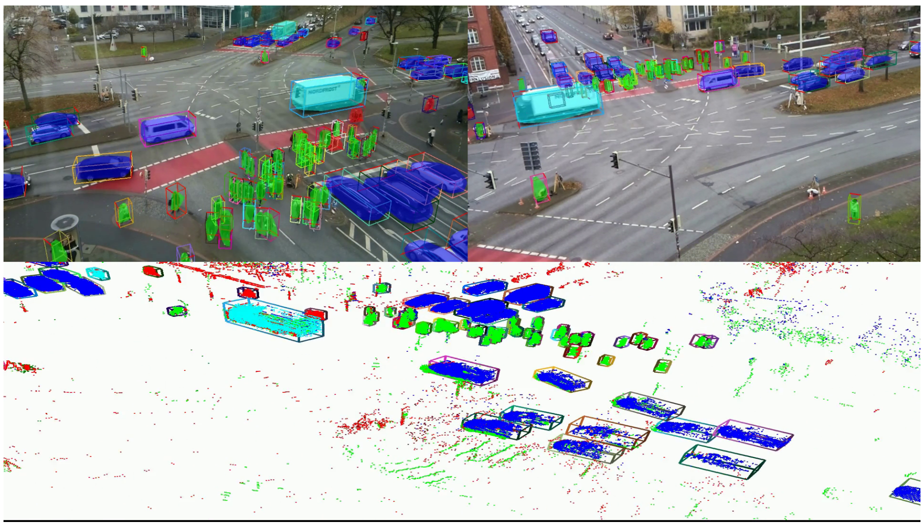

Figure 1: Impression

1 File Structure

```
LUMPI ............................................ ROOT DIRECTORY
  ├─ meta.json
  └─ MeasurementX ............. X IS THE MEASUREMENT NUMBER 0-6
       ├─ lidar
       │    ├─ 000000.ply
       │    ├─ 000001.ply
       │    ├─ ....
       │    ├─ 012000.ply
       │    └─ ......
       ├─ cam
       │    ├─ 5 ................................... CAMERA DEVICE ID
       │    │    └─ video.mp4
       │    ├─ 6
       │    └─ 7
       └─ background ........................... DISTANCE THRESHOLD
            ├─ XX.npy .............. XX IS SEASON NUMBER OF A LIDAR
            ├─ XY.npy ....... XY IS SEASON NUMBER OF A OTHER LIDAR
            └─ meta_background.txt ...................... RESOLUTIONS
```

2 Sensors

The dataset was recorded utilizing different camera and LiDAR setups. In this section, the used hardware is shortly introduced.

2.1 Cameras

The measurement was conducted with the following three different cameras. Table 1 summarizes the technical information of the cameras:

- Raspberry Pi Camera Module v2 (*PiCam*): This camera features an 8.08 megapixel Sony IMX219 image sensor with a size of 4.60 mm (1/4"), a resolution of 3280 × 2464 pixels and a pixel size of 1.12 µm. It offers a horizontal field of view (FOV) of 62.2 degrees. The output video resolution was set to 1640 x 1232 pixels, recording with a frame rate of 25 as well as 30 Hz [9].
- Dream Chip Technologies ATOM One (*ATOMOne*): This camera is based on a 2.35 megapixel Sony IMX174 image sensor with a size of 13.4 mm (1/1.2"), a resolution of 1936 x 1216 pixels and a pixel size of 5.86 µm. It offers a horizontal field of view of 89 degrees. The output video resolution was set to 1920 x 1080 pixels, recording with a frame rate of 50 Hz [4].
- Xiaomi Yi Action Camera (*YiCam*): This camera includes a 16.35 megapixel Sony IMX206 image sensor with a size of 7.77 mm (1/2.3"), a resolution

of 4672 x 3500 pixels and a pixel size of 1.34 µm. It offers a horizontal field of view of 155 degrees. The output video resolution was set to 1920 x 1080 pixels, recording with a frame rate of 30 Hz [13].

Table 1: Overview: Cameras

Camera	ID	Resolution	Frame rate	FOV (Horiz.)
PiCam	6,7	1640 x 1232 px	25/30 Hz	62.2°
ATOMOne	8	1920 x 1080 px	50 Hz	89°
YiCam	5	1920 x 1080 px	30 Hz	155°

2.2 LiDARs

During the measurement the following four different LiDAR models were used:

- Velodyne VLP-16 (*VLP16*): This LiDAR possesses 16 channels in a vertical field of view of 30° with an uniform vertical resolution of 2°. The horizontal resolution is 0.2°, the measurement range 100 m and the range accuracy ± 3 cm [12].

- Velodyne HDL64-S3 (*HDL64*): This 64-channel LiDAR operates with a predominantly downward pointing vertical FOV of around 26.5° (-24.6° to +1.9°) and a vertical resolution of about 0.4°. The horizontal resolution is 0.16°, the range 120 m and the range accuracy ± 2 cm [11].

