

# The EDI Multi-Modal Simultaneous Localization and Mapping Dataset (EDI-SLAM)

Peteris Racinskis <sup>\*</sup>, Gustavs Krasnikovs , Janis Arents  and Modris Greitans 

Institute of Electronics and Computer Science (EDI), LV-1006 Riga, Latvia; gustavs.krasnikovs@edi.lv (G.K.); janis.arents@edi.lv (J.A.); modris\_greitans@edi.lv (M.G.)

\* Correspondence: peteris.racinskis@edi.lv

**Abstract:** This paper accompanies the initial public release of the EDI multi-modal SLAM dataset, a collection of long tracks recorded with a portable sensor package. These include two global shutter RGB camera feeds, LiDAR scans, as well as inertial and GNSS data from an RTK-enabled IMU-GNSS positioning module—both as satellite fixes and internally fused interpolated pose estimates. The tracks are formatted as ROS1 and ROS2 bags, with separately available calibration and ground truth data. In addition to the filtered positioning module outputs, a second form of sparse ground truth pose annotation is provided using independently surveyed visual fiducial markers as a reference. This enables the meaningful evaluation of systems that directly utilize data from the positioning module into their localization estimates, and serves as an alternative when the GNSS reference is disrupted by intermittent signals or multipath scattering. In this paper, we describe the methods used to collect the dataset, its contents, and its intended use.

**Keywords:** robotics; SLAM; computer vision; LiDAR; dataset; ROS



Academic Editor: Juanle Wang

Received: 1 November 2024

Revised: 3 December 2024

Accepted: 5 January 2025

Published: 7 January 2025

**Citation:** Racinskis, P.; Krasnikovs, G.; Arents, J.; Greitans, M. The EDI Multi-Modal Simultaneous Localization and Mapping Dataset (EDI-SLAM). *Data* **2025**, *10*, 5. <https://doi.org/10.3390/data10010005>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

While in a traditional static, industrial setting planning a robot's trajectory can be treated as an open-loop problem, operating a mobile robot autonomously in a previously unseen environment requires a means of perceiving its surroundings and localizing itself. If onboard sensors are utilized to place the robot with respect to a model of the environment, but this model is built incrementally by using these position and orientation estimates, we arrive at the circular Simultaneous Localization and Mapping (SLAM) problem, which is one of the core challenges in robot autonomy [1]. Considering the wide range of potential applications for a SLAM system, there has been no shortage of research interest in this direction over the decades, resulting in functional systems using sensors such as LiDAR [2], depth imagery [3], RGB images [4], stereo imagery, and IMU data [5].

The need to evaluate systems—both individually and competitively, as part of the iterative development process and for comparison—has also provided ample incentive for the collection and release of public SLAM datasets [6]. However, due to the wildly diverging requirements faced by developers of systems intended for various use cases, these come in many mutually incompatible formats. Moreover, for any given target application, few—if any—will approximate the anticipated input data distribution very well. Finally, the collection of these data is expensive and time-consuming, with considerable storage space requirements. Taken all together, we assert that there is still a pressing need for more publicly available SLAM datasets, collected in a wide range of environments.

To this end, we are publishing EDI-SLAM, a collection of calibrated sensor tracks primarily intended for the development, testing, and evaluation of LiDAR-, visual-, and

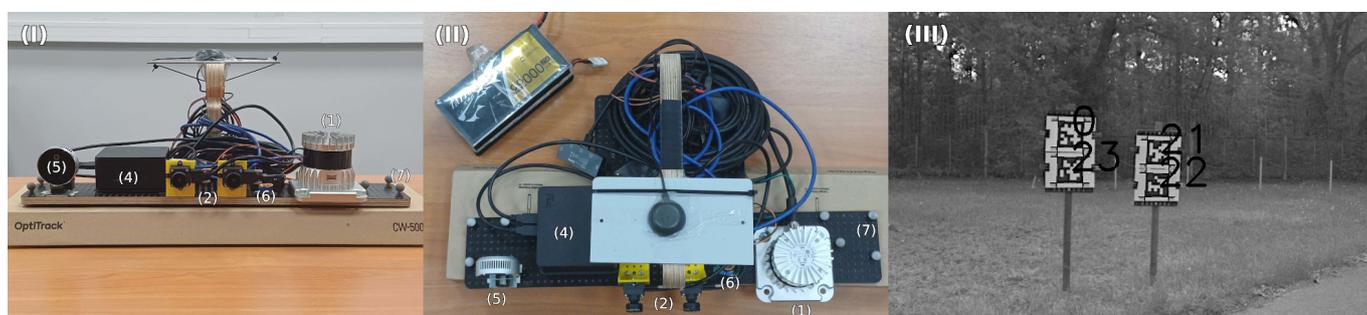
stereo-inertial SLAM systems with a focus on unstructured outdoor environments, which may involve choosing to also integrate GNSS data in environments where it may be intermittent and unreliable. To aid with performing semantic inference on the data, all images are collected in color. At the writing of this article, this contains a mix of recordings originally made for internal use in ongoing SLAM system development, with and without ground truth positioning data, collected in three environments, as pictured in Figure 1:

- The EDI courtyard and its surroundings—an urban landscape, a mix of structured built-up scenery and some dense vegetation, notable for poor-quality GNSS positioning data due to occlusions and multipath scattering;
- An open-field landscape, with high-quality GNSS data but segments that make frame-to-frame tracking difficult due to a lack of reliable tracking features;
- A forest road network—long, straight tracks with intermittent GNSS data and highly repetitive scenery, recorded on a vehicle.



**Figure 1.** The three types of scenery represented in the initial release of the dataset: (I) *courtyard*—the EDI courtyard and surroundings, a semi-structured urban environment; (II) *saga*—an open field with good GNSS coverage but a shortage of reliable tracking features; (III) *ropazi*—a forest road network.

