Astrobee ISS Free-Flyer Datasets for Space Intra-Vehicular Robot Navigation Research

Suyoung Kang[1], Ryan Soussan[2,3], Daekyeong Lee[4], Brian Coltin[2,3],
Andres Mora Vargas[2,3], Marina Moreira[2,3], Kathryn Hamilton[2], Ruben Garcia[2,3], Maria Bualat[2],
Trey Smith[2], Jonathan Barlow[2,3], Jose Benavides[2], Eunju Jeong[4], and Pyojin Kim[5]

***Abstract*—We present the first annotated benchmark datasets for evaluating free-flyer visual-inertial localization and mapping algorithms in a zero-g spacecraft interior. The Astrobee free-flying robots that operate inside the International Space Station (ISS) collected the datasets. Space intra-vehicular free-flyers face unique localization challenges: their IMU does not provide a gravity vector, their attitude is fully arbitrary, and they operate in a dynamic, cluttered environment. We extensively evaluate state-of-the-art visual navigation algorithms on these challenging Astrobee datasets, showing superior performance of classical geometry-based methods over recent data-driven approaches. The datasets include monocular images and IMU measurements, with multiple sequences performing a variety of maneuvers and covering four ISS modules. The sensor data is spatio-temporally aligned, and extrinsic/intrinsic calibrations, ground-truth 6-DoF camera poses, and detailed 3D CAD models are included to support evaluation. The datasets are available at: https://astrobee-iss-dataset.github.io/.**

***Index Terms*—Data Sets for SLAM, Space Robotics and Automation, SLAM, Autonomous Vehicle Navigation.**

I. Introduction

VISUAL navigation, such as visual-inertial odometry (VIO) and visual simultaneous localization and mapping (vSLAM), is one of the fundamental building blocks in robotics, autonomous vehicles, and augmented reality (AR) applications for positioning the 6 degrees of freedom (DoF) camera pose of a given query image. Due to its importance, various indoor and outdoor datasets [1]–[3] from autonomous vehicles to drones have been published over the past decade and have performed as *de facto* benchmark baselines, enabling the advancement of visual navigation research. Thanks to various datasets, state-of-the-art VIO and SLAM methods [4]–[7], which utilize multiple-view geometry and/or deep learning, show promising 6-DoF camera motion tracking results in a variety of indoor and outdoor environments on Earth.

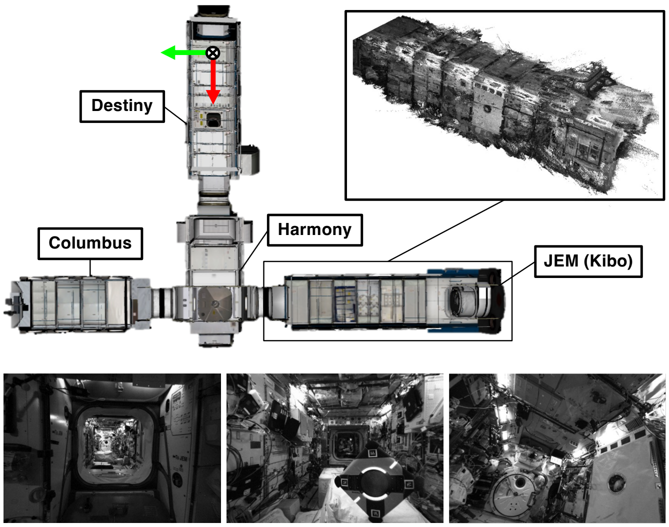


Fig. 1. Illustration of the Astrobee datasets. Dense 3D reconstruction of JEM (top) with our ground-truth generation pipeline. Astrobee, a free-flying intra-vehicular robot (IVR) designed to autonomously navigate on the International Space Station (ISS), acquires monocular image sequences (bottom) and IMU measurements across four ISS modules (top).

Several recent studies [8], [9] have focused on developing free-flying robots (NASA Astrobee, JAXA Int-Ball, DLR CIMON, etc.) that perform a variety of intra-vehicular activities in microgravity [10] on the International Space Station (ISS). We will increasingly need smart, self-run space robots to keep an eye on things while humans are away in future spacecraft like the Gateway lunar space station [11]. While accurate and robust in-spacecraft navigation is critical for the missions of intra-vehicular robots, a dataset for visual navigation in a spacecraft has not yet existed. When considering challenges for visual navigation, intra-vehicular environments differ from indoor and outdoor scenes on Earth due to the nature of spacecraft: absence of a gravity vector, severe lighting changes, environmental changes/discontinuities between ISS modules, and occlusions from unorganized cargo bags.

To address these issues, we introduce new visual navigation datasets acquired in the challenging interior environments of ISS, which have not been covered by existing datasets. The datasets are captured by the Astrobee free-flying robots [8] onboard the ISS since 2019 during intra-vehicular activities, including interior environmental surveys (e.g., systems in-

Manuscript received: August 5, 2023; Revised: November 20, 2023; Accepted: January 4, 2024. This paper was recommended for publication by Editor Ashis Banerjee upon evaluation of the Associate Editor and Reviewers' comments. *(Corresponding author: Pyojin Kim.)*

[1] Department of Electronics Engineering, Sookmyung Women's University, Seoul 04310, South Korea. 1913084@sookmyung.ac.kr

[2] NASA Ames Research Center, Moffett Field, CA, USA. {ryan.soussan,brian.coltin,andres.moravargas, marina.gouveiamoreira,kathryn.browne, ruben.m.garciaruiz,maria.bualat,trey.smith, jonathan.s.barlow,jose.v.benavides}@nasa.gov

[3] KBR Inc., Houston, TX, USA.

[4] Department of Mechanical Systems Engineering, Sookmyung Women's University, Seoul 04310, South Korea. {swbubl,eunju0316}@sookmyung.ac.kr

[5] School of Mechanical Engineering, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea. pjinkim@gist.ac.kr

Digital Object Identifier (DOI): see top of this page.

spection, monitoring, and sound level measurements). They consist of images from a forward-facing monocular navigation camera (NavCam), IMU measurements in microgravity, and pseudo ground truth 6-DoF camera poses generated from our sequential mapping and localization pipeline [10], [12]. Our main contributions are as follows:

- We present twenty-three new Astrobee datasets for benchmarking in-cabin visual navigation methods on the ISS with pseudo ground truth 6-DoF camera poses as well as 3D CAD models.
- We present detailed evaluations of state-of-the-art VO/SLAM methods showing their limitations in these challenging intra-vehicular activities on the ISS.

