

# NTU VIRAL: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint

Journal Title  
XX(X):1–10  
©The Author(s) 2016  
Reprints and permission:  
[sagepub.co.uk/journalsPermissions.nav](http://sagepub.co.uk/journalsPermissions.nav)  
DOI: 10.1177/ToBeAssigned  
[www.sagepub.com/](http://www.sagepub.com/)

SAGE

Thien-Minh Nguyen<sup>1</sup>, Shenghai Yuan<sup>1</sup>, Muqing Cao<sup>1</sup>, Yang Lyu<sup>1</sup>, Thien Hoang Nguyen<sup>1</sup>, Lihua Xie<sup>1</sup>

## Abstract

In recent years, autonomous robots have become ubiquitous in research and daily life. Among many factors, public datasets play an important role in the progress of this field, as they waive the tall order of initial investment in hardware and manpower. However, for research on autonomous aerial systems, there appears to be a relative lack of public datasets on par with those used for autonomous driving and ground robots. Thus, to fill in this gap, we conduct a data collection exercise on an aerial platform equipped with an extensive and unique set of sensors: two 3D lidars, two hardware-synchronized global-shutter cameras, multiple Inertial Measurement Units (IMUs), and especially, multiple Ultra-wideband (UWB) ranging units. The comprehensive sensor suite resembles that of an autonomous driving car, but features distinct and challenging characteristics of aerial operations. We record multiple datasets in several challenging indoor and outdoor conditions. Calibration results and ground truth from a high-accuracy laser tracker are also included in each package. All resources can be accessed via our webpage [https://ntu-aris.github.io/ntu\\_viral\\_dataset/](https://ntu-aris.github.io/ntu_viral_dataset/).

## Keywords

Dataset, Aerial Robot, Autonomous System, Simultaneous Localization and Mapping

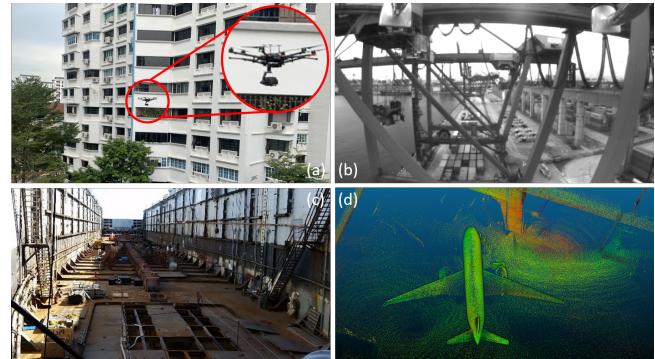
## 1 Introduction

Over the years, autonomous systems have made significant progress. One of the most crucial factors for these advancements can be attributed to the public data suites. On one hand, these datasets can waive the inhibitive requirements on budget and manpower, e.g. hardware development, calibration, field operations, etc. for individual researchers. Hence, researchers can easily investigate new navigation schemes and put them to test across a variety of scenarios and environments. On the other hand, the benchmark tools also help streamline the verification, evaluation and comparison between proposed methods, thus allowing navigation methods to be fairly rated and ranked based on a common basis.

Indeed, there are many datasets for a variety of tasks on autonomous vehicles, e.g. object recognition, stereo depth perception, scene understanding, traffic analysis, etc. In this paper we only focus on the datasets that can be used to investigate navigation capability of aerial vehicles (AVs) in GPS-denied environments, Fig. 1 illustrates a range of such environments where the application of autonomous AV system is being investigated. We review some of the datasets that are deemed most relevant to our targeted applications above. Table 1 gives a summary of these datasets and their features, to the best of our knowledge.

### 1.1 Related works: the ground-air dichotomy

From Table 1, it is clear that there exist two distinct groups of datasets. For those collected from cars or ground robots, a large number of sensors is often present, especially lidar. For example, the MIT DARPA dataset is comprised of 12 2D



**Figure 1.** Typical inspection applications for autonomous AVs in GPS-denied environments: (a) building complex, (b) cranes, (c) cargo ships (d) hangar

lidars and 1 64-channel lidar, along with a significant number of cameras. Over the years, there appears to be a decline of interest in 2D lidar (whose role has been gradually replaced by 3D lidar), along with a reduced number of cameras. This reflects the trajectory of autonomous driving research in the last decade, where in the beginning sensors had limited capability and variety, thus quality was compensated by quantity. As the field progresses, sensors have become much

<sup>1</sup>Nanyang Technological University, Singapore

#### Corresponding author:

Lihua Xie, School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore, 639798  
Email: lhxie@ntu.edu.sg

**Table 1.** Notable public datasets divided by their ground and air focuses.

Dataset	Sensor used				Ground truth	Mobile Platform
	IMU	Camera	Lidar	UWB		
MIT DARPA, <a href="#">Huang et al. (2010)</a>	N/A	4 Point Grey: 4x376x240 @10Hz 1 Point Grey: 752x480 @22.8Hz	3D Velodyne HDL-64E @15Hz 12 2D-SICK @75Hz	N/A	GPS/INS	Car
Ford Campus, <a href="#">Pandey et al. (2011)</a>	acc/gyr @100Hz	LadyBug 3: 6x1600x600 @8Hz	3D-Velodyne HDL-64E @10Hz 2 2D-Riegl LMS	N/A	GPS/INS	Car
KITTI, <a href="#">Geiger et al. (2013)</a>	acc/gyr @10Hz	2 Point Grey (gray): 2x1392x512 @10Hz 2 Point Grey (color): 2x1392x512 @10Hz	3D-Velodyne HDL-64E @10Hz	N/A	RTK GPS/INS	Car
NCLT, <a href="#">Carlevaris-Bianco et al. (2016)</a>	acc/gyr @100Hz	LadyBug 3: 6x1600x1200 @5Hz	3D-Velodyne HDL-32E @10Hz 2 2D-Hokuyo @10-40Hz	N/A	RTK-GPS LIDAR SLAM	Wheeled Robot
Oxford RobotCar, <a href="#">Maddern et al. (2017)</a>	acc/gyr @50Hz	BumbleBee: 2x1280x960 @16Hz 3 Grasshopper2: 3x1024x1024 @11.1Hz	2 2D-SICK @50Hz 3D-SICK @12.5Hz	N/A	GPS/INS	Car
KAIST Urban, <a href="#">Jeong et al. (2019)</a>	acc/gyr @200Hz	FLIR (color): 2x1280x560 @10Hz	2 3D-Velodyne-16 @10Hz 2 2D-SICK @100Hz	N/A	SLAM	Car
Newer College, <a href="#">Ramezani et al. (2020)</a>	acc/gyr @650Hz	D435i (Infrared): 2x848x480 @30Hz	3D-Ouster-64 @10Hz	N/A	6DOF ICP	Handheld
NTU VIRAL (Ours)	acc/gyr/mag @385Hz	IDS (gray): 2x752x480@10Hz,	2 3D-Ouster-16 @10Hz	4 on UAV 3 anchors	3D Laser Tracker	UAV
UMA-VI, <a href="#">Zuñiga-Noël et al. (2020)</a>	acc/gyr @250Hz	BumbleBee: 2x1024x768 @12.5Hz IDS (gray): 2x752x480 @25Hz,	N/A	N/A	SLAM	Handheld
UZH FPV, <a href="#">Delmerico et al. (2019)</a>	acc/gyr @500/1000Hz	Fisheye stereo: 2x640x480 @30Hz mDAVIS: 346x260 + event @50Hz	N/A	N/A	3D Laser Tracker	UAV
TUM VI, <a href="#">Schubert et al. (2018)</a>	acc/gyr @200Hz	IDS (gray): 2x1024x1024 @20Hz	N/A	N/A	6DOF MoCap	Handheld
Upenn Fast Flight, <a href="#">Sun et al. (2018)</a>	acc/gyr @200Hz	FLIR (gray): 2x960x800 @40Hz	N/A	N/A	GPS	UAV
Zurich Urban MAV, <a href="#">Majdik et al. (2017)</a>	acc/gyr @10Hz	GoPro (color): 1920x1080 @30Hz	N/A	N/A	Aerial Photogrammetry	UAV
EuRoC, <a href="#">Burri et al. (2016)</a>	acc/gyr @200Hz	2 MT9V034: 2x752x480 @20Hz	N/A	N/A	6DOF MoCap / 3D Laser Tracker	UAV

