MUN-FRL: a visual-inertial-lidar dataset for aerial autonomous navigation and mapping

Thalagala, Ravindu G; De Silva, Oscar; Jayasiri, Awantha; Gubbels, Arthur; Mann, George KI; Gosine, Raymond G

Publisher's version / Version de l'éditeur:
https://doi.org/10.1177/02783649241238358
The International Journal of Robotics Research, 2024-04-16

Data Paper

The International Journal of Robotics Research
2024, Vol. 0(0) 1–14

DOI: 10.1177/02783649241238358
journals.sagepub.com/home/ijr


MUN-FRL: A Visual-Inertial-LiDAR Dataset for Aerial Autonomous Navigation and Mapping

Ravindu G Thalagala[1], Oscar De Silva[1], Awantha Jayasiri[2], Arthur Gubbels[2], George KI Mann[1] and Raymond G Gosine[1]

Abstract

This paper presents a unique outdoor aerial visual-inertial-LiDAR dataset captured using a multi-sensor payload to promote the global navigation satellite system (GNSS)-denied navigation research. The dataset features flight distances ranging from 300 m to 5 km, collected using a DJI-M600 hexacopter drone and the National Research Council (NRC) Bell412 Advanced Systems Research Aircraft (ASRA). The dataset consists of hardware-synchronized monocular images, inertial measurement unit (IMU) measurements, 3D light detection and ranging (LiDAR) point-clouds, and high-precision real-time kinematic (RTK)-GNSS based ground truth. Nine data sequences were collected as robot operating system (ROS) bags over 100 mins of outdoor environment footage ranging from urban areas, highways, airports, hillsides, prairies, and waterfronts. The dataset was collected to facilitate the development of visual-inertial-LiDAR odometry and mapping algorithms, visual-inertial navigation algorithms, object detection, segmentation, and landing zone detection algorithms based on real-world drone and full-scale helicopter data. All the data sequences contain raw sensor measurements, hardware timestamps, and spatio-temporally aligned ground truth. The intrinsic and extrinsic calibrations of the sensors are also provided, along with raw calibration datasets. A performance summary of state-of-the-art methods applied on the data sequences is also provided.

Keywords

Dataset, aerial autonomy, full-scale aircraft, drones, visual-inertial-LiDAR odometry and mapping

Received 26 June 2023; Revised 23 November 2023; Accepted 05 January 2024

1. Introduction

Last-mile goods delivery using drones is expected to bring about disruptive changes in logistical operations in the near future (Chen et al., 2022; Boysen et al., 2020). Rapid technological advancements and new legislation frameworks pave the way for large-scale implementation of last-mile goods delivery (Miranda et al., 2022; Pei et al., 2022; Aurambout et al., 2019). Corporations such as Amazon, Google, and DHL (Perreault and Behdinan, 2021) are testing autonomous drone delivery methods as an efficient alternative to traditional delivery methods to cope with the growing demand. Along with drones, full-scale aircraft are being investigated due to their high payload-carrying capability, operational safety, and longer flight times (Cohen et al., 2021). Autonomous flight of such platforms requires mapping the surrounding environment, means to perform collision avoidance, and means to plan and execute normal and emergency landing procedures (Yang and Wei, 2021; Cohen et al., 2021).

Autonomous navigation and mapping applications typically use a combination of visual (cameras), inertial measurement units (IMUs), light detection and ranging (LiDAR) sensors, radio detection and ranging (Radar), and global navigation satellite system (GNSS) receivers in their navigation pipeline. This sensor combination is prevalent in autonomous driving architectures (Zhang and Singh, 2014; Jiao et al., 2022; Ding et al., 2020), where the control system requires both the surrounding map and the current pose relative to the surroundings to plan its maneuvers safely. Multi-sensor fusion algorithms that are popular in this domain include visual-inertial odometry (VIO) (Qin et al., 2019; Gomaa et al., 2020; Thalagala et al., 2021), visual-inertial simultaneous localization and mapping (VI-SLAM) (Song et al., 2022), visual-inertial-LiDAR (VIL) odometry and mapping (Shan et al., 2021, 2020; Nguyen et al., 2021),

[1]Intelligence Systems Lab, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL, Canada
[2]Flight Research Lab, National Research Council of Canada, Ottawa, ON, Canada

Corresponding author:
Ravindu G Thalagala, Intelligence Systems Lab., Faculty of Engineering and Applied Science, Memorial University of Newfoundland, 240 Prince Phillip Drive, St. John's, NL A1C 5S7, Canada.
Email: rgthalagala@mun.ca

and radar aided multi-sensor fusion methods (Liang et al., 2020; Adolfsson et al., 2022).

The development of these algorithms requires detailed testing and benchmark performance comparison. Publicly available datasets with ground truth are often utilized for this purpose. Several public datasets are available for autonomous driving algorithm development, such as *KITTI* (Geiger et al., 2013), *NCLT* (Carlevaris-Bianco et al., 2015), and *4Seasons* (Wenzel et al., 2020). These datasets have a complete set of sensor data required to implement VI-SLAM or VIL architectures with critical metadata such as time synchronization between sensors and ground-truth data for benchmark performance. Additionally, these datasets were captured using vehicles in realistic driving scenarios, having long path lengths, varied lighting characteristics, and changing environmental conditions.

Compared to autonomous driving datasets, only a few are available for aerial multi-sensor fusion algorithm development. *EuRoC* (Burri et al., 2016) and *Zurich-Urban* (Majdik et al., 2017) are aerial datasets collected on experimental platforms focused on VI-SLAM developments with only camera, IMU, and ground-truth data; they excluded LiDAR sensor data. In *EuRoC* the data collection was carried out in indoor environments. *NTU-VIRAL* (Nguyen et al., 2021) dataset contains outdoor dataset with a VIL-capable sensing suite. The dataset trajectories include areas of open space, a parking lot, and an indoor environment comprising trajectory lengths of $100 - 500m$. *NTU-VIRAL* dataset serves as a state-of-the-art benchmark for drone VIL navigation algorithm development for comparable short trajectories and scenery. However, when considering last-mile goods delivery applications, the expected trajectories will be much longer with added sensor degradation sources such as platform vibrations, higher altitudes, and changing sensor availability, particularly in the case of full-scale helicopters.

This paper presents a multi-sensor VIL aerial dataset captured on both a small-scale drone and a full-scale aircraft. The recorded data is intended to support the development of sensor fusion algorithms for last-mile goods delivery applications. Two batches of data were collected using a payload unit attached to a *DJI-M600* hexacopter drone and an NRC *Bell412* ASRA helicopter as shown in Figure 1. The payload unit consists of monocular cameras, an IMU, a 3D LiDAR scanner, a real-time kinematic (RTK) enabled GNSS receiver, and a *Jetson AGX Xavier* graphics processing unit (GPU) as the processing unit. The dataset has real flight platform sensor degradation characteristics like vibrations, operational flight altitudes and speeds, covering areas of urban towns, airports, highways, and prairies. The dataset contains time-synchronized post-processed ground truth to support benchmark performance comparison. To the best of the authors' knowledge, this work presents the first publicly available full-scale vertical takeoff and landing (VTOL) vehicle visual-inertial-LiDAR aerial dataset with ground truth, suitable for VIL algorithm development. The contributions of this paper can be summarized as follows.

Figure 1. MUN Sensor Payload Flight Test Configurations: NRC Bell 412 ASRA Nose Mount (Top/Bottom Left); MUN DJI-M600 air-frame mount top right.

- A novel publicly available visual-inertial-LiDAR dataset that includes full-scale aerial platform data with ground truth for VIL navigation algorithm development.
- The dataset incorporates flight trajectories up to 5 km in outdoor environments to suit navigation algorithm development for last-mile goods delivery applications.
- Evaluation results of state-of-the-art VIL navigation algorithms applied on the dataset to validate the suitability of the dataset to serve as a VIL navigation system design benchmark.

The *MUN-FRL* dataset, which includes the flight sequences, post-processed RTK-GNSS ground truth, synchronization information, calibration parameters, and raw calibration sequences, are made available in the public repository: https://mun-frl-vil-dataset.readthedocs.io/en/latest/.

2. Related work

Publicly available datasets have significantly contributed to the development of multi-sensor fusion algorithms. We categorize these datasets based on the type of platform (i.e., ground and aerial datasets) and the sensor suite (i.e., visual-inertial datasets and visual-inertial-LiDAR datasets). The visual-inertial datasets are discussed in the scope of this work due to their high significance in multi-sensor aerial navigation research. Whereas only a handful of visual-inertial-LiDAR datasets *NTU-VIRAL* and *TartanAir* (Wang et al., 2020) are available for aerial platform navigation algorithm development. A summary of these multi-sensor navigation datasets is provided in Table 1.