We use the fiducial marker method described in [7] and depicted in Figure 2 as the ground truth localization reference at a number of locations, whereby we pre-position plates with Apriltag [8] markers and survey the location of their corners. This provides a reference that is independently distributed with regard to the IMU-GNSS sensor module’s estimates, allowing for meaningful evaluation of SLAM systems that fuse GNSS data in environments where the satellite positioning signal is intermittent or has been distorted by environmental factors. Due to the ubiquity of the *Robot Operating System* (ROS) [9] robotics framework, we have elected to store the dataset in the form of ROS bag files, which greatly simplify usage through the playback mechanism. Bags are provided in both the ROS1 and ROS2 [10] file formats.



**Figure 2.** (I) Front and (II) top views of the sensor package, with some of the sensors indicated. (III) One of the images processed for reference pose measurement in the *courtyard\_gt* track, showing a tracking gate with detected marker indices and their corners drawn in black.

## 2. Background

### 2.1. Conventions

The outputs of a SLAM system, and therefore the ground truth annotations of this dataset, are expressed as **poses**—proper rigid transformations from  $\mathbb{R}^3$  to itself. The trajectory pose reference files in this dataset use the quaternion formulation:

$$\begin{aligned} T_a^b &= (\mathbf{t}, \mathbf{q}) \\ \mathbf{t} &\in \mathbb{R}^3 \\ \mathbf{q} &= (q_x, q_y, q_z, q_w) \in \mathbb{H} : |\mathbf{q}| = 1 \\ T_a^b(\mathbf{x}_a) &= \mathbf{x}_b \end{aligned} \quad (1)$$

where  $\mathbf{t}$  is a translation vector and  $\mathbf{q}$  a rotation unit quaternion representing these components of a transformation with regard to a fixed reference frame in Euclidean space, with  $q_w$  being the real coefficient. However, in calibration files and to describe computations, we use the pose matrix notation:

$$T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \text{SE3} \subset \mathbb{R}^{4 \times 4} \quad (2)$$

where  $R \in \text{SO3}$  is a rotation matrix. The notation convention adopted in this paper is that the transformation  $T_a^b$  applied to a vector  $\mathbf{x}_a$  in basis  $a$  (the child frame) produces  $\mathbf{x}_b$  in  $b$  (the parent frame).

When working with GNSS data, it is important to understand the geometrical model used. Many commercial receivers such as the *Xsens MTi-680G RTK* (Xsens Technologies B.V., Enschede, The Netherlands) used in this work provide position outputs in *WGS84* ellipsoid coordinates  $(\phi, \psi, h)$  [11] where  $\phi, \psi$  are the angular latitude and longitude, while the altitude  $h$  is expressed in terms of normal height above the reference ellipsoid. To compare these to position estimates obtained by a SLAM system, conversion to a Euclidean coordinate system is necessary, such as the *WGS84* Earth-centered Earth-fixed (ECEF) Euclidean frame [12].

In addition, by default, the *Xsens* module outputs orientation in a local Earth-aligned reference frame referred to as east–north–up (ENU), where the  $x$  coordinate axis points east,  $y$  points north (in the case of this real time kinematic (RTK)-enabled device, true north [13]), and  $z$  points normal to the surface [14]. This can be changed in the device configuration, but we have elected to retain this setting for applications that require a gravity-aligned reference frame. The conversion between this and the ECEF orientation is therefore a function of  $\phi$  and  $\psi$ , though over small distances, it maybe be treated as a static calibration. When computing the GNSS reference pose track, we use a spherical approximation at each ECEF position  $\mathbf{t}$ , expressed in matrix form as follows:

$$\begin{aligned} \hat{\mathbf{z}} &= \frac{\mathbf{t}}{\|\mathbf{t}\|} \\ \hat{\mathbf{x}} &= \frac{\hat{\mathbf{Z}} \times \hat{\mathbf{z}}}{\|\hat{\mathbf{Z}} \times \hat{\mathbf{z}}\|} \\ \hat{\mathbf{y}} &= \hat{\mathbf{z}} \times \hat{\mathbf{x}} \\ R_{ENU}^{ECEF}(\mathbf{t}) &= \begin{bmatrix} \hat{\mathbf{x}}(\mathbf{t}) & \hat{\mathbf{y}}(\mathbf{t}) & \hat{\mathbf{z}}(\mathbf{t}) \end{bmatrix} \end{aligned}$$

$$T_{xsens}^{ECEF}(\mathbf{t}) = T_{ENU}^{ECEF}(\mathbf{t})T_{xsens}^{ENU} = \begin{bmatrix} R_{ENU}^{ECEF}(\mathbf{t}) & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} R_{xsens}^{ENU} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (3)$$

where  $\hat{\mathbf{Z}}$  is the ECEF z-axis unit vector,  $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$  are the unit vectors of the coordinate axes in the rotated reference frame, and  $R_{xsens}^{ENU}$  is the orientation as reported by the measurement device. The latter are also available in the bag files, for direct use.

## 2.2. SLAM Datasets

For a broad overview of the SLAM problem, its formulation, and approaches in solving it and more discussion on existing datasets, we direct the reader to our prior review of the field in [1]. Furthermore, an exhaustive resource on publicly available SLAM datasets can be found at [6]. Among those that share some overlap in terms of structure or purpose with ours are the following:

