Acta Astronautica 238 (2026) 414–423

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

EPOS-Lid: Lidar benchmark dataset for pose estimation during non-cooperative rendezvous

Léo Renaut, Ksenia Klionovska *, Maximilian Albracht, Heike Frei

German Aerospace Center (DLR), Spaceflight technology, Münchener Str. 20, Weßling, 82234, Germany

ARTICLE INFO

Keywords:
Lidar benchmark dataset
Pose estimation
Space rendezvous

ABSTRACT

Lidar sensors are increasingly studied for space rendezvous applications, in particular for on-orbit servicing or active debris removal missions. With their active measurement principle, they provide accurate 3D point clouds which enable precise pose estimation. While simulated lidar data is relatively simple to generate under ideal conditions, real lidar point clouds can present high levels of noise and reflections. There is the need for representative lidar data to train and test pose estimation methods for non-cooperative space rendezvous scenarios. This work introduces *EPOS-Lid*, an openly available lidar benchmark dataset for this task. It comprises a synthetic dataset for training pose estimation methods, and real lidar point clouds collected at the European Proximity Operations Simulator (EPOS). Further, it is demonstrated with evaluation of benchmark methods how the datasets can be used for training and testing pose initialization and pose tracking methods.

1. Introduction

The capability to rendezvous with a non-cooperative satellite or a potentially inactive space debris, is at the center of On-Orbit Servicing (OOS) and Active Debris Removal (ADR) missions. In both cases, precise relative navigation information is necessary to enable a safe and precise approach towards the target object. The active satellite performing the approach is referred to as the chaser. In the last phase of the rendezvous, e.g. at relative distances below 30 m, the chaser needs to be able to estimate not only its relative position to the target, but also its relative attitude.

To estimate its relative position and attitude (pose) to the target autonomously, the chaser is equipped with electro-optical sensors. A first type of sensors are passive sensors, such as optical cameras, sometimes in a stereo-configuration, or infrared cameras. While such sensors are relatively cheap and have low power requirements, they have to handle the strongly varying light conditions in orbit [1]. In particular, optical cameras are dependent on the Sunlight illumination to capture images. They are blind on the night side, and strong reflections can lower the image quality on the day side [1].

A second type of sensors are active sensors such as Time-of-Flight (ToF) cameras or lidar (light detection and ranging). While ToF cameras based on modulated light principles are limited in range [2,3], systems that use a direct range measurement approach, commonly referred to as flash lidars, are capable of achieving kilometer-scale ranges and are frequently employed in space applications [4,5]. Likewise, scanning lidars achieve similar ranges, and have been used for space rendezvous missions such as the first commercial OOS with the Mission Extension Vehicle (MEV) [6]. Flash or scanning lidars are relatively independent of the external light conditions, and enable to capture precise 3D point clouds, making them valuable sensors for relative navigation in close range [7]. Not all point cloud data is necessarily obtained from a lidar, but lidar point cloud data presents several advantages due to the underlying laser scanning technology, in particular: high precision, robustness to external conditions, and long range. For this reason, this work focuses on lidar generated point cloud data.

Lidar pose estimation is usually split in two steps, pose initialization and pose tracking [7,8]. Pose initialization consists in finding a relative pose of the target when no a priori estimate is available, apart from a possible coarse estimate of the relative position. Classical methods for lidar pose initialization rely on polygon matching algorithms [9, 10], feature-based matching [8,11,12], or template matching [13,14]. Instead of hand-crafted descriptors, neural networks have been used in hybrid algorithms, in combination with polygon matching [15] or feature-matching methods [16,17]. Finally, direct pose estimation with neural networks consists in training either a Convolutional Neural Network (CNN) to estimate a pose from a depth image [18–20], or a point-based network using directly the lidar point cloud [21].

After initialization, pose tracking aims at estimating the pose over consecutive point clouds. For tracking, precise alignment of the source point cloud with a 3D model of the target is achieved with methods

* Corresponding author.
E-mail address: KSENIA.KLIONOVSKA@DLR.DE (K. Klionovska).

https://doi.org/10.1016/j.actaastro.2025.09.030
Received 28 April 2025; Received in revised form 30 July 2025; Accepted 8 September 2025
Available online 15 September 2025

such as Iterative Closest Point (ICP) [22]. Such methods enable precise alignment, but require an initial estimate to converge. For the first point cloud, the initial estimate is computed by a pose initialization method, and for subsequent point clouds, the tracking solution of the previous point cloud is used. For pose tracking, ICP or one of its variants is widely used [2,3,9,13]. Alternatively, to reduce the computational time, an approach based on a modified NDT algorithm has been proposed [23]. All these methods assume that a 3D model of the target satellite is available to be matched with the received point clouds. Such a 3D model is either known before the mission, or is the result of an inspection flight around the target satellite [24]. Other approaches consist in performing model-less pose estimation, i.e., in progressively constructing a point cloud model of the target satellite by aggregating consecutive lidar scans [25,26].

A key factor for the development, evaluation and comparison of pose estimation methods is the quality and availability of the data on which the algorithms are evaluated. To this aim, for camera-based pose estimation, several image datasets have been published [27–30]. With the advances in image rendering technology, professional rendering tools can be tailored for use in spacecraft rendezvous operations. Proenca and Gao [27] publish URSO, a fully synthetic image dataset of around 5000 images of the Dragon and Soyuz spacecraft. Likewise, the SwissCube dataset [28] contains 50 K images of the ClearSpace One satellite rendered with high-fidelity. Recently, the DLVS3 dataset [31] has been published, containing an impressive number of 640 K rendered images of the Hubble Space Telescope for pose estimation. Each image also contains depth information, which can be used as ground truth but does not model the specifics of depth sensors such as a lidar.

A major challenge when training deep learning methods for pose estimation is achieving Simulation to Reality (Sim2Real) transfer, i.e., keeping the same pose estimation performance when evaluating a model trained on synthetic data on real-world data [32]. To this aim, hybrid datasets, containing both simulated images and real images, have been published [29,30,33]. In particular, the SPEED dataset [30] contains both 15 K simulated images of the Tango spacecraft, based on the illumination conditions observed in orbit, and around 300 images of the same satellite gathered at a robotic rendezvous facility. Likewise, SPEED+ [33] is a great extension of the previous dataset, comprising around 60 K synthetic images, and nearly 10 K images from the robotic facility, where particular focus is laid on achieving realistic lightning conditions. Similarly to his predecessor dataset, SPEED+ is of great use for comparison and benchmarking of camera-based pose estimation algorithms [32,34,35]. Likewise, the ENVISAT image datasets contain both 16 K synthetic images, and 16 K real images gathered at a hardware-in-the-loop facility [36].

The field of lidar based-pose estimation for spacecraft rendezvous is also moving towards neural network based methods [17,19–21]. However, to the best of our knowledge, no public datasets for lidar pose estimation are currently available. To address this need, this work presents *EPOS-Lid* – a lidar benchmark dataset for relative spacecraft navigation consisting of both real point clouds collected at the European Proximity Operations Simulator (EPOS) 2.0, and simulated point clouds.[1] The synthetic lidar datasets can be used for training neural network based pose estimation methods, while the real dataset serves as a test benchmark. In addition, the datasets acquired during the rendezvous trajectories at EPOS can be used to test and validate full relative navigation pipelines, including tracking over consecutive point clouds. Furthermore, an exemplary pose estimation method is presented and tested on this dataset, demonstrating that *EPOS-Lid* can be used to develop accurate and robust lidar navigation methods.

The remainder of this paper is structured as follows: Section 2 presents the EPOS facility and the lidar simulator used to generate the real and synthetic datasets, and provides an overview of the pose estimation methods used in the evaluation. The content of the lidar datasets is described in Section 3. In Section 4, we present the results of the pose estimation method trained and evaluated on the presented benchmark. Finally, Section 5 concludes the paper.

[1] The datasets are openly available under https://www.dlr.de/en/rb/research-operation/research-projects/flight-dynamics-navigation-and-orbital-sustainability/on-orbit-servicing/epos-lid-dataset.

