Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Boxi: Design Decisions in the Context of Algorithmic Performance for Robotics

Jonas Frey[⋆,1,3] Turcan Tuna[⋆,1] Lanke Frank Tarimo Fu[⋆,2] Cedric Weibel[1] Katharine Patterson[1]
Benjamin Krummenacher[1] Matthias Müller[1] Julian Nubert[1,3] Maurice Fallon[2] Cesar Cadena[1] Marco Hutter[1]
[1] ETH Zurich [2] University of Oxford [3] Max Planck Institute for Intelligent Systems
[⋆] Equal contribution. Contact: jonfrey@ethz.ch

Fig. 1: *Boxi* can be deployed handheld, on a legged robot, or a wheeled robot across a variety of environments.

***Abstract*—Achieving robust autonomy in mobile robots operating in complex and unstructured environments requires a multimodal sensor suite capable of capturing diverse and complementary information. However, designing such a sensor suite involves multiple critical design decisions, such as sensor selection, component placement, thermal and power limitations, compute requirements, networking, synchronization, and calibration. While the importance of these key aspects is widely recognized, they are often overlooked in academia or retained as proprietary knowledge within large corporations. To improve this situation, we present *Boxi*, a tightly integrated sensor payload that enables robust autonomy of robots in the wild. This paper discusses the impact of payload design decisions made to optimize algorithmic performance for downstream tasks, specifically focusing on state estimation and mapping. *Boxi* is equipped with a variety of sensors: two LiDARs, 10 RGB cameras including high-dynamic range, global shutter, and rolling shutter models, an RGB-D camera, 7 inertial measurement units (IMUs) of varying precision, and a dual antenna RTK GNSS system [1]. Our analysis shows that time synchronization, calibration, and sensor modality have a crucial impact on the state estimation performance. We frame this analysis in the context of cost considerations and environment-specific challenges. We also present a mobile sensor suite ‘cookbook‘ to serve as a comprehensive guideline, highlighting generalizable key design considerations and lessons learned during the development of *Boxi*. Finally, we demonstrate the versatility of *Boxi* being used in a variety of applications in real-world scenarios, contributing to robust autonomy. More details and code: https://github.com/leggedrobotics/grand_tour_box**

I. Introduction

Designing a capable perception payload is fundamental to the success of mobile robotics systems. The performance of perception algorithms is inherently upper bounded by the quality of the input sensor data, which might be corrupted by aleatoric sensor noise and epistemic design issues, regardless of the applied algorithm. Therefore, it is key to select the appropriate sensors while considering time synchronization, sensor calibration, latency, communication, thermal compensation, and hardware design decisions to enable the next generation of mobile robots to operate reliably in complex environments. Previous studies have emphasized the critical role of time synchronization [75, 70, 7], rolling shutter correction [62], SLAM performance under various conditions [45, 76], and accurate intrinsic and extrinsic calibration [56, 26, 16, 30, 44] in determining downstream task performance.

To this end, we introduce *Boxi*, a compact, standalone sensing payload designed to enable the development of autonomy solutions for mobile robots, with a particular focus on quadrupedal legged robots. *Boxi* integrates best practices in mechanical, electrical, and software engineering, as well as system design, and was co-developed in collaboration with Leica Geosystems. We leverage *Boxi* to establish a direct link between design decisions and algorithmic performance on downstream tasks such as robot localization.

System Name	Cameras			Depth		IMUs			Other Sensors	GNSS Solution	Reference	Sync	Robot
	GS	RS	HDR	Stereo	LiDAR	Nav.	Tact.	Co.					
SubT-MRS[77]	4x ✔	✘	✘	✘	2x ✔	✘	✘	2x ✔	FLIR Boson	RTK	Scanner	HW, SW	Versatile
CatPack[33, 54]	✘	4x ✔	✘	✘	1x ✔	✘	✘	1x ✔	FLIR Lepton 3.5 thermal	✘	-	HW, SW	Legged
VersaVIS [70]	2x ✔	2x ✔	✘	1x✔	1x ✔	✘	✘	3x ✔	Thermal	✘	-	HW	Versatile
Shepard's Stick[35]	1x ✔	2x ✔	✘	1x ✔	1x ✔	✘	✘	2x ✔	-	✘	-	SW	Handheld
FusionPortable[36]	2x ✔	2x ✔	✘	✘	1x ✔	✘	1x ✔	3x ✔	2x Event Camera	RTK	RTK/MOCAP/Scanner	SW	Versitale
Phasma[31]	5x ✔	1x ✔	✘	✘	1x ✔	✘	✘	1x ✔	-	✘	Scanner	HW, FPGA	Hilti
VBR[6]	2x ✔	✘	✘	✘	1x ✔	✘	1x ✔	1x ✔	-	RTK	Scanner / RTK	HW, SW	Rome
Botanic[44]	4x ✔	✘	✘	✘	2x ✔	✘	1x ✔	2x ✔		✘	Scanner	HW (STM32 MCU)	Scout V1.0
PIRVS [76]	2x ✔	✘	✘	✘	✘	✘	✘	1x ✔	-	✘	-	HW	Versatile
(Industrial) BLK2Go [40]	3x ✔	1x ✔	✘	✘	1x ✔	✘	✘	1x ✔	-	✘	no	HW	Handheld
WHU-Helmet[42]	✘	✘	✘	✘	1x ✔	1x ✔	✘	✘	-	RTK	GNSS	SW	Helmet
EverySync [75]	1x ✔	✘	✘	✘	1x ✔	✘	✘	1x ✔	-	RTK	RTK	HW, SW	Wheeled Robot
S3E [24]	✘	2x ✔	✘	✘	1x ✔	✘	✘	1x ✔	UWB	RTK	INS / MOCAP	HW	Wheeled Robot
TartanDrive [68, 61]	2x ✔	✘	✘	1x ✔	3x ✔	✘	1x ✔	1x ✔	-	RTK	RTK, INS	SW	Offroad Car
MCD [47]	2x ✔	✘	✘	1x ✔	2x ✔	✘	✘	3x ✔	UWB	✘	Scanner	SW	Handheld / Wheeled
Boxi (Ours)	3x ✔	5x ✔	3x ✔	1x ✔	2x ✔	1x ✔	2x ✔	4x ✔	1x Barometer	RTK	INS, RTS, GNSS	HW, SW	Legged Robot

* IMUs are categorized into Navigation (Nav.), Tactical (Tac.), and Industrial & Consumer (Co.) grades. Cameras are categorized into global shutter (GS), rolling shutter (RS), and high-dynamic range (HDR).

TABLE I: Overview of Existing Sensor Payloads. Symbols: ✔= present, ✘= absent.

The design of *Boxi* is driven by four primary objectives: (1) developing a state-of-the-art mobile robot sensor suite, (2) fostering a better understanding between payload design decisions and algorithmic performance, (3) allowing the collection of large scale multi-sensor datasets with high-quality calibration and time synchronization and (4) enabling autonomous mobile navigation. Based on these objectives, we equipped *Boxi* with accessible state-of-the-art sensors featuring two spinning LiDARs (repetitive and non-repetitive), 10 cameras (high-dynamic range, global shutter, and rolling shutter models), a stereo-based depth camera, 7 inertial measurement units (IMUs) of varying precision and cost (5 USD to 12 000 USD), as well as a dual antenna RTK-GNSS system [1] and tight integration to Leica Geosystems MS60 Total Station (Reflector Prism & Receiver Link). Compared to autonomous driving payloads, *Boxi* is more compact and lightweight, but shares similar design requirements provided in [28]. It is based on the principle that payload design must consider algorithmic performance [76]. To support and extend this notion to general robotics, we present a sensory payload *'cookbook'* that summarizes design decisions and lessons learned, enabling the development of future perception systems beyond *Boxi*.

Further, we showcase how *Boxi*, being *SLAM-ready* [76], in addition, can be classified as *autonomy-ready* by performing onboard state estimation, mapping, and navigation. At the core of this work, we use *Boxi* to conduct a rigorous study to investigate the state estimation performance across multiple dimensions (e.g., accuracy, robustness to time, and extrinsic calibration) with respect to hardware and software design decisions in 7 different environments.

We begin by analyzing the performance of different sensor modalities using popular and state-of-the-art approaches for each modality (LiDAR, Camera, Proprioception, and GNSS). We then compare different camera types (high dynamic range, global shutter, and rolling shutter) in terms of image quality and Visual Interial Odometry (VIO) performance. In addition, we compare the performance impact of IMUs (5 USD to 12 000 USD) on dead-reckoning and LiDAR Interial Odometry (LIO) performance. Lastly, we simulate time synchronization and extrinsic calibration errors to investigate the effect on state estimation. Through the steps described above, we will address the following research question: *Which sensors should be purchased given the mapping accuracy requirements, budget constraints, and deployment environment, and how important are extrinsic and intrinsic calibration, as well as time synchronization, in achieving the desired accuracy?* To foster further research, the full hardware and software design is made available open-source.

In summary, the main contributions of this work are:

- Design, development, and evaluation of *Boxi*, a versatile multi-modal sensor payload optimized for data collection on mobile robotic platforms.
- Ablation studies on the effect of modalities, time synchronization accuracy, and extrinsic calibration accuracy across multiple real-world environments.
- Compiling a comprehensive *cookbook*, featuring insights, lessons learned, and best practices for designing payloads for robotic platforms.

II. Related Works

A. Payloads for Autonomous Driving

The development of mobile mapping systems initially focused on integration into wheeled platforms due to their capacity to accommodate heavy sensors and their ability to drive long distances [28]. These systems often incorporated a combination of cameras, LiDAR sensors, IMU, and GNSS—a sensor suite that continues to form the foundation of our work. One of the milestones in the development of autonomous driving was marked by the introduction of the KITTI dataset [28], which featured comprehensive data from cameras, LiDAR, IMU, and RTK GNSS, and which continues to be relevant in the field today. Subsequent data collection platforms focused on semantic understanding [15] and temporal changes [46]. While the previous works were primarily driven by academia, multiple industry-driven sensory payloads were used to release high-quality datasets, such as NuScenes [11], Waymo [63], and Argoverse [13]. These datasets often feature similar sensor configurations; however, detailed information on calibration, time synchronization, and sensor payload design is not publicly available. More recent efforts have introduced modern sensors such as 360° Field of View (FoV) radar [9] or event cameras [27].

B. Compact Vision Payloads

Within the last decade, a variety of more mobile and lightweight payloads based on multi-camera setups were intro-

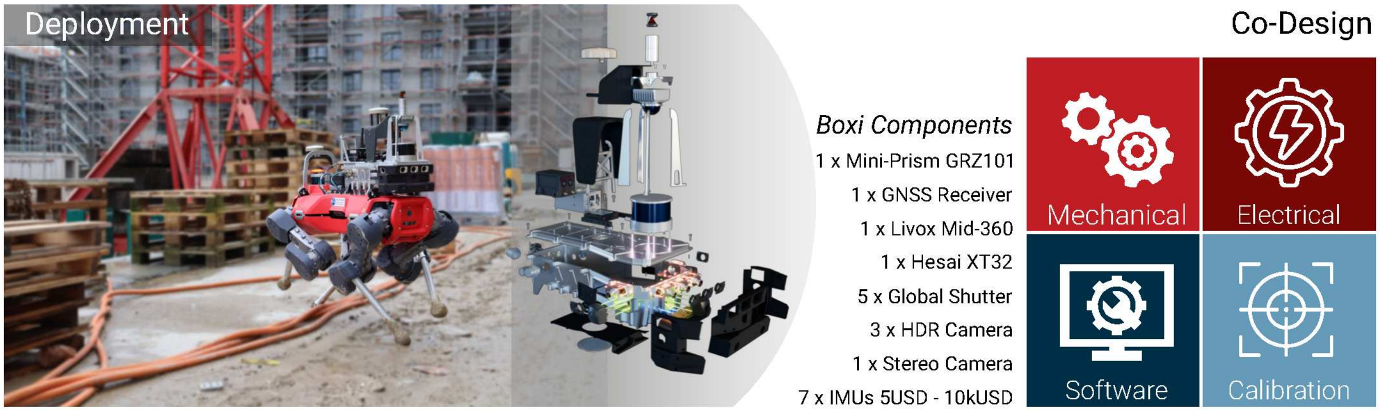

Fig. 2: Co-design of hardware, electrical systems, software, and calibration is essential for developing a robust sensor payload.

duced. The authors in [58] introduced the DLR 7, a stereo camera system with FPGA-based time synchronization, enabling handheld operation at just 800 g. Similarly, Nikolic et al. [48] developed an ultra-lightweight unit integrating stereo cameras and an IMU with FPGA-based synchronization and exposure time compensation, tailored for online visual feature extraction. Zhang et al. [76] presented the PIRVS system, which directly integrated the SLAM algorithms on-board, while considering the latency between the image exposure period and CPU processing. Expanding on these designs, Tschopp et al. [70] introduced VersaVIS, an open, versatile multi-camera and IMU sensor suite with precise hardware synchronization, which later evolved into the commercial Sevensense CoreResearch [59]. Although all previous work highlights the importance of calibration and time synchronization and includes FPGAs or ISPs for processing or triggering, no systematic experimental evaluation of their impact on state estimation performance is provided.

1) Compact Multi-Modal Sensor Payloads: The development of more compact multi-modal sensor suites has been of interest in industry and research [64, 75, 54] driven by their promising potential to enhance precision and resilience in challenging environments. An industrial example is the Leica Geosystems BLK2Go [21], a tightly integrated handheld scanning device, which is equipped with three global shutter cameras, an IMU, and an 830 nm laser scanner (420k points/s) that provides measurements up to 3 mm accuracy.

EverySync [75] is a multi-sensor payload focusing on time synchronization accuracy. EverySync relies on the methods developed in VersaVIS [70] to perform triggering and time synchronization. Moreover, the authors of EverySync provided an extensive analysis of accurate time synchronization and achieved a time synchronization accuracy of 1 ms between GNSS, visual, and LiDAR sensors. Similarly, LiDAR-Visual and Inertial payloads [31, 54, 35] showed the importance of time synchronization and extrinsic calibration between modalities for SLAM and state estimation purposes. Liu et al. [44] integrated and demonstrated a multi-LiDAR, multi-camera, and multi-IMU setup with precise time synchronization up to 1 µs using Precision Time Protocol (PTP) and a self-designed STM32 MCU triggering board. Recently, Nguyen et al. [47] integrated a sensor payload and proposed using a continuous time B-spline trajectory representation for reference pose estimation. They showed that this formulation mitigates the impact of time synchronization errors for SLAM and state estimation evaluation.

III. Overview

A. How to read the paper

This paper is structured as follows. Section III-B details *Boxi*'s key specifications, providing the foundation for the algorithmic performance analysis provided in Section IV, which is self-contained and highlights the most relevant experiments and findings. Finally, Section V serves as a comprehensive *cookbook*, describing the design of *Boxi* and presenting key lessons learned as well as generalizable findings for the design of future sensor payloads.

B. Boxi Overview

Boxi is designed to enable autonomy and provide a comprehensive understanding of the payload design decisions on downstream task performance. This requires the integration of different sensing modalities and sensor types. Therefore, *Boxi* includes exteroceptive sensors such as two LiDARs and a stereo camera setup for geometric perception. In addition, the payload is equipped with high-dynamic range, global shutter, and rolling shutter cameras for visual perception, and various tiers of IMUs for inertial sensing. *Boxi* provides time synchronization accuracy ranging between 5 ms to 10 µs (sensor-dependent Section V-D2), and accurate extrinsic calibrations (cam-to-cam $0.3\,\mathrm{mm}@1\sigma$, cam-to-LiDAR $1.0\,\mathrm{mm}$ corner detection accuracy) between all cameras and LiDARs. Precise extrinsic calibration is achieved by CNC machining *Boxi* out of an aluminum mono-block with a manufacturing precision of $0.05\,\mathrm{mm}$ in combination with state-of-the-art calibration algorithms (customized version of Kalibr [56, 26], IMU intrinsic [8], LiDAR extrinsic [25]).

The cameras and LiDARs on *Boxi* are carefully positioned to capture the near-field surroundings of the robot, which is crucial for locomotion and navigation as opposed to a reality capture setup. The complete sensor payload weighs 7.1 kg, making it

	Name	Description	Power	est. Price
LiDARs				
(Livox)	Livox Mid-360	vFoV: 59°, Range: 0.1 m, 40 m, 10 Hz, Accuracy: ±0.02 m	6.5 W (avg)	749 USD
(Hesai)	Hesai XT-32	vFoV: 31°, Range: 0.05 m, 120 m, 10 Hz, Accuracy: ±0.01 m	10 W (avg)	3000 USD
Cameras				
(CoreResearch)	Sevensense CoreResearch	1440×1080 @ 10 fps, FoV: 126°×92.4°, 71.6 dB, GS, 1.6 MP	12 W (max)	2000 USD
(HDR)	TierIV C1	1920×1280 @ 30 fps, FoV: 120°×80.0°, 120 dB, RS, 2.5 MP	1.7 W (max)	900 USD
(ZED2i)	Stereolabs ZED2i	1920×1080 @ 30 fps, FoV: 126°×92.4°, 64.6 dB, RS, 4 MP	2 W (max)	500 USD
Reference				
(AP20)*	Modified Leica AP20	MS60 Total Station Bluetooth Communication & Synchronization	—	5000 USD
(Prism)	Mini-Prism GRZ101	Static Accuracy ±2 mm (3σ), hFoV 360°, vFoV ±30°	—	1000 USD
(CPT7)	Novatel Span CPT7	Dual RTK GPS, Intertial Explorer, TerraStar	18 W(max)	40 000 USD
	3GNSSA-XT-1	Multiband GPS Antenna [BeiDou, L-Band, GLONASS, GPS, Galileo, Navic]	1 W (each)	
IMUs				
(HG4930)	Honeywell HG4930	100 Hz, ±400 ° s^{-1}, ±20 g_{grav}, GBI 0.25 ° h^{-1}, ABI 0.025 mg	2 W (max)	12 000 USD
(STIM320)	Safran STIM320	500 Hz, ±400 ° s^{-1}, ±5 g_{grav}, GBI 0.3 ° h^{-1},	2 W (max)	8000 USD
(AP20-IMU)	Proprietary IMU (AP20)	200 Hz	—	Mid-Range
(ADIS)	Analog Devices ADIS16475-2	200 Hz, ±500 ° s^{-1}, ±8 g_{grav}, GBI 2.5 ° h^{-1}, ABI 0.0036 mg	—	500 USD
(Bosch)	Bosch BMI085 (CoreResearch)	200 Hz, ±2000 ° s^{-1}, ±16 g_{grav}	—	5 USD
(TDK)	TDK ICM40609 (Livox)	200 Hz, ±2000 ° s^{-1}, ±32 g_{grav}	—	5 USD
(ZED2i-IMU)	Unknown (ZED2i)	400 Hz, ±1000 ° s^{-1}, ±8 g_{grav}	—	—

TABLE II: *Boxi* Components: GS=GlobalShutter, RS=RollingShutter. Used camera settings are provided under their respective sections. ABI=Accelerometer Bias Instability, GBI=Gyro Bias Instability. * The (AP20) requires a Leica MS60 Total Station at a cost of 50 000 USD.

suitable for deployment with a harness or with adaptable mounting options that can be deployed on different robotic platforms. The sensor arrangement is optimized for near-field perception (stereo camera baseline of 12 cm) and long-range using a high-accuracy repetitive LiDAR. Throughout this work, each sensor will be referred to by its abbreviation outlined in Table II.