more compact and efficient, along with better software and algorithms.

For aerial platforms, also referred to as *Unmanned Aerial Vehicles (UAVs)*, *Micro Aerial Vehicles (MAVs)*, *aerial robots* or *drones* in the literature, we can see a clear contrast to the ground-based datasets. In this case, due to the payload constraint, 3D lidar is often left out of consideration. Instead, the focus is often on high frame-rate camera systems for self-localization capability at high speed and aggressive maneuvers, often under low lighting conditions, with the goal of pushing the capability of Visual-Inertial Odometry (VIO) and vision-based Simultaneous Localization and Mapping (SLAM) systems to cover ever more extreme conditions.

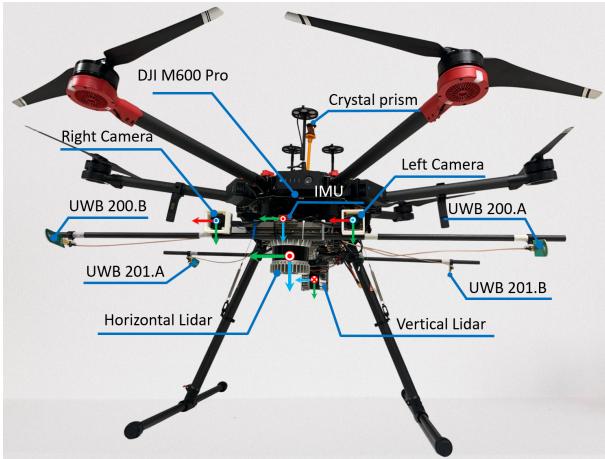
It can be seen that the dataset we are presenting in this paper falls in the middle of the two aforementioned classes. For those datasets using cars and wheeled robots, the vehicle's trajectory is often close to the ground plane, without much abrupt translation or rotation motions. This is an advantage that has been well exploited in many navigation systems, especially the lidar-centric ones. However, they may no longer be effective with drastic changes in roll, pitch and vertical motions commonly seen in a UAV dataset. Besides, due to their long endurance, ground vehicle datasets can be conducted over a long period of time and a large environment, and have access to GPS. In these cases GPS data can be used for loop closure and/or ground truth, however it has significant error and reduces the reliability and comparability of the analysis. In contrast, the UAV datasets often have a shorter timespan and cover a smaller environment, which make them less

representative of real-world scenarios. However, for semi-controlled environment, UAV datasets can feature some artificial landmarks, i.e. visual markers or ranging anchors, to help reduce localization drift, which is applicable in most targeted industry inspection applications. Moreover, we can also employ high-accuracy laser tracking methods to provide ground truth, which is desirable for a stringent analysis and comparison of the localization methods.

## 1.2 A new dataset for autonomous drone.

Despite the aforementioned differences in the two classes of datasets, there is no doubt on their usefulness for the intended applications. As mentioned earlier, for the applications of aerial vehicles in our interest (Fig. 1), there appears to be an absence of compatible public dataset for the relevant scenarios. This motivates us to construct a novel sensor setup and offer a new benchmark suite for advanced autonomous aerial systems that can be adopted by the industry in the near future. Fig. 2 gives an overview of sensor setup.

Briefly speaking, we employ a high-frequency IMU and a stereo camera rig typical of UAV systems (Sec. 2.1, Sec. 2.3). Next, thanks to the current advances in lidar hardware that have brought down its size and price, we find that it is due time 3D lidar can be popularly used on aerial platforms, thanks to many benefits of 3D lidar over a pure vision system. Hence, we integrate two 3D lidars on our UAV platform. Different from previous datasets with 3D lidar mounted on ground vehicles, ours features a much more complex motion in 3D space with frequent drastic rotational and translational moves. More details on the lidars can be found in Sec. 2.2.



**Figure 2.** The drone setup used for the data collection. A DJI M600 Hexacopter is used to carry the payload comprising of 1 IMU, 2 lidars, 2 Cameras, 4 UWB ranging nodes, and a crystal prism to be tracked by a total station to provide ground truth.

In addition to the aforementioned sensors, as highlighted in Table 1, our dataset also features UWB range measurements. Indeed, it is a consensus that drift is an intrinsic feature of onboard self-localization. However, our recent works have shown that by using range measurements to some fixed landmarks in an environment, one can reduce, or even eliminate such drift Nguyen et al. (2020b,a). For semi-controlled environments such landmarks can be easily deployed without much effort. More information on this sensor is given in Sec. 2.4.

Finally, we employ a laser tracking system to provide high-accuracy<sup>1</sup> ground truth for the position estimate. The experiments are conducted at various environments on the campus of Nanyang Technological University (NTU), including both indoor and outdoor conditions. The datasets are also recorded in rosbag format, which can be directly used on Robot Operation System (ROS). We also include some codes that can be used for the analysis of the localization information using our datasets. Several plug-and-play examples of using state-of-the-art localization methods with our datasets such as OpenVINS Geneva et al. (2020), VINS-Fusion Qin et al. (2018), A-LOAM Zhang and Singh (2018), LIO-SAM Shan et al. (2020), M-LOAM Jiao et al. (2020) are also provided. All of these resources can be found on our data suite’s web page [https://ntu-aris.github.io/ntu\\_viral\\_dataset/](https://ntu-aris.github.io/ntu_viral_dataset/).