- *NTU VIRAL (A Visual–Inertial–Ranging–Lidar Dataset for Autonomous Aerial Vehicles)* (2022) [15] uses a similar sensor setup (LiDAR, IMU, a pair of cameras) and also employs an external ground truth annotation method. Unlike us, they provide two LiDAR scanners located at right-angle planes to each other, which is more important in an aerial vehicle, but do not include GNSS data in their tracks. Their ground truth localization system is not designed for long, open-ended trajectories, instead utilizing a set of stationary ultra-wideband (UWB) radars to continuously localize the drone in a confined working area.
- *The Multi-Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception* (2018) [16], as the name suggests, is primarily intended for use in the development of systems utilizing event cameras. Despite treating the LiDAR primarily as a ground truth estimation tool, it nevertheless provides both LiDAR and GNSS data in addition to the primary stream of event camera and grayscale images. Furthermore, some of the trajectories in this dataset are also on the kilometer scale in length, enabling use in similar target applications in comparison to ours.
- The *KITTI* [17] and *KITTI-360* [18] datasets provide painstakingly annotated benchmarks for semantic scene understanding, and therefore greatly differ from our data both in scale and purpose. However, these do provide a similar sensor suite to ours alongside reliable ground truth localization.
- The *TUM-VI* [19] visual odometry benchmark provided the inspiration for our portable sensor package collection method. However, it (alongside many other popular datasets collected in a similar manner) does not include LiDAR data. Moreover, they only provide GNSS-invariant ground truth annotation at the starts and ends of trajectories, where motion tracking equipment can be deployed.
- The *Malaga 2009* dataset [20] also consists of calibrated LiDAR, camera, and GNSS readings, with the GNSS data being used to provide a ground truth trajectory. Furthermore, they conduct a much more extensive analysis of the ground truth uncertainty metrics. The greatest practical differences between their approach and ours are the platform, collection environment and purpose. We use a hand-held sensor package for most recordings, which is often lower, slower, and less stable (particularly in orientation); we collect data in less structured and/or GNSS-inhibited environments; we also target the evaluation of systems that directly fuse GNSS data, necessitating GNSS-independent ground truth measurements.

What differentiates EDI-SLAM from most currently existing datasets is the intended use case and collection environment. We emphasize wholly or partially unstructured environments, long trajectories (on the order of kilometers) where drift accumulation poses a considerable challenge in spatial consistency and loop closure detection, and sensor

data suitable for use with semantic segmentation algorithms (pinhole projection color imagery). We provide valid ground truth annotation for SLAM systems which may use any combination of LiDAR, IMU, RGB, and GNSS data in their localization estimates. Our collection methodology is platform-agnostic, being collected using a portable sensor package which can be carried by hand or mounted on a vehicle, such as a car in the case of the *ropazi* track. We believe our dataset fills a gap in the currently available resources—low-to-the-ground, relatively low-velocity recordings from unstable platforms (e.g., lightweight UGVs) in mixed urban, rural, and forest terrain, specifically featuring LiDAR data. The additional ground truth reference obtained using the markers also enables the evaluation of SLAM systems which directly fuse the onboard GNSS data in partially GNSS-denied environments.

### 3. Materials and Methods

#### 3.1. Hardware Setup

All recordings were made with an evolved version of the hand-portable sensor package first discussed in [7] and depicted in Figure 2. It mounts the following sensors and equipment, with numbers corresponding to their tags in Figure 2:

1. *Ouster OS1 rev7* 32-line mechanical LiDAR, which outputs point clouds at 10 Hz and IMU data at 100 Hz.
2.  $2 \times$  *Basler Dart 1920-160uc* global shutter cameras with 4 mm lenses. These are used with a software trigger for stereo applications, produce a center-cropped image at  $480 \times 752$  pixels, and were configured to collect RGB images at 30 frames per second for the recordings.
3. *Xsens MTi-680g* RTK GNSS-IMU navigation unit (occluded in the image).
4. *Intel nuc* compact form factor general-purpose computer, running *Ubuntu Linux 20.04 LTS* and *ROS1 Noetic*, to serve as the ROS master and perform data collection;
5. *Intel RealSense L515* depth camera (not used for this application);
6. Reflective markers for testing and calibration in the *Optitrack* motion tracking system;
7. A voltage regulator that can supply the LiDAR and PC from a 6S Li-Po battery or direct current power supply.

The official ROS drivers [21] are used to interface with the LiDAR. To employ the *MTi-680g* with RTK, we use a local fork of the now-superseded driver at [22]. We provide our own RTK base station located on EDI grounds. The cameras are operated through a ROS front-end for the *Pylon SDK* [23] developed for internal use, though ROS drivers are available [24]. For time synchronization, the *Ouster OS1* ROS driver was configured to use the host system's time stamps, the *Xsens* was initialized with UTC time from the *nuc* in the manufacturer's *mtmanager* software, and the images are stamped with the software trigger time.

#### 3.2. Calibration

Figure 3 shows the reference frames of the sensor package and the calibrated transforms between them, with frames belonging to different physical devices being coded in different colors. *os\_sensor* is the parent frame of the point clouds published by the LiDAR. The LiDAR's internal calibration—transformed between *os\_sensor*, *os\_imu*, and *os\_lidar*—is provided by the device's ROS driver and does not require user calibration. To obtain the camera-intrinsic—as well as the camera-to-camera-, camera-to-LiDAR-, and camera-to-*xsens*-extrinsic calibration matrices—we use *kalibr* [25] on the same recording of the sensor package in motion with a  $6 \times 6$  *Apriltag* [8] grid in view. The intrinsic and camera-to-camera parameters are obtained using the multi-camera calibration tool, whereas both the camera-to-*xsens* and camera-to-LiDAR calibration is performed in camera-to-imu

mode—we use the *Ouster’s* built-in IMU for reference. The LiDAR-to-xsens transform is then obtained as

$$T_{xsens}^{os\_sensor} = T_{os\_imu}^{os\_sensor} (T_{os\_imu}^{cam_0})^{-1} T_{xsens}^{cam_0} \tag{4}$$