IV. Algorithmic Performance

A. Real-World Datasets

We evaluated our hardware and software decisions on data from seven real-world environments collected in the wild (Figure 3), which are part of the GrandTour dataset. All data were gathered by the legged robot ANYmal [34] equipped with *Boxi* and teleoperated by an expert human. The selected scenarios are diverse in lighting and include indoor and outdoor settings, confined spaces, wide open spaces, and both structured and unstructured scenes. The deployments take between 3 min-7 min and cover distances around 300 m and more details are available in Appendix A.

B. Algorithms

As a representative of multi-camera visual-inertial state estimation, we use OKVIS2 [41], which is configured to perform loop closures. Moreover, a customized version of the state-of-the-art LiDAR-inertial geometric-observer based odometry pipeline Direct LiDAR Inertial Odometry (DLIO) [14] is used to represent tightly coupled LiDAR-inertial methods. In addition, we use the kinematic-inertial estimator TSIF [5] to represent the performance of proprioceptive state-estimation methods. Alongside these modules, the post-processed GNSS-IMU solution of the Inertial Explorer [2] is provided as an industrial positioning solution for outdoor datasets. The summary of the algorithms is provided in Table III.

Algorithm	Modality	Description
OKVIS2 [41]	Visual-Inertial	Multi-Camera VIO with Loop Closure
DLIO [14]	LiDAR-Inertial	Tightly coupled Filter-based LIO
TSIF [5]	Kinematic-Inertial	Proprioceptive Filter-based
Inertial Explorer [2]	GNSS-Inertial	Industrial offline INS

TABLE III: Summary of the State Estimation Algorithms

C. State Estimation - Effect of Modality

Different sensors provide complementary information that allows a robot to operate in various environments that are challenging for one or more sensor modalities. The goal of this analysis is to present a rough estimate of how different modalities perform across diverse environments, helping to select the most suitable sensor for specific environments and state estimation requirements. Before interpreting the results, it is crucial to understand that the performance of each modality is highly dependent on the motion pattern (e.g., motion blur, saturation effects) and the environment (e.g., LiDAR degradation, smoke, symmetry, lighting, visual features, GNSS denial). It is not possible to provide fully generalized claims that apply to all possible scenarios, but we can deduce guiding principles.

We compare DLIO [14] (fuses data from Hesai and HG4930), OKVIS2 [41] (fuses data from CoreResearch-Stereo pair and HG4930), GNSS [2] (IE-TC more details in Appendix B) and TSIF [5]. Table IV and Table V presents

Mission Name	Kinematic [5]	LiDAR [14]	GNSS [2]	Camera [41]
Terrace	0.009 ± 0.016	0.02 ± 0.016	0.003 ± 0.003	0.033 ± 0.031
Research Station	0.007 ± 0.012	0.021 ± 0.017	0.008 ± 0.010	0.027 ± 0.046
Mountain Ascent	0.016 ± 0.040	0.015 ± 0.013	0.004 ± 0.003	0.025 ± 0.021
Hike	0.024 ± 0.031	0.023 ± 0.019	0.004 ± 0.003	0.017 ± 0.014
Excavation Site	0.014 ± 0.019	0.020 ± 0.014	0.003 ± 0.002	0.038 ± 0.044
Demolished Building	0.017 ± 0.019	0.015 ± 0.009	0.025 ± 0.040	0.031 ± 0.032
Warehouse	0.016 ± 0.047	0.024 ± 0.023	nan	0.045 ± 0.042
Average	**0.015**	0.02	**0.008**	0.031

TABLE IV: Modality Comparison in RTE per dataset.

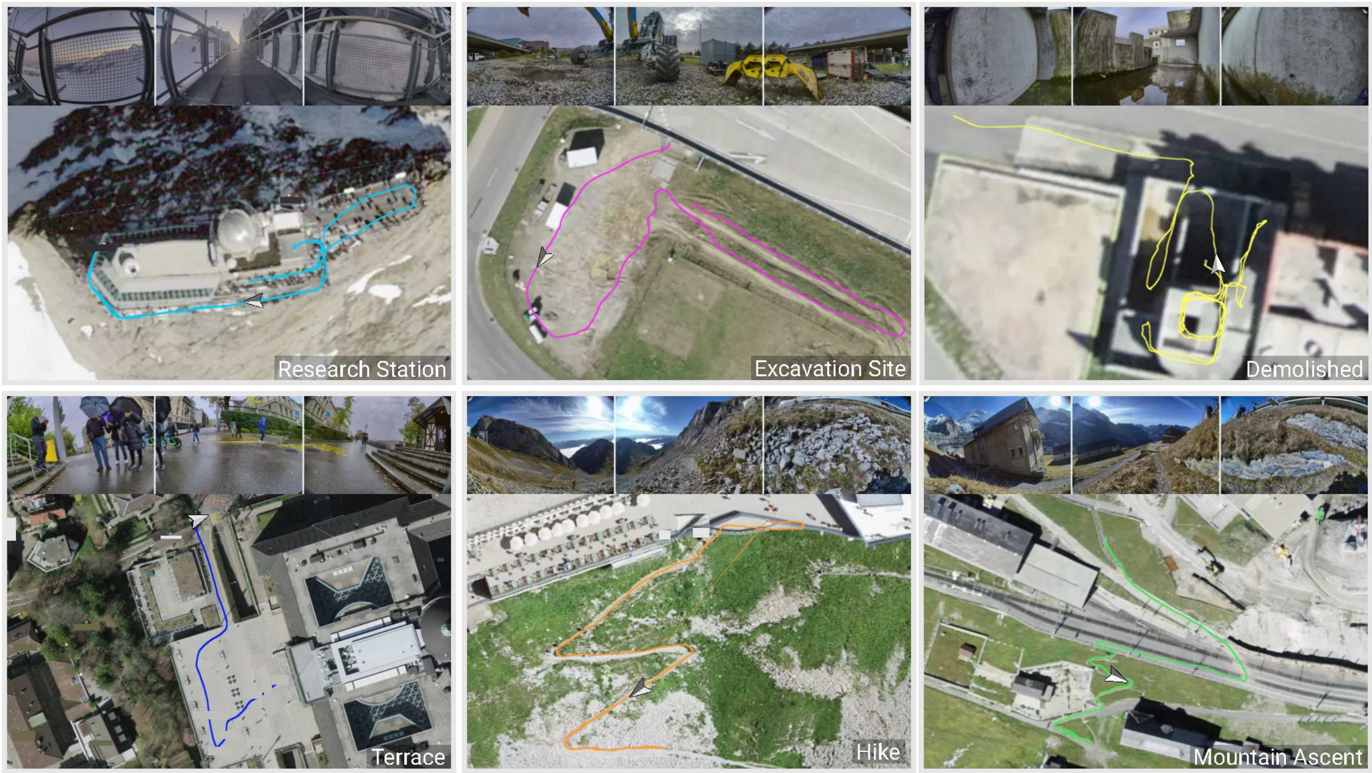

Fig. 3: Subset of the GrandTour Datasets Overview: (top-row) left, front, and right HDR camera images; (bottom-row) satellite image with projected GNSS path and camera image viewpoint; The warehouse dataset is not displayed.

Mission Name	Kinematic [5]	LiDAR [14]	GNSS [2]	Camera [41]
Terrace	0.309 ± 0.113	0.023 ± 0.015	0.01 ± 0.004	0.524 ± 0.208
Research Station	0.154 ± 0.091	0.025 ± 0.012	0.138 ± 0.08	0.198 ± 0.104
Mountain Ascent	0.589 ± 0.368	0.046 ± 0.019	0.016 ± 0.006	0.563 ± 0.184
Hike	1.353 ± 0.559	0.038 ± 0.019	0.059 ± 0.021	0.243 ± 0.105
Excavation Site	0.360 ± 0.284	0.021 ± 0.012	0.016 ± 0.007	0.372 ± 0.212
Demolished Building	0.266 ± 0.110	0.019 ± 0.013	0.113 ± 0.082	0.118 ± 0.064
Warehouse	0.429 ± 0.242	0.063 ± 0.047	nan	0.419 ± 0.180
Average	0.49 ± 0.40	**0.03 ± 0.02**	0.06 ± 0.06	0.35 ± 0.17

TABLE V: Modality Comparison in ATE per dataset.

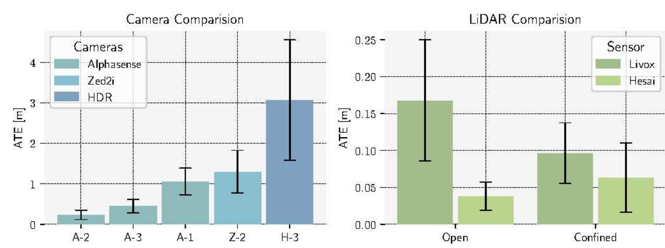

Fig. 4: Camera and LiDAR Comparison using OKVIS2 (HG4930) and DLIO (HG4930). (A-2, A-3, A-1, Z-2, H-3) indicates respective camera configuration (see Section IV-E). The Camera is evaluated on the *Mountain Ascend* dataset while the LiDAR comparison is provided on the *Hike* (Open) and *Warehouse* (Confided) datasets.

Relative Trajectory Error (RTE) and Absolute Trajectory Error (ATE) results respectively across all seven tested environments compared against the total station (TPS) measurements. DLIO, consistently outperforms OKVIS2 and kinematic-based state estimation in terms of ATE. With sufficient satellite coverage, the offline-optimized GNSS solution provides highly accurate performance, both locally (RTE) and globally (ATE). In terms of RTE, kinematic-based state estimation performs well, verifying its common use for control.

D. State Estimation - LiDAR Comparison

In the following, all Absolute Position Error (APE) and Relative Position Error (RPE) error metric computations are done with respect to our 6-Degrees of Freedom (DoF) ground truth pose estimate, which is obtained by fusing TPS, IMU, and Inertial Explorer post-processed poses (see Appendix B). We test how the Hesai, an accurate long-range LiDAR, compares against the lower-cost and lightweight Livox. For this comparison, we selected an open (*Hike*) and confined environment (*Warehouse*) and reported the results in Figure 4. The *Hike* dataset contains limited geometric features compared to the *Warehouse* dataset. Across both datasets, the Hesai outperforms the Livox. However, in the dataset with a large featureless space, the gap is more significant.

E. State Estimation - Camera Comparison

In this study, we compare the accuracy of different cameras for VIO using OKVIS2. We evaluated multiple different camera configurations:

- CoreResearch: (A-1) Single, (A-2) Stereo, (A-3) Triple
- HDR: (H-1) Single, (H-3) Triple
- ZED2i: (Z-1) Single, (Z-2) Stereo,

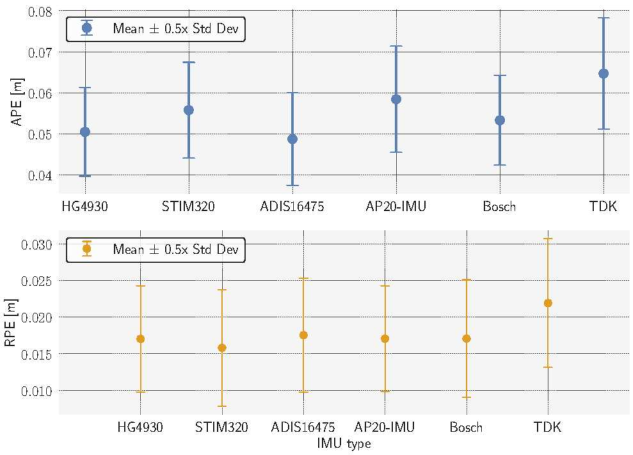

Fig. 5: Effect of different IMUs on the performance of DLIO module on the *Mountain Ascend* dataset. The generated APE and RPE errors are visualized.

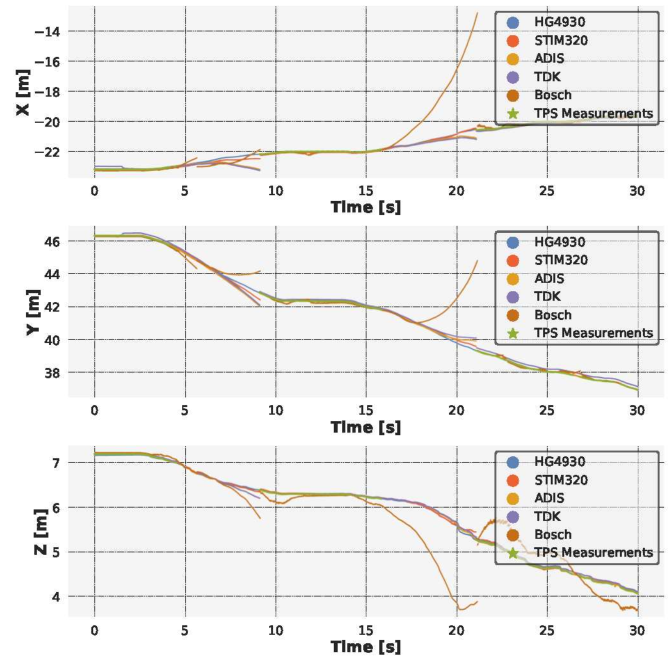

Fig. 6: IMU Dead-Reckoning during TPS measurement dropout (4.5 s-9 s and 17 s-22 s).

with the number representing the camera configuration. Detailed mounting and FoV specifications are available in Section V-C1. We evaluated the performance in the *Mountain Ascent* dataset (see Figure 4). Both the H-1 and Z-1 configurations diverged, highlighting challenges with rolling shutter cameras. The CoreResearch unit, utilizing global shutter cameras, consistently delivers the best accuracy across all environments. For the HDR image, we provide OKVIS2 with the timestamp of the center line in the middle of the exposure period; however, the first line is captured approximately 15 ms earlier, while the last line is captured 15 ms later. Our results indicated that not accounting for the rolling shutter effect significantly degrades performance. The ZED2i rolling shutter camera is connected via USB (unlike the HDR camera, which uses GMSL2). Therefore, it does not allow for accurate timestamping and is limited to the accuracy of the USB serial buffer readout. The three HDR cameras have limited overlapping FoV compared to the ZED2i stereo-pair. This explains the worse performance of the three HDR cameras in comparison to the two ZED2i cameras. Our results highlight the need to account for the rolling shutter effect, and our paired multi-camera dataset enables rigorous future studies and algorithmic development.

F. State Estimation - Effect of IMUs

IMUs are fundamental to state estimation due to their high-rate measurements and their ability to perceive angular velocity alongside linear acceleration. Numerous IMUs with different noise characteristics and costs are available to researchers. While the general relation between noise characteristics and algorithmic performance is well-understood, how this relation can manifest itself in real-world scenarios has received limited attention [20, 49]. To bridge this gap, we tested six different IMUs using DLIO [14] with Hesai LiDAR again on the *Mountain Ascend* dataset. Figure 5 shows only a marginal difference in APE for different IMUs. We hypothesize that this is the case due to the tightly coupled LiDAR point cloud registration. The point cloud registration corrects the inaccuracies in the IMU state propagation and ensures the correct bias estimation.

To analyze the impact of IMU noise characteristics independent of the tightly-coupled LiDAR processing, we conducted a dead-reckoning experiment. The TPS measurements are fused with IMU measurements using Holistic Fusion (HF) [51], similar to our ground truth pose estimate explained in Appendix B. During a TPS measurement dropout, IMU dead-reckoning provides insight into the performance difference based on the noise characteristics. The results are shown in Figure 6. TPS measurements are unavailable between 4.5 s to 9 s as well as from 17 s to 22 s. As seen in Figure 6, the tactile-grade HG4930 performs the best, resulting in a smooth alignment with the TPS measurement after the dropout period. On the other hand, consumer-grade IMUs diverge after a short period (around 5 s) of dead-reckoning time. For a fair comparison, Holistic Fusion was provided with the noise characteristics specified in the IMU datasheet.

G. State Estimation - Ablations

The importance of accurate time synchronization and extrinsic sensor calibration has been highlighted by the community as discussed in Section I and Section II. Despite this common knowledge, the required time synchronization precision and calibration accuracy are often vaguely stated. Therefore, we provide an ablation study on how different time and extrinsic calibration offsets affect performance.