The rest of this paper is organized as follows: Sec. 2 presents our hardware setup and the intended purpose for each sensor. Sec. 3 presents the description of the environments and the flight tests. Sec. 4 describes the organization, format, definitions and convention used in the datasets. Sec. 5 explains the calibration procedures. Sec. 6 describes our recommendation for evaluation. Sec. 7 lists out the known issues in our implementation.

## 2 Sensor setup

A DJI M600 Pro<sup>2</sup> hexacopter is used to carry the sensor setup (Fig. 2). Table 2 provides a summary of the sensors and their corresponding specifications. All of the messages

are timestamped by their *publish time* on ROS. Below, we provide a more detailed description of these sensors.

### 2.1 IMU:

A VectorNav VN100<sup>3</sup> rugged IMU is employed as the main inertial sensor in our system. It is also chosen to be the center of the *body frame* where the extrinsics of other sensors are referenced to. The sensor is configured to publish data at 400Hz, though the effective rate is roughly 385Hz. Note that besides the angular velocity and acceleration measurements, the topic /imu imu also contains the orientation estimate by the device’s internal Extended Kalman Filter, which also fuses magnetometer with the aforementioned measurements.

### 2.2 3D lidars:

In this work, two 16-channel OS1 gen1<sup>4</sup> lidars are configured so that the so-called *horizontal lidar* can scan the objects in the front, back, left, and right sides of the UAV, while the other so-called *vertical lidar* can scan the ground, front and back sides. Thus, the two can complement each other well to maintain good observation on the environment. Note that each lidar has an internal IMU that outputs angular rate and acceleration measurements at 100Hz rate. Each lidar is loaded with the latest V2 firmware for additional fields such as reflectivity, time, and ambient. These quantities are also recorded in the bag files.

### 2.3 Stereo cameras

Two uEye 1221 LE<sup>5</sup> monochrome-global-shutter cameras are mounted on the drone, facing directly forward. The two are synchronized by an external trigger to capture and publish image at almost the same time. *Typically, the difference in the timestamps of the images triggered at the same time is below 3 ms.* This is necessary if one needs to set a threshold to synchronize the images in the buffer for stereo processing such as in VINS-Fusion Qin et al. (2018).

### 2.4 UWB ranging sensors

The UWB ranging network is setup in a similar manner with our previous works Nguyen et al. (2020b, 2021a, 2018). Two Humatics P440 UWB radios<sup>6</sup>, given the IDs 200 and 201 in the network, are mounted on the UAV, each has two antennae A and B extended to the four corners. Hence we have a total of 4 ranging nodes on the UAV: 200.A, 200.B, 201.A, 201.B (see Fig. 2). Another three UWB nodes with IDs 100, 101, 102 are used as anchors. As such in total we have 12 UAV-to-anchor ranging pairs.

The ranging sequence is programmed in a way such that all 12 ranging pairs are executed equally. Each ranging step takes 25 ms. Hence with only one UAV ranging node,

<sup>1</sup>[https://github.com/ntu-aris/ntu\\_viral\\_dataset/blob/gh-pages/docs/Leica\\_Nova\\_MS60\\_DS.pdf](https://github.com/ntu-aris/ntu_viral_dataset/blob/gh-pages/docs/Leica_Nova_MS60_DS.pdf)

<sup>2</sup><https://www.dji.com/sg/matrice600-pro>

<sup>3</sup><https://www.vectornav.com/products/VN-100>

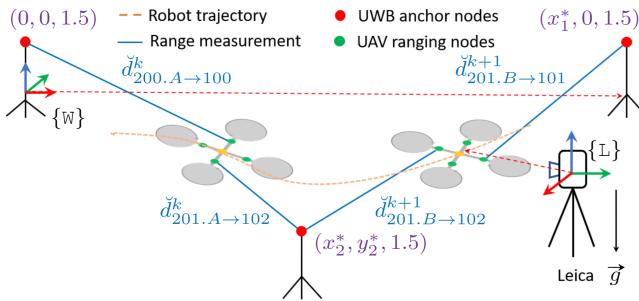
<sup>4</sup><https://ouster.com/products/os1-lidar-sensor>

<sup>5</sup><https://en.ids-imaging.com/store/ui-1221le-rev-2.html>

<sup>6</sup><https://humatics.com>

**Table 2.** The sensors used in this dataset and their corresponding specifications

No.	Sensor	Model	Topic name	Message Type	Rate (Hz)
1	IMU (Sec. 2.1)	VectorNav (VN100)	/imu imu	sensor_msgs/Imu	385
			/imu/magnetic_field	sensor_msgs/MagneticField	385
			/imu/temperature	sensor_msgs/Temperature	385
2	Horizontal Lidar (Sec. 2.2)	OS1-16 Gen1	/os1_cloud.node1 imu	sensor_msgs/Imu	100
			/os1_cloud.node1 points	sensor_msgs/PointCloud2	10
3	Vertical Lidar (Sec. 2.2)	OS1-16 Gen1	/os1_cloud.node2 imu	sensor_msgs/Imu	100
			/os1_cloud.node2 points	sensor_msgs/PointCloud2	10
4	Camera 1 (Sec. 2.3)	uEye 1221 LE (monochrome)	/left/image_raw	sensor_msgs/Image	10
5	Camera 2 (Sec. 2.3)	uEye 1221 LE (monochrome)	/right/image_raw	sensor_msgs/Image	10
6	UWB Sensors (Sec. 2.4)	Humatics P440	/uwb_endorange_info	uwb_driver/UwbRange	68.571
			/uwb_exorange_info	uwb_driver/UwbEcho	5.714
			/nodes_pos_sc	nav_msgs/Path	5.714
7	3D Laser Tracker (Sec. 2.5)	MS60 Leica TotalStation	/leica/pose/relative	geometry_msgs/PoseStamped (orientation is not set)	20

**Figure 3.** Illustration of the ranging scheme and the external frame of references. Refer to Sec. 2.4 for detailed descriptions.

ideally one can expect 40 Hz of ranging measurements on the topic `/uwb_endorange_info`. However, in our case, there are two ranging nodes (we program node 200 to range to one anchor while 201 ranges to another), thus we can obtain twice the amount of range measurements, i.e. 80 Hz. Nevertheless, we reserve some steps in the sequence to let the anchors range among themselves and broadcast the measurements. Therefore the rate of the UAV-to-anchor ranges, i.e. the `/uwb_endorange_info` topic is reduced to 68.571 Hz, while the rate of anchor-to-anchor ranges, published over the `/uwb_exorange_info` topic, is 5.714 Hz, as shown in Tab. 2. Fig. 3 gives a simple illustration of the ranging incidents over two time steps. Note that due to the fact that the transmission power of the UWB depends on the relative orientation between the antennae of the two nodes, some ranging steps may fail to return, or return an unreliable measurements. Thus the effective rate of UWB measurement can vary through time, depending on the position of the UAV relative to the anchors during flight.