To the best of our knowledge, this is the first visual-inertial dataset obtained from outer space for intra-vehicular robots (IVR). We hope our Astrobee datasets serve as standards for comparison, facilitating the progress of IVR navigation inside facilities such as the ISS, Gateway, and future Commercial LEO Destinations (CLDs).

II. Related Work

Numerous datasets [1]–[3] containing a wide range of indoor and outdoor visual scenarios on Earth have been released in computer vision and robotics communities over the past decade, but there is no dataset for autonomous navigation of IVRs in a spacecraft yet.

A variety of free-flying robots in space orbital stations are being actively researched for photographing operations [9], [13], microgravity robotics research platforms and automation of ISS maintenance tasks [8], [14], and autonomous assistance system to aid astronauts [15]. The ability to precisely locate intra-vehicular robots and create accurate maps inside the ISS forms the basis for their utilization, regardless of the scenarios.

The NASA SPHERES [14], the predecessor of Astrobee, can only perform 6-DoF positioning on the ISS with the help of pre-installed ultrasonic beacons in the SPHERES work envelope. Although the JAXA Int-Ball [9] employs an in-cabin visual navigation approach with optical navigation cameras, it requires the two stereoscopic markers to be installed at the planned position. The localization accuracy of the marker-based navigation method depends heavily on how well the artificial markers are visible within the camera's field of view (FOV). While the DLR/Airbus CIMON [15] is on the move on the ISS, a dual 3D camera sensor builds a map of the surrounding environments based on SLAM algorithms. However, its VIO/SLAM algorithms and navigation accuracy have not been reported in detail, and CIMON is only tested in the Columbus laboratory module. The NASA Astrobee freely navigates inside the ISS, originally employing the MSCKF [16]-based visual-inertial navigation method [12], and not relying on any artificial markers or auxiliary devices. In order to further improve the navigation performance of Astrobee, semantic mapping [17] has been recently studied and a visual-inertial graph-based localization (AstroLoc) [10] algorithm has replaced the MSCKF implementation. While this has improved localization accuracy, AstroLoc sometimes fails to track the 6-DoF ego-motion due to severe lighting changes [18], abrupt in-place rotations, and navigation between ISS modules.

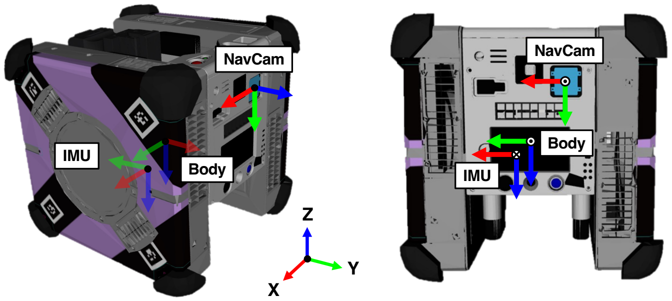


Fig. 2. The Astrobee free-flying IVR used for dataset collection. A NavCam provides monocular image sequences inside the ISS at 15 Hz. Angular rates and acceleration measurements from the IMU are logged at 250 Hz. The pseudo ground truth of camera poses for all images is generated through our offline SfM pipeline.

TABLE I
Sensors in Astrobee and Ground-Truth Accuracy

Sensor	Model	Rate (Hz)	Characteristics
Camera	DFM 42BUC03-ML	15	1280×960, Global Shutter
IMU	Epson M-G362PDC1	250	MEMS
Position	Our SfM Pipeline	15	Accuracy:≈ 5 cm

Overall, research on the navigation of IVRs in spacecraft is quite sparse while up-to-date real-time VIO/SLAM methods such as ORB-SLAM3 [4], DROID-SLAM [7], and VINS-Mono [6] are being actively validated on robotic platforms and Earth-based datasets. We hope our Astrobee dataset stimulates new and exciting research for the autonomous navigation of in-spacecraft free-flying robots in the upcoming era of expanded human presence in space by providing challenging scenarios for VIO and SLAM-based methods.

III. The Astrobee Datasets

The Astrobee free-flying robots have been operating onboard the ISS since 2019. We describe the characteristics of the Astrobee datasets as well as the sensor setup of Astrobee and the data acquisition/calibration processes. We explain how to generate pseudo ground truth camera poses for all images with our sequential mapping and localization algorithms.

A. Astrobee Platform

The Astrobee free-flying robots, equipped with a suite of six commercial off-the-shelf (COTS) external sensors [8], are utilized on the ISS for data collection as shown in Fig. 2. Although each Astrobee has a variety of cameras such as HazCam, PerchCam, and SciCam [8], we have built a dataset with monocular images from NavCam for now, which is mainly utilized for general-purpose in-cabin visual navigation. The NavCam is mounted in a forward-facing position, which provides monocular image sequences inside the ISS at 15 Hz with a 116° field of view (FOV), fixed focus, and 1.2 megapixel (MP) resolution (for full details of Astrobee hardware, refer to [19]). Angular rates and acceleration measurements from the IMU are logged at 250 Hz in Table I. Fig. 2 illustrates the position of the NavCam and IMU, and corresponding camera and IMU body frames.