1) Perturbation on Time Synchronization: Similar to Section IV-F, we deployed DLIO fusing Hesai and HG4930 measurements. We added different constant time delays to the IMU measurements and provided APE and RPE metrics in Figure 7. A time offset less than 5 ms negligibly impacts performance. We attribute this to the fact that the HG4930

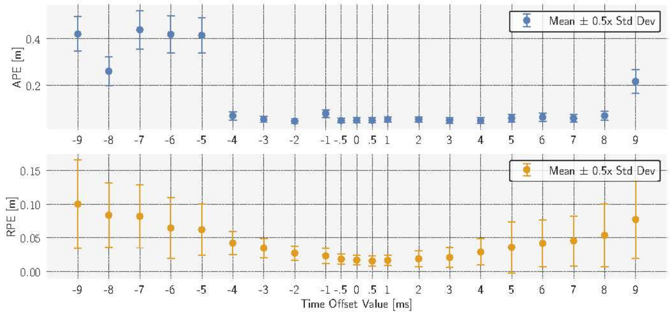

Fig. 7: Time Offset - Performance DLIO (Hesai + HG4930) in APE and RPE with offset applied to the IMU measurements.

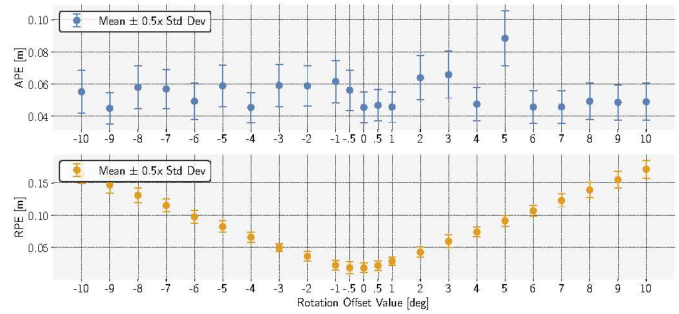

Fig. 9: Extrinsic Rotational Offset - Performance DLIO (Hesai + HG4930) in APE and RPE with an offset applied to the positive pitch angle of the extrinsic calibration.

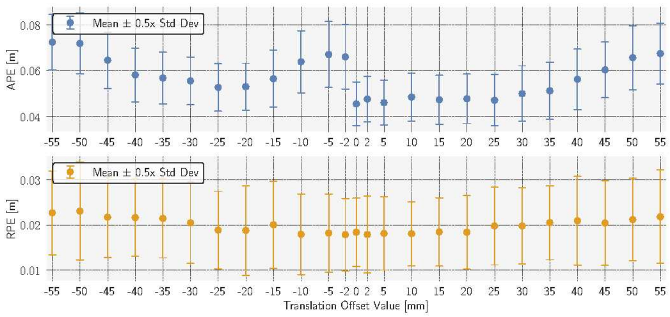

Fig. 8: Extrinsic Translation Offset - Performance DLIO (Hesai + HG4930) in APE and RPE with an offset applied to the positive X-axis of the extrinsic calibration.

Fig. 10: Image Quality Comparison. The HDR camera excels in dark and low-light conditions. The ZED2i misses details in challenging scenarios (Row 1: Tree, Row 2: Sky). The CoreResearch struggles with overexposure and misses crucial details that are essential for localization.

IMU operates at $100\,\mathrm{Hz}$, with a respective measurement period of $10\,\mathrm{ms}$, which is larger than the time offset. We validated the accuracy of the IMU time synchronization in Section V-D2 and further found that different IMUs have varying offsets, which can be potentially attributed to internal filtering.

2) Perturbation on Extrinsic Calibration: Another fundamental property of a well-designed sensor payload is the availability of reliable and accurate extrinsic calibration between sensors. Similar to Section IV-G1, we use DLIO and perturb the extrinsic calibration in translation and rotation and provide the results in Figure 8 and Figure 9, respectively.

Below $5\,\mathrm{mm}$ translation offset, DLIO performance is minimally impacted (Figure 8). Moreover, as the offset increases, the tightly coupled LiDAR-Inertial method suffers from this systematic error, and the LiDAR registration cannot converge. DLIO does not perform online extrinsic calibration. The translation offset might be negated by the point cloud registration process employed after the IMU state propagation.

The rotational component of the extrinsic calibration is more sensitive to perturbations, and errors as small as 2° cause the estimates to diverge (Figure 9). This observation aligns with the fact that the Hesai is a long-range LiDAR, and the far-away measurements are affected more by the rotational offset. Importantly, DLIO runs a geometric observer to estimate and propagate the state; this geometric observer has convergence characteristics in rotation and subsequently in translation, which explains the robustness against the perturbations.

3) Summary: Using *Boxi*, we demonstrated that performance across modalities varies among the tested environments. Naively replacing images recorded from a global shutter camera with rolling shutter images results in a significant performance drop. Similarly, switching from a long-range high-accuracy LiDAR to a lower-accuracy short-range LiDAR leads to a significant decrease in performance. While using a tactile-grade IMU does not directly lead to superior state estimation for DLIO, in the dead reckoning scenarios, the lower noise tactile-grade IMU can temporarily mitigate drift effectively. Our findings show that DLIO is robust to small time-delay errors (up to $2\,\mathrm{ms}$) and large translation offsets. However, accurate rotational extrinsic calibration remains critical to ensure precise state estimation.

H. Perception - Mapping

To showcase that *Boxi* is not only a SLAM-ready payload but also an autonomy-ready payload, we integrated on-board mapping and traversability analysis. Namely, we integrated a previously established elevation mapping [22] framework that combines visual and geometric information (see Figure 11-d). This framework relies on accurate timestamps and precise extrinsic calibration to associate camera images with the map. We choose HDR camera images to be projected onto the

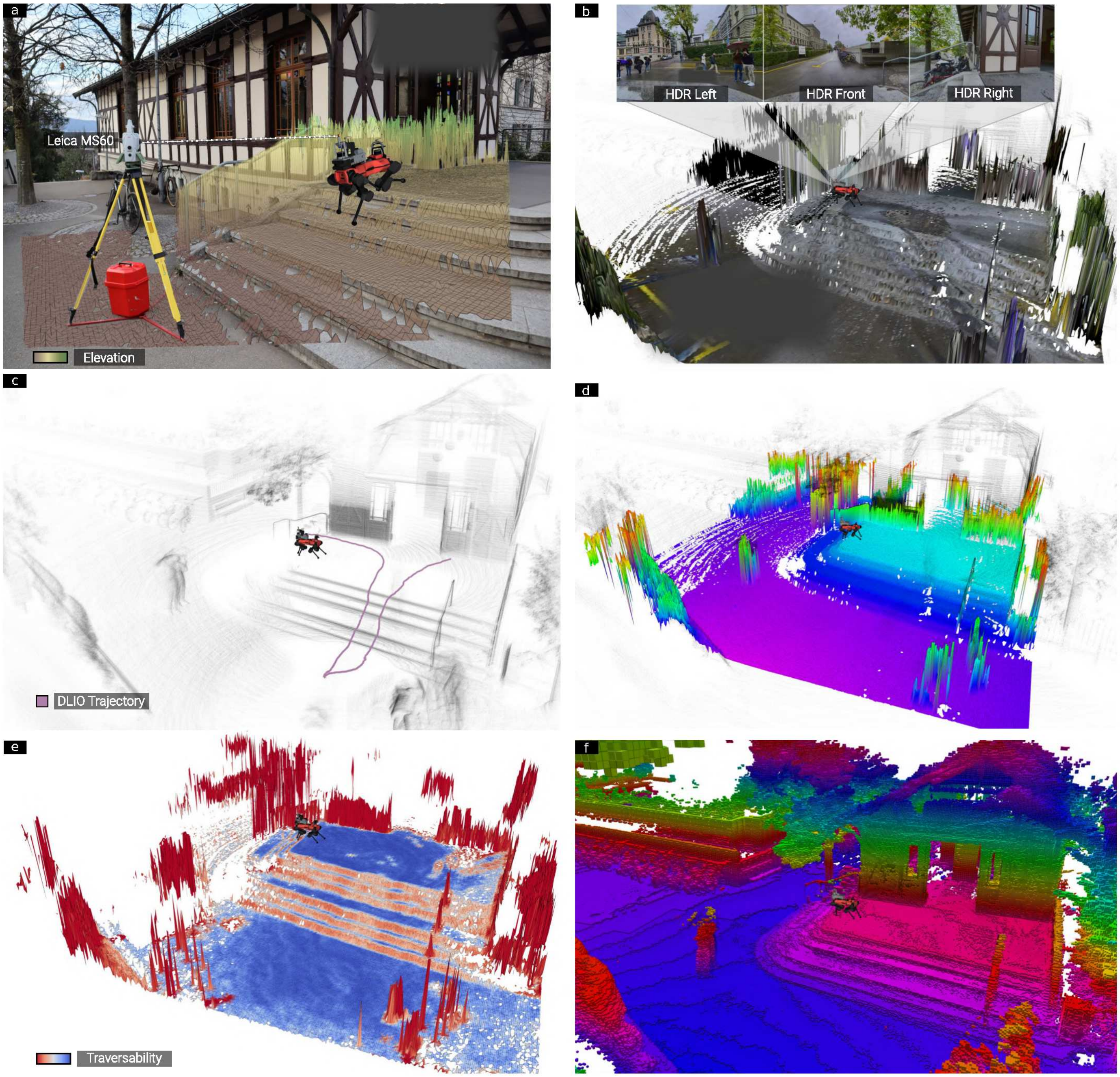

Fig. 11: Example Deployment of Boxi: (a) TPS setup used for the ground truth pose generation. (b) Accurate intrinsic and extrinsic camera calibration enables precise projection of RGB data onto the colorized elevation map. (c) Accumulated point cloud and trajectory visualization based on the registration by the DLIO module. (d) The 2.5D elevation mapping shows the smoothness of the local geometry perception. (e) Traversability analysis derived from the elevation map. (f) 3D Volumetric map generated using the Wavemap [57] module.

map for their superior image quality. While various works have previously shown that different cameras yield varying performance for state estimation, in Figure 10, we show the influence of different lighting conditions on image quality. It is clear that the HDR camera consistently provides the best visual image quality across all environments, which is crucial for effective mapping and scene understanding. Additionally, we demonstrate how the elevation map can be utilized to assess traversability risks for a legged robot (Figure 11-e).

We also integrated the 3D volumetric mapping framework, Wavemap [57], which generates a detailed volumetric representation of the scene and is capable of handling overhanging obstacles (Figure 11-f). The 3D representations and elevation map can be used by the robot for downstream path planning.

I. Limitations - Algorithmic Performance

Even when evaluating methods over a large number of datasets, it is challenging to make generalizable claims that hold across environments, motion characteristics, and different systems. However, the above represents our best attempt to quantitatively assess the sensitivity of different hardware design decisions on state estimation performance.

We have tuned the parameters of DLIO to the best of our knowledge to achieve competitive performance across all datasets. However, for OKVIS2, we did not fine-tune the parameters for each camera setting. In addition, using a framework that directly incorporates rolling shutter compensation could offer deeper insights, as performance is significantly affected when rolling shutter effects are not accounted for. Additionally, for many experiments, we do not measure effects in isolation — e.g., changing the shutter type alone is not feasible, as the underlying imaging sensor differs between cameras. While we ensured that resolution and FoV were comparable (e.g., between HDR and CoreResearch), other factors, such as dynamic range and sensor characteristics, still vary. While we can provide accurate extrinsic and intrinsic calibration for the camera and LiDARs, we found preliminary evidence for a time synchronization offset between IMUs and other sensors within the IMU periods. However, since our evaluation across multiple datasets did not reveal a strong dependence of DLIO on time synchronization, we remain confident that the presented results are valid. Further time synchronization validations are available in Section V-D2.

V. Cookbook to a Sensor Payload

In this section, we provide a generalizable cookbook to design a sensor payload, covering the most important requirements. We explain at first general requirements (Section V-A), followed by the component selection (Section V-B), mechanical design (Section V-C), electronics and communication (Section V-D), calibration (Section V-E), software (Section V-F), and our lessons learned (Section VI).

A. Boxi Payload Requirements

We required *Boxi* to rely only on an external power source and provide a communication line for simple and seamless integration into different platforms without requiring substantial adaptations or modifications. In addition to being compact, the system should be flexible and expandable, allowing for the retrofitting of additional sensing modalities. A rigid and stiff chassis is essential for protecting all components as well as reliable extrinsic calibration, which must remain consistent over time. For *Boxi*, the observability of the kinematic chain between the robot and the sensor payload is required, therefore rendering hardware low-pass filtering, such as vibration dampeners, not suitable.

The payload should incorporate mature, proven, and production-ready sensors, which have already been tested in real-world scenarios. To enable on-board autonomy, the *Boxi* has to offer sufficient computational power. This includes adequate CPU and GPU capabilities as well as enough bandwidth, RAM, and storage to handle the collection of multi-hour datasets. In accordance with the sensor and compute requirements, the total power consumption of the payload should not exceed $300\,\mathrm{W}$ to guarantee long-term deployment with limited battery capacity. Similarly, the weight of the payload is not allowed to exceed $10\,\mathrm{kg}$ to allow for hand-held and general-purpose legged robot operation.

All components must be synchronized and intrinsically calibrated up to industry standard accuracy. Each sensor must have an appropriate FoV to enable autonomous navigation, as this is the primary task of the payload, and the FoV between the sensors should overlap adequately. Finally, the system should be designed to be rugged. It must handle occasional falls and impacts while operating in all-weather conditions, and must be dust-proof and waterproof to the IP65 standard.

B. Component Selection

1) Cameras: To select the best vision suite, we considered *Sensor & Lens*, *Interfacing & Control*, and *Software*. Despite many important attributes such as resolution, frame rate, and FoV being typically taken into consideration, two equally important aspects are often neglected: lens distortion characteristics and depth of field. The former leads to a distorted version of an image compared to a pin-hole camera model, and the latter determines the range at which an object is in focus. The desired focus range depends on the application and can range from centimeter-scale for inspection tasks to $10\,\mathrm{m}$-$100\,\mathrm{m}$ for autonomous driving.

The camera sensor and shutter mechanism (global or rolling) determine the image quality, as well as the sensor's sensitivity, sensed spectrum, and dynamic range. Although a rolling shutter is more cost-effective, it can introduce additional complexity on the software processing side for certain types of environments and motion characteristics.

The camera interface determines how the captured images are transmitted to the host PC, with common options being USB, Gigabit Multimedia Serial Link version 2 (GMSL2), and Ethernet. Each connection type has unique benefits or downsides, e.g., additional latency in image transmission time or increased compute requirements on the host PC. Among the available interfaces, GMSL2 is currently the emerging industry standard for automotive driving cameras. However, integrating GMSL2 may require additional effort compared to simpler options such as Ethernet and USB, as GMSL2 requires additional deserialization hardware.

Additionally, it is crucial to understand the camera's time synchronization capabilities and whether it supports hardware triggering. For advanced applications, synchronizing the camera exposure with the timestamp of other modalities is desirable. In multi-camera setups, particularly for stereo depth estimation, synchronized exposure and triggering may be required. Lastly, the availability of software drivers (e.g., in ROS1, ROS2, Python, or C++) influences the ease of integration, and the availability of in-built white balancing and color correction features can save significant development effort.

We selected industry-standard cameras across three types: global shutter, off-the-shelf stereo, and high-dynamic range. The three selected camera systems are detailed in Appendices C to E.

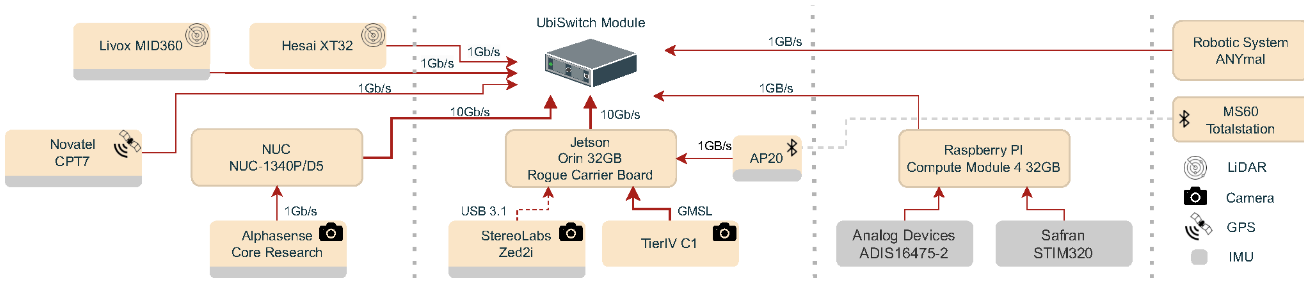

Fig. 12: Communication Overview: The camera sensors are directly connected to the processing compute units to reduce network load.

Camera - Recipe

☑ Interface (*GMSL, Ethernet, USB, ...*)
☑ Camera Sensor (*Size, FPS, Shutter, Dynamic Range, ...*)
☑ Lens (*Distortion, FoV, ...*)
☑ Timing (*Synchronization, Triggering, Latency*)
☑ Driver Support & Host Device Requirements

2) LiDARs: There exists a variety of LiDAR types, each offering trade-offs in range, accuracy, intensity sensing, multi-return features, wavelength, scan pattern, FoV, point density, laser disparity, beam divergence, noise characteristics, and cost. Most of these trade-offs are straightforward; however, we would like to highlight laser disparity as a key factor, where higher disparity may be desirable for navigation applications, e.g., to detect twigs and branches within a forest, while low disparity is favorable for localization. Further, time synchronization is especially crucial for LiDAR. Unlike cameras, which have exposure times of multiple milliseconds, LiDARs effectively produce point estimates in time.

For *Boxi*, we chose spinning LiDARs over solid-state LiDARs as the latter typically have a smaller FoV, are less mature, and are designed mainly for autonomous driving. *Boxi* primarily targets legged robots operating at lower velocities but higher accelerations and with omnidirectional movement, therefore benefiting from the extended FoV of spinning LiDARs. More details about the two selected LiDARs are available in Appendix F.