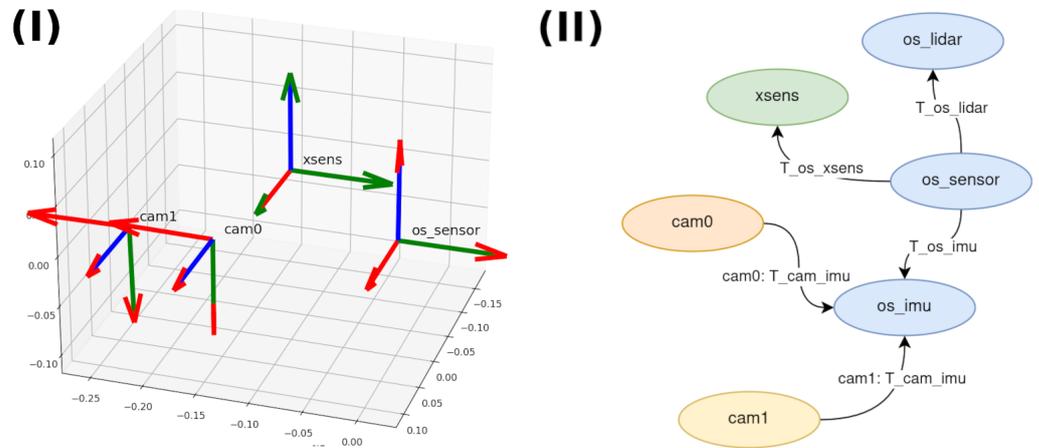


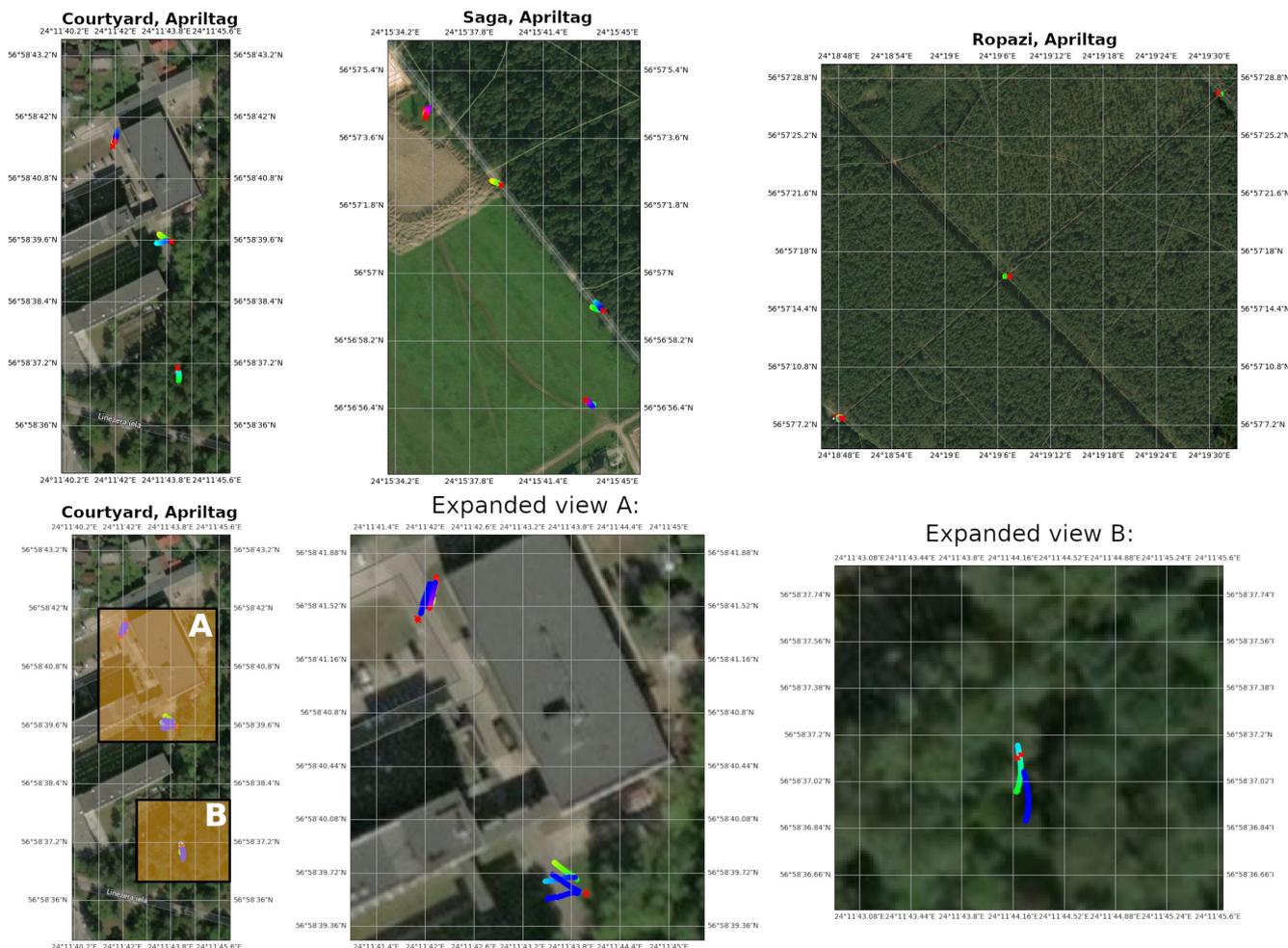
Figure 3. (I) A view of the sensor frames. (II) The transform graph.

### 3.3. Collection

Over the course of developing the methodology used in this dataset, we performed dozens of collection runs over the spring and summer of 2024, most of which were not prepared for use with external reference poses and many only include partial sensor sets. Two such runs, *courtyard\_no\_gt* and *saga\_no\_gt*, were deemed to be of sufficient quality to be included in the public dataset. However it must be noted that the calibration provided with these was performed later, during the collection of the annotated tracks, and is therefore not as accurate. The main part of the dataset as it is provided at initial release consists of the three trajectories pictured in Figure 4. Two different ways of carrying the sensor package were employed during collection—both *courtyard* tracks and *saga\_gt* were collected by hand; for *saga\_no\_gt* and *ropazi\_gt*, it was attached to the roof of a car using an ad hoc fixture, to ease data collection over long distances.