Ground platform datasets *KITTI* (Geiger et al., 2013), *NCLT* (Carlevaris-Bianco et al., 2015), *Oxford RobotCar* (Maddern

Table 1. Publicly Available Dataset for Multi-Sensor Navigation Algorithm Development.

					Sensors		
Dataset	Year	Platform	Ground truth	Domain	Cameras	LiDARs	IMUs
KITTI (Geiger et al., 2013)	2013	Car	RTK-GNSS INS	Ground	4×PointGrey @10 Hz	VLP-64E @10 Hz	acc/gyro @10 Hz
NCLT (Carlevaris-Bianco et al., 2015)	2015	UGV	RTK-GNSS LiDAR-SLAM	Ground	LadyBug @5 Hz	3 × 3D HDL-32E @10 Hz 2 × 2D-Hokuyo @40 Hz	acc/gyro @10 Hz
EuRoC (Burri et al., 2016)	2016	AscTec Quadcopter	6DOF Vicon MoCap	Aerial	2×MT9V034 @20 Hz	×	acc/gyro @200 Hz
Oxford RobotCar (Maddern et al., 2016)	2016	Car	×	Ground	BumbleBee: @16 Hz 3×Grasshoper2 @11.1 Hz	2×SICK 2D@ 50 Hz 1×SICK 3D@ 12.5 Hz	acc/gyro @50 Hz
Zurich Urban (Majdik et al., 2017)	2017	ArDrone	Aerial Photogrammetry	Aerial	GoPro @30 Hz	×	acc/gyro @10 Hz
TUM-VI (Schubert et al., 2018)	2018	Handheld	6DOF MoCap @Start/end	Ground	IDS @20 Hz	×	acc/gyro @200 Hz
Complex urban (Jeong et al., 2019)	2019	Car	×	Ground	FLIR @10 Hz	2 × 3D HDL-16 @10 Hz 2 × 2D-SICK @100 Hz	acc/gyro @200 Hz FOG @1000 Hz
UMA-VI (Zuñiga-Noël et al., 2020)	2020	Handheld	VIO @12.5 Hz	Ground	Bumblebee2 @12.5 Hz Stereo IDS uEye @25 Hz	×	acc/gyro@250 Hz
Newer college (Ramezani et al., 2020)	2020	Handheld	LiDAR-SLAM	Ground	Intel-D435i @30(Hz)	OS1-64 @100(Hz)	acc/gyro@100 Hz acc/gyro@400 Hz acc/gyro@250(Hz)
LVI-SAM (Ramezanet al., 2020)	2020	Handhel	GNSS	Ground	FLIR @20(Hz)	VLP-16 @10(Hz)	acc/gyro@500(Hz)
ALTO (Cisneros et al., 2022)	2022	Helicopter	RTK-GNSS	Aerial	IDS imaging	×	acc/gyro@200 Hz
VPAIR (Schleiss et al., 2022)	2022	Fixed wing	RTK-GNSS	Aerial	Generic-RGB	×	acc/gyro@25 Hz
Mid-air (Fonder et al., 2019)	2020	Simulated Quadcopter	Simulated GNSS	Aerial	Simulated-camera	×	Simulated acc/gyro
TartanAir (Wang et al., 2020)	2020	Simulated Aerial vehicle	Simulated GNSS	Aerial	Simulated-camera	Simulated-LiDAR	Simulated acc/gyro
NTU-VIRAL (Nguyen et al., 2021)	2021	M600	3D laser Tracker	Aerial	IDS @10 Hz	2×Ous.-16 @10 Hz	acc/gyr/mag@ 385 Hz
MUN-FRL (ours)	2022	Helicopter	RTK-GNSS	Aerial	FLIR @20 Hz	VLP-16 @10 Hz	acc/gyr/@400 Hz

et al., 2016), *Complex Urban* (Jeong et al., 2019), *Newer College* (Ramezani et al., 2020), and *LVI-SAM* (Shan et al., 2021) are heavily cited for visual-inertial-LiDAR navigation algorithm development. All these datasets contain at least one LiDAR scanner, monocular or stereo camera, IMU, and GNSS receiver. It is important to note that *TUM-VI* (Schubert et al., 2018), *UMA-VI* (Zuñiga-Noël et al., 2020), *Newer College*, and *LVI-SAM* datasets mentioned in Table 1 were collected using handheld sensor rigs while other ground platform datasets in Table 1 were collected with sensor suite mounted on a vehicle allowing to capture actual driving conditions, driving speeds, and sensor degradation sources such as platform vibrations.

The benchmark datasets available for VIL algorithm development generally include additional supporting information such as time synchronization between sensors, sensor calibration parameters, and ground truth. Time synchronization between multiple sensors reduces the estimation drift caused by the timing inconsistencies of individual sensor clocks (Faizullin et al., 2022). The datasets *NCLT*, *Oxford RobotCar*, and *Complex Urban* provide timestamps that can be used to synchronize sensing data using received time stamps. However, due to inherent clock drift, the sensors are not guaranteed to capture data synchronously with each other. Additionally, the received time stamps include message transfer delays when recording data. Datasets such as *KITTI*, *Newer College*, and *LVI-SAM* report hardware time synchronization between the clocks of each sensor, which accurately synchronizes sensor data with time stamps references to a common clock, for example, GNSS clock.

Accurate calibration of intrinsic and extrinsic parameters reduces estimation drift in multi-sensor fusion algorithms (Schubert et al., 2018). This may include parameters related to camera intrinsic calibration, camera-IMU extrinsic calibration, camera-LiDAR extrinsic calibration, and GNSS-IMU extrinsic calibration. These critical parameters are provided in datasets such as *KITTI*, *NCLT*, *Oxford RobotCar*, *Complex Urban*, *Newer College*, and *LVI-SAM*.

Ground-truth data is essential for the performance evaluation of multi-sensor fusion algorithms. Ground platform datasets typically provide ground-truth data as GNSS positions. Coarse meter-level GNSS ground-truth data was provided in *Oxford RobotCar* and *NCLT* datasets. The datasets such as *Complex Urban*, *KITTI*, *Newer College*, and *4Seasons* (Wenzel et al., 2020) provided centimeter-level RTK-GNSS platform positioning information.

When considering aerial platforms, visual-inertial datasets are more prevalent, with most aerial datasets considering a minimum payload sensing suite, that is, stereo or monocular camera and IMU. The *EuRoC* dataset contains hardware time-synchronized stereo camera and IMU data captured in indoor environments with changing trajectory speeds and lighting conditions. The dataset provides post-processed ground truth with a sub-centimeter level accuracy using a motion capture system, calibration information, and raw calibration data. The intended application of *EuRoC* dataset is GNSS-denied indoor navigation. *Zurich Urban* captured a single 2 *km* long outdoor dataset using a tethered micro aerial vehicle (MAV) in an urban environment. The *Zurich Urban* dataset contains software time-synchronized data with meter-level GNSS position ground truth. The *Mid-air* dataset (Fonder et al., 2019) provides a visual-inertial dataset using an Unreal engine-based simulator, resulting in photo-realistic synthesized data to support navigation algorithm development. The *ALTO* dataset (Cisneros et al., 2022) and *VPAIR* dataset (Schleiss et al., 2022) were collected using a helicopter and a light aircraft, respectively, reaching altitudes up to 300 m. These datasets are intended for visual place recognition and localization tasks, providing GNSS-inertial navigation system (INS) ground truth location data, IMU data, laser altimeter measurements, and RGB images captured from a downward-facing camera. The trajectories covered in these datasets span ranges of up to 100 km.

Compared to ground platform datasets, there is a significant shortage of aerial platform datasets available for VIL algorithm development. Payload weight limitations have mainly restricted the use of LiDAR scanners on MAV platforms. The *TartanAir* dataset generates a synthetic aerial VIL dataset using an Unreal engine-based simulator to support SLAM applications. The dataset includes stereo RGB images, depth images, segmentation data, optical flow information, camera poses, and LiDAR point clouds. Recently published *NTU-VIRAL* dataset utilized a moderately heavy payload ($\sim 5kg$) attached to a DJI-M600 drone to capture datasets capable of VIL algorithm development. The sensing suite contains two 3D LiDARs, a stereo camera, IMU, ultra wide band (UWB) ranging sensors, and a laser tracker (Leica Nova MS60 MultiStation) based ground truth positions. The dataset has many sequences in different outdoor and indoor environments. *NTU-VIRAL* contains hardware time-synchronized sensing data, centimeter-level post-processed ground truth, calibration information, and sequences in different environment conditions. The flight trajectories cover six different environments with areas ($\sim 100m \times 100m$) that were ideally suitable for building inspection applications. Utilizing this dataset for application scenarios such as last-mile goods delivery would be sub-optimal since it lacks the kilometer-range long trajectories, corresponding altitudes, and platform dynamics seen in operational scale drone delivery type applications.