LiDAR - Recipe

☑ Type (*Spinning, Solid State...*)
☑ Characteristics (FoV, Accuracy, Rate, Distance, Points/s)
☑ Timing (*Synchronization, Triggering, Latency*)
☑ Driver Support & Host Device Requirements

3) IMUs: IMUs can be classified into different categories: consumer/industrial, tactical, and navigation grade based on their pricing and stability. Most IMUs used in mobile robots are consumer-grade IMUs and are based on MEMS technology.

Key performance metrics include gyroscope and accelerometer bias instability, hysteresis, temperature drift, aging effects, as well as the random walk characteristics of angular and linear velocity noise. However, datasheet specifications often do not reflect the actual conditions of the target use case. As a result, other error sources, such as temperature dependence or specific motions, can become the dominant contributors to overall measurement errors in real-world applications. When choosing an IMU, it is important to match the robot motion profile with the input range, specifically the maximum acceleration and angular rate, to avoid saturation. In addition, the sampling rate provided for IMU is typically crucial for the motion characteristics of the target application. Generally, for all IMUs, it is crucial to ensure that the measurement aggregation configuration and point-of-percussion [56] correction is available to the user. Different IMUs require different communication protocols (Serial, SPI, I2C, CAN, Ethernet), which, in combination with time synchronization and setting up on-device filtering chains, can become a multi-week engineering effort. When selecting an IMU, we recommend considering software support, bias instability, aging effects, temperature sensitivity, and the availability of low-level drivers provided by the manufacturers, which can significantly reduce integration efforts. More details of all selected IMUs are provided in Appendix G.

IMU - Recipe

☑ Time synchronization requirements
☑ Gyroscope and Accelerometer instability
☑ Maximum Acceleration and Angular Velocity Rating
☑ Software Support - Reducing Integration Effort

4) GNSS: Global Navigation Satellite System (GNSS) provides drift-free position estimation, making it indispensable for global navigation. Typical position accuracy ranging from $1\,\mathrm{m}$ to $10\,\mathrm{m}$. Many receivers support external atmospheric correction data to compute a Real-Time Kinematic (RTK) solution, which can significantly improve the overall accuracy to approximately $1\,\mathrm{cm}$ in the best case. GNSS receivers typically support single-antenna or dual-antenna setup; the latter not only yields improved positional accuracy, but also provides heading estimation. The selection of antennas and receiver capabilities ultimately dictates which satellite constellations can be used. Moreover, the considerations of electromagnetic interference (EMI) turned out to be critical when selecting an antenna placement, as noted in Section V-D1. The atmospheric correction data can be received from one or more nearby base stations via the online GNSS correction providers or a radio link from a local base station. In environments where real-time communication to the base station is not possible, Precise Point Positioning (PPP) solutions can be employed. With PPP, correction data is received through single-path communication via satellite over a large geographic area, resulting in an

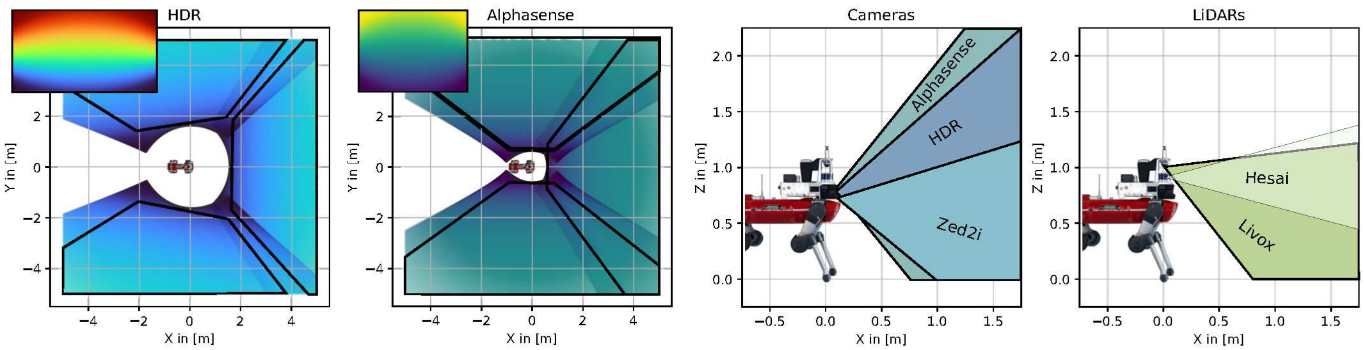

Fig. 13: Visualization of LiDAR and Camera FoV: *(Far Left)* Top-down view of HDR FoV projected on flat ground plane. The black outline represents the rectified FoV; The distorted FoV overlaps while the undistorted one does not overlap. *(Left Center)* Top-down view of the CoreResearch camera FoV projected on the ground plane. Overlap in both distorted and undistorted FoV. *(Right Center)* Side view of FoV for CoreResearch, HDR, and ZED2i cameras, with ZED2i tilted downward at a 15° angle and rotated 180° in its image axis for better space management within *Boxi*. *(Far Right)* FoV of Hesai and Livox LiDARs.

accuracy up to $2.5\,\mathrm{cm}$. More details about our GNSS receiver and postprocessing software are available in Appendix H.

GNSS - Recipe

- ☑ Dual or Single Receiver (Baseline)
- ☑ Single or multi-band support
- ☑ Online Corrections (None, RTK, PPP)
- ☑ EMI (Antenna Placement, Receiver Placement)
- ☑ Postprocessing Support (Correction data)
- ☑ Synchronization against GNSS Time
- ☑ Connectivity & Driver Support & Host Device Requirements

5) Network Switch: Reliable communication between Ethernet-enabled sensors and computers requires a switch or a managed/unmanaged router. Key considerations include the number of ports, port speeds, form factor, cost, and switching capacity. One desirable feature is the hardware support for PTP time synchronization (IEEE 1588v2), as using a non-compliant switch can increase time synchronization errors. More details about the selected switch are provided in Appendix I.

Switch - Recipe

- ☑ Management (active or passive)
- ☑ Switching Capacity, Speed, Size
- ☑ PTP-enabled

6) Compute: To establish a versatile research platform capable of deploying various algorithms, we require both a CUDA-accelerated NVIDIA GPU and an x86 CPU architecture. Due to the large amount of raw sensory data, two computers are required to perform the required raw data processing and allow for online image compression (see Section V-F). This, in addition to the demand for accurate time-stamping and other constraints, led us to integrate three compute modules, namely a NUC, Jetson, and Raspberry Pi (see Figure 12). More details for each selected module are provided in Appendix J.

Compute - Recipe

- ☑ Workload Analysis (CPU, GPU)
- ☑ Reduce Compute Units to Minimum
- ☑ Hardware Acceleration (*NVEC H.265, GPU*)
- ☑ Connectivity (*Ethernet, USB-X, SPI, GMSL, PCI-E* ..)
- ☑ Time synchronization capabilities (PPS, PTP)

C. Mechanical

The entire payload has been designed with a strong focus on positioning accuracy and stiffness while minimizing weight.

1) Component Placement: In general, the component placement requires balancing multiple, often conflicting, objectives. The most important criterion is the safety of each component, specifically the LiDAR sensors and the GNSS antennas, which require a rollover cage. Apart from thermal, dust-proofing, and water-proofing considerations, the FoV for each sensor is essential for the placement of the exteroceptive components. During autonomous operation, each sensor should capture a comprehensive but overlapping view of the environment, which is also key for calibration and multi-modal algorithms. For example, sufficient overlap between LiDARs and cameras allows calibration of the extrinsic parameters using inexpensive targets rather than requiring a dedicated calibration room. Figure 13 illustrates the overlapping FoV between all cameras and LiDARs. When mounted on an ANYmal robot with a standing height of approximately $0.8\,\mathrm{m}$, the cameras capture the ground in front of the feet of the robot starting at a distance of $0.75\,\mathrm{m}$. This setup effectively balances peripheral coverage and ground visibility for local navigation. The front-facing CoreResearch stereo pair has a baseline of $11\,\mathrm{cm}$, which results in accurate depth estimates up to a distance of $20\,\mathrm{m}$ and enables a direct comparison to the ZED2i with rolling shutter.

The Livox is mounted upside down, allowing for accurate estimation of the terrain directly around the robot, which is crucial for 3D near-field mapping. The Hesai is mounted horizontally on top of the camera bundle for optimal FoV for localization.

In addition, the primary GNSS antenna is rigidly connected to the lid through standoffs and a copper-shielded element.

Our empirical findings showed that the EMI resulting from the LiDARs interferes with the GNSS signal (see Section V-D3).

Component Placement - Recipe

☑ Interference Analysis
☑ Field of View Analysis
☑ Protection & Ease of Calibration

2) Tolerances and Assembly: To meet the $10\,\mathrm{kg}$ weight requirement while maintaining structural integrity, *Boxi* (base and lid) is machined from a single AL-6061 T4 aluminum mono-block with a minimum wall thickness of $2\,\mathrm{mm}$. This material offers a favorable strength-to-weight ratio and enables in-house manufacturing.

All parts follow ISO 2768-1 with a fine tolerance class ($0.05\,\mathrm{mm}$). A full 3D CAD model was created to account for connector spacing and cabling constraints. Positioning pins are used at all interfaces, including between the lid and body and for all sensor mounts, to minimize offsets due to screw backlash and machining tolerances. A protective cage is mounted on the lid to house the Livox unit and prism while shielding the LiDARs from impact.

Tolerances and Assembly - Recipe

☑ Material (*Weight, Stiffness, Number of Interfaces*)
☑ Thermal behavior of the Material
☑ Manufacturing (*Cost, Precision, Quantity*)
☑ Positioning Pins, Thread-locking Adhesives, Torque
☑ Consider Assembly (Plugging in Connectors)

3) Thermal Considerations: The primary thermal load within *Boxi* originates from the two main compute modules and the DC-DC converter. To balance cooling performance, simplicity, and cost-effectiveness, we selected a convection-based cooling strategy. All heat-generating components were strategically placed at the rear, while cameras and LiDAR sensors were placed at the front of *Boxi* to limit the adverse effects of thermal expansion. To improve thermal management, cooling fins were integrated beneath the two compute modules, complemented by a single radial fan to actively circulate air.

To verify the efficacy of the cooling system and better understand the impact of temperature changes on the deformation of *Boxi* and, in turn, the effect on extrinsic calibration, we conducted a thermal Finite Element Method (FEM) simulation under maximum thermal load. The maximum heat generation from all significant sources adds up to $92\,\mathrm{W}$ dissipation. The FEM simulation results shown in, Figure 14, indicate a low overall deformation in the sensor area. Under maximum load, the relative rotation between the sensors remains within $0.004°$, which is less than the machine tolerance and the accuracy of the calibration. The simulation validates our thermal design and shows that aluminum AL6061-T4 is sufficient, even though it has double the thermal expansion coefficient of regular stainless steel [3].

We also verified the thermal design of *Boxi* under normal load (recording, see. Section V-F) with an ambient temperature of $19\,°\mathrm{C}$. We monitored the maximum temperature of the housing using a Fluke Thermal Camera and measured the internal CPU

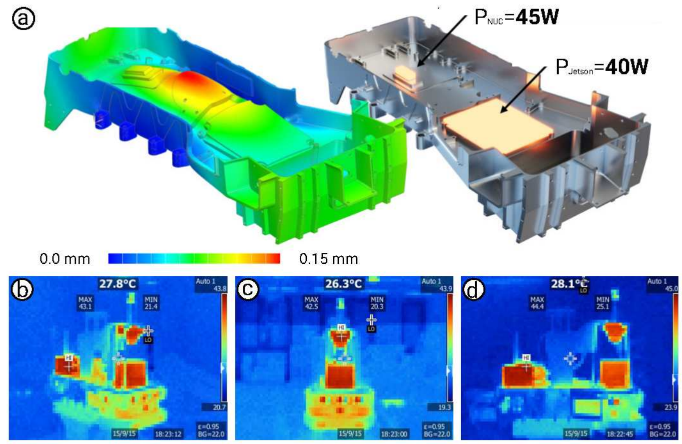

Fig. 14: (a) Thermal deformation from heat generated by the compute modules. The deformation is amplified and visualized for clarity, with the mounting points assumed to remain at a constant temperature. (b-d) Thermal imaging of the box during recording.

temperature of the NUC at $55\,°\mathrm{C}$ and the Jetson at $51\,°\mathrm{C}$. The entire housing remained well below $30\,°\mathrm{C}$, while the fan operated at $30\,\%$ speed. The CPT7 with $43\,°\mathrm{C}$ and all cameras and LiDARs with a temperature below $40\,°\mathrm{C}$ remained well within the operation specifications.

Thermal Considerations - Recipe

☑ Heat Sources (*Compute Units, Sensors*)
☑ Ambient Conditions (*Temp. Range, Humidity Levels*)
☑ Thermal FEM Simulation (*Angular and Linear Deformation*)

4) Advanced Topics: As *Boxi* is intended to be mounted on a legged robotic platform, we also considered topics such as vibrations, weatherproofing, and fall protection. The details of these topics are provided in Appendix O.

5) Recommendations and Reflections: The dust- and waterproof design proved to be highly effective, particularly with the fins strategically placed on the exterior. Integrating an off-the-shelf modular cable pass-through was both simple to integrate and offered flexibility. When designing rollover cages, it is crucial to always assume the worst-case scenario and identify potential failure points under various conditions. While the compact form-factor fan was desirable, it generates substantial noise at high RPMs.

We initially used the manufacturer's camera FoV to design the housing. However, this specification was applied to rectified images, which led to the chassis obstructing a significant portion of the image because of the fisheye effect. Manufacturing the base body from a single block of aluminum significantly improved stiffness, precision, and thermal properties while simplifying the design. However, this approach may be impractical and prohibitively expensive for commercial applications.

D. Electronics and Communication

1) Power: *Boxi* supports a wide input voltage range from $18\,\mathrm{V}$ to $60\,\mathrm{V}$ for compatibility across platforms. Power is distributed via a custom PCB integrating two step-down converters: a $12\,\mathrm{V}$, $300\,\mathrm{W}$ (Traco TEQ 300-4812WIR) and a $5\,\mathrm{V}$, $15\,\mathrm{W}$.

To mitigate EMI and ground loop issues, we used ground-isolated converters, added input filter coils, and implemented a star ground layout. Distinct connector types were chosen for different power levels to prevent misconnection and ensure electrical safety. We recommend analyzing required voltage levels and power consumption, considering the inrush currents of all components, and hot-plug battery exchange if desired.

Power - Recipe

- ☑ Power Analysis (*Voltage Levels, Power, Inrush Current*)
- ☑ Distribution (*Cabling, Polarized Connectors*)
- ☑ Best practices to prevent EMI

2) Communication and Synchronization: Extending on the basic knowledge on communication and time synchronization provided in Appendix K, components that operate on independent clocks require active synchronization to mitigate drift. Typical consumer-grade crystal oscillators drift at a rate of $50\,\mathrm{ppm}$ (parts per million), leading to $29.9\,\mathrm{ms}$ drift per $10\,\mathrm{min}$. Two common synchronization protocols are PTP (e.g., `ptp4l`, `phc2sys`) and NTP (e.g., `chrony`). PTP is generally preferred when reliable network connections are available, as it can achieve nanosecond-level precision. As mentioned before, the GNSS module CPT7 is used as a clock reference and PTP Grandmaster clock. Since the switch in *Boxi* does not support hardware PTP synchronization (IEEE 1588v2; see Section V-B5), we conducted a $43\,\mathrm{h}$ continuous, time synchronization drift analysis, showcasing high-quality time-synchronization over the entire period. More details and results on this experiment can be found in Appendix L.

a) Synchronization - Camera and LiDAR: The time synchronization of both LiDARs has been established with the IEEE 1588v2 (2008) standard PTP protocol, which indicates PTP clock accuracy $\leq 1\,\mu\mathrm{s}$ and PTP clock drift $\leq 1\,\mu\mathrm{s}\,\mathrm{s}^{-1}$. Both LiDARs provide a per-point timestamp for accurate integration to downstream tasks. Different from the LiDARs, cameras require an exposure time of around $10\,\mathrm{ms}$ for normal lighting conditions. Ideally, the timestamp should correspond to the midpoint of the exposure for each pixel. In the case of our rolling shutter HDR cameras, one needs to calculate the exposure time for every i-th horizontal line of the image separately:

$$i \times \left(\frac{1}{30 \times 1400} \right) = i \times 23.8\,\mu s \tag{1}$$

Thus, the last line is exposed exactly $30.464\,\mathrm{ms}$ after the first. Similarly to the LiDAR case, compensating for this different trigger and readout time improves state estimation and requires precise synchronization [63]. Manufacturer data is not available for the ZED2i, regarding exposure time or global shutter compensation.

b) Synchronization - IMUs: All 7 IMUs available within *Boxi* are time synchronized to a varying level of precision. The Raspberry Pi Compute Module 4 handles the time synchronization ADIS and STIM320 using hardware timestamping via custom-written kernel modules. The other IMUs are synchronized by the manufacturer-provided drivers. More information about the IMU time synchronization details can be found in the Appendix M.

c) Synchronization - Verification: When it comes to time synchronization, one of the essential factors is the validation of the claimed accuracy. While tools such as `ptp4l` provide the estimated time tracking offset [74] (see Appendix L), this is harder to achieve for sensors synchronized with other means. Since we have IMUs connected to each PC, the time synchronization accuracy can be validated by aligning the angular velocities of the IMUs over time, similar to the approaches in [23, 31]. We have developed a standalone, open-source tool for verifying time synchronization between two IMU time series measurements at arbitrary rates. We ran the verification tool across all recorded missions to obtain a confidence interval per mission using the HG4930 as a reference IMU. We applied the tool to three $30\,\mathrm{s}$ snippets per axis for each mission. The results are shown in Table VI and indicate that the offset between all IMUs can be estimated up to a standard deviation of around $0.5\,\mathrm{ms}$ and is constantly below $4.5\,\mathrm{ms}$, which can be attributed to the exposure time of the IMU and internal time delays. The details of the time synchronization validation tool and procedure are provided in Appendix N. For verification of the camera time synchronization, there is the option to hardware trigger an external LED; however, this procedure is time-intensive and out of the scope of this work [70]. Lastly, all IMUs, and LiDARs operate in free-running mode. It was sufficient for our use case to only trigger cameras of the same type simultaneously.