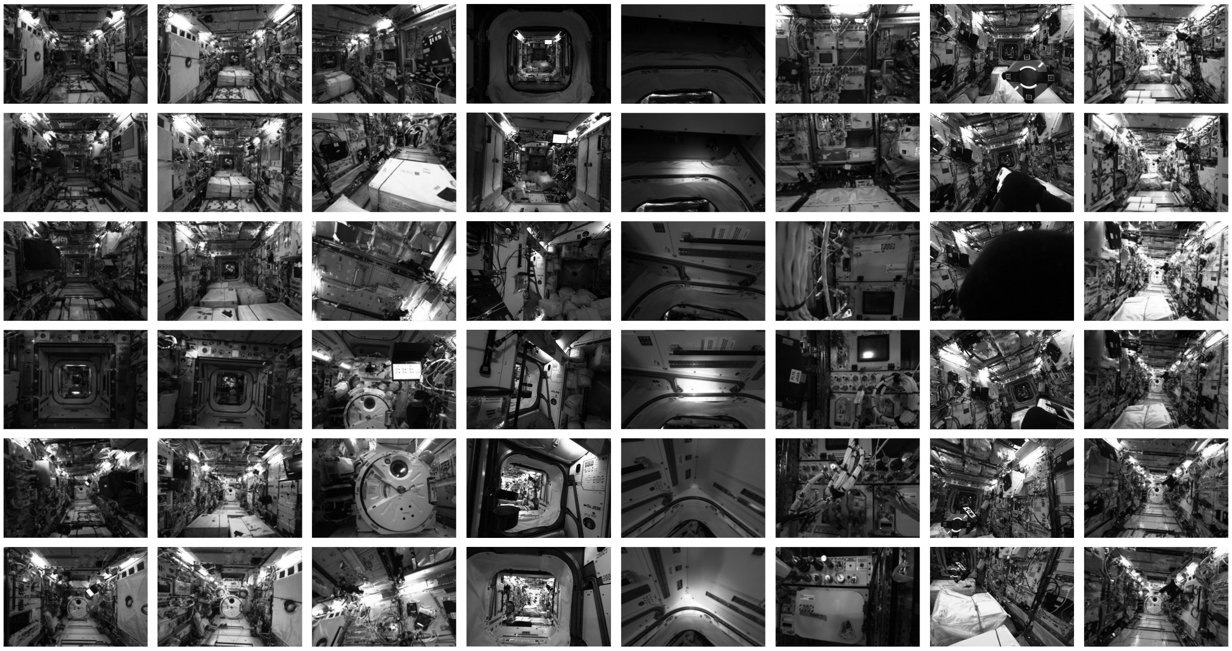


Fig. 3. Example images from NavCam to demonstrate the diversity in our Astrobee datasets. Each column represents a sequence of intra-vehicular activities of Astrobee free-flying robots currently operating on the ISS. They include lighting changes within Kibo, abrupt in-place 3D rotations, free flights between ISS modules, hatch/wire inspection tasks, and occlusions by other Astrobee robots.

The Astrobee robots make extensive use of the open-source Robot Operating System (ROS), which is used for all internal communications between various Astrobee software components. Visual and inertial data are timestamped and recorded onboard the Astrobee as rosbag files with ROS facilities. Since there are no sensors or beacons on the ISS to localize Astrobee directly, we generate pseudo ground truth later through our offline Structure-from-Motion (SfM) process with an accuracy of about five cm [18] in Table I.

B. Dataset Description

Our datasets consist of various intra-vehicular activities of Astrobee free-flying robots currently operating on the ISS between May 13, 2019 and July 14, 2022. The Astrobee IVRs perform tasks and operate from the “Kibo” Japanese Experiment Module (JEM) through the “Harmony” Node 2 module to the “Destiny” U.S. Lab module on the ISS as shown in Fig. 1. In the Astrobee datasets, they fly autonomously between ISS modules and conduct tasks including interior environmental surveys (e.g., indoor mapping and sound level measurement) inside Kibo, which is the largest ISS module (cabin free space $\sim 8 \times 2.25 \times 2.25$ m). Environmental changes (e.g., reconfiguration of experiments, individual light sources moving, and stowage containers being attached to the deck) of Kibo over time are also reflected in the Astrobee datasets. Fig. 3 provides visual examples of the proposed Astrobee datasets. We re-organize Astrobee datasets into four categories according to the scenarios: *Calibration*, *Testing and Debugging*, *Free Flight*, and *Intra-Vehicular Activity*. Table II shows statistics over the 23 sequences of the Astrobee dataset.

We briefly summarize the recorded sequences according to these categories.

1) Calibration: To calibrate the various cameras and the IMU installed in Astrobee, we record fixed targets attached to the docking station while the Astrobee moves around it. We employ Kalibr [20], an open-source calibration toolbox. The fixed targets include both an 11×6 checkerboard with 25.4mm square size and ALVAR fiducials [21] (see Sec. III-D for calibration details).

2) Testing and Debugging: These sequences are intended to test and debug the separated movement of each axis in 6-DoF camera motions along and around the principal axes of NavCam when evaluating visual navigation and localization algorithms. In the *rpy* (roll-pitch-yaw) sequences, the camera rotates primarily around the principal axes, the camera frame, with little translational motions. Similarly, in the *dock* sequences, the camera primarily moves along the Z-axis (forward/backward), from the docking station, with occasional minor rotational motions during the redocking process.

3) Free Flight: These sequences consist of the nine datasets recorded by Astrobee while freely flying within Kibo and/or between modules inside the ISS. In some sequences, Astrobee robots fly autonomously and come back to where they started, containing several loop closures to allow visual SLAM algorithms to recognize previously visited spaces and utilize them to reduce accumulated drift error. We also record the “return journey” sequences with the camera at different viewing angles, facing forward, upward, leftward, and rightward under three different lighting conditions (dark, middle, and light due to changes in date). The *USL2JEM* and *JEM2USL* sequences include autonomous free flights from JEM to USL and vice versa in light-changing environments.