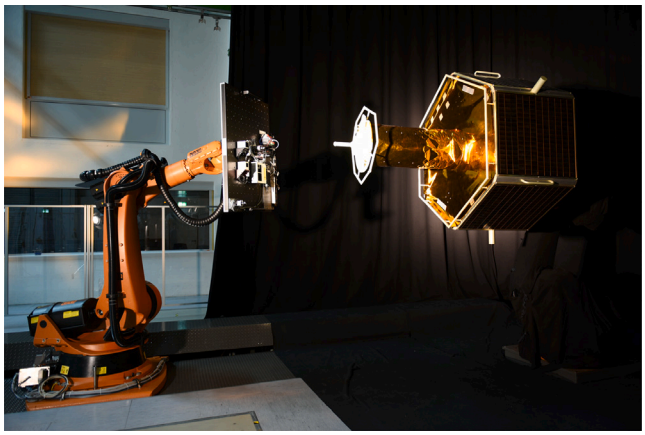


Fig. 1. EPOS facility with two industrial robots. The left one carries a sensor adapter plate and the right one a mockup of a target satellite.

2. Methods

2.1. European Proximity Operations Simulator

2.1.1. Facility

The European Proximity Operations Simulator (EPOS) is a hardware-in-the-loop testbed for simulation of rendezvous missions or other close proximity scenarios [37,38]. The facility, located at German Space Operations Center (GSOC), serves for validation and test of optical sensors and of guidance, navigation and control systems.

EPOS consists of two industrial robots with each six degrees of freedom: a KUKA KR100HA (robot 1) mounted on a linear rail-system of 25 m length and a KUKA KR240-2 (robot 2) mounted on the ground at one end of the rail. Fig. 1 presents a test scenario used for creation of the lidar benchmark dataset *EPOS-Lid*: Robot 1 carries a sensor adapter plate, where different optical sensors are mounted including a Livox Mid-40 lidar (see Section 2.1.2). Robot 2 carries a mockup of an exemplary target satellite with real surface materials such as golden MLI and solar panels. The sunlight is simulated by a spotlight; two ARRI floodlights with 5 kW and 12 kW are available at the facility. Robot 2 is surrounded by a black Molton curtain and carpet and the robot's arm is wrapped with a Molton fabric.

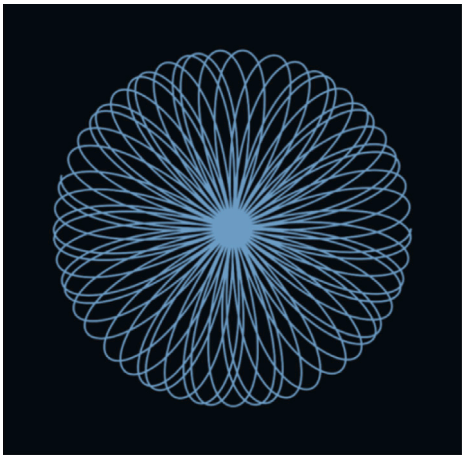
A PC-based monitoring and control system generates commands such that the two EPOS robots simulate the desired relative motion of the two spacecrafts. The relative trajectory depends on the specific test case or application. Different datasets (see Section 3.1) can be generated. The EPOS facility can be used for both open and closed loop simulations. The performance of, for example, a pose estimation method can be assessed by comparing the pose estimation results with the robots' logged trajectory (ground truth).

2.1.2. Lidar sensor

A Livox® Mid-40 sensor is used as close range rendezvous sensor for the lidar benchmark dataset *EPOS-Lid*. This sensor is widely employed in the automotive domain but is also well suited for a rendezvous test laboratory like EPOS since real space qualified lidar sensors are very expensive and have long manufacturing and delivery times.

Fig. 2(a) shows the lidar at the EPOS facility, mounted on the sensor adapter plate of robot 1 which simulates the chaser. The Livox Mid-40

(a)



(b)

Fig. 2. Livox® Mid-40 used as rendezvous lidar for *EPOS-Lid*; (a) lidar mounted on one robot of the EPOS facility, (b) scanning pattern of the lidar.

Table 1
Specification parameters of the Livox® Mid-40 lidar [39].

Parameter	Value
Dimensions	88 mm × 69 mm × 76 mm
Mass	approx. 760 g
Power	10 W
Laser wavelength	905 nm
Laser class	Class 1 (IEC60825-1)
Detection range	260 m
Field-of-view	38.4 deg
Range precision (1σ @ 20 m)	2 cm
Angular precision	0.1 deg
Beam divergence	0.28 deg × 0.03 deg
Point rate	100,000 points/s

has a non-repetitive scan pattern, visualized in Fig. 2(b). Table 1 shows some technical parameters of the lidar.

It is assumed that the initial calibration from the lidar manufacturer is already precise, so that the intrinsic calibration parameters of the lidar are not modified. However, when installed at the robotic facility, the extrinsic parameters must be calibrated, i.e., it is necessary to perform the hand-eye calibration of the sensor. This is achieved by collecting point clouds of the target at different distances and for different orientations. Since the EPOS facility is calibrated, the relative pose of the robots to each other is known precisely, and it is only to find the pose of the lidar with respect to the robot carrying it, robot 1.

The calibration is achieved by applying a local optimization procedure relying on smoothed NDT registration [23], similar to ICP. Starting from an initial guess of the hand-eye calibration parameters, these parameters are refined by simultaneously minimizing the distance of all captured calibration point clouds to the expected target point cloud. The only difference with a local point cloud matching method for pose estimation is which pose is being optimized: Instead of searching the optimal target pose, the pose of the satellite mock-up with respect to the robot equipped with the sensor plate is known from the ground truth data of the facility. The pose being optimized is the pose of the lidar sensor with respect to the sensor plate.

The Mid-40 sensor is from the automotive domain, not qualified for space, and has a very specific scanning pattern. However, the underlying laser scanning technology is identical to space-grade sensors. The lidar will be configured to capture approximately 10 000 points on the target satellite per scan, see Section 3.1. Such a point density is high but comparable, for example, with the density of the flash lidar used for the OSIRIS-REx mission [5], which has a detector array of 128 × 128 pixels. Likewise, a constant number of points per scan is a setting that is representative of a space scenario, since for example Jena-Optronik RVS 3000™ scanning lidar [15] can dynamically adapt its field-of-view to fit the target. Most importantly, the laser rays interact with the real materials on the target mock-up, as presented in the previous Section 2.1.1. This enables to observe artifacts in the point cloud data that are also expected to be observed in space, such as reflections on the target's MLI and solar panels. These will be described in Section 3.1.

2.2. High-fidelity lidar simulator

To generate training data in sufficient amount, a high-fidelity lidar simulator developed in previous work [20,21] is used. The suggested approach consists in training a neural network for pose estimation on synthetic data only, and testing it on real lidar data collected at EPOS. Since the synthetic point clouds differ from the real ones, a loss in precision in pose estimation might be observed when evaluating a neural network on real data compared to an evaluation on synthetic data. This difference is known as the domain gap [33,40]. Trying to bridge the domain gap and achieving a successful Simulation To Reality (Sim2Real) transfer is one of the challenges in lidar pose estimation.

There are several reasons for training only on synthetic data, rather than for example on a mix of synthetic and real data gathered at EPOS. A lidar simulator allows to quickly generate a large amount of point clouds in every possible pose configuration. While data collection is possible at a hardware-in-the-loop facility, achieving the same quantity and diversity of poses is more challenging in that setting. Also, in an operational scenario, the geometry of the target satellite might be unknown, and a 3D model only available after an inspection phase. In such a case, a lidar simulator enables to generate a dataset and train a new pose estimation method within very short time, what would not be possible at a physical facility. Another factor is that the approach of training only on synthetic data, and testing on real data, enables to validate that the method is able to generalize to different data. If the data on which the method is trained was acquired under the same conditions as the test data, then no guarantee would be provided that the pose estimation is able to adapt to different conditions, such as the conditions that might be observed in space.