IMU	**x-Axis** $\mu \pm \sigma$ [ms]	**y-Axis** $\mu \pm \sigma$ [ms]	**z-Axis** $\mu \pm \sigma$ [ms]	**AVG** $\mu \pm \sigma$ [ms]
ADIS	1.41 ± 0.16	1.33 ± 0.09	1.35 ± 0.32	1.36 ± 0.21
Bosch	3.29 ± 0.14	3.20 ± 0.16	3.35 ± 0.36	3.28 ± 0.25
AP20-IMU	1.03 ± 0.13	0.85 ± 0.14	1.71 ± 0.40	1.20 ± 0.45
HG4930	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
TDK	-4.43 ± 0.22	-4.34 ± 0.18	-3.14 ± 1.74	-3.97 ± 1.16

TABLE VI: Evaluation of the IMU Software Alignment Tool across datasets, with three alignment runs on different sub-trajectories per dataset. We provide alignment results for each axis, along with the mean and standard deviation.

Synchronization - Recipe

- ☑ Bandwidth and Latency
- ☑ Precision Time Protocol (PTP) (Preferred)
- ☑ Network Time Protocol (NTP) (Unreliable Network)
- ☑ Hardware Timestamping (Linux Kernel Modules)
- ☑ Verification Tools

3) Electro Magnetic Interface (EMI): EMI mitigation in *Boxi* focused on minimizing both emissions and internal interference. Analog signal integrity was prioritized by placing sensitive components, such as the GNSS receiver, outside the enclosure and using additional EMI absorbers. Testing showed that maximizing the distance between the GNSS antennas and the main housing, avoiding direct contact with the metal structure, and adding a copper ground plane significantly reduced interference. Inside the enclosure, all signal lines use shielded cables where possible. An exception is the $1\,\mathrm{Gbps}$ Ethernet link, which uses unshielded twisted pairs to simplify routing. Two openings in the aluminum housing on the side panels are fitted with 3D-printed plastic parts, enabling the mounting of Wi-Fi antennas within *Boxi*. This design represents a trade-off between compact integration and EMI mitigation. As an alternative, the WiFi antennas can be routed outside the box.

EMI - Recipe

☑ EMI Emission Requirements
☑ Signal Integrity (GNSS Sensitive)
☑ Shielding, Grounding, Minimize Cable Length

E. Calibration

Here, we detail the intrinsic and extrinsic calibrations of all sensors, covering the calibration methods, procedures, and verification techniques. These calibrations comprise the intrinsic and extrinsic calibration of all cameras with respect to each other (the bundle of all cameras), the extrinsic calibration of each LiDAR, jointly, to all CoreResearch cameras, the extrinsic calibration of each IMU to the front-facing global shutter cameras, and the prism position in the reference frame of the camera bundle. Additionally, the IMU drift parameters are calibrated.

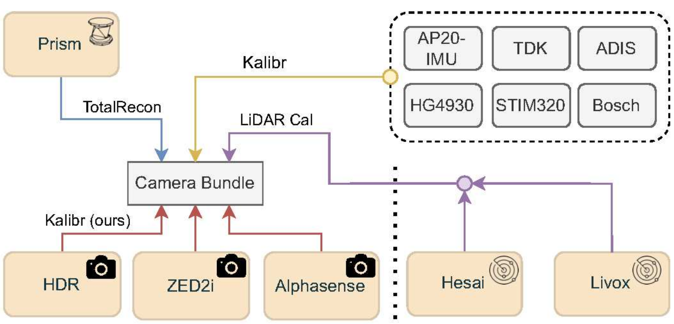

Fig. 15: An overview of the calibrations done. The arrows indicate the transforms that are explicitly calibrated.

Given the large number of available sensors — 2 LiDARS, 10 cameras, 7 IMUs, and a reflector prism — we carefully designed the calibration method, setup, and procedure.

1) Calibration Chain: Central to the calibration of our system are the intrinsic and extrinsic calibrations of the 10 cameras. Since cameras cover a wide FoV, sufficient overlap is available to robustly calibrate the extrinsic parameters by detecting a calibration target placed in various poses. The camera bundle calibration forms the foundation for calibrating all other sensors in the subsequent steps of the calibration chain Figure 15.

2) Camera Intrinsic and Extrinsic: For camera intrinsic and extrinsic calibration, we re-implemented Kalibr [26, 56] to allow for online calibration and feedback during the recording session. The calibration can be decomposed into an intrinsic and extrinsic phase. During the intrinsic phase, the operator moves an AprilGrid [52] in front of all cameras until sufficient samples are collected (see Appendix P1). Then, in the extrinsic phase, it is required to statically place the target at different locations around the *Boxi* or move the payload itself until sufficient static samples are collected to avoid time synchronization errors and motion blur (see Appendix P1). We periodically estimate the covariance of the intrinsic and extrinsic parameters in real-time to provide feedback on the sufficiency of the data collected. We used the Kannala-Brandt [37] equidistant distortion model for the wide-angle fisheye CoreResearch [53, 64] and HDR cameras, and the rad-tan distortion model for the ZED2i. More details on why we chose the AprilGrid can be found in Appendix P2. Similarly, details on reprojection errors across all cameras are provided in Appendix P3

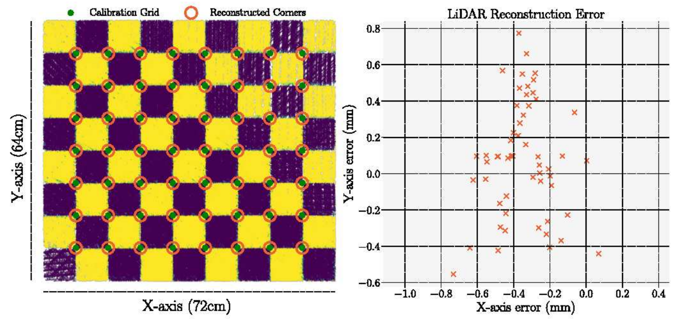

Fig. 16: LiDAR reconstruction of the calibration target using camera-detected poses and the optimized extrinsic calibration. The reconstructed corner locations are sub-millimeter accurate.

In Table VII, we show the numerical evaluation of the estimates of the extrinsic and intrinsic parameters, which are also presented live to the user during the calibration procedure. The standard deviation of the translation and rotation for all cameras is excellent. The total intrinsic and extrinsic calibration procedure takes around 5 min compared to the default Kalibr workflow of multiple hours.

3) Camera to IMU: Before calibrating the camera-IMU extrinsic parameters, we collect static data of the IMUs over 18 h to model the Allan variances of each of the IMUs using [8]. We then perform camera-IMU extrinsic calibration using the front-facing global shutter CoreResearch cameras using Kalibr [26]. To collect camera data capable of capturing the dynamic motion required for the camera-IMU calibration accurately, we set the camera frame rate to 17 Hz, with an exposure time of 1 ms. With this exposure time, we also employ uniform artificial lighting of the static calibration target to aid in the detection of the calibration target. More details about the motions and individual IMUs can be found in the Appendix Q

4) LiDAR to Camera Extrinsic: To efficiently perform the LiDAR to camera extrinsic calibration, we used a variant of the intensity alignment extrinsic calibration DiffCal [25]. Unlike plane matching, which requires many samples spanning different orientations, this intensity-matching method can recover the calibration using a single sample. Additionally, intensity matching allows one to incorporate partial observations of the calibration target in the LiDAR point cloud, which is typical for our Hesai LiDAR with a narrow FoV. Allowing for partial observations enables the collection of calibration data closer to the cameras, where the detected poses are more accurate.

For this calibration, we used a different calibration target formed of a checkerboard pattern. Each LiDAR is individually calibrated against the 5 CoreResearch cameras. Using the

	Translation (m)				Rotation (°)				Error (px)	
Camera	x	y	z	σ_t	Roll	Pitch	Yaw	σ_r	Mean	Med
`CoreResearch-Front-Center`	–	–	–	–	–	–	–	–	0.32	0.25
`CoreResearch-Front-Right`	0.0544	0.0011	0.0002	0.0001	0.09	0.13	-0.22	0.005	0.20	0.16
`CoreResearch-Front-Left`	-0.0559	-0.0004	0.0009	0.0001	0.04	-0.01	0.96	0.005	0.28	0.20
`CoreResearch-Left`	-0.1019	0.0115	-0.0681	0.0002	84.10	-80.23	-93.59	0.008	0.24	0.20
`CoreResearch-Right`	0.1041	0.0144	-0.0693	0.0002	79.62	80.68	90.57	0.010	0.39	0.31
`HDR-Front`	0.0002	0.0337	0.0076	0.0001	9.89	0.23	0.29	0.006	0.52	0.44
`HDR-Left`	-0.1028	0.0479	-0.0671	0.0002	135.55	-75.47	-135.43	0.010	0.39	0.34
`HDR-Right`	0.1021	0.0478	-0.0684	0.0003	134.93	76.10	136.21	0.012	0.39	0.34
`ZED2i-Left`	0.0594	0.0759	0.0202	0.0002	-0.07	-0.03	-179.34	0.014	0.63	0.56
`ZED2i-Right`	-0.0601	0.0744	0.0203	0.0002	-0.00	-0.08	-179.44	0.016	0.57	0.48

TABLE VII: Typical results from a full camera-camera intrinsic and extrinsic calibration. The spatial transforms are with respect to CoreResearch-Front-Center, hence is has no entries for these values as it is the origin. Note that we do not impose a stereo constraint on the ZED2i camera during calibration, yet the result is precisely consistent with the stereo baseline of the camera.

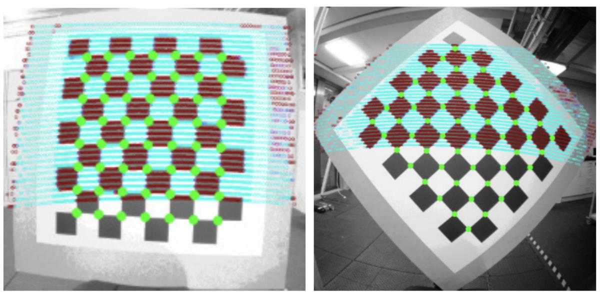

Fig. 17: Example calibration data samples used for the LiDAR to camera calibration. Hesai LiDAR points are overlayed on the image, and colored by each point's intensity channel. The calibration target reflects the LiDAR beams with intensity contrast between the black and white squares seen in the camera image.

calibrated extrinsic parameters between the cameras, the detected calibration target poses from each camera are fused and jointly calibrated with respect to each LiDAR, without refining the intra-camera extrinsic parameters. More details on the calibration target and data collection can be found in Appendix R.

Figure 16 shows a dense, reconstructed point cloud of the calibration target, colored by the intensity channel. This reconstruction has the coordinate system of the calibration target as its origin. From this dense reconstruction, we use a line-based corner detection algorithm (similar to [17]) to detect the locations of the checkerboard corners in the reconstruction. Figure 17 shows the sub-millimeter consistency between the detected corners and their true positions on the canonical calibration target. In Figure 17, we show example overlays of the LiDAR points on the calibration target, after optimization. Another example of camera-LiDAR overlay during rapid motion is provided in Appendix R1.

5) Camera to Prism: Due to the lack of existing methods, we developed a custom calibration procedure, *TotalRecon*, to calibrate the front-facing CoreResearch cameras to the total station prism. Using TPS measurements and camera detections of a static calibration target, we solve for a 9-DoF state: the prism position relative to the camera, and the 6-DoF pose of the target with respect to the total station. To ensure accurate target detection, the camera is positioned within 1 m of the target. The total station tracks the prism while *Boxi* is held static at various poses around the target using a tripod. Sufficient samples are collected to excite all translational and rotational degrees of freedom. We validate the calibration by checking the consistency of prism estimates across all cameras, transforming them to the front-center camera frame using known extrinsics. The routine yields 3 mm cross-camera consistency and achieves accuracy below the mechanical mounting tolerance.

Calibration - Recipe

- ☑ Calibration Requirements and Chains
- ☑ Available Calibration Methods (CAD, Tools)
- ☑ Calibration Targets and Procedure Requirements
- ☑ Calibration Intervals
- ☑ Change of Calibration (Time, Temperature)
- ☑ Calibration Verification

F. Software

The hardware and software design of *Boxi* were co-developed. To ensure ease of use for researchers, the software is based on Ubuntu 20.04 LTS with ROS1 Noetic and ROS2 Humble, managed via Docker. A custom kernel was used on the Raspberry Pi Compute Module 4 to enable IIO support and accurate kernel timestamping. For all ROS1 components, we ran a single `roscore` instance per PC and enabled inter-PC communication using the fkie multi-master package [66], which utilizes gRPC under the hood.

1) Recording: In line with best practices established in previous research [28, 53], we only run the required sensor drivers and prioritize the recording of raw driver data to minimize artifacts, minimize processing time during recording, and preserve future processing flexibility. As an example, the UDP packets are recorded for Hesai LiDAR instead of point clouds, and the serialized SVO2 files are recorded for the ZED2i RGB-D camera to maintain maximum fidelity. We implemented a distributed recording system, which stores data on the Jetson, NUC, and CPT7, minimizing network bandwidth and distributing load across machines. Specific details about image compression and recording details can be found in Appendix R1. In summary, recording for 10 minutes results in 14.4 GB HDR, 5.77 GB CoreResearch, 1.0 GB ZED2i, 1.14 GB Hesai, 1.76 GB Livox, and >100 MB for all IMU and other auxiliary status, diagnostic data, totaling at around 25 GB per 10 minutes of deployment.

During this recording, the Jetson exhibited higher CPU usage (72.0%) compared to the NUC (49.2%), while only the Jetson showed GPU activity (7.0%). The NUC utilized the network at $58.2\,\mathrm{Mbps}$, whereas the Jetson showed no network activity. Conversely, the Jetson exhibited higher disk write bandwidth at $18.7\,\mathrm{Mbps}$, compared to $13.5\,\mathrm{Mbps}$ on the NUC.

Software - Recipe

☑ Co-development of Hardware and Software
☑ Testing of all Software on Prototype (Desk Setup)
☑ Ensuring Driver and Hardware Compatibility
☑ Adaptable and Modular Software Design
☑ Hardware Acceleration (Video Compression, GPU)
☑ Simplicity & Maintainability & Documentation

G. Limitations - Boxi Design

Starting with the mechanical design, *Boxi* lacks a robust rollover cage to protect the system under the full load of a legged robot tipping over. While the lid provides sufficient stiffness for mounting, it cannot absorb the energy generated during falls, especially under the robot's full weight.

To address this issue, the development of a newly designed protective rollover cage is essential. The cage should be fully decoupled from *Boxi* and directly mounted onto the robot, offering enhanced protection and durability. During our thermal simulation, we mainly considered the heat generated by the compute units. However, the heat generated by individual sensors is also non-neglectable and currently not taken into consideration.

While we can measure highly accurate time synchronization via PTP offsets, starting the development with more verification tools in mind would have helped. The developed IMU alignment tool confirmed good time synchronization. Further, we did not implement verification, such as active LED targets, to validate the manufacturer's rolling shutter effect compensation.

The decision to use the Jetson Orin with the compact ConnectTech carrier board was motivated by its smaller form factor. However, the absence of real-time interrupt pins resulted in a Raspberry Pi being required to manage the STIM320 and ADIS IMUs. This introduced further complexities, including a custom Raspberry Pi PCB design and PTP synchronization. In retrospect, avoiding the integration of IMUs such as the ADIS and STIM320 due to their limited ROS1/ROS2 driver support—would have significantly reduced integration efforts and further improved the maintainability of the software stack. Comprehensive environmental testing, including shock and vibration resistance assessments, has yet to be performed. Although extrinsic and intrinsic calibrations have demonstrated stability over time, broader environmental validation remains a critical area for future work.

VI. Lessons Learned and Discussions

1. **Hardware and software should be jointly developed**: The interdependence of software and hardware turned out to be the most difficult part to get right. An exhaustive analysis of software and hardware requirements and the interdependence between components is essential.

2. **Select components for the application**: While technical specifications in datasets are important, having access to good software support and easy integration can be more important and justify additional cost, weight, size, or worse specifications.

3. **Time synchronization targeted for a specific application**: Time synchronization requirements might differ based on the application and the software. State-estimation and SLAM tasks might require µs or ms level time synchronization between LiDAR and IMUs.

4. **Develop software before hardware**: We recommend implementing the software early in the design process to identify any compatibility issues.

5. **Develop verification tools for everything**: How can you verify that the software functions as intended? The development of tools to ensure time synchronization and data integration is essential.

6. **Perform rapid-prototyping during development**: We recommend building a cheap mock-up setup and 3D-printed prototypes before moving to advanced CNC milling.

7. **Calibration targeted for you application**: Define your calibration requirements, and which tools and setups are required to achieve it. An overlapping FoV, use of cheap targets, or avoiding expensive calibration room setups was desirable for us.

8. **Keep it simple**: What is better than a ptp-enabled switch? Using no switch. We recommend using as few compute units, sensors, and mechanical components as possible. Only implement custom PCBs if they significantly simplify the design. The same applies to the software: e.g. integrating ROS1 and ROS2 inherently increased the complexity.

VII. Conclusion

In conclusion, *Boxi* has proven to be an invaluable tool, enabling the collection of the GrandTour dataset with high-quality ground truth reference data. Through the algorithmic performance analysis (Section IV), we demonstrated that sensor payload design should consider the target application, software availability, and accessible components.

Out of the selected components, NovAtel SPAN CPT7 and Leica Geosystems components (custom AP20 and MS60) stood out as vital components, due to their excellent software support and performance, despite their cost. The design of the power distribution board also contributed to a clean assembly and minimized EMI problems, with the placement of the GNSS antenna being the only challenge we encountered.