4) Intra-Vehicular Activity: We prepare diverse intra-vehicular activities and survey scenarios conducted onboard free-flying space IVRs. Some sequences pose challenges due to severe lighting changes, environmental changes/discontinuities between modules, and occlusions caused by cargo bags or other Astrobee robots. The fifth

TABLE II
LIST OF ASTROBEE DATASET SEQUENCES

Sequence Name	Duration [s]	Avg. Trans. Speed. [m/s]	Avg. Rot. Speed. [deg/s]
Testing and Debugging			
td_roll	63	0.009	2.958
td_pitch	75	0.007	2.514
td_yaw	50	0.006	3.753
td_dock	98	0.028	0.417
Free Flight			
ff_return_journey_forward	402	0.032	2.155
ff_return_journey_up	413	0.032	2.563
ff_return_journey_down	398	0.030	2.400
ff_return_journey_left	303	0.041	3.068
ff_return_journey_right	328	0.033	2.750
ff_return_journey_rot	108	0.114	12.746
ff_JEM2USL_dark	32	0.148	9.162
ff_USL2JEM_bright	92	0.056	1.566
ff_nod2_dark	296	0.009	2.616
ff_nod2_bright	265	0.010	2.639
Intra-Vehicular Activity			
iva_kibo_trans	229	0.047	0.796
iva_kibo_rot	196	0.054	2.402
iva_hatch_inspection1	403	0.008	1.234
iva_hatch_inspection2	521	0.011	0.971
iva_watch_queenbee	236	0.009	2.285
iva_robot_occulusion	192	0.010	2.748
iva_ARtag	62	0.083	3.312
iva_badlocal_rotation	313	0.009	2.650
iva_badlocal_descend	244	0.005	0.335

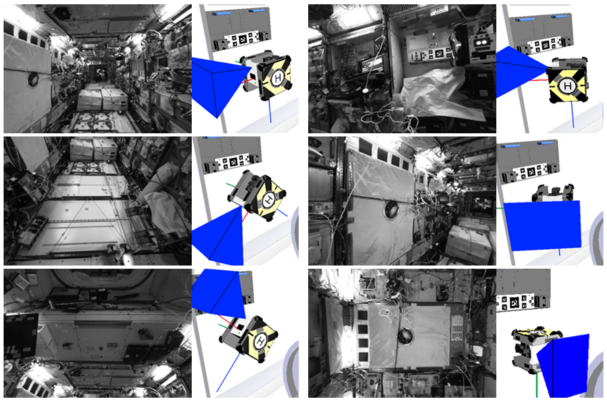


Fig. 4. Various orientations of NavCam viewing frustums attached to the Astrobee in the *Free Flight* category. Each pair from left to right represents forward, down, up, right, left, and rot sequences in "ff_return_journey_".

column of Fig. 3 shows example images in *hatch inspection* sequences while observing the hatch seal and shooting light to inspect it. The *wire inspection* sequences record the walls in JEM very closely, resulting in some wires causing severe occlusions as shown in the sixth column of Fig. 3. In the seventh column of Fig. 3, severe occlusions and moving dynamic objects occur in the images by another Astrobee robot performing different missions.

C. Dataset Format

We specify the format and conventions in which sensor data, ground-truth, and calibration parameters are reported. We provide each Astrobee sequence in the TUM RGB-D format [1] for compatibility with existing evaluation and dataset parsing tools. Each sequence is compressed as a single zip file containing a description file, images, calibration parameters, ground-truth, and IMU measurements as illustrated in Fig. 6.

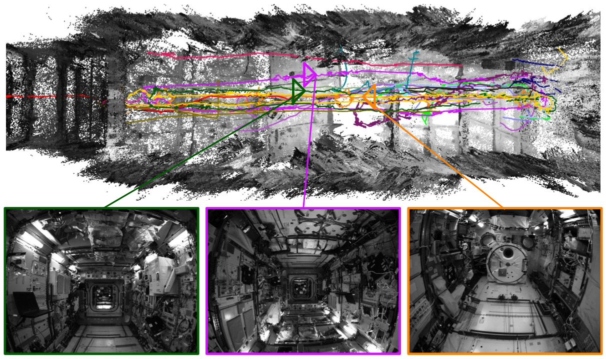
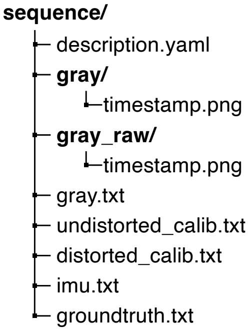


Fig. 5. Astrobee traces of our dataset recordings on the ISS (top). Colors encode each Astrobee dataset sequence. NavCam images (bottom) taken from the locations marked as camera frustums (dark green, purple, and orange) for each sequence show the interior of Kibo. The purple pose is rolled approximately 180° relative to the dark green pose, thus the image in purple is upside down, which is an interesting IVR maneuver observed in the spacecraft.



```
sequence/
  ├ description.yaml
  ├ gray/
  │   └timestamp.png
  ├ gray_raw/
  │   └timestamp.png
  ├ gray.txt
  ├ undistorted_calib.txt
  ├ distorted_calib.txt
  ├ imu.txt
  └ groundtruth.txt
```

Fig. 6. Structure of the provided zip files and their location that stores each Astrobee sequence. Note that 'sequence' is a placeholder.

- `description.yaml`: a text file containing an overall description of the sequence (the name of the robot used, the original rosbag file, and the date of recording, etc.),
- `gray/`: a folder containing all undistorted grayscale images (PNG format, 1 channel, 8-bit per channel),
- `gray_raw/`: a folder containing all raw (distorted) grayscale images with FOV lens distortion [21],
- `gray.txt`: a text file with a consecutive list of all grayscale images (format: timestamp filename),
- `undistorted_calib.txt`: a text file containing camera intrinsic parameters for undistorted images (format: fx fy cx cy),
- `distorted_calib.txt`: a text file containing camera intrinsic parameters for distorted images with FOV lens distortion coefficient (format: fx fy cx cy w),
- `imu.txt`: a text file containing the timestamped gyro and accelerometer measurements expressed in IMU body frame (format: timestamp wx wy wz ax ay az),
- `groundtruth.txt`: a text file containing the ground-truth 6-DoF camera poses for all grayscale images stored as the timestamped translation vector and unit quaternion expressed in ISS world coordinate frame (format: timestamp tx ty tz qx qy qz qw).

In addition to these zip files organized by activity scenarios in the spacecraft, we also release the original raw rosbag file recorded during the real flight of Astrobee on the ISS.