3. Sensor setup

We have developed a platform-agnostic multi-sensor payload unit, enabling us to capture datasets with similar sensors on different platforms. This design allows for testing and evaluation using a drone, which benefits from fewer regulations and test flight preparations than full-scale aircraft. The sensors necessary for VIL navigation algorithm implementation are already available in small form factors, enabling them to be used in interchangeable operations.

The payload houses two cameras, a LiDAR scanner, an IMU, a GNSS receiver, a long-range (LoRA) RTK correction information receiver, and a *Jetson AGX Xavier* GPU. A GNSS

Table 2. Detailed Payload Sensor Specifications.

Sensor type	Type	Unit
LiDAR	Velodyne	
Model	VLP-16	
Channels	16	
Range	100	*m*
Vertical FOV	30	deg
Horizontal FOV	360	deg
Frequency	10	*Hz*
IMU	Xsens	
Model	MTi-30 AHRS	
Frequency	400	*Hz*
Gyro noise density	0.03	$^o/s/\sqrt{Hz}$
Accel noise density	60	$\mu g/\sqrt{Hz}$
Mag RMS noise	0.5	*mGauss*
Camera - down	FLIR - USB	
Model	BFS-U3-16S2M-BD	
Resolution	1440 × 1080	*pixel*
Frequency	20	*Hz*
Readout method	Global shutter	
Camera - front	FLIR - GigE	
Model	BFS-PGE-04S2C-CS	
Resolution	720 × 540	*pixel*
Frequency	20	*Hz*
Readout method	Global shutter	
Lenses - both cameras	Computar	
Lens	A4Z2812CS-MPIR	
Focal length	2.8-10	*mm*
Horizontal field of view	127.6	deg
Vertical field of view	65.0	deg
GNSS	simpleRTK2B	
Receiver	u-blox ZED-F9P	
Base station	Reach RS2	
RTK communication	LoRa radio	
Position output	NMEA	
RTK solution	5	*Hz*

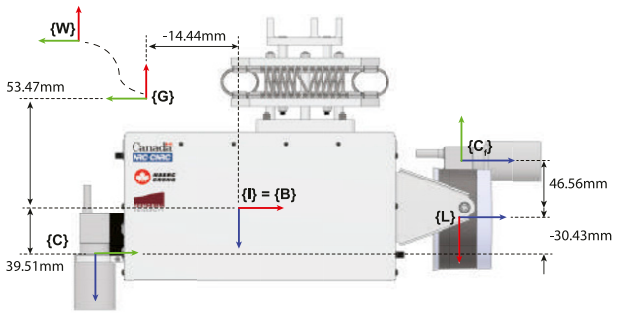


Figure 2. Sensor payload, key coordinate frames, and dimensions.

base station was securely positioned at a known location for RTK correction information telemetry. The detailed specifications of sensors used in the payload are summarized in Table 2.

The LiDAR of the payload is rigidly attached, facing downwards. The two cameras are rigidly mounted with one front-facing and the other downward-facing, as shown in Figure 2. In the *MUN-FRL* dataset and its evaluations, we have limited the data feed to the downward-looking camera as it provided the most information for navigation algorithms and stable estimation throughout the flight trajectories.

Figure 2 illustrates the coordinate frames associated with the sensors. The X, Y, and *Z* axis are represented by red, green, and blue, respectively, and defined as follows: the IMU frame, denoted as $\{I\}$, is equivalent to the body frame $\{B\}$, the front camera frame is denoted as $\{C_f\}$, the down camera frame is denoted as $\{C\}$, the LiDAR frame is denoted as $\{L\}$, the GNSS receiver antenna frame is denoted as $\{G\}$, and the East-North-Up (ENU) frame of reference is denoted as $\{W\}$. Figure 3 shows the sensor frames rigidly connected with the IMU coordinate system $\{B\}$. The dotted line represents the changing relative pose $\mathbf{T}_{WG}$ during flight.

The convention used to describe the spatial transformations of the sensors is as follows. A transformation matrix $\mathbf{T}_{BA} \in SE(3)$, transforms point coordinates $\mathbf{p}_A \in \mathbb{R}^3$ defined in frame $\{A\}$ to point coordinate defined in frame $\{B\}$, that is, $\mathbf{p}_B = \mathbf{T}_{BA}\mathbf{p}_A$.

4. Sensor calibration

This section presents the calibration methods used to determine intrinsic and extrinsic parameters for each sensor of the payload unit. Calibration was performed in four main steps: camera intrinsic, IMU intrinsic, camera-IMU extrinsic, and camera-LiDAR extrinsic calibration. The resulting calibration parameters were stored in a *"calibration.yaml"* file for each batch of data captured on the *Bell412* and *DJI-M600* platforms.

4.1. Camera intrinsic calibration

The intrinsic parameters of the cameras include the camera projection matrix and the lens distortion model parameters. These were found using the camera calibration tool provided in the *VINS-Fusion* package

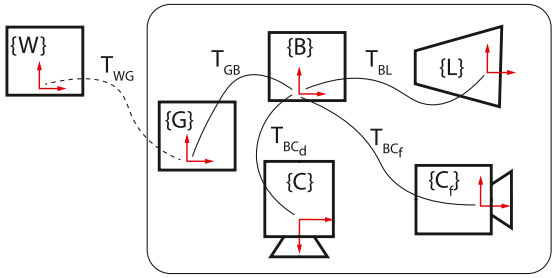


Figure 3. Schematic view of sensor coordinate frames (red) and corresponding transformations (black).

(Qin et al., 2019) following the instructions mentioned therein. These camera calibration parameters are included in the *"calibration.yaml"* file.

4.2. IMU intrinsic calibration

IMU measurements are commonly assumed to be corrupted by white noise with standard deviation σ_n and a random walk noise with standard deviation σ_w. The Xsens MTi-30 data sheet provides nominal values for these parameters, requiring refinement for accurate probabilistic modeling of IMU measurements. Therefore, a separate calibration has been carried out using the method described in Schubert et al. (2018). The refined parameters are included in the *"calibration.yaml"* file.

4.3. Camera-IMU calibration

The spatial transformation between camera and IMU, and time-offset (t_d) between camera and IMU measurements were found using the open source *Kalibr* package Furgale et al. (2013). Spatial transformation matrix $\mathbf{T}_{BC}$ and time offset t_d are included in the *"calibration.yaml"* file.

4.4. Camera-LiDAR calibration

The spatial transformation between the camera and LiDAR ($\mathbf{T}_{CL}$) was found using Matlab and our custom robot operating system (ROS) node. A calibration dataset was created by capturing a set of point-cloud (*pcd*) files and image files at the same instant of time. A checkerboard pattern was used as the calibration target when capturing calibration datasets. The calibration was first carried out using the Matlab LiDAR calibration toolbox to find an initial calibration. Then, we used our ROS node to overlay the pcds on images and manually refined the calibration. The projected point cloud should align with the checkerboard if the calibration was performed correctly, as shown in Figure 4(a). To account for any final adjustments that typically happen prior to flight, the calculated transformation matrices were fine-tuned by overlaying and aligning point clouds on feature-rich images taken from data capture missions as shown in Figure 4(b). The final transformation matrix $\mathbf{T}_{CL}$ is included in the *"calibration.yaml"* file.

4.5. IMU-GNSS calibration

IMU-GNSS extrinsic calibration, that is, $\mathbf{T}_{GB}$, was obtained from the computer-aided design (CAD) models relevant to GNSS antenna placement since this accuracy was sufficient for ground-truth generation purposes. The calculated $\mathbf{T}_{GB}$ is included in the *"calibration.yaml"* file.

