Standard Datasets for Autonomous Navigation and Mapping: A Full-Stack Construction Methodology

Yuanzhi Liu, Yujia Fu[†], Minghui Qin[†], Yufeng Xu[†], Bin Cui, Kunhua Liu, Fengdong Chen, Wei Tao, Michiel Vlaminck, Bart Goossens[*], Poly Z.H. Sun, and Hui Zhao[*]

***Abstract*—The development of intelligent Vehicles (IVs) requires extensive standard datasets for training, benchmarking, and improvement. Autonomous Navigation and Mapping (ANM), as a critical technology for IVs, imposes exceptionally high demands on dataset construction. This is significant in its requirements for comprehensive sensor calibration, precise time synchronization, and accurate generation of ground truth. Besides, the whole construction workflow also demands intricate knowledge and sophisticated practices, necessitating lengthy learning curves for researchers to attain proficiency. The above challenges have led to a slow production of qualified datasets, directly constraining the advancement of ANM. However, so far, an investigation focused on a mature construction methodology of ANM dataset is still missing. This paper strives to fill the gap. Specifically, based on our systematic reviews and extensive practices, for the first time, a full-stack construction methodology of ANM dataset is proposed, including modules of platform construction, sensor calibration, time synchronization, ground truth generation, synthetic data production, and benchmark criteria, with detailed techniques and methodological routes provided in each step. Several long-standing issues are resolved within the methodology. Importantly, we introduce versatile calibration and synchronization frameworks that attain up to us-level and mm-level precision. Besides, we propose a full-scenario ground truth system that can generate scene-map and trajectory at cm-level accuracy. To verify the effectiveness of our methodology, we design a high-quality dataset and benchmark multiple state-of-the-art algorithms on it. The successful workflow demonstrates that our methodology can significantly reduce the research threshold and help individuals and institutions to construct datasets in a standardized way.**

***Index Terms*—Navigation, Mapping, Standard Datasets, Intelligent Vehicle, Construction Methodology.**

Manuscript received September 26, 2023; revised 21 December, 2023; revised 11 January, 2024; accepted 24 January, 2024. This work is supported by the National Key R&D Research Program of China under grant 2018YFB1305005. *([†]Equal second-contribution: Yujia Fu; Minghui Qin; Yufeng Xu.) ([*]Corresponding author: Hui Zhao; Bart Goossens.)*

Yuanzhi Liu, Yujia Fu, Yufeng Xu, Bin Cui, Wei Tao, and Hui Zhao are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, 200240 Shanghai, China (e-mail: lyzrose@sjtu.edu.cn; yujiafu@sjtu.edu.cn; xuyufeng@sjtu.edu.cn; cuibin728@sjtu.edu.cn; taowei@sjtu.edu.cn; huizhao@sjtu.edu.cn). Yuanzhi Liu is also with the IPI-Ghent University (email: yuanzhi.liu@ugent.be).

Minghui Qin is with the Horizon Robotics, 100094 Beijing, China (e-mail: minghui01.qin@horizon.cc).

Kunhua Liu is with the School of Mechanical and Automotive Engineering, Qingdao University of Technology, 266520, Qingdao, China (e-mail: liukunhua@qut.edu.cn).

Fengdong Chen is with the School of Instrumentation, Harbin Institute of Technology, 150001 Harbin, China (e-mail: chenfd@hit.edu.cn).

Bart Goossens and Michiel Vlaminck are with the imec-IPI-Ghent University, 9000 Gent, Belgium (e-mail: bart.goossens@ugent.be; michiel.vlaminck@ugent.be).

Poly Z.H. Sun is with the School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zh.sun@sjtu.edu.cn).

I. Introduction

The development of intelligent vehicles (IVs) requires extensive standard datasets for training, benchmarking, and improvement [1], [2]. Benefitting from publicly available datasets, profound changes have taken place in IV field over the last decade, giving rise to diverse applications [2] such as autonomous driving [1], unmanned mining [2], [3], service robots [1], and intelligent logistics and transportation [2]. Autonomous Navigation and Mapping (ANM), as a pivotal technology for IVs, places exceptionally high demands on dataset construction compared to other tasks. This is significant in its requirements for comprehensive sensor calibration, precise time synchronization, and accurate generation of ground truth. Besides, the whole construction workflow also demands intricate knowledge and sophisticated practices, necessitating lengthy learning curves for researchers to attain proficiency. These challenges have led to the slow production and deficient quality of datasets, constraining the advancement of ANM.