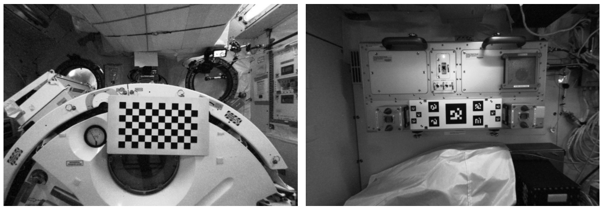


Fig. 7. Example images containing calibration targets in Kalibr [20] for NavCam calibration. The checkerboard pattern attached to the Kibo airlock (left) and ALVAR fiducials installed in the Astrobee docking station (right).

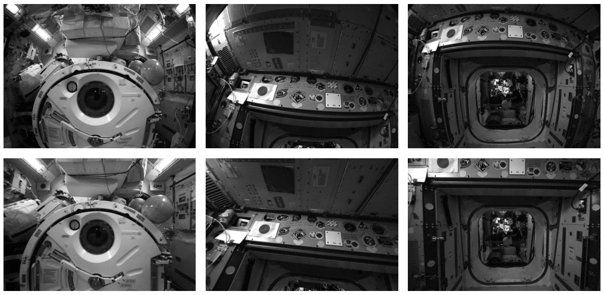


Fig. 8. The NavCam images comparison before (top) and after (bottom) undistortion given by camera calibration with the FOV lens distortion model. Target lines inside the ISS modules on the original images (top) are corrected to straight lines after undistortion (bottom).

D. Camera Calibration

The proposed Astrobee datasets contain raw (distorted) image data, intrinsic and extrinsic calibration parameters. We calibrate the intrinsic of the camera (NavCam) and the camera-IMU extrinsic prior to dataset collection with Kalibr [20], which is a ROS-friendly and automatic calibration tool to support various types of targets, projection models, and lens distortion models (for full details, refer to [21]). Images containing ALVAR fiducials and checkerboard targets from NavCam can be found in the *Calibration* category in the dataset as shown in Fig 7.

It is noteworthy that we employ the field-of-view (FOV) lens distortion model [22] rather than the radial-tangential because the distortion in the images becomes so severe (see Fig 8). The FOV model is more suitable for a fish-eye lens camera and its nonlinear distortion. We calibrate the FOV distortion coefficient, the FOV angle ω of the ideal fish-eye lens, with Kalibr [20], and can obtain the undistorted images in Fig 8. We provide both the original distorted images (`gray_raw/`) with the camera intrinsic and FOV distortion (`distorted_calib.txt`) and the undistorted images (`gray/`) with the intrinsic of the camera (`undistorted_calib.txt`) all together for the convenience of researchers.

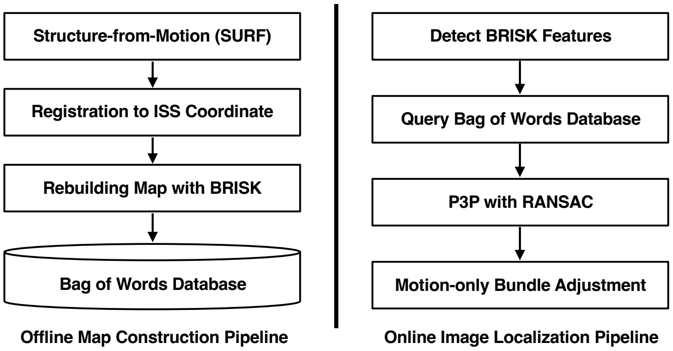

Structure-from-Motion (SURF)
Registration to ISS Coordinate
Rebuilding Map with BRISK
Bag of Words Database
Offline Map Construction Pipeline
Detect BRISK Features
Query Bag of Words Database
P3P with RANSAC
Motion-only Bundle Adjustment
Online Image Localization Pipeline

Fig. 9. Our offline 3D map construction and localization pipeline for generating pseudo ground truth camera poses for visual navigation benchmark.

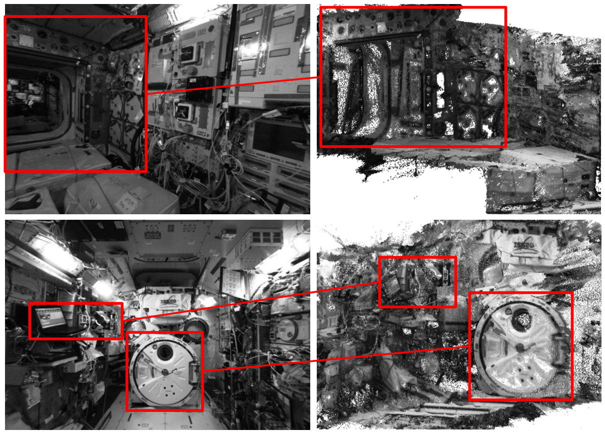


Fig. 10. A view from the NavCam showing the inside of Kibo (left) and a dense point cloud with our ground-truth 6-DoF camera poses (right). Objects in the red boxes (hatch, wires, laptop, and another Astrobee) and Kibo airlock are 3D reconstructed well from the dense point cloud (right), showing our ground-truth generation is qualitatively accurate.

E. Ground Truth Generation

The pseudo ground truth creation pipeline relies on the offline map building and online localization pipeline [12] shown in Fig. 9. To generate a map, the mapping pipeline first detects SURF features and matches in the provided set of images before using incremental bundle adjustment (BA) to optimize both the camera poses and 3D positions of the matched features. It then registers the map to the ISS world coordinate frame using a 3D ISS model as shown in Fig. 1. The accuracy of our ISS maps is about 9 cm on average (for more detailed quantitative metrics and analysis, refer to the supplementary materials). For online use, the pipeline rebuilds the resulting SURF map as a BRISK map by detecting BRISK features and re-running BA with the camera poses fixed before building a hierarchical vocabulary tree using the BRISK features for fast image matching. BRISK features are less accurate but faster to compute and enable the localizer to more quickly detect and match features between images during live activities.

Pseudo ground truth is generated for new data using the same procedure used for updating existing maps. We first create a SURF map for the new data and merge it with an existing map by matching new SURF feature detections to existing ones before re-running BA with the existing map's camera poses fixed. After running BA, the resulting registered poses for the new images are then used as ground truth. Since the new data is downsampled before being added to remove low movement and in-place rotation sequences that often lead to errors during BA, we also use our localization pipeline to match downsampled images to the updated ground truth map, relying on P3P with RANSAC with further motion-only BA using the inlier features from RANSAC. Fig 10 shows a highaccuracy dense reconstruction with COLMAP [23] with the ground-truth 6-DoF camera poses we generated.