5. MUN-FRL dataset

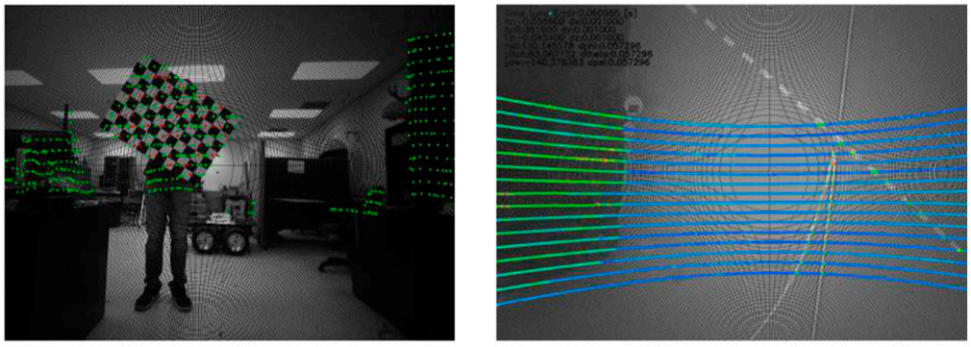
5.1. Dataset description

The dataset includes two batches of data sequences collected by interchanging the multi-sensor payload unit between a *DJI-M600* drone and the NRC *Bell412* ASRA helicopter. Inside the payload unit, all the sensors were connected to a *Jetson AGX Xavier* GPU running ROS. The sensors were publishing data as ROS messages and then recorded as a single ROS bag file. The ROS bags were saved using an internal solid-state drive (SSD) connected to the *Jetson AGX Xavier*.

5.2. Sequence description

The data sequences in this study attempt to mimic last-mile goods delivery missions by starting at one point and ending at another with lengths of up to 5 km. Some sequences include partially overlapping trajectories and multiple loops. For example, Figures 5 and 6 display the two trajectories of *Bell412* sequences.

The *DJI-M600* data batch consists of three sequences obtained at various locations in the vicinity of St. John's,



(a) Matlab LiDAR calibration verification by projecting the point cloud on the image.

(b) The LiDAR point cloud alignment verification on a snapshot of the Bell412-1 sequence.

Figure 4. Camera-LiDAR calibration methodology (Left: Initial calibration using Matlab toolbox, Right: Fine-tuning the calibration using dataset images.).

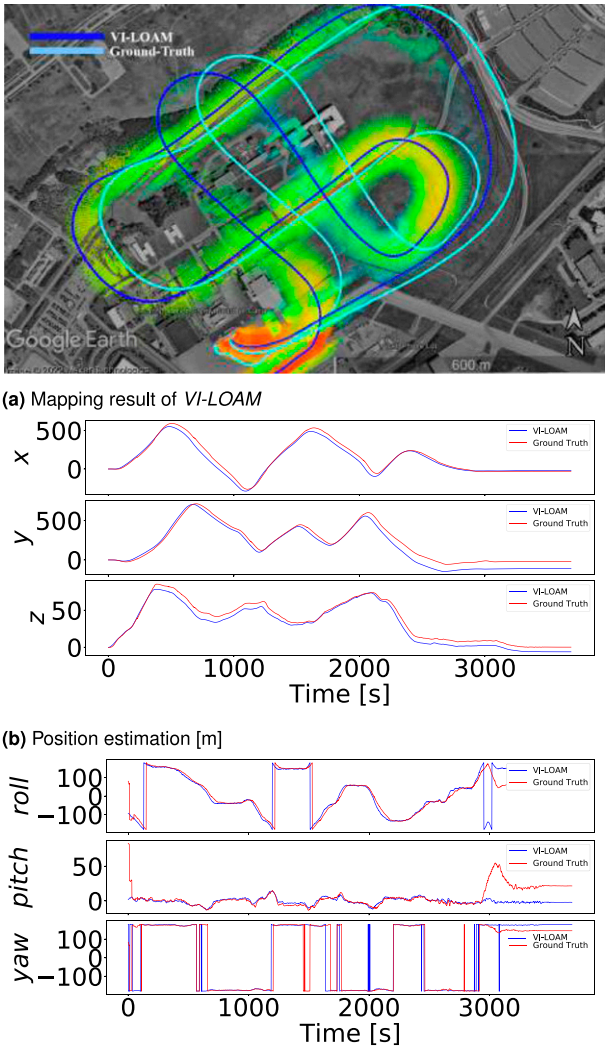


Figure 5. *VI-LOAM* results on the Bell412-6. Useful LiDAR points are only available at the beginning and at the end of the dataset.

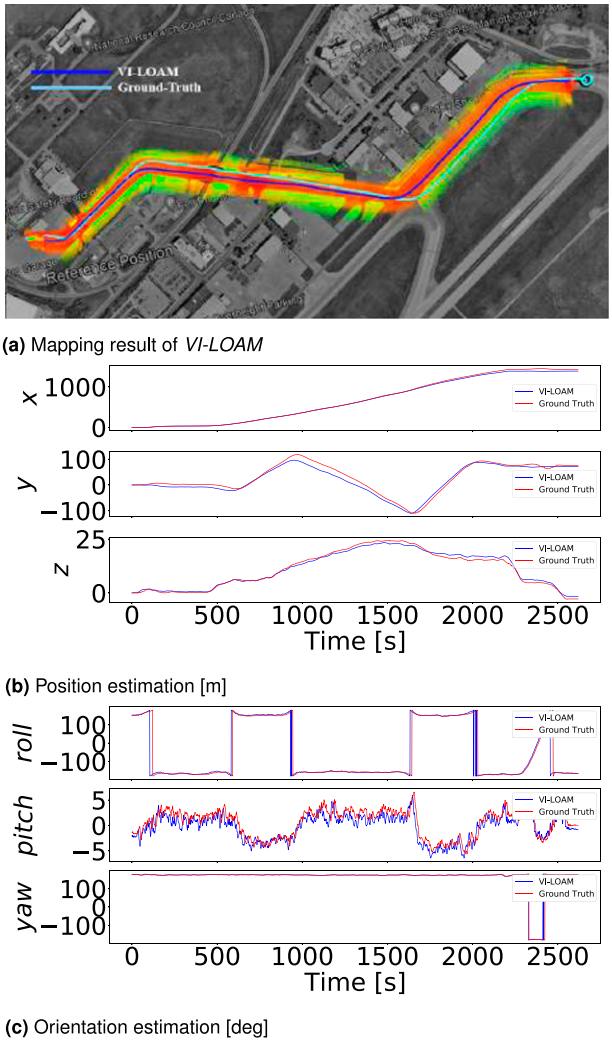


Figure 6. *VI-LOAM* results on the Bell412-1 dataset. The terrain is predominantly flat since this is a low altitude flight over the main taxiway from the hanger.

Newfoundland, Canada. The *Bell412* data batch consists of six sequences that cover urban areas, airports, and highways in the vicinity of the Ottawa International Airport, Ontario, Canada. Table 3 presents a detailed description of each dataset sequence and Figure 10 shows sample images from datasets.

5.3. Data format

5.3.1. ROS bag files. For every sequence in the dataset, we provide two ROS bag files, one raw bag file, and a synchronized bag file. Raw bag contains the real-time captured data, that is, before ROS message timestamp adjustment using the hardware-synchronized universal time coordinate (UTC) timestamps. Synchronized bag file contains the corrected message timestamps achieved by post processing of the dataset. Details related to each ROS topic of a raw bag file and a synchronized bag file are summarized in Table 4.

5.3.2. Calibration files. In the *MUN-FRL* dataset, the calibration parameters are different between *DJI-M600* and *Bell412* sequences. Therefore, for ease of use, we provide calibration parameters in a *"calibration.yaml"* file for each sequence of the dataset. The file includes the intrinsic parameters of the camera and IMU, and extrinsic parameters between camera-IMU, camera-LiDAR, and IMU-GNSS sensors. The transformation matrices are provided following the standard OpenCV matrix convention. Additionally, we provide calibration data sources that were used for IMU noise quantification, camera calibration, camera-IMU calibration, and camera-LiDAR calibration applicable for both the *DJI-M600* sequences and the *Bell412* sequences of the dataset.

5.4. Time synchronization scheme

The synchronization of sensors in the payload unit was achieved through hardware time synchronization using the

Table 3. Dataset Organization and Sequence Descriptions.