Due to the construction threshold, existing high-quality ANM datasets were mainly proposed by established research institutions and companies. Among the most renowned are KITTI [4], TUM-RGBD [5], and EuRoC [6], which have become the indispensable references of current ANM research. Additionally, there also emerged many newer datasets in recent years, such as Oxford RobotCar [7], KAIST Urban [8], Baidu ApolloScape [9], and Waytous Automine [10], further complementing diverse real-life factors and scenario types. However, though fulfilling the fundamental testing demands, existing datasets are still far from sufficient in terms of quality and volume to trigger giant next-stage technological breakthroughs. For instance, in KITTI dataset, part of the system was synchronized by software, which introduced a sub-10ms temporal error among sensors. In KAIST Urban, due to the interference by high-rise buildings, the trajectory ground truth generated by Differential Global Navigation Satellite System (D-GNSS) can hardly achieve sub-dm accuracy. Besides, datasets focused on corner cases are still seriously lacking, thus cannot comprehensively reflect the real world. In such circumstances, the community is in urgent needs of a mature methodology to broaden the base of contributors and standardize the quality and process for dataset construction.

This paper strives to fill the gap. Specifically, based on our systematic reviews and extensive practices, for the first time, a full-stack construction methodology of ANM datasets is proposed, including modules of platform construction, sensor calibration, time synchronization, ground truth generation, synthetic data production, and benchmark criteria, with detailed techniques and methodological routes provided in each step. Several key issues are resolved within the methodology. Importantly, we introduce versatile calibration and synchronization frameworks that can achieve up to us-level and mm-level precision in complex vehicle settings. Besides, we propose a full-scenario ground truth system that can generate scene-map and trajectory both at cm-level accuracy. As a validation, we design a meticulous robot platform and build a high-quality dataset within an indoor-outdoor integrated scenario. Our assessment confirms that its key metrics have achieved leading-level of the field. We also benchmark multiple state-of-the-art (SOTA) ANM algorithms on it, proving a high versatility of the dataset. The successful workflow demonstrates that our proposed methodology can significantly reduce the research threshold and facilitate individuals and institutions to construct datasets in a standardized way. To date, the most relevant work regarding the topic of vehicle data could be [11] (Kang *et al.*) published by IEEE Transactions on Intelligent Vehicles in 2019. It comprehensively overviews the publicly available datasets for autonomous driving, mainly serving as a dictionary for dataset selection. Different from this work, our paper is focused on the construction methodology and inherent techniques of ANM datasets.

The main contributions of this paper are as follows:

- We propose a full-stack methodology for the construction of standard ANM datasets, covering modules from platform construction to benchmark criteria, with detailed techniques and methodology routes provided. This is the first paper to investigate ANM datasets at technical level.
- We tackle several long-standing challenges in the field, including the proposal of versatile calibration and synchronization frameworks and an integrated navigation and mapping ground truth system. This directly bridges the gap for high-quality and standardized dataset creation.
- We construct a multi-sensory robot platform and design a high-quality dataset in an indoor-outdoor connected scenario. The key dataset metrics are evaluated and multiple SOTA algorithms are successfully executed and benchmarked, demonstrating the effectiveness and versatility of our proposed methodology.

Website: https://github.com/robot-pesg/Standard-Data-ANM

II. Related Works

A. Autonomous Navigation and Mapping (ANM)

ANM arises in the context that intelligent vehicles demand autonomous navigation and mapping to facilitate path planning, scene recognition, autonomous operations, and other transportation tasks. The technology of ANM encompasses several key elements, including Dead Reckoning (DR) [12], Sensor Odometry (SO) [13], Simultaneous Localization and Mapping (SLAM) [14], Visual-Inertial Navigation System (VINS) [15], and Light Detection and Ranging (LiDAR) Mapping System (LMS) [16]. While their underlying principles are highly interrelated, their specific applications and emphases could differ. Among these, DR/SO/SLAM serve as foundational technologies, whereas VINS/LMS are specialized implementations tailored to particular sensor modalities and application scenarios.

DR/SO plays a fundamental role in autonomous navigation. It involves the step-by-step estimation of the vehicle's position and orientation based on previously determined motion states [12]. Traditionally, for ground vehicles, DR is typically implemented by fusing wheel odometry and inertial sensors [17]. However, such system may be impractical for non-standard locomotion platforms like aerial and aquatic vehicles. Furthermore, it is well recognized that wheel odometry can lose precision due to slippage and may perform poorly on rugged terrains [18]. In response to these challenges, perceptual sensors have gained significant research attention to achieve robuster performance, commonly referred to as Sensor Odometry (SO). SO encompasses solutions like Visual Odometry (VO) [18], Visual-Inertial Odometry (VIO) [15], LiDAR Odometry (LO) [19], and more, which are also regarded as modern DR techniques. Note that, DR is subject to cumulative errors, and as a result, it is often integrated with external positioning aids in practice (such as Global Navigation Satellite System (GNSS) [12]).