IV. Visual Navigation Benchmark

We benchmark six state-of-the-art visual odometry (VO) and SLAM algorithms on the Astrobee datasets. We compare both deep learning and classical VO/SLAM methods with AstroLoc [10], which is the visual-inertial localizer currently employed in the Astrobee robots inside the ISS.

A. Evaluated Methods

AstroLoc [10], a graph-based monocular visual-inertial localizer that runs on the limited computing platform of Astrobee on the ISS, plays a key role as a baseline in the visual navigation benchmark. AstroLoc [10] in VIO mode only uses VIO information, while localization mode also uses map-based measurements. We choose ORB-SLAM3 [4] and DSO [5], representative methods of geometry-based classical VO/SLAM. ORB-SLAM3 [4] is one of the most representative feature-based visual SLAM methods, building a map and localizing 6-DoF camera poses in real-time based on ORB features. DSO [5] is one of the representative direct VO/SLAM approaches that utilize image intensity gradients such as image edges. For the deep learning-based VO/SLAM approaches, we select DROID-SLAM [7], DF-VO [24], and SC-SfMLearner [25]. We run all VO/SLAM methods made publicly available by the original authors from their official GitHub pages with default (initial) parameter settings. Note that AstroLoc is carefully tuned for the Astrobee through a parameter sweep to perform its mission. To make a fair comparison of the motion tracking performance of each SLAM method, we disable loop closing and relocalization steps in ORB-SLAM3 [4] and DROID-SLAM [7]. NLC denotes 'no loop closing' in Table III. Monocular image sequences from NavCam are used as input for the evaluation, and we utilize the pre-trained models for each data-driven approach on the official GitHub pages.

B. Evaluation Metrics

We employ the Umeyama algorithm [26] to align the estimated and ground-truth trajectories, following [1], [27]. Accordingly, two aligned trajectories share the same scale and starting points. Given the ground-truth and estimated positions that are associated with the timestamps, we compute the absolute trajectory error (ATE) [1] and the absolute rotation error (ARE) [28]. To evaluate the overall accuracy of the estimated trajectories, we compute the root mean square of the absolute trajectory error and the absolute rotation error. Furthermore, we utilize success rate (SR) [29], the percentage of successfully localized images within a certain threshold (0.3m, 5°) with respect to the ground-truth, which is very effective for evaluating the success/failure ratio.

C. Discussion

Table III quantitatively shows the results of six state-of-the-art visual odometry and SLAM methods on the Astrobee ISS datasets. For each sequence, we highlight the smallest error in bold, and the second best is underlined. We also visualize the trajectory estimation results of each VO/SLAM approach compared to the ground-truth as shown in Fig 11.

First, we observe that classical geometry-based methods such as ORB-SLAM3 and AstroLoc outperform the deep learning-based VO/SLAM approaches. In the classical geometry-based methods, they are accurate in the order of ORB-SLAM3, AstroLoc, and DSO, and their ATE values are 0.119 m, 0.139 m, and 0.196 m on average, respectively. On the other hand, the deep learning approaches show inaccurate 6-DoF positioning results overall, except for DROID-SLAM results in some sequences. The average ATE of the DROID-SLAM is 0.469 m, whereas DF-VO and SC-SfMLearner are 1.433 m and 2.896 m, respectively. Overall, ORB-SLAM3 with default parameters and the fine-tuned AstroLoc are the first and second most stable on the ISS for most Astrobee sequences with an accuracy of less than 0.3 m error. DROID-SLAM demonstrates superior accuracy by a large gap from the existing deep learning-based VO/SLAM approaches, and sometimes achieves the most accurate results among six SOTA methods while it depends on heavy GPU computation. DF-VO and SC-SfMLearner show inaccurate and inconsistent 6-DoF motion tracking results due to poor generalization of deep learning-based approaches.

We report the absolute rotation error (ARE) of each VO/SLAM method in Table IV because of the importance of rotational motion tracking in VO/SLAM [30]. Similar to Table III, we observe that classical geometry-based methods outperform deep learning-based approaches. The average ARE values of ORB-SLAM3, AstroLoc, and DSO are 0.605°, 0.609°, and 1.277°, respectively. On the other hand, deep learning approaches generally show inaccurate rotational results, except for DROID-SLAM in some sequences. The average ARE value of DROID-SLAM is 0.523°, while DF-VO and SC-SfMLearner exhibit much higher error values of 2.131° and 24.761°. Overall, DROID-SLAM, ORB-SLAM3, and AstroLoc demonstrate the highest stability for rotational motion tracking on the ISS, with an accuracy of less than 1° error. DROID-SLAM exhibits remarkable accuracy, surpassing the other DL-based methods by a significant margin.

When comparing across SLAM methods with the relocalization step disabled, ORB-SLAM3 is the most accurate, followed by DROID-SLAM and AstroLoc, and their average ATE values are 0.131 m, 0.485 m, and 0.595 m. The average ARE values of ORB-SLAM3, DROID-SLAM, and AstroLoc are 0.425°, 0.548°, and 1.036°, respectively. AstroLoc in VIO mode clearly has lower accuracy when operating without the pre-built ISS maps because it limits factors and optimization times to perform graph-based optimization on Astrobee's limited computing power. ORB-SLAM3, based on the ORB feature, shows the most accurate and consistent estimation results regardless of the relocalization step.