Figure 4. Cont.



**Figure 4.** Satellite maps with reference positions—made with GNSS-IMU (top) and visual marker tracking gates (middle). Expanded views (bottom) of marker-derived reference poses (solid blue), markers (red), and corresponding GNSS-IMU estimates (gradient).

### 3.4. Ground Truth Measurement

Before each collection run, a number of tracking gates (see Figure 2, item III) is placed along the planned route. One is always placed at the start of the track, while the others are spaced to maximize their value in measuring anticipated tracking drift. Additionally, in the case of the EDI courtyard track, the gates are placed in locations where significant GNSS degradation is predicted. Each consists of four markers mounted in a broadly rectangular pattern on two stands. Each marker corner is uniquely identified, and its position is surveyed, providing  $xy$  measurements in the *LKS92* geodetic coordinate system [26] and heights referenced to the *LV'14* quasi-geoid model [27]. These are converted first to lat, lon, and alt coordinates and then to positions in ECEF. The reference poses corresponding to each image with a sufficient number of corner detections (in this case, 16) are then independently estimated using the *SQPNP* perspective- $n$ -point algorithm [28], implemented in *OpenCV* [29]. As is clearly seen in the middle row of maps of Figure 4, the tracking gate method produces a much sparser set of reference poses. However, as previously shown in [7] and discussed below, these are of higher accuracy—and sometimes substantially so—compared with the GNSS-IMU track. Specifically, when evaluated under laboratory conditions with motion capture equipment as a reference, the mean error of the *perspective- $n$ -point* ground truth measurement method was assessed at 0.018 m in translation and 0.43° in rotation.

## 4. Structure, Contents, and Usage

### 4.1. The Dataset

The EDI-SLAM dataset is hosted on EDI's domain at [30] and has been made available for download under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License [31] as of the writing of this article. At initial public release, it consists of five sensor tracks collected at three different locations, totaling around 250 GB of raw sensor data. ROS1 and ROS2 versions of the bags with a subset of the topics—renamed to establish an interface agnostic to future changes in the collection methodology or sensor models—have been created. Ground truth reference poses are provided for three of the tracks, one at each location, and the original raw recordings are also available. Finally, a script for using the reference poses as well as example outputs discussed in Section 4.2 are included. The directory structure is laid out as follows:

```

- ground_truth
  └─ <location>_gt
     ├── ecef.txt           # marker corner locations in ECEF
     ├── poses_abs.csv      # camera poses computed with pnp
     ├── poses_ahru.csv     # xsens poses converted to ECEF
     └─ tag_observations.zip # images used for pnp with detected markers
- raw_bags
  └─ <location>_<ground truth status>.bag # ROS1 bags are files
- ROS1
  └─ <location>_<ground truth status>
     ├── calibration.yaml
     └─ <location>_<ground truth status>.bag # ROS1 bags are files
- ROS2
  └─ <location>_<ground truth status>
     ├── calibration.yaml
     ├── <location>_<ground truth status>.bag # ROS2 bags are directories
     ├── metadata.yaml
     └─ <location>_<ground truth status>.bag.db3
- example
  ├── compute_error_metrics.py # script for computing ATE, RPE, RMSE
  ├── T_xsens_cam0.csv         # calibration matrix used in the examples
  └─ metrics_<location>_<ground truth status>.txt

```

Depending on the intended application, potential users can therefore choose to bulk-download only the relevant type of recording—raw, reformatted ROS1 or ROS2. Only the *poses\_abs.csv* and *poses\_ahru.csv* files are necessary for evaluation; the marker corners and tag observations only being saved to aid in the identification and mitigation of potential errors and outliers. The reference poses are stored in rows of

$$(t, x^{ecef}, y^{ecef}, z^{ecef}, q_x^{xecef}, q_y^{xecef}, q_z^{xecef}, q_w^{xecef}) \quad (5)$$

where  $t$  is the timestamp.  $x, y, z$  are the translation coordinates and  $q_x, q_y, q_z, q_w$  are the rotation coordinates, all given in the ECEF frame. All of the ROS1 and ROS2 bags contain the same topics, with the same types, listed in Table 1. As stated in the notes column, *cam0* is always on the left and *cam1* is on the right. The frames of the point cloud and IMU topics are also given. The *gnss\_pose* contains the output of the *xsens* positioning filter before conversion in a surface-aligned orientation frame. Table 2 summarizes the sizes of the

respective tracks. Track length  $l$  and average velocity  $v$  estimates were produced for the ground-truth-annotated tracks, using the *Xsens* positions  $\mathbf{t}_{Xsens,i}^{ECEF}$  by taking

$$l = \sum_{i=1}^{n-1} \|\mathbf{t}_{Xsens,i}^{ECEF} - \mathbf{t}_{Xsens,i+1}^{ECEF}\| \quad (6)$$

where  $n$  is the number of the *Xsens* fused pose estimates and

$$v = \frac{l}{t(n) - t(1)} \quad (7)$$

where  $t(i)$  is the timestamp of the  $i$ -th observation.