The anchors are placed on tripods and they are adjusted to have the same height from the ground. Thus, their nominal coordinate is 1.5m. If we select anchor 100 to be at the origin, and anchor 101 to be on the +x direction, and +z direction is from the ground to the anchor nodes, then we can establish

a coordinate system using the right hand rule (the frame  $\{W\}$  in Fig. 3). Using this arrangement and the distances from the topic `/uwb_exorange_info` we can easily estimate the coordinates  $x_1^*$ ,  $x_2^*$ ,  $y_2^*$ . The coordinates of the anchor nodes are published in the `/nodes_pos_sc` topic.

## 2.5 Ground truth

A Leica Nova MS60 MultiStation<sup>7</sup> is used to track a crystal prism mounted on the top of the UAV to provide ground truth for position estimate. Note that the coordinate frame of this ground truth system is aligned with gravity during the startup, and its z axis points at the opposite direction of the gravity (Fig. 3). Moreover, since there is a significant displacement between the prism and the body frame's origin, the accuracy analysis needs to take this into account. Sec. 6 discusses this issue in details.

## 3 Dataset characteristics

The datasets are divided into three groups based on the environments, namely the EEE, SBS and NYA sequences. The environments include both indoor and outdoor locations on NTU campus. Fig. 4 presents some overviews of these environments. Tab. 3 gives a brief summary of the sequences and their statistics.

### 3.1 The EEE sequences

Three datasets are collected at the carpark in the center of the School of EEE, NTU, hence named the *EEE sequences*. The area is surrounded by tall building blocks on all side which can be conducive for lidar-based SLAM. Also, reliable visual features can be detected on nearby objects such as trees, road markings, and buildings (see Fig. 4a).

<sup>7</sup><https://Leica-geosystems.com/en-sg/products/total-stations/multistation/Leica-nova-ms60>



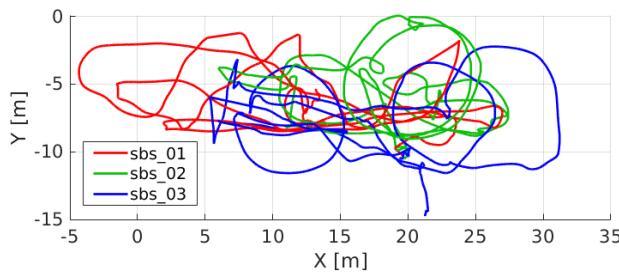
**Figure 4.** Environments where the datasets are collected

**Table 3.** Some statistics of each sequence

Name	Time	Path Length	Average Vel.	Average Ang. Vel
eee_01	398.7 s	237.07 m	0.677 m/s	0.119 rad/s
eee_02	321.1 s	171.07 m	0.585 m/s	0.101 rad/s
eee_03	181.4 s	127.83 m	0.800 m/s	0.184 rad/s
nya_01	396.3 s	160.30 m	0.463 m/s	0.150 rad/s
nya_02	428.7 s	249.10 m	0.657 m/s	0.141 rad/s
nya_03	411.2 s	315.46 m	0.821 m/s	0.125 rad/s
sbs_01	354.2 s	202.30 m	0.625 m/s	0.101 rad/s
sbs_02	373.3 s	183.57 m	0.573 m/s	0.145 rad/s
sbs_03	389.3 s	198.54 m	0.601 m/s	0.120 rad/s

### 3.2 The SBS sequences

The *SBS sequences* are collected at an open square next to the School of Biological Sciences, NTU. This area is surrounded by some low-rise buildings with large glass surfaces. Also, visual features may only be detected on objects far way, which can produce noisy depth. The drone's trajectories in these sequences are shown in Fig. 5.



**Figure 5.** Trajectories of the AV in the SBS sequences as recorded by the Leica laser tracker.

### 3.3 The NYA sequences

The auditorium at NTU, shown in Figure 4c, represents a challenging environment for visual SLAM. These datasets feature some significant challenges. For e.g. semi-transparent surfaces can be an issue for lidar SLAM, while flight dynamics and low lighting conditions are difficult for visual SLAM. Moreover, we also notice significant multi-path effects and signal loss on the UWB measurements (Fig. 6).

## 4 Dataset format

### 4.1 Files organization

The datasets are recorded into rosbag files during the flight, and saved on the memory of the drone's onboard computer (DJI Manifold 2C). The ground truth data is recorded on a separate computer, and is later temporally synchronized and merged with the bag from the drone's onboard computer, yielding a single rosbag for each flight test.

Each rosbag file is accompanied by a set of files containing the calibration parameters for each sensor. Fig. 7 illustrates the content of such a calibration file. These files document the coefficients of the camera model, sensor-to-body coordinate transforms, IMU noise, camera-IMU time delay, and can be parsed using opencv and ROS APIs. In addition, some accessories to utilize the pointclouds and UWB data are also provided with links on the data suite's web page.

### 4.2 Units and timestamps

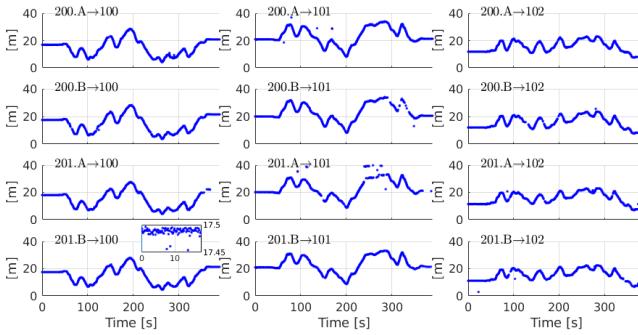
All of the measurements are reported in SI units except the timestamps of the messages are in `ros/Time`<sup>8</sup> format, which can be found in the `header.stamp` field of the messages. The timestamps are assigned by each sensor's driver when the message is published over ROS.

In addition, the timestamp of each lidar pointcloud message matches with the *the end-time of the scan*. Moreover, the pointcloud also contains the timestamp of each point relative to *the start-time of the scan*. This will be useful if one seeks to perform deskew operation on the pointcloud.