A detailed description of the lidar simulator is provided in [20]. Its main characteristics are as follows: The lidar simulator is implemented as a ray-tracer. The simulator reproduces the characteristics of the Livox Mid-40 sensor. The rosette scan pattern of the Mid-40 (see Fig. 2(b)) is achieved by a deflection of the ray through two inclined prisms rotating at different angular rates. This behavior is reproduced in the simulator following the modeling by [41]. In addition, the reflectivity of the materials is accounted for. In particular, the solar panels and

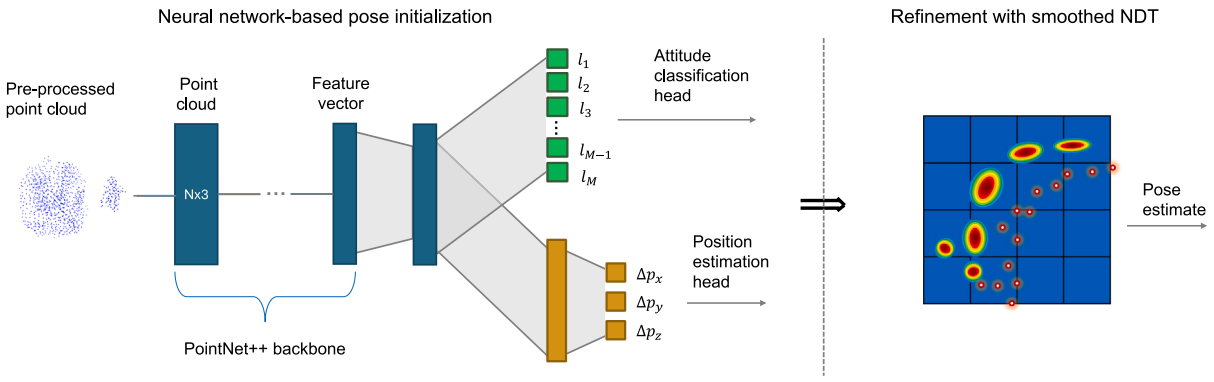


Fig. 3. Overview of the pose initialization and tracking methods evaluated in this work.
Source: Adapted from [21,23].

golden Multi-Layer Insulation (MLI) sheets on the satellite mock-up installed at EPOS (Fig. 1) are highly reflective. Since the exact reflectivity properties of these materials are unknown, they can be randomized in the simulator. To simulate reflection effects and ghost points, double reflections are also modeled, i.e., the reflection of a ray on a first surface, which then hits a second surface before coming back. Higher order reflections (triple or more) are not simulated, for efficiency. Sensor noise is also modeled, such as range errors and beam divergence errors [20]. The noise is added according to the sensor specifications in Table 1. Finally, the simulator accounts for motion blur, i.e., the relative displacement of the target during the duration of a scan, which leads to a distorted point cloud. Motion blur is modeled assuming that the relative motion during the scan time is linear, i.e., that the velocity and angular rates are constant.

2.3. Benchmark lidar pose estimation method

2.3.1. Pose estimation

To demonstrate that the lidar datasets can be used for training and evaluation of lidar-based pose estimation, we provide an overview of the pose initialization and tracking method used as benchmark in this work. A schematic representation of the logic of the pose initialization and refinement pipeline is presented in Fig. 3.

As detailed in Section 1, lidar pose estimation is frequently split in pose initialization, and pose tracking. For pose initialization, no prior estimate is available. In this work, we perform pose initialization with a point-based neural network. This method is introduced and described in detail in [21]. The same overall pose estimation pipeline is used in this work. For completeness, we recall the main steps of this pose initialization method.

A point-based neural network directly processes 3D data of the point cloud. The first step is to center and down-sample the point cloud, since the network expects a fix input size of the lidar scan. Centering is performed by computing the trimmed centroid of the point cloud [20,21]. Compared to a regular centroid, a fraction $p = 0.25$ of the highest and lowest outliers in the points cloud are removed, before computing the centroid of remaining points. This logic enables to filter out outliers which could negatively impact the position of the centroid. The input size of the network is set to 1024 point. To down-sample the point cloud up to only 1024 points, a kd-tree version of the Farthest Point Sampling (FPS) method is implemented [42]. Compared to a naive FPS implementation, this enables faster processing on a Central Processing Unit (CPU). In case the original point cloud contains less than 1024 points, points are randomly duplicated in the scan to reach the desired input size. However, since the point clouds captured by the Livox Mid-40 are dense, this is never the case for the EPOS test data.

After pre-processing, the point cloud is processed by a 3D neural network, in this case a PointNet++ [43] backbone. Compared to the original backbone, the model is optimized for runtime by reducing the number and size of point clusters used at each clustering step. The network architecture is detailed in [21]. Afterwards, the feature vector produced by the neural network is processed by two prediction heads, one for attitude classification, and the other for position estimation. The attitude classification accounts for the symmetries of the target, i.e., it only estimates the attitude corresponding to one of the symmetric equivalents of the target attitude (see next section, Fig. 4). Finally, since the point cloud was initially centered, the position estimation delivered by the neural network is added to the initially estimated position of the centroid to retrieve the unscaled position estimation. The details of the training procedure on the *EPOS-Lid* synthetic datasets are presented in Section 4.1. An important aspect in the training is the addition of point cloud specific data augmentation layers, which increase the robustness of the network to unseen data. As presented in the original work, these layers comprise jitter, deformation, removal of points, and addition of outliers in the data.

The initial pose estimate is usually coarse, which is why a pose refinement method is used afterwards [7,8]. In previous work, we have developed a pose tracking method based on a modified version of the NDT algorithm [23]. This method shows better efficiency and robustness than ICP for satellite pose tracking [23]. The smoothed NDT tracking algorithm is used in two ways: First, it enables to refine the initial pose estimate. Second, when considering consecutive point clouds, the pose initialization only needs to be performed on the first point cloud. For all subsequent scans, the pose estimate coming from the previous scan can be used as an initial estimate to start the tracker [23].

Pose tracking is performed by matching a 3D model of the target satellite with recorded point clouds. The reference point cloud sampled from the geometrical model of the target is also provided along with the datasets, under the name `target_model.3d`. When using the smoothed NDT tracker, this reference point cloud is used to compute a NDT representation of the target. This NDT representation takes the form of a probabilistic representation, as illustrated on the right in Fig. 3: For each voxel of the 3D space, the distribution of points in this voxel (represented by a heatmap on the figure) is computed. Next, starting from an initial guess of its relative pose, the source point cloud (represented by the white points on the figure) is iteratively aligned with the target point cloud to maximize its likelihood under the assumption that it follows the probability distribution of the target point cloud. The tuning parameters of the NDT algorithm used in this work (cell size, number of iterations, termination threshold) are provided in Section 4.1.

2.3.2. Pose error definition for a symmetrical spacecraft

The target spacecraft mounted at EPOS, and considered as a use-case for the datasets, was presented in Fig. 1. The coordinate frame of the target is defined in Fig. 4(a). The coordinate frame is centered at the

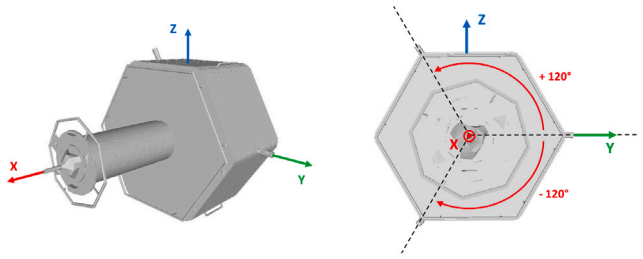


Fig. 4. (a) Coordinate frame of the target satellite. (b) Symmetries of the target satellite around the roll axis X, when viewed from the front.
Image source: [21].

center-of-mass of the satellite. The X-axis is defined by the direction of the front tower of the satellite. The main body of the satellite has a hexagonal shape. Additionally, three antennas and handlebars are positioned on the edges of the hexagon. As shown in Fig. 4(a), the Y-axis of the satellite passes through one of the small antennas mounted on an edge of the hexagon. Finally, the Z-axis is defined such that the coordinate frame is orthonormal and follows a right-handed convention.