**Table 1.** Topics and message types in the bagfiles.

Topic	Type	Notes
/points	sensor_msgs/msg/PointCloud2	<i>os_sensor</i>
/gnss	sensor_msgs/msg/NavSatFix	
/imu_lidar	sensor_msgs/msg/Imu	<i>os_imu</i>
/imu_xsens	sensor_msgs/msg/Imu	<i>xsens</i>
/gnss_pose	geometry_msgs/msg/PoseStamped	lat, lon, alt, ENU
/camera_left/image_raw	sensor_msgs/msg/Image	<i>cam0</i>
/camera_right/image_raw	sensor_msgs/msg/Image	<i>cam1</i>

**Table 2.** Sensor measurement counts, lengths, and velocities in the tracks where available.

Track	LiDAR	Images (per Camera)	$l$ , m	$v$ , m/s	size, GB
<i>courtyard_gt</i>	5599	17,152	747.36	1.32	42.9
<i>saga_gt</i>	7905	24,344	1099.22	1.36	60.9
<i>ropazi_gt</i>	11,673	35,375	4223.57	3.59	88.8
<i>courtyard_no_gt</i>	4366	13,458	-	-	33.6
<i>saga_no_gt</i>	3320	10,315	-	-	25.8

#### 4.2. Use in Evaluation

Given two sets of timestamped poses, there exist many statistics one may compute to compare them. In the SLAM context, two common benchmarks are Relative Pose Error (RPE) and Absolute Trajectory Error (ATE), formalized in [32]. Additionally, in cases such as SLAM-GNSS fused systems, a simple root mean square position error (RMSE) may be of interest. In the initial release version of the dataset, we include a script to compute RPE, ATE, RMSE, as well as local ENU-aligned XY projections of the ATE and RMSE. We also take advantage of the fact that two sets of reference poses are already present in the data to provide a worked example of using the dataset.

Given two corresponding sequences of poses in SE3 ( $T_t$ ), ( $Q_t$ ), where  $Q$  is the reference, and a delay interval  $\Delta$ , the root mean square RPE is computed as

$$\Theta_{i,i+\Delta} = (Q_i^{-1}Q_{i+\Delta})^{-1}(T_i^{-1}T_{i+\Delta})$$

$$\epsilon_{rpe} = \frac{1}{|K|} \sum_{i \in K} \|\mathbf{t}(\Theta_{i,i+\Delta})\|^2 \quad (8)$$

where  $K$  is the index set of the discrete sequences and  $\mathbf{t}(T)$  is the translation component of a pose. As is apparent from the expression, this statistic measures the amount of local deviation—over any given interval of a fixed length  $\Delta$ —between the two trajectories. Computing the ATE requires first estimating an aligning transform  $S$ , which is found by

minimizing the distance between corresponding translation points. The expression is then given as

$$\epsilon_{ate} = \frac{1}{|K|} \sum_{i \in K} \|\mathbf{t}(Q_i^{-1}ST_i)\|^2 \quad (9)$$

while the final metric, trajectory RMSE, is simply a special case found by not performing the alignment step—in effect, setting  $S$  to be the identity matrix

$$\epsilon_{rmse} = \frac{1}{|K|} \sum_{i \in K} \|\mathbf{t}(Q_i^{-1}T_i)\|^2 \quad (10)$$

Special care needs to be taken in cases when the two sequences  $T_t, Q_t$  are not given in the same reference frame. We adopt the convention whereby, given zero error, the calibration matrix  $C$  is given such that the following equation holds

$$Q_i = T_i C \quad (11)$$

which, in the case of aligning  $Xsens$  filtered GNSS-IMU pose estimates with the marker-derived poses of the left camera, is given by the transform  $T_{cam0}^{xsens}$ .

Finally, while RPE is invariant to external rotations (left-multiplied transforms cancel by construction), the directional components of ATE and RMSE may be examined separately in an external reference frame—potentially of interest when evaluating systems integrating GNSS, which may perform worse in estimating altitude than latitude or longitude, as shown below. To obtain error estimates in the XY-plane only,  $\epsilon_{ate,XY}$  and  $\epsilon_{rmse,XY}$ , we bring all poses  $Q_i, T_i$  into the ENU frame computed at ground truth centroid  $\mathbf{c}_Q$ , as in Equation (3)

$$\mathbf{c}_Q = \frac{1}{|K|} \sum_{i \in K} \mathbf{t}(Q_i)$$

$$Q_i^{ENU} = (T_{ENU}^{ECEF}(\mathbf{c}_Q))^{-1} Q_i^{ECEF}$$

$$T_i^{ENU} = (T_{ENU}^{ECEF}(\mathbf{c}_Q))^{-1} T_i^{ECEF} \quad (12)$$

and set the  $z$  components of all translation vectors to 0 before computing the ATE and RMSE as before.

The script `compute_error_metrics.py`, in the `example` directory of the dataset, computes all of the error statistics given above when provided with three inputs: an inferred track file, a reference track file, both formatted as per Equation (5), as well a calibration matrix  $C$  in its own `.csv` file. For each timestamp  $t$  in the inferred track, we find two timestamps  $\tau_1, \tau_2$  in the reference track, such that

$$\tau_1 \leq t < \tau_2$$

$$|\tau_i - t| < \eta \quad (13)$$

where  $\eta$  is the timing tolerance parameter, arbitrarily set to 0.2 s in the example—equivalent to two LiDAR scan periods. This ensures that only estimates inside intervals covered by the reference track are included in the error calculations. We interpolate the reference  $Q_t$  as