Folder	Sequence	Alt [m] *	Sequence description	Files
DJIM600	Quarry-1	59	Uneven feature rich terrain with rocks and cliffs	<*sequence*>.bag <*sequence*>_synced.bag calibration.yaml
	Quarry-2	51	Similar to Quarry-1 but with a longer trajectory	
	Lighthouse	28	Feature rich terrain with helipad, ocean waves, snowy rocks	
Bell 412	Bell412-1	25	Low altitude flight along a predominantly flat taxiway	
	Bell412-2	114	Rolling start trajectory over trees, roads and flat ground	
	Bell412-3	154	Predominantly over flat ground and areas with trees	
	Bell412-4	112	Trajectory over flat ground, trees, buildings, and runway	
	Bell412-5	171	Trajectory over highways, rail tracks, flat ground, and trees	
	Bell412-6	168	Loopin trajectory mostly over buildings, highways, and runways	
Calibration			Stationary imu messages recorded for 3 hours	imu_noise.bag
	DJI-M600		IMU-Camera calibration bag used in Kalibr toolbox	kalibr.bag
	Bell412		Set of checkerboard target for camera intrinsic calibration	/checkerboard images
			Set of synced images and PCDs for camera-LiDAR calibration	/synced PCDs and images

* Alt means the maximum altitude point of the trajectory measured in meters.

Table 4. Descriptions of ROS Topics Included in the MUN-FRL Dataset.

Sensor	Message Type	Topic Name	Description	Frequency
LiDAR	sensor_msgs/PointCloud2	`/velodyne_points`[1,2,3]	raw velodyne pointcloud	10 Hz
	sensor_msgs/TimeReference	`/time_ref_scan`[1]	hardware time at pcd trigger	
Camera	sensor_msgs/Image	`/camera/image_mono`[1,2,3]	Mono8 format	20 Hz
	sensor_msgs/Image	`/camera/image_color`[1,2,3]	RGB8 format	
IMU	sensor_msgs/Imu	`/imu/data`[1,2,3]	raw data	400 Hz
	geometry_msgs/Vector3Stamped	`/imu/mag`[1,2,3]	raw magnetometer data	100 Hz
	sensor_msgs/ImuStamped	`/imu/data_stamped`[1]	raw data with trigger timestamp	400 Hz
	sensor_msgs/TimeReference	`/imu/time_ref`[1]	hardware time at IMU trigger	400 Hz
	sensor_msgs/TimeReference	`/imu/time_ref_cam`[1]	hardware time at camera trigger	20 Hz
	sensor_msgs/TimeReference	`/imu/time_ref_pps`[1]	hardware time at PPS	1 Hz
GNSS	sensor_msgs/NavSatFix	`/fix`[1,2,3]	raw RTK-GNSS	5 Hz
	nmea_msgs/Sentence	`/nmea_sentence`[1]	nmea string RTK-GNSS	
	sensor_msgs/NavSatFix	`/fix_ppk`[2,3]	post processed RTK-GNSS	
Ground Truth 6DOF	nav_msgs/Odometry	`/ins_ppk`[2,3]	6DOF odomtery pose	5 Hz
	nav_msgs/Odometry	`/ins_frl`[2]	6DOF odomtery pose	
	nav_msgs/Odometry	`/ins_dji`[3]	6DOF odomtery pose	

* All the post-processed ground-truth topics are synchronized using UTC timestamp and available in <*name*>_synced.bag.
[1] available in the unsynced ROS bag of all sequences.
[2] available in the synced ROS bag of Bell412 sequences.
[3] available in the synced ROS bag of DJI-M600 sequences.

pulse per second (PPS) signal from the GNSS receiver. The PPS signal is an absolute time reference from the GNSS satellite clock, which reduces the clock drift of the sensors. The PPS signal was fed into the synchronization management module (SMM), which generated a *GPGGA NMEA* (Faizullin et al., 2022) message for the LiDAR, allowing it to publish point-cloud messages with synchronized time stamps. The IMU also supplied the PPS signal to synchronize its clock, which then triggered the camera.

The IMU publishes a `/imu/time_ref` topic, including the hardware timestamp of the IMU's internal clock for the published data. The point-cloud acquisition hardware time and the camera trigger pulse hardware timestamp are published on `/time_ref_scan` and `/imu/time_ref_cam` topics, respectively. The timestamp of the received PPS message from the GNSS receiver is published on the `/imu/time_ref_pps` topic. These hardware timestamps can accurately time reference sensor messages within a navigation algorithm. Alternatively, the synced bag of a sequence, which has the post-processed messages with corrected UTC stamps of the sensors, can be used. The block diagram of the hardware synchronization is shown in Figure 7.

5.5. Ground-truth data

In order to facilitate VIL algorithm evaluation, we provide four types of GNSS-based ground-truth data for all our data sequences. First, we provide the `/fix` ROS topic, which contains the raw RTK-GNSS message available in both raw

and synced bags. Second, we provide post-processed kinematic (PPK) RTK-GNSS messages stored in synced bags. The PPK solution for all our sequences, when processed with RTKPost from RTKLib (RTKLib, 2023), consistently achieves a quality index (Q) exceeding 95% and maintains a signal-to-noise ratio (SNR) continuously surpassing 40 on average. Third, we provide the post-processed GNSS multi-sensor 6DOF pose (position and orientation) in synced bags. For this purpose, the solution of the *VINS-Fusion* package (Qin et al., 2019)

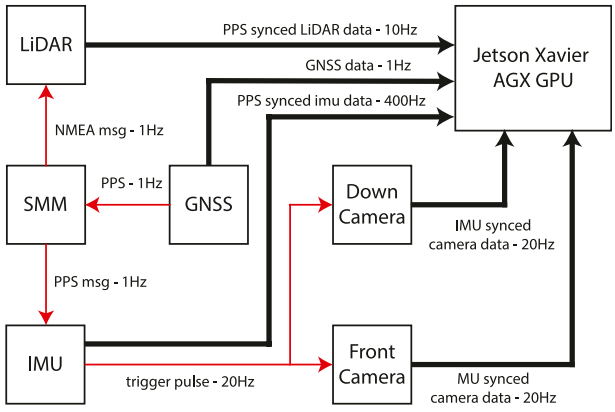


Figure 7. PPS time synchronization scheme.

with PPK GNSS factors enabled for each sequence is saved as a ROS *Odometry* message. As the fourth ground truth source, we provide the 6DOF pose (position and orientation) information communicated by the aerial platform's navigation system, that is, *DJI-M600* and *Bell412*, stored in synced bags. Details of all available ground-truth topics are summarized in Table 4.

It is recommended to use `/fix_ppk` for position ground truth evaluation and `/ins_ppk` for 6DOF pose ground truth evaluation as they are accurate RTK-GNSS based ground truth sources. Additionally, the `/fix_ppk` and `/ins_ppk` topics do not have corruptions due to base station initialization errors or momentary communication loss with the base station during operation.

6. Dataset evaluation

To assess the dataset suitability as a benchmark for multi-sensor fusion algorithms, we evaluated its performance using state-of-the-art estimation algorithms, including a visual-inertial algorithm *VINS-Fusion* (Qin et al., 2019), a Lidar-only odometry algorithm *A-LOAM* (Cao and Cui, 2019), a LiDAR-inertial odometry algorithm *FAST-LIO2* (Xu et al., 2022), and a visual-inertial and LiDAR combined algorithm *VI-LOAM* (Didula et al., 2022). *VI-LOAM* is a combination of *VINS-Fusion* and the LiDAR-only odometry algorithm *A-LOAM* to achieve a combined VIL solution. The results of our evaluation are presented in Table 6. Note that the *VINS-fusion* algorithm used here is capable of incorporating GNSS and loop closure constraints. However, these updates are disabled when calculating the solution in Table 6, that is, only the visual-inertial solution from the algorithm is shown. The " × " indicates the algorithm failed for the corresponding sequence.

6.1. Evaluation criteria

To evaluate the performance of multi-sensor fusion algorithms on our data sequences, we incorporated commonly used evaluation criteria (Schubert et al., 2018; Qin et al., 2019), which are presented below.

6.1.1. RMSE position [m]. The root mean squared error (RMSE)-position is computed by comparing the tracked positions $\widehat{\mathbf{p}}_i$ with the corresponding GNSS ground-truth positions $\mathbf{p}_i$, using the following formula:

$$e_p = \sqrt{\frac{1}{I_{gt}} \sum_{i=1}^{I_{gt}} \|\mathbf{p}_i - \widehat{\mathbf{p}}_i\|} \tag{1}$$

The total number of ground-truth data indices is denoted by I_{gt}. To perform a length-normalized evaluation of the results, the RMSE drift [%] is calculated by dividing the RMSE position [m] by the length of the sequence.