Moreover, we plan an Autonomy *Boxi* version with a reduced set of sensors, which will include the two LiDARs, three HDR cameras, and a single updated next-generation NVIDIA Jetson processor, alongside a PTP-enabled switch for improved synchronization. These refinements will further enhance *Boxi*'s capabilities, ensuring it remains a versatile and reliable platform for future research and applications in robotics. Lastly, a drawback of *Boxi* is the lack of auxiliary modalities such as UWB, Radar, event, and thermal cameras, which can improve the usefulness and effectiveness of the sensor payload in various environments.

References

[1] Cpt7 product sheet, 2024. URL https://ppmgmbh.com/Download/Produkte/NovAtel/SPAN_CPT7/Datenblatt/Datasheet_NovAtel_CPT7.pdf. Accessed: 2025-01-21.

[2] Inertial explorer, 2024. URL https://docs.novatel.com/Waypoint/Content/Inertial_Explorer/Overview_of_IE.htm. Accessed: 2025-01-21.

[3] The Engineering ToolBox (2005). Metals - temperature expansion coefficients, 2005. URL https://www.engineeringtoolbox.com/thermal-expansion-metals-d_859.html.

[4] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 10 2023. URL https://github.com/ceres-solver/ceres-solver.

[5] Michael Bloesch, Michael Burri, Hannes Sommer, Roland Siegwart, and Marco Hutter. The two-state implicit filter recursive estimation for mobile robots. *IEEE Robotics and Automation Letters*, 3(1):573–580, 2017.

[6] Leonardo Brizi, Emanuele Giacomini, Luca Di Giammarino, Simone Ferrari, Omar Salem, Lorenzo De Rebotti, and Giorgio Grisetti. Vbr: A vision benchmark in rome. *arXiv preprint arXiv:2404.11322*, 2024.

[7] A. Bry, C. Richter, and N. Roy. Spatial and temporal calibration of multi-sensor systems with application to stereo vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3863–3870, 2015.

[8] Russell Buchanan. Allan variance ros, November 2021. URL https://github.com/ori-drs/allan_variance_ros.

[9] Keenan Burnett, David J Yoon, Yuchen Wu, Andrew Z Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith YK Leung, et al. Boreas: A multi-season autonomous driving dataset. *The International Journal of Robotics Research*, 42(1-2):33–42, 2023.

[10] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[11] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[12] Alexander Carballo, Jacob Lambert, Abraham Monrroy, David Wong, Patiphon Narksri, Yuki Kitsukawa, Eijiro Takeuchi, Shinpei Kato, and Kazuya Takeda. Libre: The multiple 3d lidar dataset. In *2020 IEEE intelligent vehicles symposium (IV)*, pages 1094–1101. IEEE, 2020.

[13] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019.

[14] Kenny Chen, Ryan Nemiroff, and Brett T Lopez. Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 3983–3989. IEEE, 2023.

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 2, page 1, 2015.

[16] Andrei Cramariuc, Aleksandar Petrov, Rohit Suri, Mayank Mittal, Roland Siegwart, and Cesar Cadena. Learning camera miscalibration detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4997–5003. IEEE, 2020.

[17] Igor Cvisic, Ivan Markovic, and Ivan Petrovic. Enhanced calibration of camera setups for high-performance visual odometry. *Robotics Auton. Syst.*, 155:104189, 2022. URL https://doi.org/10.1016/j.robot.2022.104189.

[18] Effie Daum, Maxime Vaidis, and François Pomerleau. Benchmarking ground truth trajectories with robotic total stations. *arXiv preprint arXiv:2309.05134*, 2023.

[19] Timothy A. Davis and Yifan Hu. The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1), December 2011. ISSN 0098-3500. doi: 10.1145/2049662.2049663. URL https://doi.org/10.1145/2049662.2049663.

[20] Mattia De Agostino, Ambrogio Maria Manzino, and Marco Piras. Performances comparison of different mems-based imus. In *IEEE/ION Position, Location and Navigation Symposium*, pages 187–201. IEEE, 2010.

[21] Graziella Del Duca and Carol Machado. Assessing the quality of the leica blk2go mobile laser scanner versus the focus 3d s120 static terrestrial laser scanner for a preliminary study of garden digital surveying. *Heritage*, 6(2):1007–1027, 2023.

[22] Gian Erni, Jonas Frey, Takahiro Miki, Matias Mattamala, and Marco Hutter. Mem: Multi-modal elevation mapping for robotics and learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11011–11018. IEEE, 2023.

[23] Marsel Faizullin, Anastasiia Kornilova, Azat Akhmetyanov, and Gonzalo Ferrer. Twist-n-sync: Software clock synchronization with microseconds accuracy using mems-gyroscopes. *Sensors*, 21(1), 2021. ISSN 1424-8220. doi: 10.3390/s21010068. URL https://www.mdpi.com/1424-8220/21/1/68.

[24] Dapeng Feng, Yuhua Qi, Shipeng Zhong, Zhiqiang Chen, Qiming Chen, Hongbo Chen, Jin Wu, and Jun Ma. S3e: A multi-robot multimodal dataset for collaborative slam. *IEEE Robotics and Automation Letters*, 2024.

[25] Lanke Frank Tarimo Fu, Nived Chebrolu, and Maurice Fallon. Extrinsic calibration of camera to lidar using a differentiable checkerboard model. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1825–1831, 2023. doi: 10.1109/IROS55552.2023.10341781.

[26] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286. IEEE, 2013.

[27] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

[28] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[29] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017.

[30] Annika Hagemann, Moritz Knorr, Holger Janssen, and Christoph Stiller. Inferring bias and uncertainty in camera calibration. *International Journal of Computer Vision*, pages 1–16, 2022.

[31] Michael Helmberger, Kristian Morin, Beda Berner, Nitish Kumar, Giovanni Cioffi, and Davide Scaramuzza. The hilti slam challenge dataset. *IEEE Robotics and Automation Letters*, 7(3):7518–7525, 2022.

[32] Xiangcheng Hu, Linwei Zheng, Jin Wu, Ruoyu Geng, Yang Yu, Hexiang Wei, Xiaoyu Tang, Lujia Wang, Jianhao Jiao, and Ming Liu. Paloc: Advancing slam benchmarking with prior-assisted 6-dof trajectory generation and uncertainty estimation. *IEEE/ASME Transactions on Mechatronics*, 2024.

[33] Nicolas Hudson, Fletcher Talbot, Mark Cox, Jason Williams, Thomas Hines, Alex Pitt, Brett Wood, Dennis Frousheger, Katrina Lo Surdo, Thomas Molnar, et al. Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge. *arXiv preprint arXiv:2104.09053*, 2021.

[34] Marco Hutter, Christian Gehring, Andreas Lauber, Fabian Gunther, Carmine Dario Bellicoso, Vassilios Tsounis, Péter Fankhauser, Remo Diethelm, Samuel Bachmann, Michael Blösch, et al. Anymal-toward legged robots for harsh environments. *Advanced Robotics*, 31(17):918–931, 2017.

[35] Edo Jelavic, Dominic Jud, Pascal Egli, and Marco Hutter. Towards autonomous robotic precision harvesting: Mapping, localization, planning and control for a legged tree harvester. *arXiv preprint arXiv:2104.10110*, 2021.

[36] Jianhao Jiao, Hexiang Wei, Tianshuai Hu, Xiangcheng Hu, Yilong Zhu, Zhijian He, Jin Wu, Jingwen Yu, Xupeng Xie, Huaiyang Huang, et al. Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3851–3856. IEEE, 2022.

[37] Juho Kannala and Sami Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28:1335–40, 09 2006. doi: 10.1109/TPAMI.2006.153.

[38] Leica Geosystems. Leica grz101 360° mini reflector. https://shop.leica-geosystems.com/de-DE/survey/accessory/geosystems/leica-grz101-360-degree-mini-reflector/buy. Accessed: 2025-01-20.

[39] Leica Geosystems. Leica nova ms60 multistation. https://leica-geosystems.com/products/total-stations/multistation/leica-nova-ms60, 2020. Accessed: 2025-01-20.

[40] Leica Geosystems AG. Leica BLK2GO - Handheld Laser Scanner, 2024. URL https://shop.leica-geosystems.com/global/leica-blk/blk2go/buy. Accessed: 19 Oct 2024.

[41] Stefan Leutenegger. Okvis2: Realtime scalable visual-inertial slam with loop closure. *arXiv preprint arXiv:2202.09199*, 2022.

[42] Jianping Li, Weitong Wu, Bisheng Yang, Xianghong Zou, Yandi Yang, Xin Zhao, and Zhen Dong. Whu-helmet: A helmet-based multi-sensor slam dataset for the evaluation of real-time 3d mapping in large-scale gnss-denied environments. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[43] Kailai Li, Meng Li, and Uwe D Hanebeck. Towards high-performance solid-state-lidar-inertial odometry and mapping. *IEEE Robotics and Automation Letters*, 6(3):5167–5174, 2021.

[44] Yuanzhi Liu, Yujia Fu, Minghui Qin, Yufeng Xu, Baoxin Xu, Fengdong Chen, Bart Goossens, Poly ZH Sun, Hongwei Yu, Chun Liu, et al. Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments. *IEEE Robotics and Automation Letters*, 2024.

[45] W. Maddern, A. Stewart, C. Linegar, and P. Newman. Real-time kinematic ground truth for the oxford robotcar dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5279–5286, 2016.

[46] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[47] Thien-Minh Nguyen, Shenghai Yuan, Thien Hoang Nguyen, Pengyu Yin, Haozhi Cao, Lihua Xie, Maciej Wozniak, Patric Jensfelt, Marko Thiel, Justin Ziegenbein, and Noel Blunder. Mcd: Diverse large-scale multi-campus dataset for robot perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22304–22313, 6 2024. URL https://mcdviral.github.io/.

[48] Janosch Nikolic, Joern Rehder, Michael Burri, Pascal Gohl, Stefan Leutenegger, Paul T Furgale, and Roland Siegwart. A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 431–437. IEEE, 2014.

[49] Simona Nobili, Marco Camurri, Victor Barasuol, Michele Focchi, Darwin Caldwell, Claudio Semini, and Maurice Fallon. Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots. In *Robotics: Science*

and System XIII, 2017.

[50] Julian Nubert, Etienne Walther, Shehryar Khattak, and Marco Hutter. Learning-based localizability estimation for robust lidar localization. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 17–24. IEEE, 2022.

[51] Julian Nubert, Turcan Tuna, Jonas Frey, Cesar Cadena, Katherine J Kuchenbecker, Shehryar Khattak, and Marco Hutter. Holistic fusion: Task-and setup-agnostic robot localization and state estimation with factor graphs. *arXiv preprint arXiv:2504.06479*, 2025.

[52] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. doi: 10.1109/ICRA.2011.5979561.

[53] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The newer college dataset: Handheld lidar, inertial and vision with ground truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4353–4360. IEEE, 2020.

[54] Milad Ramezani, Kasra Khosoussi, Gavin Catt, Peyman Moghadam, Jason Williams, Paulo Borges, Fred Pauling, and Navinda Kottege. Wildcat: Online continuous-time 3d lidar-inertial slam. *arXiv preprint arXiv:2205.12595*, 2022.

[55] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311, 2016. doi: 10.1109/ICRA.2016.7487628.

[56] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016.

[57] Victor Reijgwart, Cesar Cadena, Roland Siegwart, and Lionel Ott. Efficient volumetric mapping of multi-scale environments using wavelet-based compression. In *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.065. URL https://www.roboticsproceedings.org/rss19/p065.pdf. Code available at https://github.com/ethz-asl/wavemap.

[58] Korbinian Schmid and Heiko Hirschmüller. Stereo vision and imu based real-time ego-motion and depth image computation on a handheld device. In *2013 IEEE International Conference on Robotics and Automation*, pages 4671–4678. IEEE, 2013.

[59] Sevensense Robotics AG. Core research development kit, 2022. URL https://github.com/sevensense-robotics/core_research_manual. Accessed: 19 Oct 2024.

[60] Henrique Simas, Raffaele Di Gregorio, Roberto Simoni, and Marco Gatti. Parallel pointing systems suitable for robotic total stations: Selection, dimensional synthesis, and accuracy analysis. *Machines*, 12(1):54, 2024.

[61] Matthew Sivaprakasam, Parv Maheshwari, Mateo Guaman Castro, Samuel Triest, Micah Nye, Steve Willits, Andrew Saba, Wenshan Wang, and Sebastian Scherer. Tartandrive 2.0: More modalities and better infrastructure to further self-supervised learning research in off-road driving tasks. *arXiv preprint arXiv:2402.01913*, 2024.

[62] Matej Smid and Jiri Matas. Rolling shutter camera synchronization with sub-millisecond accuracy. *arXiv preprint arXiv:1902.11084*, 2019.

[63] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[64] Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, and Maurice Fallon. The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. *arXiv preprint arXiv:2411.10546*, 2024.

[65] Tomas Thalmann and Hans Neuner. Sensor fusion of robotic total station and inertial navigation system for 6dof tracking applications. *Applied Geomatics*, 16(4):933–949, 2024.

[66] Alexander Tiderko, Frank Hoeller, and Timo Röhling. *The ROS Multimaster Extension for Simplified Deployment of Multi-Robot Systems*, pages 629–650. Springer International Publishing, Cham, 2016. ISBN 978-3-319-26054-9. doi: 10.1007/978-3-319-26054-9_24. URL https://doi.org/10.1007/978-3-319-26054-9_24.

[67] Marco Tranzatto, Frank Mascarich, Lukas Bernreiter, Carolina Godinho, Marco Camurri, Shehryar Khattak, Tung Dang, Victor Reijgwart, Johannes Loeje, David Wisth, et al. Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge. *arXiv preprint arXiv:2201.07067*, page 3, 2022.

[68] Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2546–2552. IEEE, 2022.

[69] Maciej Trzeciak, Kacper Pluta, Yasmin Fathy, Lucio Alcalde, Stanley Chee, Antony Bromley, Ioannis Brilakis, and Pierre Alliez. Conslam: Construction data set for slam. *Journal of Computing in Civil Engineering*, 37(3):04023009, 2023.

[70] Florian Tschopp, Michael Riner, Marius Fehr, Lukas Bernreiter, Fadri Furrer, Tonci Novkovic, Andreas Pfrunder, Cesar Cadena, Roland Siegwart, and Juan Nieto. Versavis—an open versatile multi-camera visual-inertial sensor suite. *Sensors*, 20(5):1439, 2020.

[71] Turcan Tuna, Julian Nubert, Yoshua Nava, Shehryar Khattak, and Marco Hutter. X-icp: Localizability-aware lidar registration for robust localization in extreme envi-

ronments. *IEEE Transactions on Robotics*, 2023.

[72] Turcan Tuna, Julian Nubert, Patrick Pfreundschuh, Cesar Cadena, Shehryar Khattak, and Marco Hutter. Informed, constrained, aligned: A field analysis on degeneracy-aware point cloud registration in the wild. *arXiv preprint arXiv:2408.11809*, 2024.

[73] Maxime Vaidis, Mohsen Hassanzadeh Shahraji, Effie Daum, William Dubois, Philippe Giguère, and François Pomerleau. Rts-gt: Robotic total stations ground truthing dataset. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17050–17056. IEEE, 2024.

[74] Steve T Watt, Shankar Achanta, Hamza Abubakari, Eric Sagen, Zafer Korkmaz, and Husam Ahmed. Understanding and applying precision time protocol. In *2015 Saudi Arabia Smart Grid (SASG)*, pages 1–7. IEEE, 2015.

[75] Xuankang Wu, Haoxiang Sun, Rongguang Wu, and Zheng Fang. Everysync: An open hardware time synchronization sensor suite for common sensors in slam. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12587–12593. IEEE, 2024.

[76] Zhe Zhang, Shaoshan Liu, Grace Tsai, Hongbing Hu, Chen-Chi Chu, and Feng Zheng. Pirvs: An advanced visual-inertial slam system with flexible sensor fusion and hardware co-design. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3826–3832. IEEE, 2018.

[77] Shibo Zhao, Yuanjun Gao, Tianhao Wu, Damanpreet Singh, Rushan Jiang, Haoxiang Sun, Mansi Sarawata, Yuheng Qiu, Warren Whittaker, Ian Higgins, et al. Subt-mrs dataset: Pushing slam towards all-weather environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22647–22657, 2024.

Appendix

In the appendix, we provide further details on the design and development of *Boxi*.

Acknowledgments

Jonas Frey and Julian Nubert are supported by the Max Planck ETH Center for Learning Systems.

This work was supported and partially funded by Leica Geosystems, which is part of Hexagon. We would like to express our gratitude to Benjamin Müller, Benjamin Schöll, Sprenger Bernhard, Eisenreich Stefan, and Jonas Nussdorfer for their invaluable expertise in networking, calibration, time-synchronization, AP20, and their assistance in the collection of calibration data. We gratefully acknowledge the support of NovAtel Inc., Hexagon - Autonomy & Positioning division, with special thanks to Ryan Dixon for his contributions.

In addition, this work was supported by the National Centre of Competence in Research Robotics (NCCR Robotics), the ETH RobotX research grant funded through the ETH Zurich Foundation, the European Union's Horizon 2020 research and innovation program under grant agreement No 101016970, No 101070405, and No 101070596, and an ETH Zurich Research Grant No. 21-1 ETH-27. This work was supported by the Open Research Data Grant at ETH Zurich. We acknowledge that the International Foundation High Altitude Research Stations Jungfraujoch and Gornergrat (HFSJG), 3012 Bern, Switzerland, made it possible for us to carry out our experiment(s) at the High Altitude Research Station at Jungfraujoch. We also thank the custodians, Mrs. Daniela Bissig and Mr. Erich Bissig, guide Ms. Doris Graf Jud (Jungfraubahnen), and Annette Fuhrer (Jungfraubahnen) for the support of our activities during the Jungfraujoch trip.