The target spacecraft presents some symmetries, which makes an unequivocal estimation of its attitude impossible. First, the main body has a hexagonal shape. However, the antennas enable to discern more details: As shown in Fig. 4(b), the three antennas enable to discriminate between some orientations of the hexagon. Still, the spacecraft geometry model is invariant by a rotation by ±120 deg (or $\pm 2\pi/3$ rad) around its roll axis X. Therefore, given a certain attitude of the target $\overline{R}$, defined as the rotation matrix which transforms a quantity expressed in the target coordinate frame into a quantity expressed in the lidar coordinate frame, it is considered that two additional attitudes are symmetrically equivalent. These attitudes are $\overline{R}R_X(2\pi/3)$ and $\overline{R}R_X(-2\pi/3)$, where the rotation matrix around the axis X with an angle α is defined as:

$$R_X(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}. \tag{1}$$

The pre-multiplication of the reference rotation $\overline{R}$ by a rotation of $\pm 2\pi/3$ rad around the roll axis corresponds to the symmetrical equivalents for this target geometry, as illustrated in Fig. 4(b).

The objective of the lidar-based navigation is to minimize the position and attitude error between the ground truth and the estimates of the method. Let $(\overline{R}, \overline{\rho})$ be the real attitude matrix and position of the target. As previously, the attitude is the target attitude relative to the lidar coordinate frame, and the position the relative position of the target to the lidar. Consider an estimate of these two quantities (R, ρ), originating from a pose estimation method. The position error between the real and estimated position is simply:

$$\epsilon_{pos} = \|\overline{\rho} - \rho\|. \tag{2}$$

For the attitude, the usual distance function measuring the angular error between two attitudes is:

$$d(\overline{R}, R) = \arccos\left(\frac{\mathrm{tr}(\overline{R}^T R) - 1}{2}\right). \tag{3}$$

However, this attitude distance measure is not adapted to the case of a symmetrical spacecraft. Indeed, in this case, the attitudes $\overline{R}$ and $\overline{R}R_X(2\pi/3)$ would be distant from $2\pi/3$, while they are considered symmetrically equivalent for the spacecraft illustrated in Fig. 4. Therefore, the attitude error for the symmetrical spacecraft is defined in this work

Table 2
Number of point clouds per dataset.

Dataset	I	II	III	IV	V	VI
#PCs	2483	1253	1302	2428	1868	501

as the minimum distance between the estimated attitude, and one of the three symmetrical equivalents of the ground truth attitude:

$$\epsilon_{att} = \min\left(d(\overline{R}, R), d\left(\overline{R}R_X(\tfrac{2\pi}{3}), R\right), d\left(\overline{R}R_X(\tfrac{-2\pi}{3}), R\right)\right). \tag{4}$$

Together, ϵ_{pos} and ϵ_{att} enable to measure the precision of a pose estimate for this symmetrical spacecraft.

3. Datasets

3.1. EPOS datasets

The datasets are publicly available[1]. Each point cloud is stored in a file named `{number}.3d`, where `{number}` represents a changing numerical identifier. For instance, 0000.3d, 0500.3d, and 1000.3d correspond to different point clouds captured at different instances. The point cloud data is represented in the lidar coordinate system, where the first column corresponds to the X-coordinate, the second column corresponds to the Y-coordinate, and the third column corresponds to the Z-coordinate. For a practical example of how to use and visualize the dataset, including a conversion script and viewing instructions, see Appendix.

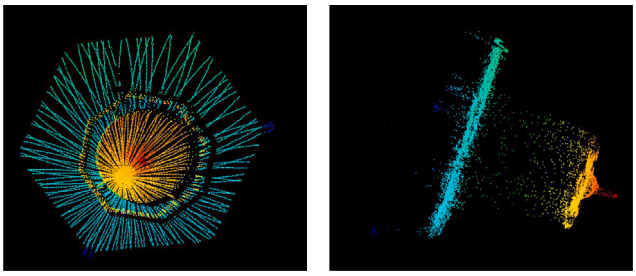
Corresponding to each point cloud file, there exists a ground truth file with the same identifier but in the format `{number}.pose`, e.g. 0000.pose, 0500.pose, and 1000.pose. The ground truth file contains the estimated pose of the target at the end of scan time in the lidar coordinate frame. The structure of this file is as follows:

- The **first line** represents the timestamp in seconds, corresponding to the Guidance, Navigation and Control (GNC) system time. For tracking purposes, the difference between consecutive timestamps provides information about the temporal spacing of point clouds, which can be used to estimate motion, such as changes in position and orientation over time.
- The **second line** contains the position of the target, expressed in the lidar coordinate frame, represented as a three-dimensional vector.
- The **third line** specifies the orientation of the target using a quaternion in the format (q_s, q_x, q_y, q_z), where q_s is the scalar component, and (q_x, q_y, q_z) are the vector components. This rotation is the rotation from the target, to the lidar coordinate frame.

There are six datasets containing real lidar point clouds collected at EPOS. The number of point clouds in each dataset is given in Table 2. The number of points in each captured point cloud remains nearly constant (~10,000 points) across all datasets. This consistency is achieved through an adaptive framerate strategy, which dynamically adjusts the lidar framerate to maintain a stable point count per scan. The front and side view of the point cloud 1000.3d from Dataset II is in Fig. 5.

The target mockup at EPOS has surface properties similar to those of a real satellite, and it was illuminated by a 5 kW sunspot, replicating realistic lighting conditions encountered in space environments. As a result, some point clouds in all three datasets may contain erroneous points due to specular and multiple reflections from the mockup's surfaces. These inaccurate points are represented by blue dots in Fig. 6. Retaining these points in the *EPOS-Lid* benchmark is essential for testing the robustness and reliability of algorithms under realistic conditions.

The chaser maintains a controlled viewing elevation and azimuth angle while the target tumbles, allowing different surfaces to be observed passively over time. Additionally, the relative distance decreases



(a) (b)

Fig. 5. Point cloud 1000.3d from Dataset II taken by the Livox® Mid-40 lidar at EPOS. (a) Front view (b) Side view.

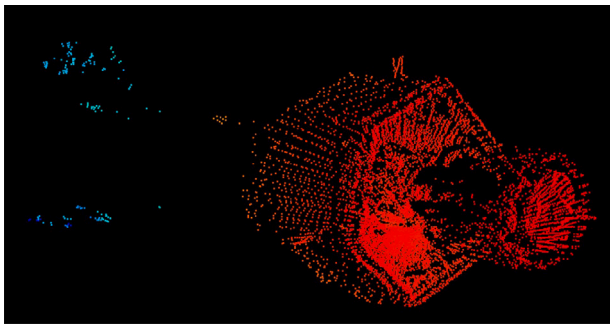


Fig. 6. Point cloud 0639.3d from Dataset III with erroneous points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

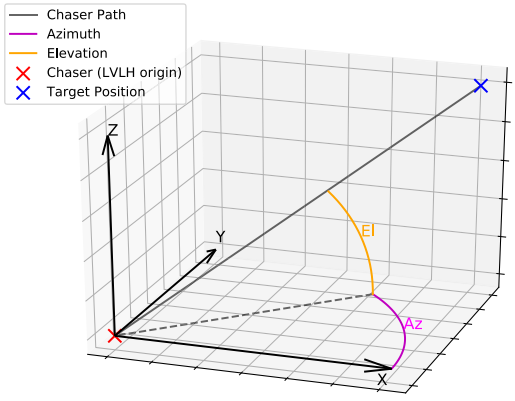


Fig. 7. Setup sketch for data collection in chaser's LVLH coordinate frame.

as the chaser gradually approaches the target during the tumbling motion. The setup for data collection is illustrated in the LVLH coordinate frame of the chaser in Fig. 7.

The target's pose is provided in the lidar coordinate frame, which is related to the chaser's LVLH frame through a transformation obtained via calibration. While the LVLH frame describes the relative motion, the lidar frame serves as the sensor's local reference for measurements. Table 3 presents the azimuth, elevation, and relative distance range for each dataset accordingly.