SLAM extends the capabilities of SO by not only tracking the vehicle trajectory but also constructing and maintaining an environmental map [14]. Relying on perceptual sensors capable of both pose estimation and scene reconstruction, SLAM constructs a map that enables loop closure upon revisiting known locations, eliminating cumulative drifts, and facilitating relocalization even without external positioning aids [14]. While SLAM is more resource-intensive due to its map maintenance compared to DR and SO, its real-time operation remains an advantage for navigation systems, providing heightened accuracy and robustness. Moreover, profit by the global consistency of the loop-closed map, dense SLAM systems can also be adapted for mapping-centric tasks [20].

VINS is an approach that fuses vision and inertial sensors to achieve lightweight and accurate navigation functionality [15]. The two modalities are designed to complement each other: inertial sensor provides high-rate ego-motion but drifts quickly, while vision sensors can provide more reliable motion estimation and correct the real-time inertial bias [21]. VINS can be implemented with both VIO [22] and Visual-Inertial SLAM (VI-SLAM) [21], and has formed a self-contained research branch in recent years [15].

LMS is a technology that utilizes LiDAR as its centric sensor to achieve high-quality 3D mapping through mobile data acquisition and registration [16]. It can be implemented through LiDAR Odometry and Mapping (LOAM) [23] and LiDAR SLAM [24]. Additionally, to achieve better precision and richer information, advanced LMSs often employ multi-sensor fusion techniques. This involves the optimization by factor graph for smoothing and mapping [24], as well as the texturization of the map by camera reprojection [20].

B. Datasets for ANM

As introduced from the above background, compared to other research areas, ANM is typically a sophisticated and intricate task. Consequently, creating the corresponding datasets

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3360273

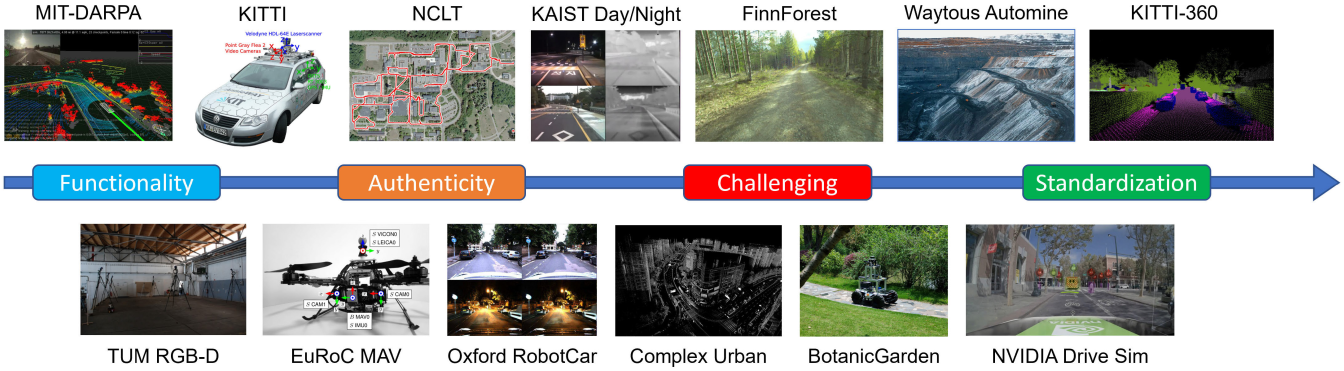


Fig. 1. The development stages and evolution trend of standard datasets in ANM field.

is quite complicated and has a high barrier to entry. Such complexity is particularly reflected in the requirement for precise and diverse sensor inputs, as well as the necessity of high-quality ground truth data for navigation and mapping. Over the past two decades, research focused on ANM datasets has been continuously expanding, emphasizing their significance as an independent and crucial research domain within the field. However, existing datasets are still insufficient in terms of scale, quality, and diversity, which limit the breakthrough of ANM and delay its transitioning to mature productivity.

We roughly categorize the development of standard datasets into four stages to review the evolution trend of this field. These stages are referred to as the Functionality Stage, Authenticity Stage, Challenging Stage, and Standardization Stage, as illustrated in Fig. 1.

Functionality Stage: During the Functionality Stage, standard datasets primarily fulfilled basic testing and evaluation purposes. The proposed datasets primarily focused on sensor availability and ground truth reliability, while the collection were mainly conducted in controlled or ideal environments. Representative datasets include MIT-DARPA [25], TUM-RGBD [5], and KITTI [4], which were extensively used for nascent ANM research. This stage gave rise to many VO and Visual-SLAM (V-SLAM) systems represented by ORB-SLAM [26] and LSD-SLAM [27], as well as LiDAR Mapping algorithms represented by LOAM [23].