Our sincere appreciation goes to William Talbot for his work in integrating the ADIS IMU. We also thank Elena Krasnova for her assistance with Blender and for helping develop the new roll cage. We are grateful to Flurin Schindele for his efforts in integrating the Box onto various robots, and to Victor Reijgwart for his assistance with integrating Wavemap. We extend our thanks to Johann Schwab for his support in integrating the TierIV C1 cameras, and to Cyrill Püntener for his expertise in network configurations. We thank Ulrich Huemer for his design of the mechanical subsystems within *Boxi*, and Thomas Mantel for his work on the mechanical integration of the Livox MID360 and Prism. Lastly, we thank Laurin Schmid and Andreas Binkert for their contributions to the integration of various 3D-printed parts.

A. Dataset Descriptions

In this section, the details of each dataset is provided.

- *Hike:* The robot descends and then ascends on loose gravel in a very large environment where optimization degradation [72] might occur. The robot traverses $330\,\mathrm{m}$ in $405\,\mathrm{s}$.
- *Warehouse:* This dataset consists of the robot walking within a confined warehouse starting from the outdoor court. The environment features items, shelves, and a roll-up door for feature-based state estimation problems. The robot traverses $202\,\mathrm{m}$ in $370\,\mathrm{s}$.
- *Research Station:* Acting as a simple dataset, the robot walks outside a research station featuring industrial metal stairs on an elevated metal grid platform. In this dataset, the robot traverses $210\,\mathrm{m}$ in $262\,\mathrm{s}$.
- *Excavation Site:* In this outdoor dataset, the robot walks around an operating excavator. This dataset contains large stones, water pits, small dirt roads, and construction site containers, which bring unique features. The robot traverses $262\,\mathrm{m}$ in $200\,\mathrm{s}$.
- *Terrace:* This dataset covers the robot walking from a cog railway station onto a terrace towards a large university building where limited features are available for short-range sensors while features are in abundance for large-scale perception. The robot traverses $200\,\mathrm{m}$ in $283\,\mathrm{s}$ seconds.
- *Demolished Building:* Walking from the outside into a search and rescue training facility through a demolished building with an open staircase and over multiple floors. The robot traverses $290\,\mathrm{m}$ in $368\,\mathrm{s}$.
- *Mountain Ascent:* The robot starts by walking the scaffolding stairs, crossing a train station, and then traversing a narrow, very steep hiking path in sunny conditions, which presents challenges based on exposure. The robot traverses $300\,\mathrm{m}$ in $384\,\mathrm{s}$.

B. Reference Ground Truth Pose Generation

The availability of a precise 6DoF ground truth robot poses as a reference is critical for evaluating many robotic applications, such as state estimation and SLAM performance. Contrary to many previous works, our deployments are indoors and outdoors, spanning hundreds of meters, and require millimeter-level precision to evaluate LiDAR-based state estimation methods. These requirements directly render LiDAR-based and only RTK-GNSS based references unsuitable [28, 9, 77]. While using Motion Capture (MOCAP) systems can provide high precision ground truth pose estimates at high frequency, these systems are restricted to a small workspace [10, 24], making them unsuitable for large-scale deployments. Many approaches [53, 64, 69, 47, 42] rely on stationary scanners to create an accurate representation of the environment and register LiDAR measurements against these maps. However, the registration accuracy is bounded by the sensor noise characteristics itself, the accuracy of the map, and the registration algorithm [47]. Moreover, iterative registration methods can not retrieve accurate poses in LiDAR-degraded environments [71, 50], and the generation of accurate maps requires significant manual effort [77] or may not even be feasible in challenging dynamic environments, such as a forest on a windy day. Recent works [73, 65, 18] used one or more TPS to obtain ground truth position estimates with millimeter-to-sub-millimeter accuracy [60]. A TPS is a device commonly employed in construction and surveying to measure the 3D position of a reflector target. By mounting a prism

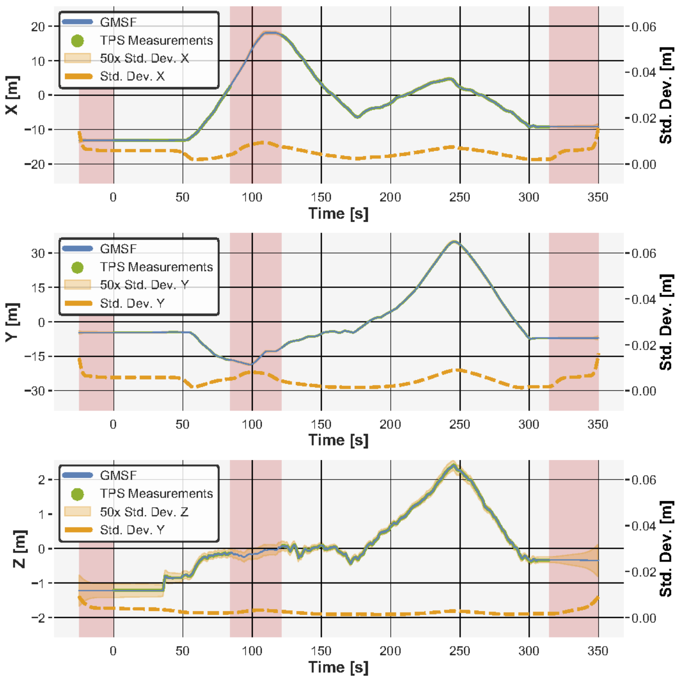

Fig. 18: Ground truth trajectory generated with Holistic Fusion (HF) for the *Excavation Site* dataset is shown with the covariance-based uncertainty region overlaid. Given that the uncertainty is on a millimeter scale, we display 50x the standard deviation as an overlay and provide the numerical value on the right axis in the dotted orange line. The red regions indicate TPS measurement dropouts.

target on a robot TPS measurements of the robot's position can be acquired during direct line-of-sight from the TPS to the robot. To overcome the direct line-of-sight limitation, our solution fuses TPS measurements with precise IMU and GNSS measurements to obtain 6DoF ground truth poses. Specifically, we use a single Leica Geosystems MS60 Total Station [39], and the Leica Geosystems 360° Mini-Prism GRZ101 [38], which has a static accuracy of around $\pm 1.5\,\mathrm{mm}$ (3σ) within 360° heading rotation and a pitch angle of $\pm 30°$. To stream the data up to $20\,\mathrm{Hz}$ from the TPS to *Boxi*, Leica Geosystems provided the Bluetooth radio handle RH18 together with an AP20 special edition. The AP20 runs a special firmware (not commercially available), which enables streaming of time-synchronized TPS measurements with time-sync jitter of $\pm 1\,\mathrm{ms}$ (3σ).

The acquired TPS measurements are then fused with HG4930 IMU measurements, and post-processed GNSS poses. The GNSS solution is provided by the proprietary NovAtel Precise Positioning Product called Inertial Explorer [2] and the NovAtel SPAN CPT7 GNSS receiver mounted on *Boxi*. Using accurate atmospheric corrections, Inertial Explorer can provide a tightly-coupled pose solution with accuracies up to $1\,\mathrm{cm}$ [2], which we will refer to as IE-TC. Inertial Explorer also produces loosely coupled (IE-LC) and a precise point positioning (IE-PPP) solution; NovAtel recommended that in most cases IE-TC is most precise solution. To fuse the TPS measurements and the IE-TC estimates, we first solve an online and then, an offline factor-graph optimization problem to obtain the fused highly accurate 6DoF robot pose, following a similar approach to PaLoc [32]. We build on the publicly available Holistic Fusion (HF) framework [51] and refer to the offline factor-graph optimization output as our ground truth robot pose estimation. Details and code will be made available open-source.

Method	RTE $\mu(\sigma)$[m]	ATE $\mu(\sigma)$[m]
HF-Live	0.0033(0.0028)	0.0117(0.0305)
HF-Optimized w/o IE	**0.0027**(**0.0021**)	**0.0021**(**0.0012**)
HF-Optimized	**0.0027**(**0.0021**)	**0.0021**(**0.0012**)
IE-TC	<u>0.0033</u>(<u>0.0023</u>)	0.0159(<u>0.0069</u>)
IE-LC	<u>0.0033</u>(0.0023)	<u>0.0149</u>(0.0072)
IE-PPP	0.0356(0.0648)	0.162(0.0118)

TABLE VIII: ATE and RTE (per $1\,\mathrm{m}$ distance traveled) for the *Excavation Site* dataset are shown (best in **bold**), and the second best is <u>underlined</u>. HS indicates the output of Holistic-Fusion in different operation modes.

1) Ground Truth Verification: For verification, we compare our ground truth poses against the TPS measurements. It is important to note that this analysis is conducted to show that the sensor fusion and factor-graph optimization steps do not alter the alignment accuracy against the TPS measurements but extend it with the HG4930 IMU measurements and the IE-TC odometry. As a result, the Holistic-Fusion-Optimized method is expected to produce the best position estimates against TPS measurements because it has privileged access to it. Here, extension refers to estimating the robot pose instead of a position even during the dropout of the TPS measurements.

The analysis is conducted for the *Excavation Site* dataset, which has consistent GNSS coverage and TPS measurements throughout the deployment except for the beginning, the end, and $84\,\mathrm{s}$ to $121\,\mathrm{s}$ time period of the mission. The error metrics ATE and RTE are obtained using evo [29]. Crucially, as we compare the accuracy against the TPS measurements, the results provided in Table VIII do not reflect the accuracy of position during the dropout periods of the TPS measurements. Table VIII shows that the HF-Optimized and HF-Optimized w/o IE solutions perform identically and produce the least error against the TPS measurements. This is expected since we configured the Holistic Fusion (HF) framework to rely on the TPS measurements when available. As seen, the HS-Live method performs worse than HF-Optimized, which builds a graph or slow batch optimization to minimize the error. Moreover, both the IE-LC and IE-TC methods have higher ATE errors than Holistic-Fusion, which uses highly accurate TPS measurements. Lastly, IE-PPP performs the worst as this method relies only on the satellite-based GNSS correction. In terms of RTE, which is a measure of trajectory smoothness, the methods perform similarly, while HF-Optimized methods perform slightly better.

During the previously mentioned drop out of TPS measurements, the HF-Optimized relies on the IE-TC poses instead of TPS. To highlight this case, the HF-Optimized output is compared against the IE-TC output, and the occlusion period is highlighted in red. Figure 18 shows the generated trajectories. As seen, the *marginal* covariance estimation correlates with the mismatch between the TPS and IE-TC measurements, and the uncertainty increases in the absence of the TPS measurements, implying deviation from the absolute true position. The marginal covariance is a measure of how well the measurements align with each other and not the true covariances.

C. Sevensense CoreResearch

The Sevensense CoreResearch is a global shutter camera unit introduced in 2020 [59]. The unit supports up to 8 cameras, with a resolution of $0.4\,\mathrm{MP}$ or $1.6\,\mathrm{MP}$. It is connected via Gigabit Ethernet (1000BASE-T, IEEE 802.3ab) and PTP time-synchronized. The cameras support hardware triggering full exposure time control and synchronization between cameras. For *Boxi*, we chose to use $5 \times 1.6\,\mathrm{MP}$ cameras (1440×1080) with three color and two monochrome cameras, with the two monochrome cameras positioned in a stereo setup. All cameras are equipped with $2.4\,\mathrm{mm}$ focal length $165.4°$ horizontal FoV lenses and run at a maximum rate of around $10\,\mathrm{Hz}$ due to bandwidth limitations on the Ethernet interface despite efforts to optimize it. This unit has been proven reliable and highly valued in academic research, being used for the latest SuBT Challenge [67] and popular datasets [31, 53, 64]. The manufacturer provides well-documented, closed-source ROS1 drivers, but ROS2 drivers are not yet available. Since the provided driver also lacked support for debayering, white balancing, and color correction, we developed an in-house solution to address this.

D. StereoLabs ZED2i

The StereoLabs ZED2i is a $12\,\mathrm{cm}$ baseline stereo rolling shutter camera unit introduced in 2023 marketed specifically for outdoor robotic operation. Through the ROS1/2 drivers provided by StereoLabs, the camera provides classical and learning-based depth estimation, but it requires an NVIDIA GPU (CUDA support) for operation. In addition, the driver provides barometric and thermal measurements at $100\,\mathrm{Hz}$ and IMU measurements at $400\,\mathrm{Hz}$ during real-time operation. The camera is connected via USB 3.0 technology, and thus, time-synchronization to the host PC (USB buffer arrival timestamp) or hardware triggering is not supported. It can provide image resolution up to 2K (2208×1242) and comes with different lens options. We chose the wide FoV $110°$ horizontal version with a $2.1\,\mathrm{mm}$ focal length lens. Crucially, the driver provided by StereoLabs allows video encoding in H.264, H.265, or lossless formats and supports serialized recording of the measurements with exceptional efficiency. As an exception, the serialized format can record the IMU measurements only up to a rate of $45\,\mathrm{Hz}$ as of SDK version 4.2.5.

E. TierIV C1

The Tier IV C1 is a $120\,\mathrm{dB}$ high dynamic range and rolling shutter camera with $2.5\,\mathrm{MP}$ chip resulting in (1920×1280) images at $30\,\mathrm{Hz}$ with a horizontal FoV of $120°$. The camera can be connected through GMSL2. The exact read time per line of the image is provided by the manufacturer, and the camera can be hardware-triggered via an f-sync signal. The camera performs automatic white-balancing and TierIV provides calibration tools as well as a custom lens correction model. Integration of the camera requires a sophisticated understanding of the Linux device tree and kernel modules to set up hardware triggering.

F. LiDARs

Given the above attributes, we integrated the Livox Mid360, a research community favorite for its affordability and unique scan pattern, making it ideal for near-field mapping due to its lower range ($40\,\mathrm{m}$) and high disparity (multiple cm at $20\,\mathrm{m}$). For mapping and localization, we integrated the Hesai XT32 sparse LiDAR, with a range of $0.5\,\mathrm{m}$ to $120\,\mathrm{m}$ and high point accuracy of $1\,\mathrm{cm}$. Both LiDARs have well-maintained drivers, PTP synchronization, intensity returns, and Ethernet connectivity. Both LiDARs are configured to aggregate measurements over a $100\,\mathrm{ms}$ period and provide point clouds at $10\,\mathrm{Hz}$. A comparison of various spinning LiDARs can be found in [12] and [43] compares a solid-state LiDAR to a spinning LiDAR.

G. IMUs

For *Boxi*, we choose to integrate three tiers of IMUs. Two (high-end) tactical-grade MEMS IMUs, namely the HG4930 and the STIM320. For the mid-range IMU, we chose the ADIS16475-2 IMU, where the predecessor model has been used for the collection of popular existing datasets [10]. The AP20-IMU is also a mid-range industrial grade IMU, however it only provides measurements, while connected to the Leica Geosystems MS60 Total Station via Bluetooth. The Livox, CoreResearch, and the ZED2i camera all ship with integrated low-cost, consumer-grade IMUs (see. Table II). We operate the IMUs on *Boxi* at a range from 100-500 Hz, excluding the ZED2i IMU dropout due to serialization.

H. GNSS

We use the NovAtel SPAN CPT7 industrial inertial navigation solution, which, though expensive (see Table II), offers a compact design, features one of the best commercially available MEMS IMU (see HG4930 in Table II), good ROS1 drivers, and provides Ethernet connectivity. The tactile grade HG4930 IMU within the CPT7 is pinned down, and the CPT7 internally uses the factory extrinsic calibration provided by Honeywell. Critically, CPT7 functions as *Boxi*'s Grandmaster clock of the PTP network, internally synchronizing with the received GPS time when available. In addition to providing the current GNSS solution and IMU measurements in real-time, the CPT7 allows for onboard data logging. The log files can be post-processed using Inertial Explorer to correct the GNSS measurements in hindsight through precise multi-constellation atmospheric corrections, obtained from the Leica Geosystems HGxN SmartNet GNSS correction service. Furthermore, we use NovAtel TerraStar-C Pro Multi-Constellation Correction PPP solution, which requires L-Band satellite communication to provide accurate pose estimation without relying on base station measurements during deployment.

I. Network Switch

According to these criteria, we selected the cost-efficient UbiSwitch Module with UbiSwitch Baseboard that supports three 10GBASE-T and 8 1GBASE-T standard connections. This switch meets most requirements, specifically the excellent form factor and rugged design. However, it lacks IEEE 1588v2 support (see Section VII).

J. Compute

For handling CPU-intensive tasks, we chose the industrial AsRock Intel i5 NUC-1340P/D5, which offers a strong balance between performance and power efficiency, with excellent single-core performance; the ITX motherboard supports dual $2.5\,\mathrm{Mbps}$ Ethernet ports, $64\,\mathrm{GB}$ of RAM, and a $4\,\mathrm{TB}$ capacity M.2 form factor SSD.

For GPU-based computing, we selected the NVIDIA AGX Jetson Orin compute module paired with the ConnectTech Rogue Carrier Board and the ConnectTech GMSL2 adapter board. The carrier board is equipped with dual $10\,\mathrm{Gbps}$ Ethernet, two USB 3.2 connections, $64\,\mathrm{GB}$ unified RAM, and a $2\,\mathrm{TB}$ M.2 SSD. However, despite supporting I2C, SPI, and dual UART interfaces, the ConnectTech Rogue Carrier Board exposes an insufficient number of interrupt pins for hardware timestamping. To address this gap, we integrated a Raspberry Pi Compute Module 4 to handle timestamping for two IMUs, manage status LEDs and buttons, and control fan speed. All computers support PTP time synchronization to efficiently synchronize with all sensors that support the protocol. We emphasize that our compute choices should not be directly applied to commercial systems without careful consideration. In commercial applications, where custom software development and other operating systems are feasible, e.g. more power-efficient ARM-based chipsets may be preferable. In hindsight, despite the better form factor of the ConnectTech carrier board, it would have been easier to integrate Nvidia's Developer Kit, reducing the number of computers to a minimum.