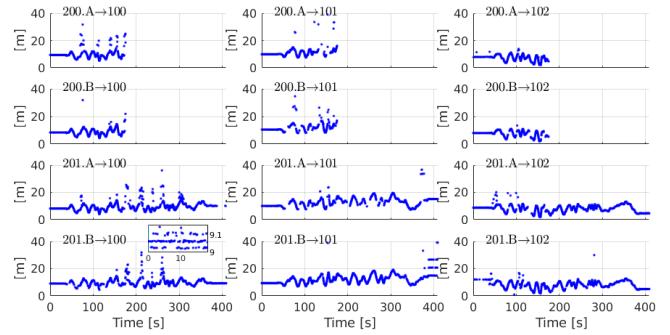
### 4.3 Message types

The message types for most sensors are standard ROS messages, however there are some custom definitions for UWB. Specifically, for the UWB messages of type `uwb_driver/UwbRange`, besides the distance measurement, the message also contains other information such as signal over noise ratio, line-of-sight self-diagnosis, antenna coordinates in the body frame, and the anchor coordinates, which are fixed by using the first messages in `/nodes_pos_sc`. The source code that defines these message can be found on our data suite's web page [https://ntu-aris.github.io/ntu\\_viral\\_dataset](https://ntu-aris.github.io/ntu_viral_dataset).

<sup>8</sup><http://wiki.ros.org/roscpp/Overview/Time>



(a) UWB ranges in sbs\_03 dataset. The zoom-in plot shows a 5 cm fluctuation in measurement when the drone was static.



(b) UWB ranges in nya\_03 dataset, showing signal loss by UWB 200 mid-flight, and multi-path effect on all ranging edges

**Figure 6.** UWB ranges in outdoor and indoor conditions

```
%YAML:1.0
dataset_name: eee_01
image_topic: "/right/image_raw"
model_type: PINHOLE
camera_name: camera
image_width: 752
image_height: 480

distortion_model: radial-tangential
distortion_parameters:
  k1: -0.300267420221178
  k2: 0.090544063693053
  p1: 3.330220891093334e-05
  p2: 8.989607188457415e-05
projection_parameters:
  fx: 4.31336426577952e+02
  fy: 4.327527965378035e+02
  cx: 3.548956286992647e+02
  cy: 2.325508916495161e+02

T_Body_Cam: !!opencv-matrix
  rows: 4
  cols: 4
  dt: d
  data: [-0.01916508, -0.01496218, 0.99970437, 0.00519443,
  0.99974371, 0.01176483, 0.01934191, 0.1347802,
  -0.01205075, 0.99981884, 0.01473287, 0.01465067,
  0.00000000, 0.00000000, 0.00000000, 1.00000000]
# ROS tf equivalence:
# <node pkg="tf2_ros" type="static_transform_publisher"
#   name="T_Body_Cam"
#   args="1.5899639721385 0.0120510416491 1.5560618537049
#         0.00519443 0.1347802 0.01465067 Body_Cam"/>

# Estimated by Kalibr https://github.com/ethz-asl/kalibr
t_shift: -0.0202463834234 # t_imu = t_cam + t_shift
```

**Figure 7.** Content of the calibration file camera\_right.yaml of the eee\_01 dataset.

#### 4.4 Coordinate transformations

For common objects such as vectors, quaternions, rotation matrices recorded in the rosbag files, the ROS conventions are followed. Here we shall explain the extrinsics defined in the calibration files.

Let us take an instance to explain the convention used throughout the datasets. In reference to Fig. 7, notice that the transform  ${}^B\mathbf{T} \in SE(3) \subset \mathbb{R}^{4 \times 4}$  is declared via the parameter T\_Body\_Cam (the "body-cam" suffix follows a parent-child order between the two frames of reference, following the convention of ROS and the EuRoC dataset). This transform consists of the rotation  ${}^B\mathbf{R} \in SO(3) \subset \mathbb{R}^{3 \times 3}$  on the upper left corner and the translation  ${}^B\mathbf{t} \in \mathbb{R}^{3 \times 1}$  on the upper right corner, i.e.  ${}^B\mathbf{T} = \begin{bmatrix} {}^B\mathbf{R} & {}^B\mathbf{t} \\ 0 & 1 \end{bmatrix}$ .

For an object A observed in the camera frame {C} with position  ${}^A\mathbf{p}$  and orientation  ${}^A\mathbf{R}$ , its corresponding position and orientation in the body frame B, denoted as  ${}^B\mathbf{p}$  and  ${}^B\mathbf{R}$ , can be obtained by:

$${}^B\mathbf{p} = {}^B\mathbf{R} {}^A\mathbf{p} + {}^B\mathbf{t}, \quad {}^B\mathbf{R} = {}^B\mathbf{R} {}^A\mathbf{R}. \quad (1)$$

The object A in this case can be a 3D feature or an object that is being tracked by the camera or the lidar, which in a typical SLAM algorithm needs to be coupled with the extrinsics in the estimation process. Moreover, A can be another coordinate system that we need to convert the coordinates from other frames of references to.

## 5 Sensor calibration

Sensor fusion often requires prior knowledge of the spatial configuration of multiple sensors. This section introduces our calibration pipelines, which is divided into two parts, namely the rigid systems and flexible systems. The cameras and IMU are mounted on the same rigid titanium-alloy-based 3D-printed parts. Thus, their spatial relation has very small dynamic variance. The other sensors are flexibly mounted as they are connected via some carbon fiber tubes or dampers. These flexible parts will have large dynamic variance.

We first calibrate the rigid body system. The stereo cameras are calibrated both intrinsically and extrinsically. We then find the IMU to camera transform based on visual-inertial alignment. For the more flexible parts, the transform may come with more considerable variance in the spatial relation. Therefore, we mostly use the VICON system to calibrate between those sensors with some help from other priors such as recognizable planes in the lab. All of the calibration results can be verified using localization or mapping packages.

### 5.1 Stereo calibration

Our first goal is to find the intrinsic and extrinsic parameters of the stereo cameras using an approach similar to [Zhang \(2000\)](#). We collect N well-synchronized stereo images in front of a static calibration chessboard to minimize the motion blur. Using pinhole camera and radial-tangential distortion models, the intrinsics are calibrated using all of the image sequences on the MATLAB calibration toolbox. We set a threshold for the reprojection error and only use

the smaller subset of images for the extrinsics. We find that a large roll of chessboard pattern will enlarge the reprojection error for this 120-degree FOV lens. Therefore, we keep a small roll factor in calibration sequences. Finally, we verify the calibration result by comparing the stereo-matching reprojected depth to lidar measurement. The result can be well aligned with the lidar. The stereo calibration image sequence is provided in the dataset portal.

## 5.2 Visual inertial calibration

We calibrate and refine the intrinsic and extrinsic parameters of IMU and cameras by using the camera-IMU calibration procedure<sup>9</sup> of the Kalibr package Furgale et al. (2013). In this procedure, the initial guess of the camera intrinsic parameters is required and we obtain them from the stereo calibration process in the previous part. The intrinsics (biases) of IMU are calibrated beforehand using the manufacturer’s software<sup>10</sup>, and the correction is applied to the raw inertial measurement. At last, we collect the visual-inertial sequence with full 6 DOF excitation and perform the visual-inertial calibration. We have test the calibration result using VINS-Mono, VINS-Fusion, OpenVINS, and we can obtain meaningful trajectories using the calibrated parameters. A 20-ms time offset between IMU and camera timestamps is found by the Kalibr package, which is also reported in the accompanying yaml file (Fig. 9). However this value is just for reference, and we do not make any modification on sensor’s timestamps. The rosbag files used for calibration can also be found at the website.