Dataset I and IV capture an azimuthal motion, transitioning from 90° to 55° and back to 90°, with a fixed elevation of 0°, and a range reducing from 17 m to 3 m. In Dataset II a one-way azimuthal shift from 90° to 70° is performed, maintaining the same elevation and a range of 20 m to 4 m. Dataset III focuses on a constant azimuth of 70° and an elevated viewing angle of 30°, with the range similarly reducing from 20 m to 4 m. In Dataset V and VI the distance reduces with a

Table 3
Azimuth, elevation and relative distance.

Dataset	Azimuth [deg]	Elevation [deg]	Range [m]
I	90→55→90	0	17 to 3
II	90→70	0	20 to 4
III	70	30	20 to 4
IV	90→55→90	0	17 to 3
V	55	0	8 to 3
VI	55	0	15 to 9

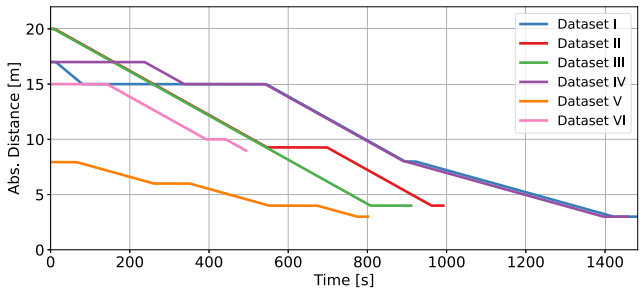


Fig. 8. Absolute distance over time between chaser and client during dataset record.

constant azimuthal shift of 55° from 8 m to 3 m and from 15 m to 9 m respectively. The progression of the absolute distance over time can be seen in Fig. 8. The relative distance between the chaser and target decreases using a straight-line approach. During an azimuth and elevation shift, the absolute distance between the target and the chaser remains constant. Although Dataset III and Dataset V have a shorter recording duration than Dataset II, they contain more point clouds due to the higher number of close-up shots. This is due to the adaptive framerate strategy, which affects the number of recorded point clouds.

In all datasets, the target has the same mass of 500 kg. The inertia tensor is $\mathbf{I} = \text{diag}(600, 500, 500)$ kg m^2, defined in the target coordinate system, as shown in Fig. 4(a). Differences in the initial angular velocity result in distinct tumbling behaviors. The Dataset I with initial angular velocity $\omega_1 = [2.0, 0.2, 0]°/s$ has a more constrained motion, primarily around the X axis. The Dataset II and Dataset III with $\omega_2 = [1.5, 1.0, 0]°/s$ have a more significant tumbling motion and therefore contain a wider variety of target poses. In the remaining datasets, the target performs a high-rate rotation about the X axis only, with angular velocity $\omega_3 = [5, 0, 0]°/s$ for Dataset IV and $\omega_4 = [8, 0, 0]°/s$ for Dataset V and VI, presenting a more challenging scenario for pose estimation. For all scenarios, it is considered that the target is uncontrolled and freely tumbling, i.e., not subject to any angular acceleration. Its motion is thus entirely defined by its initial angular velocity. Other cases, for example fuel leaks leading to perturbing angular accelerations, are not considered in the datasets.

3.2. Synthetic training dataset

The lidar simulator presented in Section 2.2 is used to generate a synthetic dataset for training of pose estimation methods. The synthetic dataset of *EPOS-Lid* contains in total 100,000 point clouds, which are split in two subsets:

- Synthetic training dataset: Contains 80,000 point clouds used for training;
- Synthetic validation dataset: Contains 20,000 point clouds used for validation, i.e., for comparing and tuning different models trained on the training dataset.

These datasets are provided together with the real EPOS datasets, and are available for download[1]. The pose and point cloud format are identical to the format for the real lidar datasets.

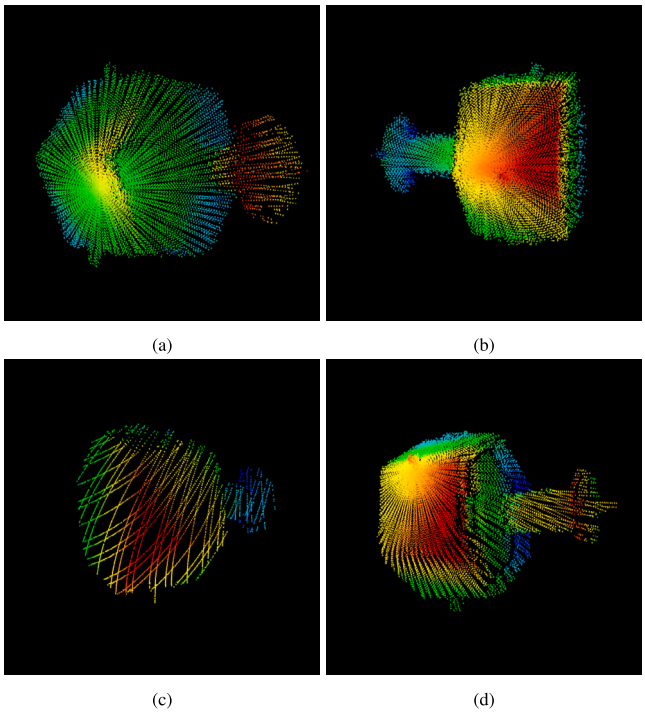


Fig. 9. Point clouds from the synthetic training dataset, captured for different poses, material properties and lidar scan times. Warm colors correspond to points closer to the sensor. (a) Point cloud number 1600; (b) Point cloud number 2400; (c) Point cloud number 2600; (d) Point cloud number 2700. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The synthetic datasets form a loose collection of point clouds, i.e., each point cloud is generated for a new relative position selected at random, not corresponding to a trajectory related to the previous point cloud. Therefore, the timestamp of each synthetic point cloud is not relevant, and set to zero in each pose file. For each point cloud, the relative range between the sensor and the target's center of mass is selected at random following a uniform distribution between 2 m and 25 m. The target is not perfectly centered in the lidar's field-of-view: The angular error between the sensor's principal direction and the direction to the target's center of mass follows a normal distribution with a standard deviation such that 3σ equal the sensor's half field-of-view (see Table 1). The attitude of the target satellite is also selected at random, such that the distribution in the attitude space is uniform.

For simulating motion blur, the relative velocities over a scan are selected following a uniform distribution bounded by a maximum of 3 cm/s and 5 deg/s. As for the real EPOS datasets, the pose in each ground truth file corresponds to the pose at the end of the scan time. The scan time is chosen to correspond to the settings of the adaptive framerate used in the EPOS experiments. It varies randomly around a value depending on the distance to the target, such that the point clouds contain in average approximately 10,000 points, but with potentially varying point cloud density.

Finally, since the exact material properties of the target are unknown, they are randomized and assigned a new value for each synthetic point cloud. This approach is known as domain randomization [44]. Through randomization, it is likely that the envelope in which the material properties are randomized contains the real values. For more details on the modeling of the reflectivity and specular characteristics of each material, we refer to [20]. Point clouds from the synthetic training datasets are shown in Fig. 9. On these point clouds, the scan pattern of the Livox lidar (see Fig. 2(b)) is clearly visible.

4. Pose estimation results

In this work, only the results of our pose estimation method are presented, but we encourage researchers developing different pose estimation methods to assess their results on the *EPOS-Lid* datasets. This will enable a common base for algorithm comparison and develeopement. In the following, the methodology and metrics used for our evaluation is detailed. The pose initialization method is evaluated on each point cloud of Datasets II and III of *EPOS-Lid*, which contain a wide variety of relative attitudes and distances. The pose tracking method is evaluated on Dataset I of *EPOS-Lid*, since it corresponds to a fly-around and approach trajectory which might correspond to the scenario of a real rendezvous mission.