- Hesai Pandar64 (*Pandar64*): The Pandar64 also provides 64 channels. However, within the vertical FOV from -25° to +15°, those are spaced non-uniformly. In the center, the angular spacing is 0.167°, but moving to the edge of the vertical FOV, it increases gradually up to 6°. The horizontal resolution is 0.2°, the measurement range between 0.3 m and 200 m and the range accuracy ± 5 cm for a range from 0.3 m to 1 m and ± 2 cm for a range from 1 m to 200 m, respectively [5].

- Hesai PandarQT (*PandarQT*): The PandarQT has 64 channels, which cover a vertical FOV of 104.2° (-52.1° to +52.1°) with a gradually decreasing resolution from 1.45° in the center to 2.338° at the edge of the FOV. The horizontal resolution is 0.6°, the measurement range between 0.1 m and 20 m and the range accuracy ± 3 cm [6].

All LiDARs have a horizontal field of view of 360° and operate with a rotation rate of 10 Hz. Apart from the Hesai PandarQT, all LiDARs are capable of direct GPS time synchronisation. Table 2 summarizes the technical specifications of the LiDAR sensors.

Table 2: Overview: LiDAR sensors

LiDAR	ID	Channels	FOV (Vert.)	FOV (Horiz.)	Res. (Vert.)	Res. (Horiz.)	Rate	Range	Range acc.	GPS time
VLP16	3,4	16	30°	360°	2°	0.2°	10 Hz	100 m	± 3 cm	✓
HDL64	0	64	26.5°	360°	0.4°	0.16°	10 Hz	120 m	± 2 cm	✓
Pandar64	1	64	40°	360°	0.167° to 6°	0.2°	10 Hz	200 m	± 2 / 5 cm	✓
PandarQT	2	64	104.2°	360°	1.45° to 2.338°	0.6°	10 Hz	20 m	± 3 cm	✗

2.3 Mobile Mapping System

In addition to the cameras and LiDARs, a van equipped with a RIEGL VMX-250 Mobile Mapping System (MMS) [10] drove multiple times in all possible configurations across the intersection. Doing so, highly accurate point clouds were gathered by two built-in RIEGL VQ-250 LiDARs, which measure up to 300,000 points per second with a ranging accuracy of ten millimeters. Simultaneously, a reference trajectory is acquired by the Applanix POS LV GNSS/IMU system, which is refined in a post-processing using correction data from the satellite positioning service SAPOS. The acquired point clouds are aligned using the adjustment strategy from [2] to generate a globally georeferenced high-density reference map point cloud, as shown in Figure **??** and hereinafter referred to as *ReferenceMapPC*.

2.4 Data collection

In this subsection, the setup of the previously introduced sensors is described for each measurement. Overall, seven measurements were conducted on three different days with varying weather conditions. Table 3 gives an overview of the sensors used in the specific measurements.

Table 3: Overview: Sensor setup during experiments

Exp. ID	0	1	2	3	4	5	6
LiDARs							
VLP-16	✓	✓	✓	✓	✓	✓	✓
VLP-16	✓	✓	✓	✓	✓	✓	✓
HDL64	✓	✓	✓	✓	✓	✓	✓
Pandar64	✗	✗	✗	✗	✓	✓	✓
PandarQT	✗	✗	✗	✗	✓	✓	✓
Cameras							
PiCam #1	✓	✓	✓	✓	✓	✓	✓
PiCam #2	✗	✗	✗	✗	✓	✓	✓
ATOMOne	✓	✓	✓	✓	✗	✗	✗
YiCam	✗	✗	✗	✗	✗	✓	(✓)
Duration (in min)	40	25	25	8	14	13	20

3 Format and Files

The dataset, the documentation and a visualization tool – as an example of how to access the data – are provided at `https://data.uni-hannover.de/de/dataset/lumpi` (DOI: 10.25835/z54qcu1b) under the Creative Commons Attribution-NonCommercial 3.0 license. In detail, the dataset includes the following files.

3.1 Orthophoto

A georeferenced aerial image with a ground sampling distance of 20 cm for the location of the *Königsworther Platz* and surrounding, provided by the city of Hanover [7], in the form of raster data: *jpg + jgw*.

3.2 Road Network Map

A georeferenced ASAM OpenDRIVE [1] map saved in the *xodr* format containing detailed information about the road network at the the *Königsworther Platz*.

3.3 Reference Point Cloud

The *ReferenceMapPC* introduced in Chapter 2.3 is stored in a binary *ply* file containing the following information:

x,y,z are the coordinates of a point in meters in the UTM coordinate frame, and

reflectance is the reflectance value provided by the sensor.

3.4 Point Clouds

The aligned point clouds of each scanner are jointly saved with 10 Hz into *ply* binary files. Each point is saved with the following information:

x,y,z are the coordinates of a point in meters,

time is the recording time in μs (each measurement starting at zero),

id is the session id of the scanner,

intensity is the reflectance provided by the sensors,

ray is the ray, which refers to the elevation angle,

azimuth is the actual rotation angle of the scanner, and

distance is the raw distance measurement of the scanner.