Authenticity Stage: During the Authenticity Stage, standard datasets strived to address the needs of ANM algorithms as they transitioned from controlled and ideal environments into daily real-world scenarios. Representative works include NCLT [28], Oxford RobotCar [7], and KAIST Day/Night [29]. They investigated the expansion of spatial-temporal scales, such as capturing larger environments, recording sequences of longer durations, and incorporating data of day/night timeslots. Furthermore, researchers began to explore updated sensor combinations suitable for real-life practice, with a particular focus on datasets tailored for VINS, represented by EuRoC [6] and TUM-VI [30]. Benefiting from these efforts, ANM methods have gained significant enhancements and iterations (*e.g.*, ORB-SLAM2 [31] and SVO2 [32]), and many novel state-of-the-art algorithms came to the fore (*e.g.*, DSO [33] and VINS-Mono [21]). However, meanwhile, as datasets are getting out of controlled and ideal environments, there is a certain reduction in data quality. For example, TUM-MonoVO [34] used LSD-SLAM [27] to generate ground truth, yet failed to achieve a qualified level of accuracy; TUM-VI [30] only provided trajectories at the start and end segments for long sequences, which were incomplete for algorithm assessment.

Challenging Stage: During the Challenging Stage, standard datasets aimed to address robustness issues, pushing algorithms to their limits by seeking out corner cases [35], [36]. Representative datasets include Complex Urban [8], FinnForest [37], TartanAir [38], ParallelEye-CS [39], [40], BotanicGarden [41], *etc*. They were typically collected under challenging scenarios, including urban canyons, congested traffics, extreme weather and illuminations, complex motion patterns, unstructured environments, and repetitive and monotonous textures. Simultaneously, this stage also gave rise to ANM systems based on novel sensor modalities, such as event vision, thermal camera, and Radar, exemplified by datasets such as Vector [42], M2DGR [43], and RADIATE [44]. In the Challenging Stage, ANM technologies were in full bloom, with traditional methods steadily improving (*e.g.*, ORB-SLAM3 [45] and LeGO-LOAM [46]), deep learning approaches coming to the fore (*e.g.*, CNN-SLAM [47] and SuMa++ [48]), novel sensor modalities and frameworks showing promise (*e.g.*, Ultimate-SLAM [49] and Radar-SLAM [50]), bringing ANM ever closer to a thorough breakthrough. However, under such circumstance, the construction of standard datasets also met great challenges. For instance, the calibration and synchronization quality were not paid full attention [43], [51], the realism of synthetic datasets was still insufficient [38], and importantly, to obtain accurate ground truth in large-scale complex scenarios [8] became fairly difficult.

Standardization Stage: Standardization Stage corresponds to the current phase, which is the Large-Scale Standardization Stage. In this phase, ANM aims to achieve key technological breakthroughs and advancements through massive amounts of standard datasets [2], [52]. The community is experiencing thorough refinement and modularization of traditional methods, maturation of deep learning approaches [2], [52], and breakthrough of novel frameworks (*e.g.*, NeRF-based localization [53] and mapping [54]). During this stage, datasets must be high-quality, all-rounded, and possess significant scale for

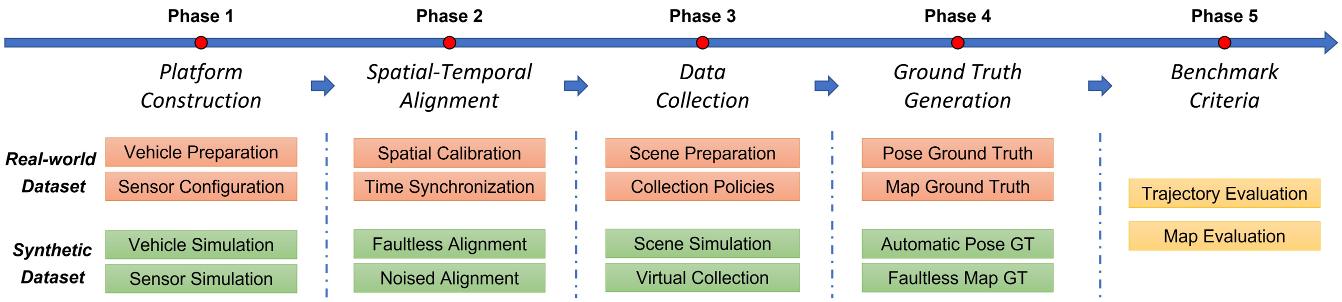


Fig. 2. Schematic graph of the construction process regarding both real-world and synthetic datasets.

training, testing, and improvement [52], [55]. Representative works include Waytous AutoMine [10], Boreas [56], KITTI-360 [57], and Scenarios Engineering (SE) framework and datasets [52], [58], [55], [59]. Additionally, synthetic datasets and simulators are gaining significant attention, with notable examples including SHIFT [60], UniSim [61], and NVIDIA Drive Sim [62]. However, due to the construction complexity and technical challenges, large-scale and high-quality datasets are still notably lacking at the present time. The community is in urgent demands of a mature and full-stack methodology to stimulate standardized construction, crowdsourcing, and realistic virtual data synthesis. This is exactly the motivation of our paper.