6.1.2. RMSE angle [deg]. This metric measures the root mean squared difference between the ground-truth orientation $\mathbf{R}_i (\in \mathbb{SO}_3)$ and the corresponding estimated orientation $\widehat{\mathbf{R}}_i (\in \mathbb{SO}_3)$, and is defined as follows:

$$e_R = \sqrt{\frac{1}{I_{gt}} \sum_{i=1}^{I_{gt}} \left\| Log_{SO_3}\left(\mathbf{R}_i \widehat{\mathbf{R}}_i^{\,T}\right) \right\|} \tag{2}$$

Ground-truth data indices are denoted by I_{gt} while the RMSE angle [%] is calculated similarly to RMSE position [%].

In order to calculate the RMSE position and RMSE angle, the */ins_ppk* topic of each sequence was used, that is, the *VINS-Fusion* solution of each sequence with PPK-GNSS factors enabled in the factor graph.

6.2. Results

This section presents the evaluation results of the data sequences. Figure 8 indicates trajectories of *Bell412* sequences and Figure 9 *DJI-M600* sequences.

Table 6 summarizes all sequences' position and orientation error results using state-of-the-art algorithms. The evaluation results presented in Table 6 exhibit comparable performance to those of existing VIL datasets, as shown in Table 5 when considering

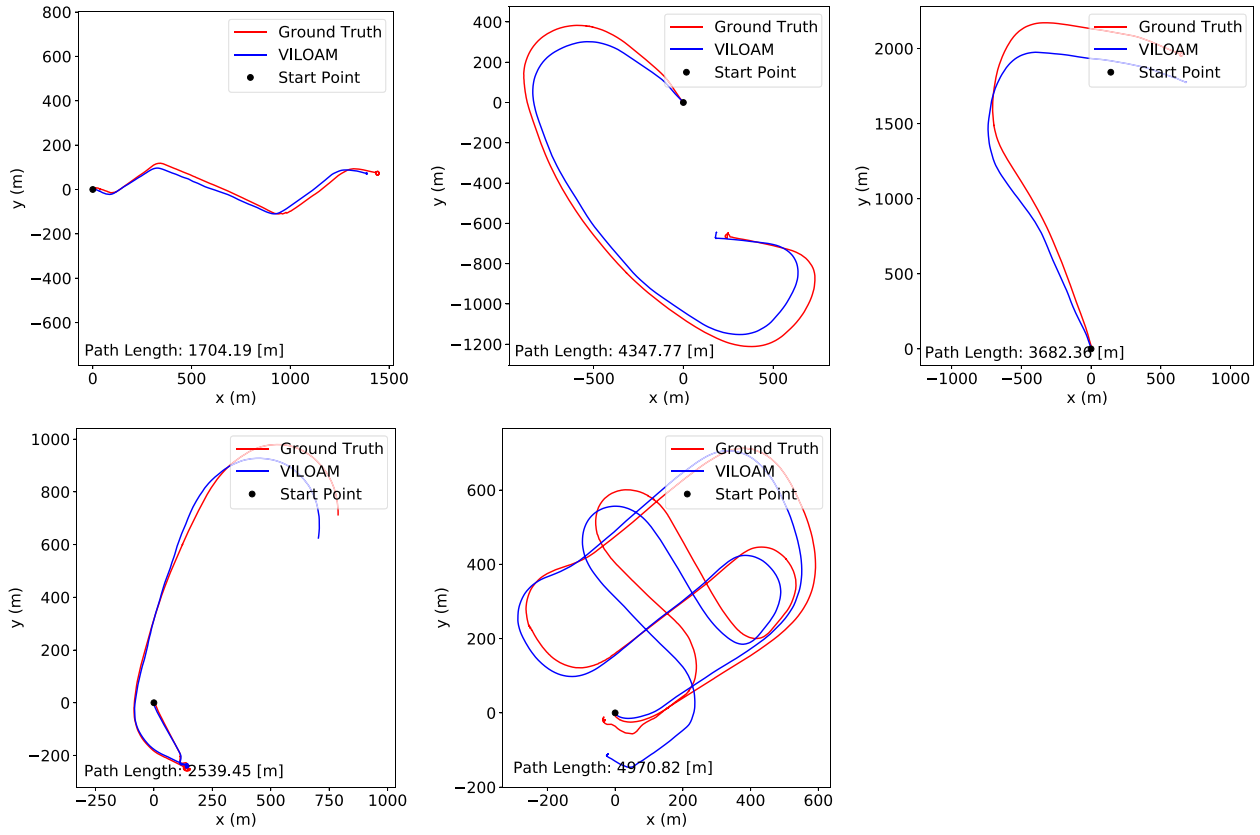


Figure 8. Trajectory estimation results of Bell412 sequences: Bell412-[1 to 3] (Top Row), Bell412-[4 and 5] (Bottom Row). The red line shows the ground truth trajectory. The presented results are obtained with real-time time-synchronized data and calibrated sensors using the *VI-LOAM*.

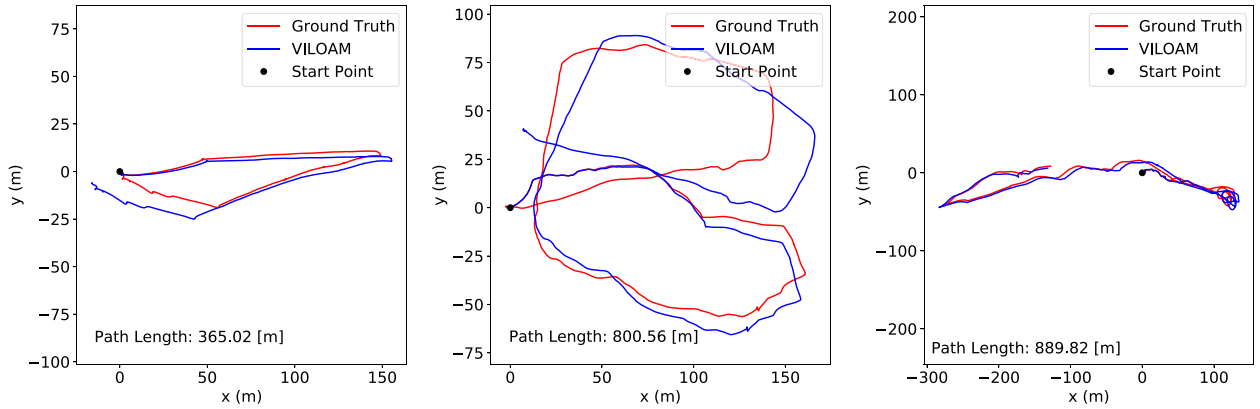


Figure 9. Trajectory estimation results of DJI-M600 sequences: Quarry-1, 2 and Lighthouse. The red line shows the ground truth trajectory. The presented results are obtained with real-time time-synchronized data and calibrated sensors using *VI-LOAM*.

RMSE drift [%]. RMSE drift [%] is a metric that captures the length-normalized drift of the solution, enabling the comparison of sequences with different lengths. For instance, our *Quarry-1* sequence demonstrates similar RMSE position drift [%] results for *VINS-Fusion* algorithm when compared to the existing *Jackel* and *eee_03* sequences. Likewise, the *Handheld* and *Bell412-3* sequences exhibit similar RMSE position [m] results for the *VINS-Fusion* algorithm.

The VI navigation algorithm, *VINS-Fusion* showed slightly higher drift in performance results for all *Bell412* sequences, as shown in Table 6. This can be attributed to the degradation of sensors on the full-scale platform related to altitude, vibrations, and flight speeds. However, when we

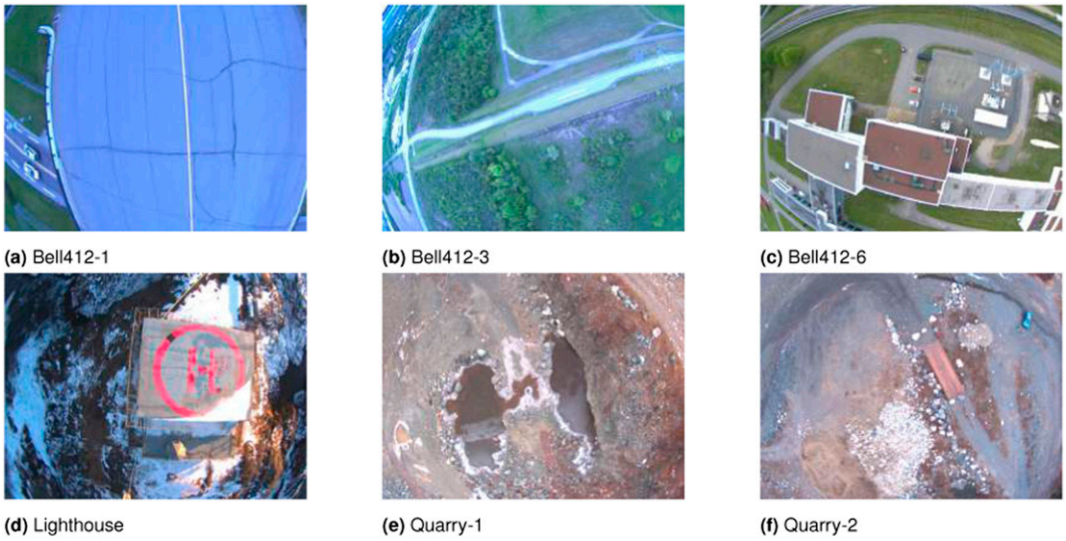


Figure 10. Sample snapshots of scenes captured in each dataset sequence.