Due to the challenging scenarios presented in the Astrobee datasets, we report SR to evaluate the overall success ratio of visual navigation in Table V. The order of SR aligns consistently with the ATE and ARE results, indicating that classical geometry-based methods outperform deep learning-

TABLE III
EVALUATION RESULTS OF ATE RMSE (UNIT: M) ON ASTROBEE DATASETS

Sequence	ff_return_journey_forward	ff_return_journey_up	ff_return_journey_left	ff_return_journey_rot	iva_kibo_trans	iva_kibo_rot	iva_ARtag
AstroLoc	**0.283**	0.066	**0.151**	0.270	0.122	**0.003**	0.080
AstroLoc (VIO mode)	1.101	0.212	0.465	0.311	0.925	1.084	0.068
ORB-SLAM3	0.432	**0.025**	0.229	0.103	0.014	0.019	0.013
ORB-SLAM3 (NLC)	0.405	0.056	0.333	0.068	0.014	0.025	0.017
DSO	0.552	0.102	0.293	0.132	0.104	0.078	0.114
DROID-SLAM	0.561	0.026	0.227	**0.014**	0.013	2.436	**0.005**
DROID-SLAM (NLC)	0.497	0.094	0.323	0.058	**0.011**	2.408	0.006
DF-VO	0.756	3.775	2.788	0.859	0.678	0.840	0.332
SC-SfMLearner	1.745	2.440	4.607	3.527	3.679	3.608	0.663

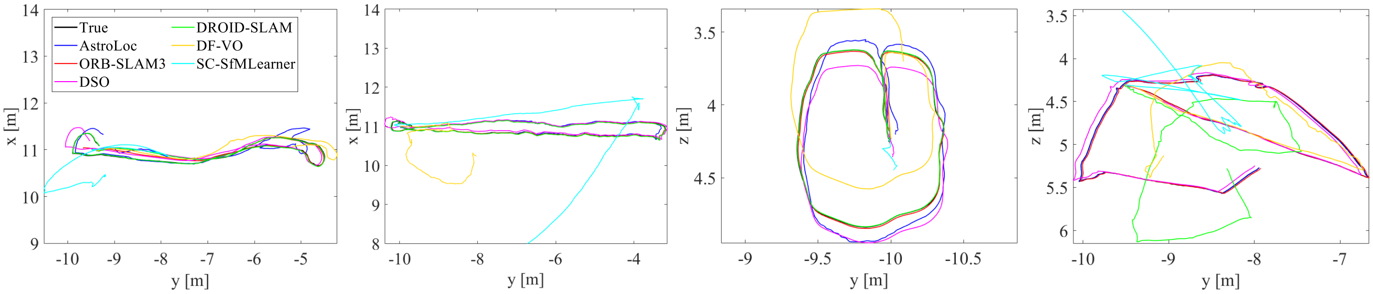


Fig. 11. Estimated trajectories of the evaluated visual navigation algorithms on free flight and intra-vehicular activity sequences from our Astrobee datasets. True (black) with our ground-truth generation is treated as ground-truth up to scale. Classical geometry-based approaches (AstroLoc, ORB-SLAM3, and DSO) show more accurate and stable trajectory estimation results than deep learning-based methods (DROID-SLAM, DF-VO, and SC-SfMLearner).

TABLE IV
EVALUATION RESULTS OF ARE RMSE (UNIT: DEGREE)

Sequence	td_roll	td_pitch	td_yaw
AstroLoc	0.546	0.401	**0.879**
AstroLoc (VIO mode)	1.439	0.790	**0.879**
ORB-SLAM3	**0.034**	0.149	1.632
ORB-SLAM3 (NLC)	0.044	**0.126**	1.105
DSO	2.080	0.145	1.609
DROID-SLAM	0.083	0.137	1.350
DROID-SLAM (NLC)	0.090	0.137	1.416
DF-VO	1.327	3.080	1.987
SC-SFM	21.325	23.364	29.595

based methods. Among the classical geometry-based methods, the average SR values of ORB-SLAM3, AstroLoc, and DSO are 0.94, 0.94, and 0.93, respectively. In deep learning-based approaches, DROID-SLAM exhibits a superior success ratio over the others, with an average SR value of 0.84, while DF-VO and SC-SfMLearner show much lower average SR values of 0.44 and 0.10, respectively.

We test VO and SLAM algorithms on the desktop computer using an AMD Ryzen 9 5950X 16-Core Processor operating at 3.4GHz with 32GB of RAM, which is about 20 times faster than Astrobee's hardware [10]. The desktop computer is also equipped with a graphic card with NVIDIA GeForce RTX 3080 to run deep learning-based methods. We report the average run times of evaluated methods in Table VI. AstroLoc is significantly faster than ORB-SLAM3 and DSO while showing accurate and precise positioning accuracy similar to the performance of ORB-SLAM3. ORB-SLAM3, which is the most accurate geometric SLAM method, runs about five times slower than AstroLoc although it includes not only visual odometry and localization like AstroLoc but also the mapping process. DROID-SLAM is the most efficient among the deep learning-based approaches without the cost of sacrificing localization accuracy. For the Astrobee free-flyers, which do not have a powerful GPU and have limited computing power, AstroLoc is currently the most appropriate visual navigation method as it shows accurate motion tracking performance similar to ORB-SLAM3 and operates much faster.

V. CONCLUSION

We present the first annotated benchmark datasets for evaluating free-flyer visual navigation algorithms in a zero-g spacecraft interior. Our datasets were captured with the Astrobee free-flyers operating inside the International Space Station (ISS). These datasets capture a wide range of interesting scenarios such as free flights and intra-vehicular activities in challenging environments like the absence of gravity, severe lighting changes, environmental changes between ISS modules, and occlusions from unorganized cargo bags. The proposed Astrobee datasets contain rectified grayscale images, IMU measurements in microgravity, and pseudo ground truth 6-DoF camera poses, etc. Furthermore, we benchmark state-of-the-art visual navigation methods, including the currently operating Astrobee localizer [10] on the ISS. We hope the Astrobee dataset will inspire new research directions in visual navigation, enabling a wide range of intra-vehicular activities and tasks in the upcoming space age for VIO and SLAM-based methods. In future work, we plan to enrich the datasets by adding more scenarios such as object detection and handrail perception, and release time-synchronized depth images from other cameras in the Astrobee free-flyers such as HazCam, PerchCam, DockCam, and SciCam.