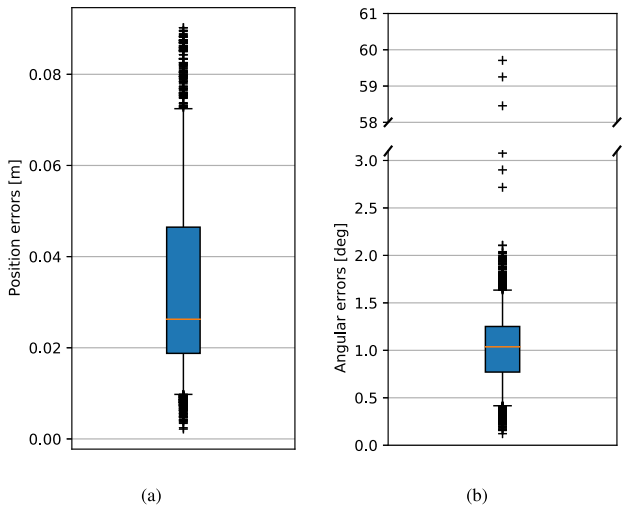
4.1. Pose initialization

The pose initialization method presented in Section 2.3 is trained on the synthetic training dataset of *EPOS-Lid*. Since the synthetic dataset is smaller than the dataset used to train the same model in previous work [21], the number of epochs is increased to 175. All other training parameters remain unchanged, including a batch size of 32, the Adam optimizer with an initial learning rate of 1e−3 and cosine decay, the data augmentation procedure, and a dropout rate of 0.3 in the top layer. The tuning of these training parameters is achieved by monitoring the evolution of the model performance on the synthetic validation dataset of *EPOS-Lid*.

After training, the models are evaluated on the real EPOS lidar datasets. The evaluation of the pose initialization results is performed on Datasets II and III of *EPOS-Lid*, since these datasets were recorded specifically to contain a wide variety of relative poses for testing initialization methods. As in [21], the PointNet++ backbone is modified to be optimized for runtime, and the results presented here are the results of the runtime optimized model. This model achieves real-time capability when evaluated on the CPU of onboard representative computing hardware. Each point cloud of the Datasets II and III is processed in isolation, i.e., the raw point cloud is passed to the pose initialization method, without an initial pose estimate. As in previous work [21], the pose initialization is followed by a refinement step with smoothed NDT [23]. The settings for the smoothed NDT refinement step are: a voxel grid size and a maximum point-to-cell distance of 7.5 cm, a maximum number of 30 iterations, and a termination threshold once the pose increment is lower than 0.05 deg and 1 mm.

The distribution of position errors (Eq. (2)) and attitude errors (Eq. (4)) is presented in Fig. 10. For both boxplots, the blue box represents the interquartile values, and the orange line the median. The whiskers extend to the 5th and 95th percentile, and outliers are marked by the black crosses. From Fig. 10(a), it is seen that all position errors are below 9 cm. The distribution of attitudes, in Fig. 10(b), presents more outliers. Amongst the 2555 point clouds of Datasets II and III, the attitude estimation error ϵ_{att} is higher than 3 deg for only 13 of them. This corresponds to an error percentage of 0.51%.

We now analyze more in detail these error cases of the pose initialization method. Out of the 13 error cases, for which the angular error of the method is above 3 deg, 12 correspond to Dataset II and only one to Dataset III. From these errors, approximately the half (6) correspond to an angular error of the pose estimation between 50 deg and 60 deg. This is also observable in the distributions of Fig. 10(b). Such an error indicates that the pose estimation method has been mislead by the symmetries of the target's hexagonal shape, see Fig. 4. Indeed, a hexagon is invariant by a rotation of ±60 deg. Thus in cases where the small antennas which would enable to discriminate between these poses are not clearly distinguishable (again, see Fig. 4), the method estimates an attitude which can be erroneous by ±60 deg. The remaining error cases are less straightforward to explain. Yet, it stands out that all these errors happen in very close range, for experiment times above 880 s, which corresponds to relative distances between the



(a) (b)

Fig. 10. Pose errors when evaluating the pose initialization followed by the NDT refinement step to every point cloud of Datasets II and III: (a) Distribution of position errors ϵ_{pos}; (b) Distribution of attitude errors ϵ_{att}. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Initialization errors (median and 99th percentile) when using the neural network based initialization followed by smoothed NDT on all point clouds of Datasets II and III.

Position error [cm]		Attitude error [deg]	
Median	99th %ile	Median	99th %ile
2.62	8.69	1.04	1.96

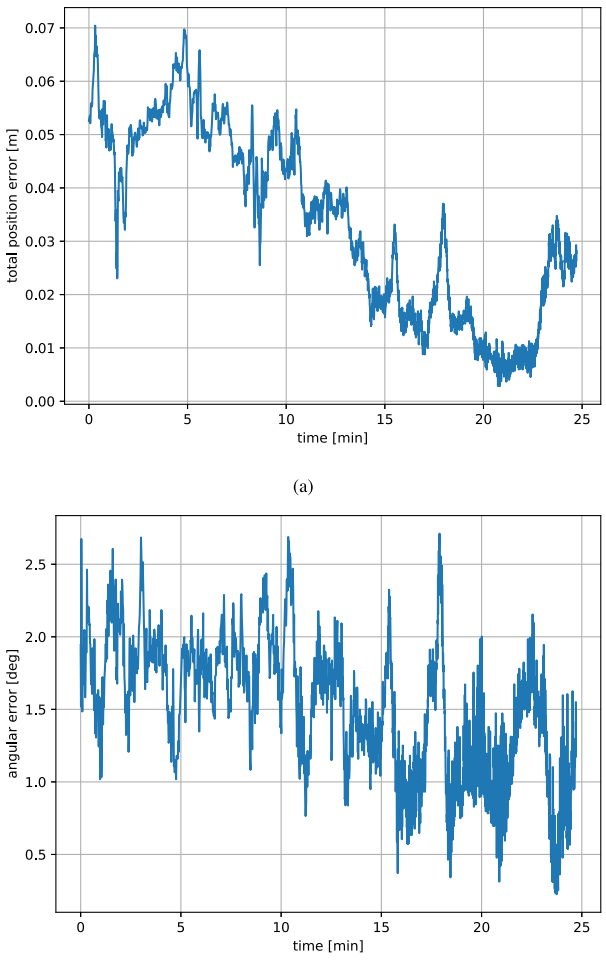
target and chaser satellite below 6 m, see Fig. 8. An explanation is that in very close range, the target is only partially in the field-of-view, so that fewer features of its shape are observable in the point cloud.

To compare this benchmark method with future models, we suggest comparing two metrics, given the distribution of position errors ϵ_{pos}, and of attitude errors ϵ_{att}. The first metric is the median value of these errors, which indicates the precision of the method. The second metric is the upper 99th percentile of the errors. This percentile enables to quantify the robustness of the pose initialization, which is essential for a critical application like pose estimation during space rendezvous. We argue that demonstrating that 99% of the pose estimates lie below a certain error threshold is a good measure to certify or evaluate a pose estimation method. These two metrics for the benchmark pose initialization method are presented in Table 4.

4.2. Pose tracking

The pose tracking evaluation is performed on the Dataset I of *EPOS-Lid*. For the first point cloud of this dataset, the pose is initialized with the initialization method presented in Section 4.1. Afterwards, for all subsequent point clouds, tracking is performed with the smoothed NDT method. As in [23], the tracking is coupled with a motion filter. For each new point cloud, the initial pose estimate to start the NDT refinement is provided by the current prediction of the filter.

The results of the pose tracking on Dataset I are presented in Fig. 11. The results are the raw results of the smoothed NDT tracker, not the filtered results. The position error, presented in Fig. 11(a), is around 5 cm during the fly-around phase, before decreasing to around 3 cm during the approach. The maximum value error is reached during the fly-around, with a position error of 7.04 cm. The attitude error of the pose estimation is presented in Fig. 11(b). The maximum angular error of 2.71 deg is reached at the beginning of the tracking.



(a)

(b)

Fig. 11. Tracking results on *EPOS-Lid* Dataset I over time: (a) Position error ϵ_{pos}; (b) Attitude error ϵ_{att}.

Table 5
Tracking errors (median and 99th percentile) when using smoothed NDT to track the target pose on the consecutive point clouds of Dataset I.