## 5.3 Other flexible parts compensation and ground truth

The dataset is captured from a payload bay mounted on our custom DJI M600 Pro drone. Therefore, the triple GPS, compass, and prism on top of the drone are not rigidly connected to the payload bay, i.e. their spatial relation can change during flight. However, this is a necessary compromise to reduce vibration effect on the quality of the lidar scan, camera image, and IMU. In light of these conditions, we first use VICON to verify most of the other extrinsic parameters by putting the VICON reflective ball at each sensor while the drone is at rest on the floor. Then we exert some force on the damped payload bay to tilt it up to some maximum angle. The maximum displacement of the prism between static configuration and maximum tilting angle is around 2cm. Based on our study, the drone only exhibits small control angles with a minimized jerk during the flight tests. Therefore, the prism measurement is relatively accurate to represent the ground truth. Note that we use hand-eye calibration to align the ground truth with onboard altitude estimate obtained from the drone’s internal barometer and IMU, which are independent from the sensors on the payload.

To obtain a bound of the error in the temporal alignment between ground truth and onboard sensors, we conduct several experiments with several state-of-the-art localization methods and calculate their Absolute Trajectory Error (ATE) with multiple versions of ground truth that are time-shifted by -0.5, -0.49, -0.48, ..., 0.5 seconds. The results are reported in Fig. 8. We can see that despite using different sensor

**Table 4.** ATE of state-of-the-art localization methods over NTU VIRAL datasets. The best odometry result is highlighted in **bold**, and the second best is underlined. All values are in meter.

Dataset	VINS-Fusion	LIO-SAM	MLOAM	VIRAL-SLAM
eee_01	0.608	<u>0.075</u>	0.249	<b>0.060</b>
eee_02	0.506	<u>0.069</u>	0.166	<b>0.058</b>
eee_03	0.494	<u>0.101</u>	0.232	<b>0.037</b>
nya_01	0.397	<u>0.076</u>	0.123	<b>0.051</b>
nya_02	0.424	<u>0.090</u>	0.191	<b>0.043</b>
nya_03	0.787	<u>0.137</u>	0.226	<b>0.032</b>
sbs_01	0.508	<u>0.089</u>	0.173	<b>0.048</b>
sbs_02	0.564	<u>0.083</u>	0.147	<b>0.062</b>
sbs_03	0.878	<u>0.140</u>	0.153	<b>0.054</b>

combinations and methods, i.e. stereo camera-IMU (VINS), lidar-IMU (LIO-SAM), and stereo camera-IMU-lidar-UWB (VIRAL SLAM), all methods can achieve minimal ATE within the [-0.1, 0.1] interval. Thus, we are confident that the temporal alignment error between ground truth and the onboard sensors is within 0.1s in the current datasets. Table 4 presents the ATE of the SLAM frameworks without time shift. A more comprehensive analysis can be found in Nguyen et al. (2021b). Fig. 9 demonstrates a result of VIRAL SLAM on one sequence.

## 6 Evaluation recommendations

From our experiments of state-of-the-art SLAM methods on NTU VIRAL, we believe some issues regarding the evaluation process merit a detailed discussion. In most cases, after applying a navigation method on our dataset, the user can obtain a trajectory consisting of  $N + 1$  position estimates of the body frame, denoted as  $\{\hat{\mathbf{p}}_{t_n}^B\}_{n=0}^N$ , and a trajectory of  $M$  ground truth samples of the crystal prism  $\{\hat{\mathbf{p}}_{t_m}^L\}_{m=0}^M$ . From these samples, the user would like to calculate the estimation accuracy of the method based on some metrics. To facilitate the use of our dataset, MATLAB scripts for this purpose are also included in the suite. We note that some previous works such as Sturm et al. (2012); Zhang and Scaramuzza (2018) might have briefly addressed some of the issues discussed below. However, we think a more formal description can be of interest to some readers.

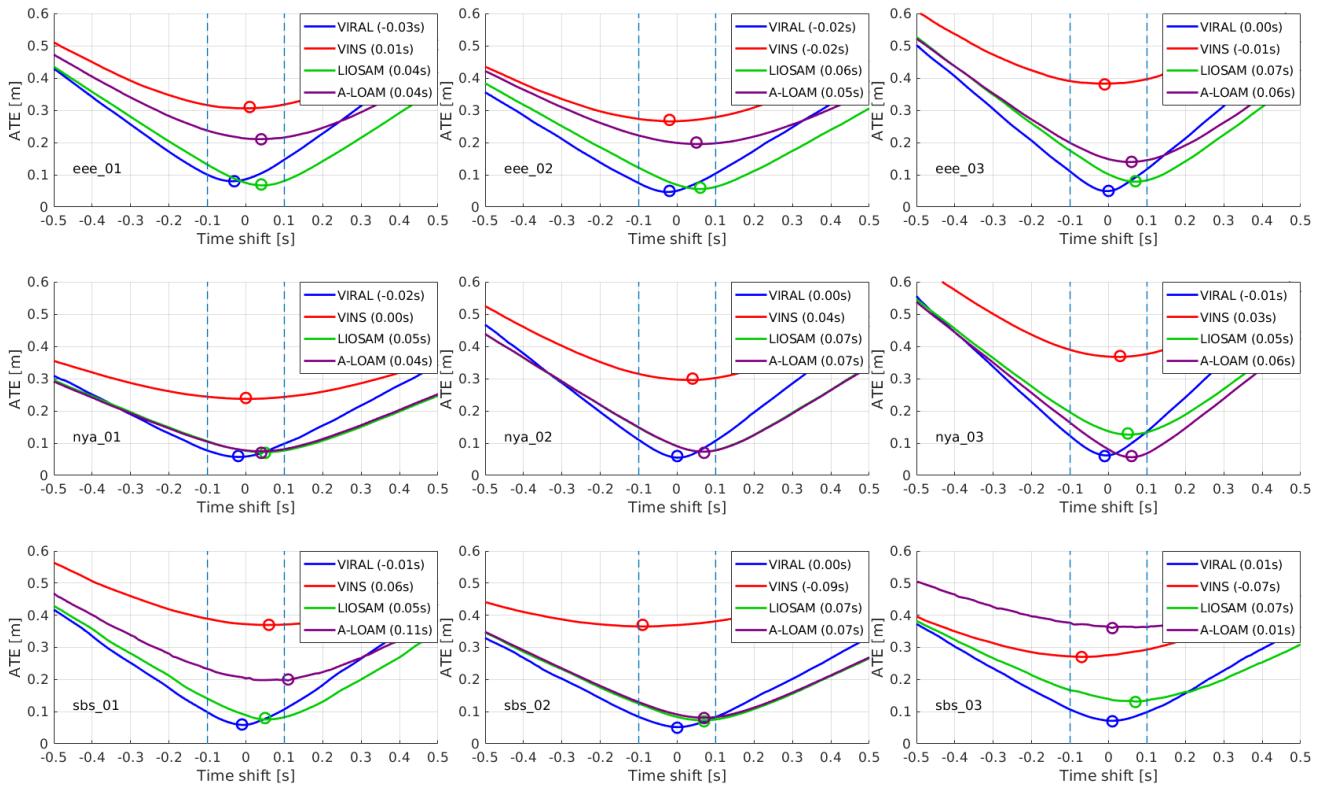
### 6.1 Are we tracking the same point?