III. General Construction Methodology

A. Dataset Structure

1) Data Sequence: Based on the workflow that ANM estimates movements and/or builds scene maps through mobile sensing, the essential part of ANM datasets is, therefore, the sequential input data consisting of navigation series (Inertial Measurement Unit (IMU), GNSS, wheel encoder, *etc.*), vision streams (grayscale, RGB, thermal, *etc.*), and LiDAR scans. For modern ANM systems, it is imperative to have at least a sequence of vision or LiDAR scanning [4] data available. Moreover, the inclusion of multi-sensor data is highly recommended, as it broadens the applicable scopes of datasets and facilitates diverse sensor fusion research.

2) Calibration Parameters: Due to the combined usage of homogeneous and heterogeneous sensors, their spatial coordinates need to be aligned into an identical frame, which is so-called spatial extrinsic calibration. The extrinsics consist of pairs of rotation matrices and translation vectors and are usually set with respect to the center of a vehicle. Besides, the internal properties of sensors should also be determined, such as camera intrinsics (world-to-pixel projection matrix and lens distortion), IMU white noise and bias, and so on.

3) Synchronized Timestamps: To ensure the reliability of multi-sensor fusion, data from different sources should be precisely aligned to an identical timeline, which is so-called time synchronization. Importantly, the timestamp should be set to the exact data sampling time, rather than the trigger or data receiving time.

4) Ground Truth: For navigation and mapping tasks, Ground Truth (GT) mainly refers to the precise ego-motions (position and/or orientation) and scene maps that are qualified for the assessment of algorithms. Importantly, the navigation GT should also be synchronized with the sequential data to avoid biased evaluation. Moreover, the extrinsics of GT with respect to the source sensors should also be applied.

5) Semantic Annotations: Semantic information is usually annotated on 2D images and 3D point clouds. For 2D images, annotations are pixel-wise [41] and are typically represented by bounding contours accompanied by semantic labels. For 3D point clouds, the annotations are point-wise [57], labeled with specific object categories. Usually, semantic annotations are not a necessary part for ANM datasets, but they are highly encouraged to facilitate semantic-related ANM research.

B. Construction Process for Standard Datesets

ANM datasets can be classified into two categories: real-world and synthetic ones. Both of them involve five main construction steps: platform construction, spatial-temporal alignment, data collection, ground truth generation, and benchmark criteria, as illustrated in Fig. 2. The first step is platform construction. In the case of real-world datasets, this entails preparing a vehicle tailored to specific working environments and equipping it with desired sensors. For synthetic datasets, a virtual platform is integrated within the software engine, involving the simulation of motion patterns and sensor models. The second step is spatial-temporal alignment. In the case of real-world datasets, this step includes two key aspects: the geometrical calibration of sensors, both intrinsic and extrinsic, and the time synchronization among sensors and host computers. For synthetic datasets, faultless alignment can be achieved by computer design. Nevertheless, given the practical challenges of achieving ideal alignment in real-world operations, it is recommended to consider both the inclusion or exclusion of alignment errors in the simulation process. The third step is data collection, which involves the preparation of testing scenarios and the definition of collection policies. It is recommended to prepare or simulate various conditions of the target scenes and collect the environment as completely as possible. The fourth step is ground truth generation. It refers to the accurate measurement of vehicle trajectory and scene map. For real-world datasets, this process is typically achieved by leveraging external high-end instruments, which hold a much higher accuracy than the tested algorithms. For synthetic datasets, ground truth can be acquired automatically from the simulator, and the accuracy is faultless. Besides, ground truth may also involve the annotation of semantics. The last step is the definition of benchmark criteria. It includes the metrics and principles for algorithm assessments [5]. At present, the benchmark criteria are relatively mature and generic, but they can also be customized depending on different evaluating dimensions and difficulty levels. In the following sections, we will begin by investigating the full-stack construction pipeline for real-world datasets. The techniques for producing synthetic data will be carried out in an individual dedicated section.

IV. Platform Construction

A. Vehicle Preparation

Preparing the vehicle involves primarily considering the testing scenarios of the dataset, as well as the quantity and size of the hardware facilities intended for deployment. Typically, the vehicles can be categorized into four main classes: ground vehicle, aerial vehicle, surface and underwater vehicle, and human-carried equipment.

1) Ground Vehicle: Ground vehicles encompass a variety of platforms such as wheeled robots [41], full-sized cars [4], motorcycles [63], trucks [10], and more. These platforms offer ample space to accommodate diverse sensor types, making them vital for research of multi-sensor fusion. Additionally, ground vehicles can typically provide wheel odometry data, further enhancing their suitability for such studies.

2) Aerial Vehicle: Aerial vehicles primarily refer to unmanned aerial vehicles (UAVs) [6]. In contrast to ground vehicles, UAVs offer increased flexibility and exhibit a 6-degree-of-freedom (6-DoF) motion pattern, posing greater challenges [64] for navigation and mapping algorithms. However, due to their compact and lightweight nature, UAVs face limitations in carrying a wide array of sensor types.