3.5 Videos

The recorded videos for each camera are separately stored as *mp4* files using the *h.264* video codec. The corresponding recording frequencies are documented in the *Metadata*.

3.6 Background

Provides thresholds for each LiDAR beam to identify the foreground. The distance threshold for each point is stored as two dimensional *numpy* array, where the ray refers to the row and the azimuth divided by the horizontal resolution refers to the corresponding column.

3.7 Metadata

Provides information about the sensor setup in *json* format. The following information for each sensor are available:

extrinsic represents the 4×4 projection matrix from sensor frame to UTM frame,

type is the type: *camera* or *lidar*,

measurementId is the id of the measurement,

sessionId is the unique id for an measurement and device combination,

fps is the recording frequency in Hz,

angles is a list of elevation angles (LiDAR sensors only), and

tvec, rvec provide the two 3×1 vectors for the projection of points into the camera image (cameras only).

The meta data is structured in 3 dictionaries:

session sensor information accessible by sessionId,

device all measurements for a device, deviceId refers to measurementId and then to sessionId, and

measurement all sensors for an measurement, measurementId refers to deviceIds and then to sessionIds.

3.8 Label Format

Each object is tracked with its 3D as well as 2D bounding box, class and mask. The 3D bounding box and track information are given as comma separated text files (csv). A 3D bounding box is encoded with id, length, width, height. The track is encoded by the frame id, object id and a projection matrix (4×4) for the bounding box pose. The 2D bounding boxes are stored in the *MOT* benchmark format [3]: frame, id, tl.x, tl.y, w, h, confidence, classId, visibility. The masks are stored in an image sequence (mp4) for each camera and are accessible by the 2D bounding box. We picked a subset of the COCO [8] classes, namely person (1), car (2), bicycle (3), motorcycle (4), bus (5) and truck (6).

4 Syncronization

All LIDARs are synchronized via GNNS puls per second and delivered a time stamp fo each point in μs. Each measurement start at time 0 with camerea frame 0. The correct camera frame for a point could be calculatet by

$$f = \lfloor x \times 10^{-6} \times y \rfloor \tag{1}$$

where: x=point time stamp
y=camera fps
f=frame

References

[1] ASAM e.v. *ASAM OpenDRIVE*. Accessed on 1st February 2022. URL: https://www.asam.net/standards/detail/opendrive/.

[2] Claus Brenner. "Scalable Estimation of Precision Maps in a MapReduce Framework". In: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '16. Burlingame, California: Association for Computing Machinery, 2016. ISBN: 9781450345897. DOI: 10.1145/2996913.2996990.

[3] Patrick Dendorfer et al. "Mot20: A benchmark for multi object tracking in crowded scenes". In: *arXiv preprint arXiv:2003.09003* (2020).

[4] Dream Chip Technologies GmbH. *ATOM one*. Accessed on 1st February 2022. URL: https://www.dreamchip.de/products/atom-one-family/atom-one.html.

[5] Hesai Technology Co., Ltd. *Pandar64 64-Channel Mechanical LiDAR User Manual*. 2021.

[6] Hesai Technology Co., Ltd. *PandarQT 64-Channel Short-Range Mechanical LiDAR User Manual*. 2021.

[7] Landeshauptstadt Hannover. *Open GeoData: Digitale Orthophotos (DOP20)*. Accessed on 1st February 2022. 2020. URL: https://www.hannover.de/Leben-in-der-Region-Hannover/Verwaltungen-Kommunen/Die-Verwaltung-der-Landeshauptstadt-Hannover/Dezernate-und-Fachbereiche-der-LHH/Stadtentwicklung-und-Bauen/Fachbereich-Planen-und-Stadtentwicklung/Geoinformation/Open-GeoData/Digitale-Orthophotos-DOP20.

[8] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[9] Raspberry Pi Ltd. *Raspberry Pi Documentation*. Accessed on 1st February 2022. URL: https://www.raspberrypi.com/documentation/accessories/camera.html#hardware-specification.

[10] RIEGL Laser Measurement Systems GmbH. *RIEGL VMX-250*. Accessed on 1st February 2022. 2012. URL: http://www.riegl.com/uploads/tx_pxpriegldownloads/10_DataSheet_VMX-250_20-09-2012.pdf.

[11] Velodyne LiDAR, Inc. *Velodyne LiDAR HDL-64E S3 HIGH DEFINITION REAL-TIME 3D LIDAR*. 2018.

[12] Velodyne LiDAR, Inc. *VLP-16 User Manual*. Accessed on 1st February 2022. 2019. URL: https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf.

[13] XiaoYi Technology Co., LTD. *YI Action Camera Technical Specs*. Accessed on 1st February 2022. URL: https://www.xiaoyi.com/en/specs_en.html.