Since the crystal prism has a displacement of almost 0.4 m from the body frame’s origin, if we naively calculate the error between the body-centered trajectory with the ground truth, there can be a bias of 0.4 m in the estimate in the worst case scenario. Thus it is recommended that the user use the pose estimate to calculate the estimated trajectory of the crystal prism, denoted as  $\{\hat{\mathbf{p}}_{t_n}^W\}_{n=0}^N$ , for comparison with ground truth. Specifically, using (1), we can obtain:

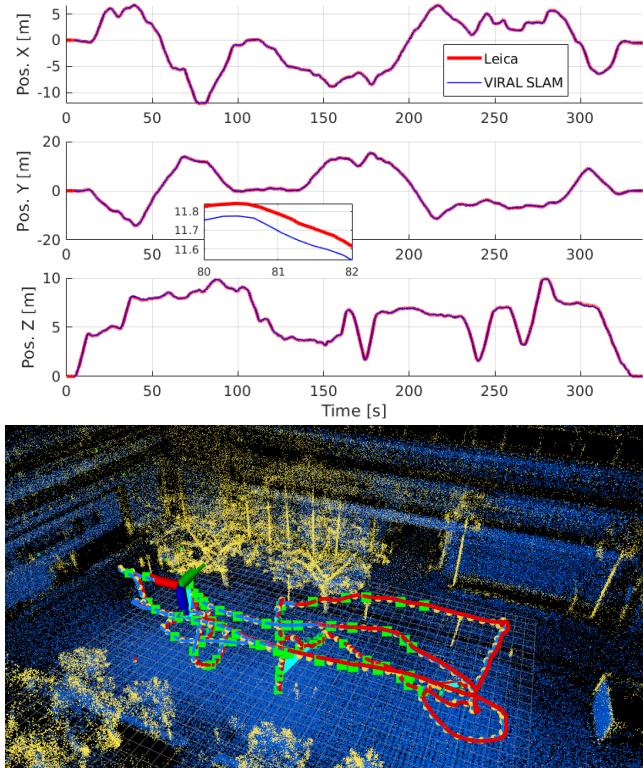
$$\hat{\mathbf{p}}_{t_n}^W = \hat{\mathbf{p}}_{t_n}^B + \hat{\mathbf{R}}_{t_n}^B \hat{\mathbf{t}}_{t_n}, \forall n \in \{0, 1, \dots, N\}, \quad (2)$$

<sup>9</sup><https://github.com/ethz-asl/kalibr/wiki/camera-imu-calibration>

<sup>10</sup><https://www.vectornav.com/resources/software>



**Figure 8.** The changes in ATE of different methods with respect to time shifts of the ground truth. Note that except A-LOAM the selected methods all feature loop closure and global optimization to eliminate the influence of drift. The minimum of each curve is marked by a circle, and the time shift that minimizes the ATE for each method is reported in the legend. It can be seen that all visual-inertial, lidar-inertial, and visual-inertial-lidar-ranging methods can achieve minimum error within the [-0.1, 0.1] interval.



**Figure 9.** VIRAL SLAM result using the eee\_01 dataset. Ground truth is made in red, the trajectory estimate in blue. The 3D map of the environment is visualized in blue and yellow points in the 3D visualisation below. Key frames are highlighted by green boxes. The ground truth and trajectory estimate are aligned using the 6DoF transformation calculated in (3).

where  $\hat{\mathbf{R}}_{t_n}$  is the orientation estimate, and  $\mathbf{t}_{t_n}$  is the translation from the body origin to the prism reported in the dataset's calibration files. Hence, we can drop the subscript  $P$  as it is implied in later parts.

## 6.2 Resampling

The goal of this task is to resample  $\{\hat{\mathbf{p}}_{t_n}\}_{n=0}^N$  and  $\{\mathbf{p}_{t_m}\}_{m=0}^M$  to obtain the sequences  $\{\hat{\mathbf{p}}_{t_k}\}_{k=0}^K$  and  $\{\mathbf{p}_{t_k}\}_{k=0}^K$  of the same length. Specifically  $\{t_k\}_{k=0}^K$  is a subset of  $\{t_n\}_{n=0}^N$  that satisfies the following

$$\exists t_s \in \{t_m\}_{m=1}^M : t_s \leq t_k \leq t_{s+1}, |t_s - t_{s+1}| < 0.1, \forall t_k.$$

Given the time  $t_s$  above, the ground truth sample at time  $t_k$  can be linearly interpolated as  $\mathbf{p}_{t_k} = \frac{t_k - t_s}{t_{s+1} - t_s} \mathbf{p}_{t_s} + \frac{t_{s+1} - t_k}{t_{s+1} - t_s} \mathbf{p}_{t_{s+1}}$ . We can further simplify the notation by denoting the trajectories as  $\{\hat{\mathbf{p}}_k\}_{k=0}^K$  and  $\{\mathbf{p}_k\}_{k=0}^K$ .

## 6.3 Coordinates transform

At the final stage, we can see that  $\{\hat{\mathbf{p}}_k\}_{k=0}^K$  and  $\{\mathbf{p}_k\}_{k=0}^K$  are still w.r.t. different coordinate frames. Hence they cannot be directly compared. Following the common public benchmarking tools by Sturm et al. (2012) and Zhang and Scaramuzza (2018), we adopt the following coordinate transform  $(\mathbf{R}_W, \mathbf{t}_W)$  to bring the two trajectories to a common frame of reference:

$$(\mathbf{R}_W, \mathbf{t}_W) = \arg \min_{(\mathbf{R}, \mathbf{t})} \left( \sum_{k=0}^K \|\mathbf{R} \hat{\mathbf{p}}_k + \mathbf{t} - \mathbf{p}_k\|^2 \right). \quad (3)$$

The above was shown to have a closed-form solution in [Umeyama \(1991\)](#), and the value of the summation is referred to as ATE of the algorithm's estimate over the dataset. A MATLAB script is also provided in this data suite to perform this task.