3) Surface and Underwater Vehicle: Surface and underwater vehicles mainly include surface vessels, unmanned surface vehicles (USVs) [51], and autonomous underwater vehicles (AUVs) [65], [66]. Similar to ground vehicles, such platforms can also carry a wide range of sensors, making them suitable for sensor-fusion studies [51]. However, their motion patterns tend to be very smooth, which can result in datasets that are less challenging to push the limits of algorithms.

4) Human-carried Equipment: Human-carried equipment mainly includes handheld rigs [30], backpacks [16], and helmets [42]. These devices are not actual vehicles but provide challenging 6-DoF motions, including sharp turns and frequent shakes [30], which can increase the comprehensiveness for algorithm testing. Such platforms are typically compact and have limited capacity, thus are not suitable for integration of a large number of sensors.

The characteristics of some commonly used platforms are listed in Table I, explaining their Degree-of-Freedom (DoF), speed, and motion properties. In practice, it is recommended to diversify the data collection platforms to create datasets with varying difficulty levels.

B. Sensor Configuration

The sensor configuration of datasets is determined by the based sensor modalities of the tested algorithms. Typically, there are six types of sensors involved for ANM applications: GNSS, IMU, Distance Measurement Instrument (DMI), Vision sensors, LiDAR, and mmWave radar, as described below.

TABLE I
Characteristics of Different Mobile Platforms

Platform	DOF	Speed	Motion Properties
Car	3	10-25m/s	Smooth forward/turn/shifting, Fast motion
Wheeled robot	3	1-5m/s	Smooth forward/turn/shifting, Slow motion
USV	3	1-5m/s	Smooth forward/turn/shifting, Slow motion
Drone	6	1-15m/s	Moderate turn/rotation, Agile motion
Handheld	6	≈1m/s	Frequent shake, Sharp turn/rotation/shifting

1) GNSS: GNSS is a system that uses satellites to determine the absolute geographic positions of GNSS devices. With a minimum requirement of 4 independent navigational satellites, by performing trilateration, the system can measure locations at meters level accuracy in open areas. In addition, by D-GNSS and Real-Time Kinematic GNSS (RTK-GNSS) technologies, the accuracy can be further improved to cm-level [4]. In practice, due to the financial considerations, moderate GNSSs are typically more suitable for algorithm development, while D-GNSSs are mainly used for generating trajectory ground truth [4].

2) IMU: IMU is an electronic device primarily utilizing accelerometers and gyroscopes to measure the acceleration, angular rate, and optionally the orientation of the host body. Depending on the precision, there are consumer-grade and high-standard IMUs. Consumer IMU is typically achieved by micro-electromechanical systems (MEMS) which is lightweight and low-cost but with poor precision and drifts quickly [21]. High-standard IMUs typically incorporate larger yet more precise gyroscopes, such as fiber optic gyroscopes (FOG), ensuring sustained precision over extended periods [8]. In practice, MEMS IMUs, known for their affordability and sufficient properties for sensor fusion [21], are the prevalent choice in most ANM applications. In contrast, high-standard IMUs are primarily reserved for tactical applications or employed as ground truth devices [8].

3) DMI: DMI is a system that uses motion sensors to estimate positional change over time. In most cases, it refers to the wheel odometry of ground vehicles that measures displacements based on rotational encoders. DMI is not a precise system and is sensitive to cumulative errors.

4) Vision Sensors: Vision sensors refer to different types of cameras that mainly capture 2D optical imaging information. There are many modalities and configurations of cameras used for ANM, including monocular, binocular, multi-camera system, fisheye, omnidirectional, thermal, event, and RGB-Depth (RGB-D) cameras, suitable for different tasks. A monocular camera is the minimum visual configuration for ANM [26], which has been widely researched due to its low-cost, compact, and easy-to-integrate features [33], [27]. However, mono-camera alone cannot recover scene scale, and may suffer from pure rotation problem [32]. Binocular camera captures synchronized image-pair with stereo disparity, enabling it to inherently recover scene depth and scale even without movement [31]. In practice, binocular-based SO/SLAM systems can already provide very good navigation performance, and are also capable of achieving dense mapping of the scene

[67]. Multi-camera, fisheye, and omnidirectional cameras are intended to increase the perception field-of-view (FoV) [68] of the system, which can strengthen not only the navigation module but also map completeness. Thermal, event, and RGB-D cameras are three novel modalities applied in this field. Thermal camera has a longer sensing spectrum than visible light and can work consistently in dark night, thick fog, and other challenging conditions [69]. Event camera is designed to capture intensity changes of each image pixel at very high rates [70]. It is especially suitable for agile and fast applications such as high-speed vehicles and drone racing [71], while it is not suitable for mapping-centric applications. RGB-D is a novel modality that combines imaging and ranging sensors [72]. It collects synchronized color and depth image-pairs and typically works within 10 meters. Furthermore, due to its sensitivity to infrared light, RGB-D sensors are primarily used for indoor applications.