$$\alpha = \frac{t - \tau_1}{\tau_2 - \tau_1} \in [0, 1]$$

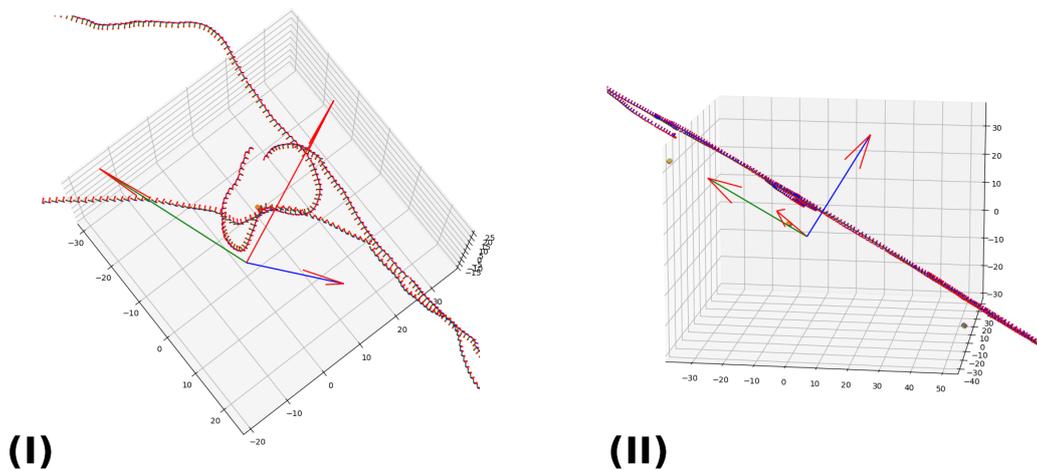
$$\mathbf{t}_t = (1 - \alpha)\mathbf{t}(Q_{\tau_1}) + \alpha\mathbf{t}(Q_{\tau_2})$$

$$\mathbf{q}_t = \text{SLERP}(\mathbf{q}(Q_{\tau_1}), \mathbf{q}(Q_{\tau_2}), \alpha)$$

$$Q_t = M(\mathbf{t}_t, \mathbf{q}_t) \quad (14)$$

where  $\alpha$  is the interpolation coefficient, SLERP is the spherical linear interpolation of quaternions [33];  $\mathbf{t}(Q)$ ,  $\mathbf{q}(Q)$  denote functions mapping a pose to its translation and quaternion rotation components, respectively; and  $M : \mathbb{R}^3 \times \mathbb{H} \rightarrow \text{SE3}$  builds a pose matrix from the interpolated vector and quaternion. To avoid distorting the RPE, we also need to discard any pairs of poses spanning an interruption in the reference track—in the example, this is performed through simply discarding any pairs of inferred poses for which the timestamp difference exceeds the timing tolerance  $\eta$  multiplied by RPE interval length  $\Delta$ .

As a worked example, we have collected the error statistics of the *Xsens* trajectory with regard to the marker poses for all three of the annotated tracks, which are shown in Table 3. The *courtyard* track clearly illustrates the point previously made in [7]: in environments with disruptions, such as multipath scattering, GNSS estimates do not provide a reliable reference, even with RTK correction (from within the same courtyard, no less). Comparing the RMSE results, it bears out that a significant contribution to the error comes in the form an incorrect altitude estimate, which is confirmed by visually examining the trajectory in Figure 5. The global transform  $S$  eliminates the contribution of steady state altitude error to the ATE metric when contrasted with RMSE, but outliers and drift still affect it. The RPE, on the other hand, seems to be largely a function of trajectory length, with longer paths that contain multiple losses and re-acquisitions of the satellite signal, creating more discontinuities.



**Figure 5.** Three-dimensional views of the *courtyard* reference track. (I) A large jump discontinuity; (II) the biased altitude estimate. The coordinate axes correspond to the ground truth centroid ENU frame  $T_{ENU}^{ECEF}(\mathbf{c}_Q)$ , while the clusters of points at the origin altitude are the surveyed marker positions.

**Table 3.** Error statistics of the *Xsens* positioning estimates with regard to the marker poses, given in  $m$ .

Track	$\epsilon_{rmse}$	$\epsilon_{rmse,XY}$	$\epsilon_{ate}$	$\epsilon_{ate,XY}$	$\epsilon_{rpe}$
<i>courtyard_gt</i>	77.291	6.873	4.000	3.193	0.008
<i>saga_gt</i>	1.305	0.452	0.114	0.099	0.024
<i>ropazi_gt</i>	8.223	2.165	3.059	2.405	0.047

## 5. Conclusions and Future Work

With this article, we introduce our public SLAM dataset, explain its collection methodology, provide usage examples, and evaluate the relative quality of the two reference pose

tracks available within. Due to ongoing research and development in the field of perception systems for autonomous robotics at EDI, we are likely to keep collecting more data which may potentially be appended to the repository. By providing an evaluation method independent of the onboard GNSS-IMU measurement unit, we enable the benchmarking of systems that fuse data from it into their positioning estimates, and also show that simply having such a sensor onboard is not sufficient for fully consistent localization. These data are currently being extensively used by EDI's researchers in developing an outdoor-focused multi-modal SLAM system and we hope that they prove similarly useful to others. A potential direction for future work that would greatly increase the value of these data is semantic annotation, in the form of terrain segmentation and object detection ground truth data.

**Author Contributions:** Conceptualization, P.R.; methodology, P.R.; software, P.R.; validation, P.R.; investigation, P.R. and G.K.; resources, P.R.; data curation, P.R. and G.K.; writing—original draft preparation, P.R.; writing—review and editing, P.R., J.A. and M.G.; visualization, P.R.; supervision, P.R.; project administration, M.G.; funding acquisition, J.A. and M.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially supported by: Latvian state research program No. VPP-EM-FOTONIKA-2022/1-0001 "Viedo materiālu, fotonikas, tehnoloģiju un inženierijas ekosistēma".

**Data Availability Statement:** Dataset publicly available at [https://edi.lv/EDI-SLAM\\_dataset](https://edi.lv/EDI-SLAM_dataset) (accessed on 31 October 2024).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

EDI	Institute of Electronics and Computer Science (Latvia)
SLAM	Simultaneous Localization and Mapping
ENU	East–North–Up surface-aligned coordinate system
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
ROS	Robot Operating System
LiDAR	Light Detection and Ranging
RGB	Red–Green–Blue images
ECEF	Earth-centered, Earth-fixed Euclidean frame
RTK	Real-Time Kinematic GNSS
ATE	Absolute Trajectory Error
RPE	Relative Pose Error
RMSE	Root Mean Squared Error

## References

1. Racinskis, P.; Ārents, J.; Greitans, M. Constructing Maps for Autonomous Robotics: An Introductory Conceptual Overview. *Electronics* **2023**, *12*, 2925. [CrossRef]
2. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278. [CrossRef]
3. Keller, M.; Lefloch, D.; Lambers, M.; Izadi, S.; Weyrich, T.; Kolb, A. Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion. In Proceedings of the 2013 International Conference on 3D Vision, Seattle, WA, USA, 29 June–1 July 2013; pp. 1–8.

4. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
5. Campos, C.; Elvira, R.; Rodr'iguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2020**, *37*, 1874–1890. [[CrossRef](#)]
6. Cho, Y. Awesome SLAM Datasets. Available online: <https://github.com/youngguncho/awesome-slam-datasets> (accessed on 27 October 2024).
7. Racinkis, P.; Arents, J.; Greitans, M. Annotating SLAM data sets with Apriltag markers. In Proceedings of the 2024 10th International Conference on Automation, Robotics and Applications (ICARA), Athens, Greece, 22–24 February 2024; pp. 438–442. [[CrossRef](#)]
8. Wang, J.; Olson, E. AprilTag 2: Efficient and robust fiducial detection. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016.
9. ROS.org. ROS/Introduction. Available online: <http://wiki.ros.org/ROS/Introduction> (accessed on 31 October 2024).
10. Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; Woodall, W. Robot Operating System 2: Design, architecture, and uses in the wild. *Sci. Robot.* **2022**, *7*, eabm6074. [[CrossRef](#)] [[PubMed](#)]
11. EPSG. WGS84—World Geodetic System 1984, Used in GPS. Available online: <https://epsg.io/4326> (accessed on 27 October 2024).
12. EPSG. WGS84—Cartesian. Available online: <https://epsg.io/4978> (accessed on 27 October 2024).
13. Xsens Technologies B.V. MTi Filter Profiles. Available online: <https://base.movella.com/s/article/MTi-Filter-Profiles-1605869708823> (accessed on 27 October 2024).
14. Xsens Technologies B.V. MTi Family Reference Manual. Available online: [https://www.xsens.com/hubfs/Downloads/Manuals/MTi\\_familyreference\\_manual.pdf](https://www.xsens.com/hubfs/Downloads/Manuals/MTi_familyreference_manual.pdf) (accessed on 27 October 2024).
15. Nguyen, T.M.; Yuan, S.; Cao, M.; Lyu, Y.; Nguyen, T.H.; Xie, L. NTU VIRAL: A Visual-Inertial-Ranging-Lidar Dataset, From an Aerial Vehicle Viewpoint. *Int. J. Robot. Res.* **2022**, *41*, 270–280. [[CrossRef](#)]
16. Zhu, A.Z.; Thakur, D.; Özaslan, T.; Pfommer, B.; Kumar, V.; Daniilidis, K. The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2032–2039. [[CrossRef](#)]
17. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *Int. J. Robot. Res. (IJRR)* **2013**, *32*, 1231–1237. [[CrossRef](#)]
18. Liao, Y.; Xie, J.; Geiger, A. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. *Pattern Anal. Mach. Intell. (PAMI)* **2022**, *45*, 3292–3310. [[CrossRef](#)] [[PubMed](#)]
19. Schubert, D.; Goll, T.; Demmel, N.; Usenko, V.C.; Stücker, J.; Cremers, D. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1680–1687.
20. Blanco, J.L.; Moreno, F.A.; González, J. A Collection of Outdoor Robotic Datasets with centimeter-accuracy Ground Truth. *Auton. Robot.* **2009**, *27*, 327–351. [[CrossRef](#)]
21. Ouster, Inc. Official ROS1/ROS2 Drivers for Ouster Sensors. Available online: <https://github.com/ouster-lidar/ouster-ros/tree/master> (accessed on 28 October 2024).
22. jiminghe. Xsens MTi ROS Driver and Ntrip Client. Available online: [https://github.com/jiminghe/Xsens\\_MTi\\_ROS\\_Driver\\_and\\_Ntrip\\_Client](https://github.com/jiminghe/Xsens_MTi_ROS_Driver_and_Ntrip_Client) (accessed on 28 October 2024).
23. Basler AG. Pylon SDKs. Available online: <https://www.baslerweb.com/en/software/pylon/sdk/> (accessed on 28 October 2024).
24. ROS.org. A ROS-Driver for Basler Cameras. Available online: [AROS-DriverforBaslerCameras](https://wiki.ros.org/AROS-DriverforBaslerCameras) (accessed on 28 October 2024).
25. Furgale, P.T.; Rehder, J.; Siegwart, R.Y. Unified temporal and spatial calibration for multi-sensor systems. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 1280–1286.
26. EPSG. LKS-92/Latvia TM. Available online: <https://epsg.io/3059> (accessed on 29 October 2024).
27. Latvijas Ģeotelpiskās Informācijas Aģentūra. Latvian Quasi-Geoid Model. Available online: <https://www.lgia.gov.lv/en/latvian-quasi-geoid-model> (accessed on 29 October 2024).
28. Terzakis, G.; Lourakis, M.I.A. A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
29. Itseez. Open Source Computer Vision Library. 2015. Available online: <https://github.com/itseez/opencv> (accessed on 31 October 2024).
30. EDI. EDI-SLAM Data. 2024. Available online: [http://edi.lv/EDI-SLAM\\_dataset](http://edi.lv/EDI-SLAM_dataset) (accessed on 31 October 2024).
31. Creative Commons. Attribution-NonCommercial-ShareAlike 4.0 International. Available online: <https://creativecommons.org/licenses/by-nc-sa/4.0/> (accessed on 1 November 2024).

32. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Abu Dhabi, United Arab Emirates, 23–27 October 2012; pp. 573–580.
33. Shoemake, K. Animating rotation with quaternion curves. In Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, San Francisco, CA, USA, 22–26 July 1985.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.