Position error [cm]		Attitude error [deg]	
Median	99th %ile	Median	99th %ile
2.24	6.41	1.36	2.48

To quantitatively evaluate the pose tracking results, the same metric as for the pose initialization is used, i.e., the median and 99th percentile values of the distributions of position and attitude errors is evaluated. These values are presented in Table 5.

5. Conclusion

This work presented *EPOS-Lid*, an openly available lidar dataset for pose estimation in non-cooperative rendezvous scenarios. The datasets comprise a synthetic and real part. The synthetic data is generated with a high fidelity lidar simulator, and can be used to train and validate neural network based pose estimation methods. The real datasets are collected at EPOS, and are meant to test lidar-based pose initialization and tracking methods. Importantly, the target satellite considered in this work is symmetric, such that a metric accounting for the symmetries is introduced for evaluating the pose estimation accuracy. Benchmark methods for pose estimation are presented to demonstrate the usability of the datasets. To ensure consistency in future evaluations, it is encouraged to use the same comparison metrics.

These datasets mark an effort to standardize and accelerate the development of lidar based pose estimation methods for space rendezvous. However, the sensor used in this work is a sensor from the automotive domain, since no space-grade lidar sensors was available at the EPOS facility at the time of this research. Future datasets might be recorded with a more representative lidar sensor, and possibly a different target satellite.

CRediT authorship contribution statement

Léo Renaut: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Ksenia Klionovska:** Writing – review & editing, Writing – original draft, Data curation, Conceptualization. **Maximilian Albracht:** Writing – review & editing, Writing – original draft, Data curation, Conceptualization. **Heike Frei:** Writing – review & editing, Writing – original draft, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Practical usage

Each point cloud file in our dataset is provided in a simple ASCII text format with the extension `.3d`. Each file contains one 3D point per line with three floating-point values. There is no header. All values are in meters. For example, the beginning of a point cloud (2462.3d) from Dataset I looks like:

```
1.3660 0.0350 -0.0490
1.3660 0.0330 -0.0470
1.3660 0.0310 -0.0450
...
```

To support practical use, we provide the following Python script in Listing 1, which converts a `.3d` file into a standard `.pcd` file. The standard `.pcd` file can be visualized using open-source tools such as `pcl_viewer` from the Point Cloud Library (PCL) [45], or further processed using any PCL-compatible pipeline.

```python
import numpy as np

def convert_to_pcd(input_file, output_file):
    points = np.loadtxt(input_file)
    with open(output_file, 'w') as f:
        f.write("# .PCD v0.7 - Point Cloud Data file
            format\n")
        f.write("VERSION 0.7\n")
        f.write("FIELDS x y z\n")
        f.write("SIZE 4 4 4\n")
        f.write("TYPE F F F\n")
        f.write("COUNT 1 1 1\n")
        f.write(f"WIDTH {len(points)}\n")
        f.write("HEIGHT 1\n")
        f.write("VIEWPOINT 0 0 0 1 0 0 0\n")
        f.write(f"POINTS {len(points)}\n")
        f.write("DATA ascii\n")
        for p in points:
            f.write(f"{p[0]} {p[1]} {p[2]}\n")

# Example usage
convert_to_pcd("2462.3d", "2462.pcd")
```

Listing 1: Python function to convert a `.3d` point cloud to the PCL `.pcd` format.

To use this tool, the PCL library must be installed on the system. Installation instructions and source code are available at: pointclouds.org *[Accessed: 3 July 2025]*. After converting one file format to another one, the file can be visualized by calling:

```
pcl_viewer 2462.pcd
```

Listing 2: Command for visualization of the converted point cloud using PCL.

The datasets presented in this paper are intended to support a wide range of applications, including 3D point cloud registration, tracking, and pose estimation. We provide raw point cloud data in `.3d` format along with corresponding ground truth annotations in `.pose` files. As the dataset is designed to be adaptable to different research goals, we do not include a fixed evaluation script. Instead, users are encouraged to develop their own benchmarking pipelines tailored to their specific algorithms and use cases. Although the dataset is provided in `.3d` format, users working with ROS can easily create their own scripts to convert the files into `sensor_msgs/PointCloud2` messages or package them into ROS bag files, depending on their specific application needs.

References

[1] L.P. Cassinis, R. Fonod, E. Gill, Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft, Prog. Aerosp. Sci. 110 (2019) 100548, http://dx.doi.org/10.1016/j.paerosci.2019.05.008.

[2] L. Liu, G. Zhao, Y. Bo, Point cloud based relative pose estimation of a satellite in close range, Sensors 16 (6) (2016) 824, http://dx.doi.org/10.3390/s16060824.

[3] K. Klionovska, M. Burri, Hardware-in-the-loop simulations with umbra conditions for spacecraft rendezvous with PMD visual sensors, Sensors 21 (4) (2021) 1455, http://dx.doi.org/10.3390/s21041455.

[4] F. Amzajerdian, V.E. Roback, A. Bulyshev, P.F. Brewster, G.D. Hines, Imaging flash lidar for autonomous safe landing and spacecraft proximity operation, in: AIAA SPACE, 2016, p. 5591, http://dx.doi.org/10.2514/6.2016-5591.

[5] J.M. Leonard, M.C. Moreau, P.G. Antreasian, K.M. Getzandanner, E. Church, C. Miller, M.G. Daly, O.S. Barnouin, D.S. Lauretta, Cross-calibration of GNC and OLA LIDAR systems onboard OSIRIS-REx, in: Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, Springer, 2022, pp. 1585–1607, http://dx.doi.org/10.1007/978-3-031-51928-4_88.

[6] M. Pyrak, J. Anderson, Performance of northrop grumman's mission extension vehicle (MEV) RPO imagers at GEO, in: Autonomous Systems: Sensors, Processing and Security for Ground, Air, Sea and Space Vehicles and Infrastructure, Vol. 12115, SPIE, 2022, pp. 64–82, http://dx.doi.org/10.1117/12.2631524.

[7] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations, Prog. Aerosp. Sci. 93 (2017) 53–72, http://dx.doi.org/10.1016/j.paerosci.2017.07.001.

[8] C. Tecchia, M. Piccinin, A. Nocerino, R. Opromolla, U. Hillenbrand, Pose acquisition of non-cooperative spacecraft from LiDAR: Comparison of methods based on point correspondences, in: AIAA SciTech Forum, 2025, p. 1416, http://dx.doi.org/10.2514/6.2025-1416.

[9] S. Ruel, T. Luu, A. Berube, Space shuttle testing of the TriDAR 3D rendezvous and docking sensor, J. Field Robot. 29 (4) (2012) 535–553, http://dx.doi.org/10.1002/rob.20420.

[10] F. Yin, W. Chou, Y. Wu, G. Yang, S. Xu, Sparse unorganized point cloud based relative pose estimation for uncooperative space target, Sensors 18 (4) (2018) 1009, http://dx.doi.org/10.3390/s18041009.

[11] J.O. Woods, J.A. Christian, LIDAR-based relative navigation with respect to non-cooperative objects, Acta Astronaut. 126 (2016) 298–311, http://dx.doi.org/10.1016/j.actaastro.2016.05.007.

[12] K. Klionovska, H. Benninghoff (now Frei), Initial pose estimation using PMD sensor during the rendezvous phase in on-orbit servicing missions, in: 27th AIAA/AAS Space Flight Mechanics Meeting, 2017.

[13] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, Pose estimation for spacecraft relative navigation using model-based algorithms, IEEE Trans. Aerosp. Electron. Syst. 53 (1) (2017) 431–447, http://dx.doi.org/10.1109/TAES.2017.2650785.

[14] W. Guo, W. Hu, C. Liu, T. Lu, Pose initialization of uncooperative spacecraft by template matching with sparse point cloud, J. Guid. Control Dyn. 44 (9) (2021) 1707–1720, http://dx.doi.org/10.2514/1.G005042.