## 7 Known issues and limitations

Despite careful design and execution of the data collection experiments, we are aware of several practical issues which pose some challenges and limit the achievable accuracy. The specifics are given below

### 7.1 Camera exposure setting

Stereo camera exposure setting is a challenging topic for any global-shutter stereo camera. Since the incoming photons to the CMOS sensor can be vastly different at different poses, the obtained image quality can also vary quite dramatically. However, stereo matching needs a set of images with consistent brightness. This dataset uses a set of fixed global shutter exposure settings for various indoor and outdoor locations, but each camera master gain is set to be automatic. Our exposure setting is usually one-third of the default auto-exposure setting for reducing the glare effect. This shorter exposure time improves the image's sharpness and reduces the motion blur at the cost of a slightly darker appearance. The darker appearance can be a challenge for any direct visual SLAM system. We think it is a necessary cost to get a sharp accurate measurement of the environment.

### 7.2 Partial loss of information

In the SBS experiments, the Leica station lost track of the prism for some short periods of time, due to its vantage point being much lower than that in other environments. The evaluation method provided by us already takes this into account in Sec. 6.2.

Besides, there was significant loss of UWB in some experiments due to low received signal strength (RSS) at certain relative position between the ranging node and the anchor. This in turn is due to the shape of the radiation pattern (see [Chen et al. \(2018\)](#) for this pattern). This is not the ideal case for a dataset, but it is a real-scenario that the researcher has to address when developing new algorithms.

## 8 Conclusion and future works

In this paper we present the datasets collected from a sensor suite typical of autonomous driving car, but equipped on a drone. The datasets are aimed at boosting the investigations into autonomous navigation of UAVs using the up-to-date technologies, which can facilitate progress in many important industries.

Despite our best efforts, there remain some issues in the development of the datasets. However, we do plan to keep our work continuously updated, and these issues will be addressed in the future. For example, the loss of UWB can be remedied by updating the UWB hardware, the tracking of ground truth can be improved with a better vantage point. Moreover, temporal alignment can be further improved by batch optimization methods. The camera hardware can also be made up-to-date with other more popular RGBD-type

sensors. Finally, more datasets will be added to the suite in the future.

## Dataset access

The dataset can be accessed at our web page [https://ntu-aris.github.io/ntu\\_viral\\_dataset/](https://ntu-aris.github.io/ntu_viral_dataset/). Other details on calibration, evaluation and updates are also presented at this site to facilitate the use of our dataset on individual's research.

## Funding acknowledgement

This work is supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation, under the Grant Call 10013 - Wallenberg-NTU Presidential Postdoctoral Fellowship 2020.

## References

- Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW and Siegwart R (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.
- Chen J, Raye D, Khawaja W, Sinha P and Guvenc I (2018) Impact of 3d uwb antenna radiation pattern on air-to-ground drone connectivity. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. pp. 1–5. DOI:10.1109/VTCFall.2018.8690726.
- Delmerico J, Cieslewski T, Rebecq H, Faessler M and Scaramuzza D (2019) Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 6713–6719.
- Furgale P, Rehder J and Siegwart R (2013) Unified temporal and spatial calibration for multi-sensor systems. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1280–1286.
- Geiger A, Lenz P, Stiller C and Urtasun R (2013) Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.
- Geneva P, Eckenhoff K, Lee W, Yang Y and Huang G (2020) Openvins: A research platform for visual-inertial estimation. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4666–4672.
- Huang AS, Antone M, Olson E, Fletcher L, Moore D, Teller S and Leonard J (2010) A high-rate, heterogeneous data set from the darpa urban challenge. *The International Journal of Robotics Research* 29(13): 1595–1601.
- Jeong J, Cho Y, Shin YS, Roh H and Kim A (2019) Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research* 38(6): 642–657.
- Jiao J, Ye H, Zhu Y and Liu M (2020) Robust odometry and mapping for multi-lidar systems with online extrinsic calibration. *arXiv preprint arXiv:2010.14294*.

- 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.
- Majdik AL, Till C and Scaramuzza D (2017) The zurich urban micro aerial vehicle dataset. *The International Journal of Robotics Research* 36(3): 269–273.
- Nguyen TH, Nguyen TM and Xie L (2020a) Tightly-coupled ultra-wideband-aided monocular visual slam with degenerate anchor configurations. *Autonomous Robots* 44(8): 1519–1534.
- Nguyen TM, Cao M, Yuan S, Lyu Y, Nguyen TH and Xie L (2020b) Liro: Tightly coupled lidar-inertia-ranging odometry. *2021 IEEE International Conference on Robotics and Automation (ICRA)*.
- Nguyen TM, Cao M, Yuan S, Lyu Y, Nguyen TH and Xie L (2021a) Viral-fusion: A visual-inertial-ranging-lidar sensor fusion approach. *IEEE Transactions on Robotics* DOI:10.1109/TRO.2021.3094157.
- Nguyen TM, Yuan S, Cao M, Nguyen TH and Xie L (2021b) Viral slam: Tightly coupled camera-imu-uwb-lidar slam. *Submitted to IEEE Robotics and Automation Letters* URL <https://arxiv.org/abs/2105.03296>.
- Nguyen TM, Zaini AH, Wang C, Guo K and Xie L (2018) Robust target-relative localization with ultra-wideband ranging and communication. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2312–2319.
- 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.
- Qin T, Li P and Shen S (2018) Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics* 34(4): 1004–1020.
- Ramezani M, Wang Y, Camurri M, Wisth D, Mattamala M and Fallon M (2020) The newer college dataset: Handheld lidar, inertial and vision with ground truth. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 4353–4360.
- Schubert D, Goll T, Demmel N, Usenko V, Stückler J and Cremers D (2018) The tum vi benchmark for evaluating visual-inertial odometry. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1680–1687.
- Shan T, Englot B, Meyers D, Wang W, Ratti C and Daniela R (2020) Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5135–5142.
- Sturm J, Engelhard N, Endres F, Burgard W and Cremers D (2012) A benchmark for the evaluation of rgb-d slam systems. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 573–580.
- Sun K, Mohta K, Pfrommer B, Watterson M, Liu S, Mulgaonkar Y, Taylor CJ and Kumar V (2018) Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters* 3(2): 965–972.
- Umeyama S (1991) Least-squares estimation of transformation parameters between two point patterns. *IEEE Computer Architecture Letters* 13(04): 376–380.
- Zhang J and Singh S (2018) Laser–visual–inertial odometry and mapping with high robustness and low drift. *Journal of Field Robotics* 35(8): 1242–1264.
- Zhang Z (2000) A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence* 22(11): 1330–1334.
- Zhang Z and Scaramuzza D (2018) A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 7244–7251.
- Zuñiga-Noël D, Jaenal A, Gomez-Ojeda R and Gonzalez-Jimenez J (2020) The uma-vi dataset: Visual–inertial odometry in low-textured and dynamic illumination environments. *The International Journal of Robotics Research* 39(9): 1052–1060.