Table 5. Existing Aerial VIL Datasets Evaluations on Publicly Available Algorithms.

		VINS-fusion[1]		A-LOAM[2]	
Dataset	Length (m)	RMSE position (m)	RMSE drift (%)	RMSE position (m)	RMSE drift (%)
Jackal	427	7.58	1.78	23.29	5.45
Handheld	391	87.11	22.28	63.78	16.31
eee_03	128	1.037	0.81	4.380	3.42
nya_03	316	1.333	0.42	0.683	0.22
sbs_03	198	1.306	0.66	4.732	2.39

used the *VI-LOAM* algorithm for the same sequences, it showed comparable performance in some sequences (i.e., *Bell412-1*) and better performance in others. This indicates that our dataset presents challenging conditions for developing VI and VIL navigation algorithms.

In our evaluation, the *VINS-Fusion* algorithm could estimate the trajectories for all the sequences until the end, except for *Bell412-2*. The rolling initialization of the *Bell412-2* sequence caused the *VINS-Fusion* algorithm to fail since it is not designed to handle such scenarios. This sequence is publicly made available to test the rolling initialization capability of VIL algorithms.

The *FAST-LIO2* algorithm estimated the trajectories for all *DJI-M600* sequences, which had structured terrains and were flown at altitudes between 20m and 60m. However, *FAST-LIO2* failed for all *Bell412* sequences. In particular, *Bell412-1* sequence had a low-altitude flight over a structure-less terrain, that is, an airport runway, which caused the algorithm to fail. The other *Bell412* sequences were comprised of high-altitude (25m – 170m) flights, resulting in low LiDAR point registrations and a lack of features in the point-clouds in mid-flight, which also led to the failure of *FAST-LIO2* algorithm.

The *VI-LOAM* combines the *A-LOAM* and *VINS-Fusion* algorithms to suit both structured and non-structured environments seen in the dataset. To validate its suitability for multi-sensor GNSS-denied navigation, the *VI-LOAM* algorithm was evaluated using our dataset. By incorporating visual, LiDAR, inertial, and compass information for state estimation, it successfully estimated the trajectories for all sequences from start to end. Example results are presented in Figures 5 and 6, showing trajectory and mapping results for the *VI-LOAM* algorithm applied to the *Bell412-1* and *Bell412-6* datasets. The *VI-LOAM* solution demonstrated improved performance compared to *ALOAM* and *VINS-Fusion* algorithms across all sequences. In the case of *DJI-M600* sequences, *FAST-LIO2* has demonstrated superior accuracy. Hence, there is a potential for future VIL developments to incorporate *FAST-LIO2* like navigation modules in the VIL pipeline.

The long-range outdoor sequences in our dataset capture the environmental conditions and characteristics for applications similar to last-mile goods delivery. It represents the sensor degradation sources of full-scale platforms, which can cause significant drift in state-of-the-art algorithms developed for ground-based or low-altitude

Table 6. State of the art Estimation Method Comparison for the Datasets.

Dataset	Length [m]	Error RMSE	VINS-Fusion[4]	A-LOAM	Fast-LIO	LVI-SAM	VI-LOAM
Quarry-1	357	position [m]	11.27	63.32	7.89	×	10.27
		angle [deg]	10.23	106.8	6.23		9.23
		position [%]	2.870	17.74	2.210		2.170
		angle [%]	0.620	29.91	1.746		0.320
Quarry-2	807	position [m]	13.26	78.54	9.87	×	11.26
		angle [deg]	13.11	114.3	10.99		12.11
		position [%]	1.641	9.730	1.109		1.041
		angle [%]	1.622	14.17	1.235		1.022
Lighthouse	890	position [m]	8.531	55.25	5.25	×	8.12
		angle [deg]	11.87	104.1	6.13		11.47
		position [%]	0.960	6.210	0.590		0.870
		angle [%]	1.331	11.69	0.688		1.001
Bell412-1	1709	position [m]	102.6	×	×	×	15.48
		angle [deg]	5.73				0.065
		position [%]	75.55				0.903
		angle [%]	4.37				0.004
Bell412-2	-	position [m]	-	×	×	-	×
		angle [deg]	-			-	
		position [%]	-			-	
		angle [%]	-			-	
Bell412-3	4336	position [m]	56.35	×	×	×	54.78
		angle [deg]	144.39				1.737
		position [%]	1.30				1.263
		angle [%]	3.32				0.045
Bell412-4	3656	position [m]	352.4	×	×	×	93.33
		angle [deg]	92.10				1.716
		position [%]	9.61				2.553
		angle [%]	2.51				0.075
Bell412-5	2138	position [m]	63.96	×	×	×	19.37
		angle [deg]	29.66				2.512
		position [%]	2.99				0.906
		angle [%]	1.16				0.117
Bell412-6	4938	position [m]	259.3	×	×	×	44.76
		angle [deg]	56.35				1.302
		position[%]	5.25				0.906
		angle [%]	1.14				0.026

[4] VINS-Fusion: Used the visual-inertial solution only without the GNSS and loop-closure measurement constraints.

applications (Zhang and Singh, 2014). Thus, our dataset can serve as a challenging benchmark test for further VIL navigation research.

7. Conclusions

This paper presents a new dataset that includes kilometer-range aerial sequences in various scenes for evaluating VIL algorithms on both small-scale and full-scale platforms using an interchangeable sensing payload unit. The dataset includes high-resolution images and LiDAR point clouds that are hardware time-synchronized with IMU and GNSS data. To facilitate evaluation, the dataset includes ground truth with centimeter-level accuracy for all sequences and data sequences for multi-sensor calibration. The dataset is publicly available with both raw and calibrated data in the following link https://mun-frl-vil-dataset.readthedocs.io/en/latest/

The test flight paths resemble last-mile goods delivery scenarios, including long trajectories and comparable altitudes, making them suitable for evaluating autonomous algorithm performance in real-world full-scale flight applications. The dataset comprises nine sequences, offering critical data for evaluating autonomous algorithm performance. Additionally, the dataset offers the possibility to evaluate algorithms related to landing zone detection, GNSS-denied navigation, and obstacle avoidance. However, it should be noted that the map resolution, limited by using VLP-16 LiDAR, is more suitable for small-scale aircraft landing zone detection. A sufficiently higher resolution map of an area should be captured for full-scale aircraft safe landing zone identification.

This work revealed several unresolved challenges that navigation algorithms face, such as the need for VIL algorithms capable of handling long-range flights, high-altitude operation, featureless environments, sensor degradation, and full-scale platform vibrations. Therefore, we believe that our dataset can serve as a challenging benchmark for assessing the capabilities of VIL algorithm developments.

Acknowledgments

The authors would like to acknowledge the contributions of Sahan Gunawardana, an undergraduate student in mechanical engineering, in conducting the mechanical design and manufacturing of the payload unit. The authors would also like to acknowledge the development and testing support of the Flight Research Lab—National Research Council of Canada and the Intelligent Systems Lab—Memorial University of Newfoundland.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Research Council of Canada's Artificial Intelligence for Logistics Program, in part by the Natural Sciences and Engineering Research Council of Canada, and in part by the Memorial University of Newfoundland.

ORCID iD

Ravindu G Thalagala https://orcid.org/0009-0007-4640-2274

References

Adolfsson D, Magnusson M, Alhashimi A, et al. (2022) Lidar-level localization with radar? The CFEAR approach to accurate, fast, and robust large-scale radar odometry in diverse environments. *IEEE Transactions on Robotics* 39: 1476–1495. DOI: 10.1109/TRO.2022.3221302.

Aurambout JP, Gkoumas K and Ciuffo B (2019) Last mile delivery by drones: an estimation of viable market potential and access to citizens across European cities. *European Transport Research Review* 11(1): 21–30. DOI: 10.1186/S12544-019-0368-2/FIGURES/20.