5) LiDAR: LiDAR is a type of optical sensor that measures the distance of an object or a surface by emitting a laser beam and counting its time of return (*i.e.*, ToF, time-of-flight) [73]. Benefiting from the active ranging mechanism, LiDAR can output very accurate and dense 3D scanning of the environment in a straightforward way. Besides, LiDAR is robust to challenging illuminations and can work in very remote distances (hundreds of meters typically), which makes it extensively used for autonomous driving [74] and mobile mapping system (MMS) [75]. Nevertheless, on the other hand, LiDAR is also an expensive sensor, which limits its application in cost-intensive scenarios. Note that, despite its exceptional ranging accuracy, LiDAR can be compromised under adverse weathers, such as dense fog, heavy rain, snowfall, and more [76].

6) mmWave Radar: mmWave radar is a special class of radar that uses millimeter-range electromagnetic waves as medium to detect the position and direction of objects. Through the Doppler effect, mmWave radar can also measure the speed of targets, which makes it preferable in object detection and tracking applications. The greatest strength of mmWave radar is that, due to the sensing spectrum, it can work in all weather conditions without losing too much ranging accuracy [76], [77]. Among different technologies, Frequency-Modulated Continuous-Wave (FMCW) radar systems are more appealing as they can provide relatively dense representations of the environment, which is necessary and proved to be effective in navigation and mapping fields [50]. This paper also chooses FMCW Radar as the focus.

In platform construction, it is encouraged to make full use of the vehicle space and equip it with as many sensor types as possible, so as to facilitate research of various ANM tasks.

C. Platform Demonstration

To provide an application demonstration for platform construction, we have designed and integrated a mobile robot system equipped with a comprehensive set of sensors, as illustrated in Fig. 3. The platform features a four-wheel differential drive wheeled robot chassis, enabling powerful and reliable operation in various terrains. Above the chassis, we design a complete set of standard aluminum profiles to mount the host computers, control modules, sensor suites, and more. For the sensor system, we have developed independent brackets using 3D printing and standard profiles, ensuring their flexibility for easy disassembly to meet the calibration and maintenance requirements. To ensure a high versatility of this platform, full-modal sensors are meticulous equipped, including stereo RGB and Grayscale cameras, spinning LiDAR, MEMS LiDAR, GNSS, high-precision IMU, consumer-grade IMU, and wheel encoders (refer to Table II for their specifications). To benefit the community, we create a detailed hardware selection and integration guideline and open-source it on our website. This eliminates the time-consuming trial and error, enabling researchers to rapidly complete the platform construction process.

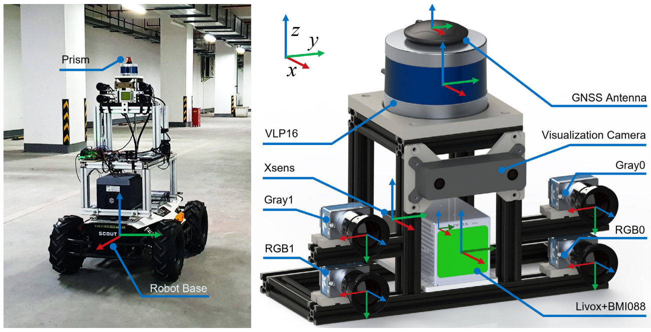


Fig. 3. The platform design and sensors coordinates of our robot system.

TABLE II
SENSORS SPECIFICATIONS OF OUR ROBOT PLATFORM

Sensor	Model	Specification
Gray Stereo	Dalsa M1930	1920x1200, 71°x56°FoV, 40Hz
RGB Stereo	DALSA C1930	1920x1200, 71°x56°FoV, 40Hz
Spinning LiDAR	Velodyne VLP16	±3cm@100m, 360°x30°FoV, 10Hz
MEMS LiDAR	Livox AVIA	±2cm@200m, 70°x77°FoV, 10Hz
Industrial IMU	Xsens Mti-680G	9-axis, sub-deg gyro, 400Hz
Consumer IMU	BMI-088	6-axis, Livox built-in, 200Hz
GNSS	Ublox ZED-F9P-01B	1.5m accuracy, 10Hz
Wheel Encoder	Scout V1.0	4WD, 3-axis, 200Hz

V. SPATIAL CALIBRATION

Complex vehicle systems typically include various types of sensors. In the whole calibration process, cameras, benefiting from their good precision and versatility, are better positioned to serve as the central nodes. This section aims to introduce a versatile calibration framework for comprehensive sensors. Three general calibration methodologies are first introduced, followed by thorough reviews and detailed explanations of camera-to-sensor calibration techniques. Finally, the overall calibration framework is carried out, which can support both calibrations from scratch and online extrinsics adjustment.