TABLE V
EVALUATION RESULTS OF SUCCESS RATE (SR) ON ASTROBEE DATASETS

Sequence	ff_return_journey_forward	ff_return_journey_up	ff_return_journey_left	ff_return_journey_rot	iva_kibo_trans	iva_kibo_rot	iva_ARtag	td_roll	td_pitch	td_yaw
AstroLoc	**0.76**	**1.00**	**0.89**	0.75	0.97	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
AstroLoc (VIO mode)	0.01	0.83	0.64	0.65	0.03	0.03	**1.00**	**1.00**	**1.00**	**1.00**
ORB-SLAM3	0.65	**1.00**	0.78	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ORB-SLAM3 (NLC)	0.67	**1.00**	0.71	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
DSO	0.57	**1.00**	0.73	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
DROID-SLAM	0.61	**1.00**	0.78	**1.00**	**1.00**	0.01	**1.00**	**1.00**	**1.00**	**1.00**
DROID-SLAM (NLC)	0.61	**1.00**	0.71	**1.00**	**1.00**	0.02	**1.00**	**1.00**	**1.00**	**1.00**
DF-VO	0.02	0.06	0.16	0.17	0.22	0.38	0.53	**1.00**	0.90	**1.00**
SC-SfMLearner	0.05	0.03	0.06	0.10	0.05	0.03	0.14	0.13	0.15	0.23

TABLE VI
COMPUTATIONAL TIME COMPARISON ON ASTROBEE DATASET

Localizer	Avg. Runtime [ms]
AstroLoc	**52.37**
ORB-SLAM3	258.24
DSO	77.20
DROID-SLAM	136.64
DF-VO	540
SC-SFMLearner	77.56

ACKNOWLEDGEMENTS

We thank the ISS program and astronauts for their support of Astrobee operations. This work was funded in part by the Game Changing Development Program in the NASA Space Technology Mission Directorate. This work was authored by employees of KBR Wyle Services, LLC under Contract No. 80ARC020D0010 with the National Aeronautics and Space Administration. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, or allow others to do so, for United States Government purposes. All other rights are reserved by the copyright owner. Pyojin Kim was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1F1A1061397).

REFERENCES

[1] J. Sturm, N. Engelhard, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of IROS*, 2012.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012.

[3] M. Burri, J. Nikolic, P. Gohl, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *IJRR*, 2016.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE T-RO*, 2021.

[5] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[6] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE T-RO*, 2018.

[7] Z. Teed and J. Deng, "DROID-SLAM: Deep visual slam for monocular, stereo, and rgb-d cameras," *NeurIPS*, 2021.

[8] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, and E. Smith, "Astrobee: A new platform for free-flying robotics on the international space station," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*, 2016.

[9] S. Mitani, M. Goto, R. Konomura, Y. Shoji, K. Hagiwara, S. Shigeto, and N. Tanishima, "Int-ball: Crew-supportive autonomous mobile camera robot on iss/jem," in *IEEE Aerospace Conference*, 2019.

[10] R. Soussan, V. Kumar, B. Coltin, and T. Smith, "Astroloc: An efficient and robust localizer for a free-flying robot," in *IEEE ICRA*, 2022.

[11] R. Carlino, J. Barlow, J. Benavides, M. Bualat, A. Katterhagen, Y. Kim, R. G. Ruiz, T. Smith, and A. M. Vargas, "Astrobee free flyers: Integrated and tested. ready for launch!" in *International Astronautical Congress 2019*, no. ARC-E-DAA-TN73951, 2019.

[12] B. Coltin, J. Fusco, Z. Moratto, O. Alexandrov, and R. Nakamura, "Localization from visual landmarks on a free-flying robot," in *IEEE IROS*, 2016.

[13] Y. Liu, L. Li, M. Ceccarelli, H. Li, Q. Huang, and X. Wang, "Design and testing of bit flying robot," in *ROMANSY 23-Robot Design, Dynamics and Control: Proceedings of the 23rd CISM IFToMM Symposium 23*. Springer, 2021.

[14] S. Nolet, "The SPHERES navigation system: from early development to on-orbit testing," in *Proc. of AIAA Guidance, Navigation and Control Conf.*, 2007, p. 6354.

[15] "CIMON: astronaut assistance system," https://www.dlr.de/en/research-and-transfer/projects-and-missions/horizons/cimon, Accessed: 2024-02-08.

[16] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *IEEE ICRA*, 2007.

[17] I. D. Miller, R. Soussan, B. Coltin, T. Smith, and V. Kumar, "Robust semantic mapping and localization on a free-flying robot in microgravity," in *IEEE ICRA*, 2022.

[18] P. Kim, B. Coltin, O. Alexandrov, and H. J. Kim, "Robust visual localization in changing lighting conditions," in *IEEE ICRA*, 2017.

[19] L. Fluckiger and B. Coltin, "Astrobee robot software: Enabling mobile autonomy on the iss," Tech. Rep., 2019.

[20] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE IROS*, 2013.

[21] T. Conceição, B. Coltin, T. Smith, A. Symington, and L. Flückiger, "Joint visual and time-of-flight camera calibration for an automatic procedure in space," in *Proc. Int. Symp. Artif. Intell., Robot. Automat. Space (i-SAIRAS)*, 2018.

[22] F. Ddvernay and O. Faugeras, "Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments," *Mach. Vis. Appl.*, 2008.

[23] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016.

[24] H. Zhan, C. S. Weerasekera, J.-W. Bian, R. Garg, and I. Reid, "Df-vo: What should be learnt for visual odometry?" 2021.

[25] J.-W. Bian, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth learning from video," *IJCV*, 2021.

[26] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE T-PAMI*, 1991.

[27] H. Li, J. Zhao, J.-C. Bazin, P. Kim, K. Joo, Z. Zhao, and Y.-H. Liu, "Hong kong world: Leveraging structural regularity for line-based slam," *IEEE T-PAMI*, 2023.

[28] P. Kim, B. Coltin, and H. J. Kim, "Visual odometry with drift-free rotation estimation using indoor scene regularities." in *BMVC*, 2017.

[29] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual slam," in *IEEE IROS*, 2020.

[30] P. Kim, B. Coltin, and H. J. Kim, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *IEEE ICRA*, 2018.