K. Cookbook: Time Synchronization Basics

To understand the relationship between time synchronization and position estimation, we compute the resulting position error for different time synchronization errors for robots moving at different constant velocities, under the assumption of a fully observable state. For a robot moving at $1\,\mathrm{m\,s^{-1}}$, achieving millimeter-level position estimation accuracy requires time synchronization errors below $10\,\mu\mathrm{s}$. The position estimation error e due to time synchronization error Δt can be approximated as: $e = v \cdot \Delta t$, where v is the robot's velocity and Δt is the time synchronization error. Similar computations can be performed for the angular velocity. As a further first principle example of the effects of time-offsets, we consider a LiDAR measurement at a distance of $25\,\mathrm{m}$ while the robot rotates at an angular velocity of $200\,^{\circ}\,\mathrm{s^{-1}}$. A timestamp offset of $1\,\mathrm{ms}$ will lead to an offset of $0.2°$ or around $8.7\,\mathrm{cm}$ error. In comparison, the vertical beam divergence of the Hesai is $0.047°$ horizontally.

Therefore, considering communication to sensors, delays in TCP/IP networks, USB controllers, serialization, and kernel scheduling can easily exceed the time precision required for millimeter-precise localization. This highlights the clear need for hardware timestamping of all sensor measurements and is particularly crucial for the long-range LiDARs, where each point must be accurately time-stamped.

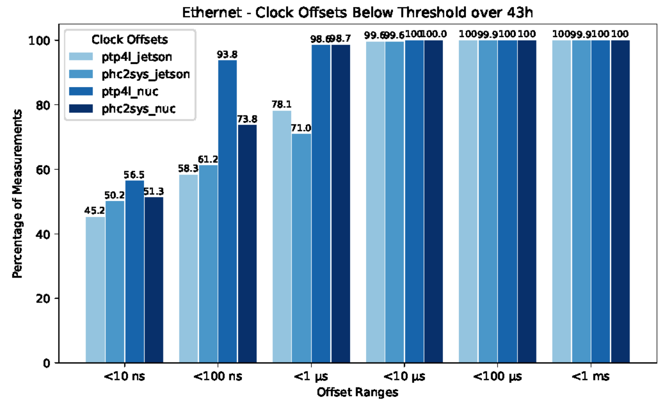

Fig. 19: PTP synchronization experiment of Jetson and Nuc synchronizing against CPT7 (the PTP Grandmaster). We report `ptp4l` and `phc2sys` offsets calculated as a percentage of samples within a given time synchronization range.

L. Cookbook: Time Synchronization PTP

To analyze the synchronization between the PCs we logged the offsets of `ptp4l` and `phc2sys` every second and summarized the results in Figure 19. The `ptp4l` offset represents the difference between the built-in clock of the network interface card (NIC) of the respective Ethernet port and the CPT7 Ethernet interface Grandmaster clock. The `phc2sys` offset, on the other hand, measures the discrepancy between the Ethernet interface clock and the system time of the respective PC. This `phc2sys` offset consistently remains below $10\,\mu\mathrm{s}$. The absence of IEEE 1588v2 standard synchronization can, in rare cases of packet loss, lead to incorrect estimations of the line delay. This, in turn, may result in improper clock adjustments, manifesting as outlier measurements seen in the `phc2sys` offset for the Jetson exceeding $1\,\mathrm{ms}$ in rare events. Peak offset readings do not inherently indicate poor clock synchronization, as the proportional-derivative (PD) controller compensates for such offsets. Similarly, inaccurate line delay measurements do not directly imply a failure in time synchronization. Our experiment remained within the normal bandwidth limits of the network switch, so the impact of bandwidth saturation was not assessed. While integrating a low-drift MasterClock was considered, it was excluded due to CPT7's inability to function as a time receiver and overall weight constraints.

M. Cookbook: Time Synchronization IMUs

Time synchronization for the STIM320 IMU is handled via custom kernel modules that use GPIO interrupts triggered on falling edges (`IRQF_TRIGGER_FALLING`). Each interrupt invokes a kernel-space ISR that captures precise timestamps using `ktime_get_real_ts64()` and stores them in a ring buffer. These timestamps are accessed from userspace via a *sysfs* interface, allowing safe retrieval and clearing by the front-end driver for integration with ROS. This setup requires kernel-level expertise in GPIOs, IRQ handling, timestamping, and *sysfs* design. Fixed sensor latencies, such as low-pass filter delays, must be compensated manually. In hindsight, synchronizing the IMU to an external trigger would offer cleaner integration than free-running operation. By contrast, the ADIS IMU leverages the Linux IIO subsystem, which natively supports timestamping. However, enabling this functionality on the Raspberry Pi Compute Module 4 requires non-trivial kernel modifications and device tree configuration.

N. Cookbook: Time Synchronization Validation

We developed a time synchronization verification tool that calculates the time offset between data streams of IMU measurements. It first transforms the angular velocity readings of the IMUs into a common reference frame by compensating for different orientations using their respective extrinsic calibration. In the second step, a single axis of the transformed measurements is selected, and the signal is linearly interpolated to a frequency of $500\,\mathrm{Hz}$, the maximum rate collected with *Boxi*. A correlation analysis between the two angular velocity signals provides an initial time offset estimate between the IMUs. We then refine this estimate using gradient-based optimization. This involves iteratively interpolating the original measurement signals based on the current time-offset estimate. We minimize the mean squared error (MSE) between the two interpolated IMU signals until convergence.

Initially, we conceived this tool as a proof of concept to verify that IMU sensor readings fall within the expected IMU rate intervals of $10\,\mathrm{ms}$ based on the HG4930 reference IMU operating at a $100\,\mathrm{Hz}$.

The results of applying this tool across all recorded missions is provided in the main paper Table VI. We applied the tool to three $30\,\mathrm{s}$ snippets per axis for each mission to estimate the standard deviation. Since the missions were recorded across seven different days, we assume that any remaining biases originate from the internal IMU-specific triggering time. Surprisingly, we find that we can estimate the IMU offset with high accuracy, achieving an average sigma of $0.5\,\mathrm{ms}$ when averaging the estimated offset across runs, different axes, and different time intervals across deployments. The remaining time offsets most likely stem from the variations in the triggering of the IMU exposure time and internal IMU delays. This analysis highlights the importance of having robust verification tools to ensure high-quality sensor data and to understand biases within the data. The STIM320 IMU is excluded from the above analysis as during testing, it was observed that the userspace front-end might have an initial time offset while reading the *sysfs* virtual file. This offset is constant for each sequence and is characterized by the period of the IMU $2\,\mathrm{ms}$. This offset is compensated in hindsight with the tool discussed in this section.

O. Cookbook: Hardware

1) Vibrations: During operation, *Boxi* experiences constant vibrations due to the radial fan and the motion of the legged robot. Using vibration-damping mounts or isolators between the base platform and *Boxi* is prohibited by the fact that the kinematic chain must remain accurately observable. Internal vibrations can also arise from rotating components, specifically the radial fan and internally spinning LiDARs. These vibrations can interfere with the readings from the IMUs; therefore, securely mounting all components is essential. Similarly, screw-in or clip-in connectors can help prevent loose connections and prevent component failures.

2) Weatherproofing: To ensure weatherproofing, we incorporated an O-ring between the main body of *Boxi* and the lid. A modular IP67-rated cable passthrough proved to be particularly useful for the flexible integration of the components on the lid. The entire enclosure is air-tight; however, to prevent water ingress due to condensation caused by temperature fluctuations between the interior and exterior of the box, we added a pressure relief valve. In previous hardware designs, we experienced short circuits resulting from this condensation, which this design aims to mitigate.

3) Protection: We generally recommend performing a hazard analysis before the mechanical design of the payload. This analysis aims to identify risks early in the design process and, in turn, helps to design mitigation strategies. Generally, the later risks are identified, the more burdensome their mitigation becomes. To protect *Boxi* from impacts, we added a 3D-printed structure around the front and the side. The 3D structure specifically protects the exposed lenses from damage. The current dual-purpose roll-over cage protecting the LiDARs is insufficient to absorb the force generated by the robot falling. A more sophisticated protection concept could introduce a decoupled roll-over cage directly mounted to the robot or an intended breaking point between the robot and *Boxi*.

Protection - Recipe

☑ Hazard Analysis (*Fall, Shock, Water, Temperature*)
☑ Mitigation Plan (*Cage, Lenses, Damping, IP-rating*)

P. Calibration: Camera Intrinsic and Extrinsic

1) Kalibr Ceres Online Version: Initially, we experimented with the off-the-shelf tool Kalibr [26, 56] but were quickly set back by three main challenges: *i)* the batched nature of the calibration formulation, along with the information gain computation at each batch iteration made the optimization problem prohibitively slow for our use case, taking up to $6\,\mathrm{h}$; *ii)* given the wide coverage of our cameras, sub-batches of our data rarely contained observation data from all cameras, and the optimization problem typically diverged; *iii)* the offline nature of the workflow required transfer of around $100\,\mathrm{GB}$ of redundant uncompressed image data which was a bottleneck.

Guided by these challenges, we developed our own calibration workflow that provides live feedback on the validity of the calibration data collected, does not store redundant calibration data, and runs in real-time during data collection.

To make calibration feasible on-board, we delegated tasks across the running computers. The Jetson and NUC platforms that run the camera drivers would also perform calibration target detection. Only the target information is transferred

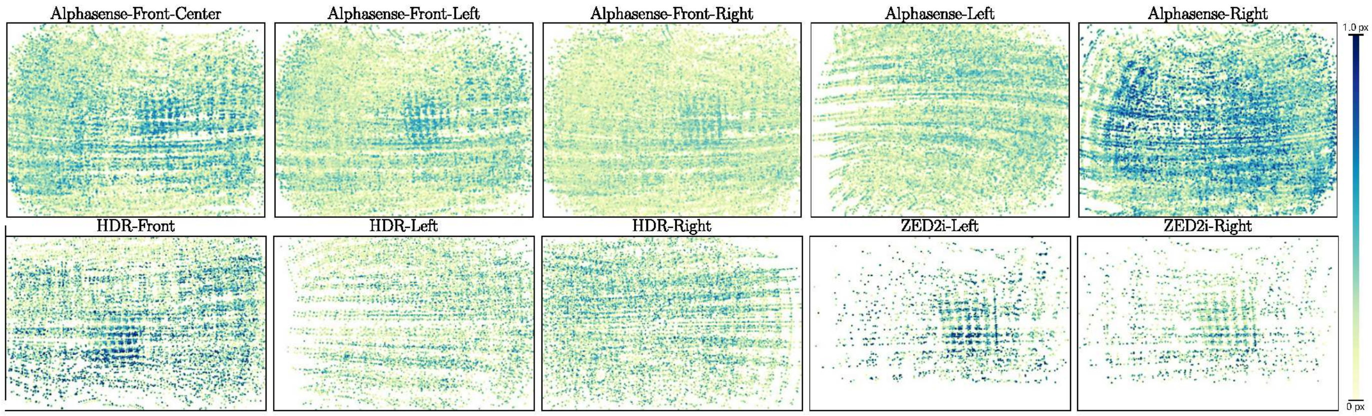

Fig. 20: Heatmap of reprojection errors across all cameras. The consistent sub-pixel reprojection array throughout the image plane verifies the selection of the respective distortion model. Units are in pixels.

to the operator PC, where the calibration optimization and live user visual feedback are run. Additionally, the calibration target detector nodes log the raw images where the calibration target was detected.

At initialization, the calibration routine estimates only the individual camera intrinsics. To achieve this, data is accumulated from all the cameras as the user moves the calibration target within the FoV of the camera bundle. Simultaneously, a separate visualization program shows a heatmap of the regions in each image that have already received calibration data. Once the program receives sufficient new observations (in this case, 300 corners), a non-linear least squares optimization, implemented using Ceres Solver [4], is run to solve for the camera intrinsic parameters using a pre-determined initial guess of the parameters.This phase temporarily halts the processing of new target detections, displaying on-screen feedback to inform the user that the program is currently busy running an optimization process. To keep the interaction close to real-time, the optimization program is run for a maximum of three iterations.

After solving the optimization problem, we estimate the diagonal of the covariance of the intrinsic parameters and the reprojection errors for each camera. The user receives feedback on this information through a 2D plot of the reprojection errors and a bar plot of the uncertainty of the intrinsic parameters. Once the maximum standard deviation of the projection parameters is less than two pixels across all cameras, the calibration program proceeds and is ready for extrinsic calibration.

Unlike the data used for intrinsic calibration, the data for extrinsic calibration requires the calibration target to remain static relative to the sensors. To ensure this, we detect a static target if its previous and next positions show less than $1\,\mathrm{mm}$ of movement. To synchronize between the cameras, we use a ring buffer to store the last 10 detected target positions, timestamps, and detections for each camera.

After the joint intrinsic/extrinsic optimization, we estimate the diagonal of the covariance of all intrinsic and extrinsic parameters, and report these to the user. The user is recommended to collect calibration data such that the estimated square root of the diagonal of the covariance matrix is under $1\,\mathrm{mm}$.

To reduce the amount of redundant calibration data collection, we first experimented with an information gain-based threshold but found that the computation of the information matrix was too time-consuming to use effectively on-board, even when using sparse matrix implementations [19]. Instead, we opted for an approximate but effective measure of information gain. We divided the image into 3×3 blocks, allocating to each block a maximum capacity of 10 corner observations. When a new observation is received (each observation contains up to 144 corner locations), the corners are mapped to their designated blocks, and if any of the mapped blocks still have the capacity for new observations, all corners of this new observation are added to the respective blocks. Note that this means that each block can exceed its set maximum capacity.

2) Calibration Target: To facilitate the convenient collection of calibration corner data close to the edges of the cameras with high distortion, we used a calibration target formed of a planar grid of AprilTag targets [52] (also known as an AprilGrid), which enables partial detection of the calibration target. The 6×6 of square patterns provide up to 144 corner detections. With squares on the sides $8.3\,\mathrm{cm}$, the target is confidently detected in the range of the calibration data collected.

3) Camera Calibration Reprojection Errors: The reprojection errors of the detected corners across all cameras are provided in Figure 20. The projection errors are, on average, below $0.5\,\mathrm{px}$ and the samples cover the full FoV.

Q. Calibration: Camera to IMU

An operator performs the calibration motion manually, smoothly moving the cameras along all three axes of translation and rotation while maintaining the view of the calibration target in the cameras. This calibration is done in post-processing, as the calibration target detector is not fast enough to run in real-time without dropping some of the images streaming at $17\,\mathrm{Hz}$.

Moreover, the STIM320 IMU has individual accelerometer axes, so we calibrate its extrinsic parameters relative to the camera accounting for the *axis-scale* using the extension to Kalibr [55]. All other IMUs are individually calibrated relative to the camera bundle, without refinements to the intra-camera

Fig. 21: Overlay of the LiDAR point cloud while moving forward at $1\,\mathrm{m\,s^{-1}}$, showcasing the consistency of the sensor calibration and the motion undistortion of the point cloud.

extrinsic parameters. The IMUs are placed in very different locations across the sensor suite, with the HG4930 roughly $30\,\mathrm{cm}$ away. From the individual calibrations, we can verify that the estimated relative transform between STIM320 and the HG4930 is consistent with the pinned-down CAD measurements.

R. Calibration: LiDAR to Camera Extrinsic

For this calibration, we used a different calibration target formed of a checkerboard pattern with 8×9 squares of size $8\,\mathrm{cm}$ with a special surface finish. Crucially, the print finish of the calibration target admits an intensity contrast in the $905\,\mathrm{nm}$ wavelength of the laser beams emitted by the LiDARs, as seen in Figure 17. Each LiDAR is individually calibrated against the 5 CoreResearch cameras. Using the calibrated extrinsic parameters between the cameras, the detected calibration target poses from each camera are fused and jointly calibrated with respect to each LiDAR, without refining the intra-camera extrinsic parameters.

The calibration data is collected by placing the target statically in various poses around the sensors in the overlapping area between the FoV of the LiDARs and the CoreResearch cameras. Our calibration program automatically detects these static segments by assessing the camera-detected calibration target motion. When the inter-frame motion is under $1\,\mathrm{mm}$, the segment is deemed static. Since the Hesai LiDAR has a sparse scanning pattern along the vertical axis, the calibration target is placed diagonally in certain poses as seen in Figure 17, to better constrain the alignment of the vertical axis of the LiDAR to the camera. As the Livox LiDAR has a non-repeating pattern, we accumulate all its points in a given static segment to create a single dense point cloud to align to the camera detections. This data collection typically takes under $10\,\mathrm{min}$.

1) LiDAR to Camera Overlay: The established time synchronization, camera intrinsic, camera-to-LiDAR, and camera-to-IMU extrinsic calibration allow us to motion-compensate the LiDAR points and accurately project them onto all CoreResearch images. In Figure 21, an example scene showcases the excellent data association that can be achieved across all directions during motion.

In our evaluation of image recording, we considered different alternatives, each offering distinct advantages and limitations. Firstly, recording images in raw format was impractical due to the large storage requirements. Lossless PNG compression, while preserving image quality, proved too computationally expensive for real-time processing of large images. MJPEG, being more efficient, offered a higher compression rate (typically around 10:1, depending on quality settings), and benefited from native support in ROS. Lastly, H.265 video compression can further decrease storage requirements up to a factor of 100 and leverage hardware acceleration on the Jetson, which significantly reduces the CPU workload.

For all CoreResearch cameras, we opted for standard ROS1 JPEG compression due to the NUC's inability to support hardware video encoding, while still being capable of handling the CPU compression workload. A notable drawback of this method is that images must be transferred through the ROS1 TCP/IP layer before being stored, introducing significant overhead due to serialization and deserialization processes. While there are more sophisticated compression methods that can run on CPU, this approach was necessary as we lacked access to the camera drivers to maximize efficiency in storing serialized image data. Similarly, for the HDR cameras, we also applied JPEG compression, but we used ROS2, which allowed us to leverage interprocess communication (IPC). This enables efficient storage of compressed images directly into MCAP files, highlighting one of the key advantages of ROS2 for our application. Lastly, for the ZED2i camera, we used the StereoLabs API to record highly serialized and efficient SVO2 files, benefiting from hardware-accelerated H.265 video compression with an SSIM $97.3\,\%$ providing the best image quality and compression ratio while using the least resources.