[15] C. Schmitt, J. Both, F. Kolb, RVS3000-3D: LIDAR meets neural networks, in: International Symposium of Artificial Intelligence, Robotics and Automation in Space, 2018, pp. 1–7.
[16] S. Zhang, W. Hu, W. Guo, 6-dof pose estimation of uncooperative space object using deep learning with point cloud, in: IEEE Aerospace Conference, AERO, IEEE, 2022, pp. 1–7, http://dx.doi.org/10.1109/AERO53065.2022.9843444.
[17] S. Zhang, W. Hu, W. Guo, C. Liu, Neural-network-based pose estimation during noncooperative spacecraft rendezvous using point cloud, J. Aerosp. Inf. Syst. 20 (8) (2023) 462–472, http://dx.doi.org/10.2514/1.I011179.
[18] O. Kechagias-Stamatis, N. Aouf, V. Dubanchet, M.A. Richardson, Deeplo: Multi-projection deep LIDAR odometry for space orbital robotics rendezvous relative navigation, Acta Astronaut. 177 (2020) 270–285, http://dx.doi.org/10.1016/j.actaastro.2020.07.034.
[19] E.A. Pensado, L.M.G. de Santos, H.G. Jorge, M. Sanjurjo-Rivo, Deep learning-based target pose estimation using LiDAR measurements in active debris removal operations, IEEE Trans. Aerosp. Electron. Syst. 59 (5) (2023) 5658–5670, http://dx.doi.org/10.1109/TAES.2023.3262505.
[20] L. Renaut, H. Frei, A. Nuchter, CNN-based pose estimation of a non-cooperative spacecraft with symmetries from lidar point clouds, IEEE Trans. Aerosp. Electron. Syst. (2024) http://dx.doi.org/10.1109/TAES.2024.3517574.
[21] L. Renaut, H. Frei, A. Nüchter, Deep learning on 3D point clouds for fast pose estimation during satellite rendezvous, Acta Astronaut. 232 (2025) 231–243, http://dx.doi.org/10.1016/j.actaastro.2025.03.009.
[22] P. Besl, N.D. McKay, A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (02) (1992) 239–256, http://dx.doi.org/10.1109/34.121791.
[23] L. Renaut, H. Frei, A. Nüchter, Lidar pose tracking of a tumbling spacecraft using the smoothed normal distribution transform, Remote. Sens. 15 (9) (2023) 2286, http://dx.doi.org/10.3390/rs15092286.
[24] H. Benninghoff, F. Rems, E.-A. Risse, P. Irmisch, I. Ernst, B. Brunner, M. Stelzer, R. Lampariello, R. Krenn, M. Reiner, et al., RICADOS-rendezvous, inspection, capturing and detumbling by orbital servicing, in: 7th International Conference on Astrodynamics Tools and Techniques, 2018.
[25] Y. Li, Y. Wang, Y. Xie, Using consecutive point clouds for pose and motion estimation of tumbling non-cooperative target, Adv. Space Res. 63 (5) (2019) 1576–1587, http://dx.doi.org/10.1016/j.asr.2018.11.024.
[26] D. Sun, L. Hu, H. Duan, H. Pei, Relative pose estimation of non-cooperative space targets using a TOF camera, Remote. Sens. 14 (23) (2022) 6100, http://dx.doi.org/10.3390/rs14236100.
[27] P.F. Proença, Y. Gao, Deep learning for spacecraft pose estimation from photorealistic rendering, in: IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 6007–6013, http://dx.doi.org/10.1109/ICRA40945.2020.9197244.
[28] Y. Hu, S. Speierer, W. Jakob, P. Fua, M. Salzmann, Wide-depth-range 6d object pose estimation in space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15870–15879, http://dx.doi.org/10.48550/arXiv.2104.00337.
[29] M.A. Mohamed Ali, A. Rathinam, V. Gaudilliere, M. Ortiz Del Castillo, D. Aouada, CubeSat-CDT: A cross-domain dataset for 6-DoF trajectory estimation of a symmetric spacecraft, in: Proceedings of the 17th European Conference on Computer Vision Workshops, 2022, http://dx.doi.org/10.1007/978-3-031-25056-9_8.
[30] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Satellite pose estimation challenge: Dataset, competition design, and results, IEEE Trans. Aerosp. Electron. Syst. 56 (5) (2020) 4083–4098, http://dx.doi.org/10.1109/TAES.2020.2989063.
[31] S. Velkei, C. Goldschmidt, K. Vass, A large-scale, physically-based synthetic dataset for satellite pose estimation, 2025, arXiv preprint arXiv:2506.12782.
[32] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière, D. Aouada, A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects, Acta Astronaut. (2023) http://dx.doi.org/10.1016/j.actaastro.2023.08.001.
[33] T.H. Park, M. Märtens, G. Lecuyer, D. Izzo, S. D'Amico, SPEED+: Next-generation dataset for spacecraft pose estimation across domain gap, in: IEEE Aerospace Conference, AERO, IEEE, 2022, pp. 1–15, http://dx.doi.org/10.1109/AERO53065.2022.9843439.
[34] A. Lotti, D. Modenini, P. Tortora, Investigating vision transformers for bridging domain gap in satellite pose estimation, in: International Conference on Applied Intelligence and Informatics, Springer, 2022, pp. 299–314.
[35] H. Yang, X. Xiao, M. Yao, Y. Xiong, H. Cui, Y. Fu, PVSPE: A pyramid vision multitask transformer network for spacecraft pose estimation, Adv. Space Res. 74 (3) (2024) 1327–1342.
[36] J. Lebreton, I. Ahrns, R. Brochard, C. Haskamp, H. Krüger, M.L. Goff, N. Menga, N. Ollagnier, R. Regele, F. Capolupo, et al., Training datasets generation for machine learning: Application to vision based navigation, 2024, arXiv preprint arXiv:2409.11383.
[37] H. Benninghoff, F. Rems, E.-A. Risse, C. Mietner, European proximity operations simulator 2.0 (EPOS) - A robotic-based rendezvous and docking simulator, J. Large-Scale Res. Facil. 3 (2017) http://dx.doi.org/10.17815/jlsrf-3-155.
[38] F. Rems, H. Frei, E.-A. Risse, M. Burri, 10-Year anniversary of the European proximity operations simulator 2.0 - looking back at test campaigns, rendezvous research and facility improvements, Aerosp. 8 (9) (2021) http://dx.doi.org/10.3390/aerospace8090235.
[39] Livox, Livox mid-40/mid-100 specs, 2025, https://www.livoxtech.com/mid-40-and-mid-100/specs (Accessed 03 February 2025).
[40] L.P. Cassinis, A. Menicucci, E. Gill, I. Ahrns, M. Sanchez-Gestido, On-ground validation of a CNN-based monocular pose estimation system for uncooperative spacecraft: Bridging domain shift in rendezvous scenarios, Acta Astronaut. 196 (2022) 123–138, http://dx.doi.org/10.1016/j.actaastro.2022.04.002.
[41] R.G. Brazeal, B.E. Wilkinson, H.H. Hochmair, A rigorous observation model for the risley prism-based livox mid-40 lidar sensor, Sensors 21 (14) (2021) 4722, http://dx.doi.org/10.3390/s21144722.
[42] M. Han, L. Wang, L. Xiao, H. Zhang, C. Zhang, X. Xu, J. Zhu, QuickFPS: Architecture and algorithm co-design for farthest point sampling in large-scale point clouds, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 42 (11) (2023) 4011–4024, http://dx.doi.org/10.1109/TCAD.2023.3274922.
[43] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017) http://dx.doi.org/10.48550/arXiv.1706.02413.
[44] K.M. Kajak, C. Maddock, H. Frei, K. Schwenk, Domain randomisation and CNN-based keypoint-regressing pose initialisation for relative navigation with uncooperative finite-symmetric spacecraft targets using monocular camera images, Adv. Space Res. 72 (7) (2023) 2824–2844, http://dx.doi.org/10.1016/j.asr.2023.02.024.
[45] R.B. Rusu, S. Cousins, 3D is here: Point cloud library (PCL), in: IEEE International Conference on Robotics and Automation, ICRA, 2011, pp. 1–4.