Boysen N, Fedtke S and Schwerdfeger S (2020) Last-mile delivery concepts: a survey from an operational research perspective. *Spectrum* 43(1): 1–58. DOI:10.1007/S00291-020-00607-8.

Burri M, Nikolic J, Gohl P, et al. (2016) The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research* 35(10): 1157–1163. DOI: 10.1177/0278364915620033.

Cao S and Cui TX (2019) GitHub - HKUST-aerial-robotics/A-LOAM: advanced implementation of LOAM. https://github.com/HKUST-Aerial-Robotics/A-LOAM.

Carlevaris-Bianco N, Ushani AK and Eustice RM (2015) University of Michigan North Campus long-term vision and lidar dataset:. 35(9): 1023–1035. DOI:10.1177/0278364915614638.

Chen KW, Xie MR, Chen YM, et al. (2022) DroneTalk: an internet-of-things-based drone system for last-mile drone delivery. *IEEE Transactions on Intelligent Transportation Systems* 23(9): 15204–15217. DOI: 10.1109/TITS.2021.3138432.

Cisneros I, Yin P, Zhang J, et al. (2022) *ALTO: A Large-Scale Dataset for UAV Visual Place Recognition and Localization*. https://arxiv.org/abs/2207.12317v1.

Cohen AP, Shaheen SA and Farrar EM (2021) Urban air mobility: history, ecosystem, market potential, and challenges. *IEEE Transactions on Intelligent Transportation Systems* 22(9): 6074–6087. DOI: 10.1109/TITS.2021.3082767.

Didula D, Oscar DS and Kusal T (2022) *VI-LOAM_ISLAB*. https://github.com/didzdissanayaka8/VI-LOAM_ISLAB.

Ding W, Hou S, Gao H, et al. (2020) LiDAR inertial odometry aided robust LiDAR localization system in changing city scenes. 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May 2020. DOI: 10.1109/ICRA40945.2020.9196698.

Faizullin M, Kornilova A and Ferrer G (2022) Open-source LiDAR time synchronization system by mimicking GNSS-clock. IEEE international symposium on precision clock synchronization for measurement, control, and communication, Geneva, Switzerland, 2–7 October 2022. DOI: 10.1109/ISPCS55791.2022.9918446.

Fonder M, Van Droogenbroeck M and Droogenbroeck VM (2019) Mid-air: a multi-modal dataset for extremely low altitude drone flights. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Piscataway, NJ: IEEE. DOI:10.1109/CVPRW.2019.00081.

Furgale P, Rehder J and Siegwart R (2013) Unified temporal and spatial calibration for multi-sensor systems. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Piscataway, NJ: IEEE. DOI: 10.1109/IROS.2013.6696514.

Geiger A, Lenz P, Stiller C, et al. (2013) Vision meets robotics: the KITTI dataset. *The International Journal of Robotics Research* 32(11): 1231–1237. DOI: 10.1177/0278364913491297.

Gomaa MA, De Silva O, Mann GK, et al. (2020) Observability-constrained VINS for MAVs using interacting multiple model algorithm. *IEEE Transactions on Aerospace and Electronic Systems* 57: 1423–1442. DOI: 10.1109/TAES.2020.3043534.

Jeong J, Cho Y, Shin YS, et al. (2019) Complex urban dataset with multi-level sensors from highly diverse urban environments 38(6): 642–657. DOI:10.1177/0278364919843996.

Jiao J, Ye H, Zhu Y, et al. (2022) Robust odometry and mapping for multi-LiDAR systems with online extrinsic calibration. *IEEE Transactions on Robotics* 38(1): 351–371. DOI: 10.1109/TRO.2021.3078287.

Liang Y, Muller S, Schwendner D, et al. (2020) A scalable framework for robust vehicle state estimation with a fusion of a low-cost IMU, the GNSS, radar, a camera and lidar. Piscataway, NJ: IEEE. DOI: 10.1109/IROS45743.2020.9341419.

Maddern W, Pascoe G, Linegar C, et al. (2016) 1 Year, 1000 Km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research* 36(1): 3–15. DOI: 10.1177/0278364916679498.

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. DOI: 10.1177/0278364917702237.

Miranda VR, Rezende AM, Rocha TL, et al. (2022) Autonomous navigation system for a delivery drone. *Journal of Control, Automation and Electrical Systems* 33(1): 141–155. DOI: 10.1007/S40313-021-00828-4/TABLES/2.

Nguyen TM, Yuan S, Cao M, et al. (2021) NTU VIRAL: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. *The International Journal of Robotics Research* 41(3): 1–11. DOI: 10.1177/02783649211052312.

Pei Z, Fang T, Weng K, et al. (2022) Urban on-demand delivery via autonomous aerial mobility: formulation and exact algorithm. *IEEE Transactions on Automation Science and Engineering* 20: 1675–1689. DOI: 10.1109/TASE.2022.3184324

Perreault M and Behdinan K (2021) Delivery drone driving cycle. *IEEE Transactions on Vehicular Technology* 70(2): 1146–1156. DOI: 10.1109/TVT.2021.3053536.

Qin T, Pan J, Cao S, et al. (2019) *A General Optimization-Based Framework for Local Odometry Estimation With Multiple Sensors*. https://doi.org/10.48550/arXiv.1901.03638

Ramezani M, Wang Y, Camurri M, et al. (2020) The newer college dataset: Handheld LiDAR, inertial and vision with ground truth. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE. DOI: 10.1109/IROS45743.2020.9340849.

RTKLib (2023) Understanding PPK solution and analyzing logs from Reach | RTK Modules. https://docs.emlid.com/reach/tutorials/post-processing-workflow/analyzing-logs/.

Schleiss M, Rouatbi F and Cremers D (2022) *VPAIR – Aerial Visual Place Recognition and Localization in Large-Scale Outdoor Environments*. https://arxiv.org/abs/2205.11567v1.

Schubert D, Goll T, Demmel N, et al. (2018) The TUM VI benchmark for evaluating visual-inertial odometry. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE. DOI: 10.1109/IROS.2018.8593419.

Shan T, Englot B, Meyers D, et al. (2020) LIO-SAM: tightly-coupled lidar inertial odometry via smoothing and mapping. *IEEE International Conference on Intelligent Robots and Systems*. Piscataway, NJ: IEEE. DOI: 10.48550/arxiv.2007.00258.

Shan T, Englot B, Ratti C, et al. (2021) *LVI-SAM: Tightly-Coupled Lidar-Visual-Inertial Odometry Via Smoothing And Mapping*. https://doi.org/10.48550/arXiv.2104.10831

Song S, Lim H, Lee AJ, et al. (2022) DynaVINS: a visual-inertial SLAM for dynamic environments. *IEEE Robotics and Automation Letters* 7: 11523–11530. DOI: 10.1109/LRA.2022.3203231.

Thalagala RG, De Silva O, Mann GK, et al. (2021) Two key-frame state marginalization for computationally efficient visual inertial navigation. *2021 European Control Conference, ECC 2021*. Piscataway, NJ: IEEE. DOI: 10.23919/ECC54610.2021.9654924.

Wang W, Zhu D, Wang X, et al. (2020) TartanAir: a dataset to push the limits of visual SLAM. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE. DOI: 10.1109/IROS45743.2020.9341801.

Wenzel P, Wang R, Yang N, et al. (2020) 4Seasons: a cross-season dataset for multi-weather SLAM in autonomous driving. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Berlin, Germany: Springer. DOI: 10.1007/978-3-030-71278-5_29.

Xu W, Cai Y, He D, et al. (2022) FAST-LIO2: fast direct LiDAR-inertial odometry. *IEEE Transactions on Robotics* 38(4): 2053–2073. DOI: 10.1109/TRO.2022.3141876.

Yang X and Wei P (2021) Autonomous free flight operations in urban air mobility with computational guidance and collision avoidance. *IEEE Transactions on Intelligent Transportation Systems* 22(9): 5962–5975. DOI: 10.1109/TITS.2020.3048360.

Zhang J and Singh S (2014) LOAM: lidar odometry and mapping in real-time. Robotics: Science and Systems, Berkeley, CA, 10 July 2014. DOI: 10.15607/RSS.2014.X.007.

Zuñiga-Noël D, Jaenal A, Gomez-Ojeda R, et al. (2020) The UMA-VI dataset: visual–inertial odometry in low-textured and dynamic illumination environments: 39(9): 1052–1060. DOI:10.1177/0278364920938439.