A. General Methodology

1) Matching-based Methods: Matching-based calibration consists of target-based and targetless approaches. The underlying principle is to identify and establish correspondences between different sensor observations [78], leveraging the mutual information to achieve alignment. Target-based methods entail the use of specific calibration targets or patterns observed by multiple sensors simultaneously. By having precise knowledge of the reference points or features, the correspondences can be established with high confidence and accuracy [79], thereby resulting in more robust and reliable calibration results. Targetless methods, instead, aim to find correspondences between sensor observations in a scene without prior knowledge [80]. They typically utilize features or key points extracted from raw sensor data, adopting techniques such as feature matching or point cloud registration to solve for the extrinsics. Even though targetless methods offer more flexibility and convenience, their accuracy may decline in less-than-ideal environments [80], [81]. Therefore, such methods are only recommended when target-based methods are not practicable [82].

2) Motion-based methods: Motion-based calibration methodologies generally consist of hand-eye calibration and sensor fusion-based calibration. Hand-eye calibration treats the motions of different rigid-connected sensors as independent processes and solves for the extrinsics by adhering to the equation AX=XB [83]. Here, A and B represent the respective sensor motions, and X denotes the extrinsics between them. While this methodology is explicit, it requires precise motion measurement or estimation for both sensors [82], which are not applicable in most cases. In contrast, sensor fusion-based calibration represents an advanced approach to the hand-eye mechanism. It treats sensor calibration as an optimization problem, where the extrinsic parameters are assigned as variables to be optimized [84]. By considering motion constraints, extrinsics, and other cost functions, the calibration can be effectively solved by minimizing the overall objective error [21].

3) Learning-based methods: Learning-based calibration includes end-to-end calibration and learning-aided calibration. In end-to-end calibration, a complete process is formulated as a single learning task, where the calibration parameters are directly predicted from the input sensor data. This approach leverages the power of deep learning models to learn the intricate relationships between sensor inputs and calibration parameters in an automatic and data-driven manner [85], [86]. By training the model on large-scale datasets that encompass diverse sensor configurations and environmental conditions, end-to-end calibration can achieve moderate calibration results [87]. On the other hand, learning-aided calibration takes a more traditional calibration pipeline and incorporates learning techniques to enhance the calibration rather than output the results in a one-stop shop [88]. For example, learning algorithms can be used to extract mutual features [89], handle challenging scenarios [90], or to correct miscalibrations [91]. Nevertheless, though bringing in great simplicity and efficiency, learning-based calibration could be less reliable, while spatial precision is crucial for standard datasets [82]. Therefore, it is typically not suitable for principled calibration from scratch.

B. Camera Calibration

Camera calibration plays a crucial role in 3D vision and is the most critical link of the calibration chain for a vehicle. It involves estimating the camera intrinsics, which describe the mapping between the 3D world and the 2D image plane, as well as the extrinsics that define the relationship between multiple cameras in a system [92]. While various camera models exist, this section focuses on the pinhole camera model due to its versatility, with the consideration that the underlying principles can apply to other camera models as well.

Camera calibration encompasses several methods, including target-based calibration using 3D [93] or planar targets [79], active motion calibration using precise external motion control [94], and self-calibration based on theories such as Kruppa equations [95] and absolute dual quadric [96]. Given the critical importance of precision and the need for a flexible calibration process, target-based methods are often considered the optimal solution. Tsai's method [93] and Zhang's method [79] are the two most widely used target-based techniques in practice. Tsai's method relies on a known 3D object with precise geometry as the target. It establishes correspondences between image pixels and 3D positions, allowing for the determination of the best projection matrix through linear least squares. The intrinsics and extrinsics are subsequently obtained through decomposition. Nonlinear least squares is then employed for refinements and to estimate the distortion parameters [93]. In contrast, Zhang's method only requires a planar target and a few observations of different orientations. It offers a closed-form solution in a more concise setup [79].

The precision of camera calibration can be evaluated by reprojection errors of feature points among the input images. An ideal calibration process typically exhibits a precision of sub-0.1 pixels [41]. Comparative studies have shown that Zhang's method exhibits greater accuracy and robustness and is more flexible in terms of target manufacturing [97]. Consequently, it is more widely utilized in practice and serves as the foundation for calibration tools in OpenCV and Matlab. At present, with the advancement and complementation of different lens models [98], camera calibration can be considered as a mature technique.

C. Camera-IMU Calibration

Inertial sensors, unlike cameras, lack environmental observations, which makes matching-based methods infeasible for extrinsics calibration. Additionally, IMU sensors alone are less precise for motion tracking, making them unsuitable for the hand-eye mechanism. As a result, camera-IMU calibration often relies on visual-inertial fusion techniques, where the extrinsics are parameterized as variables to be optimized [99]. In other words, the final extrinsics are the ones that make the sensor-fusion system achieve the best possible motion estimations.

Camera-IMU calibration consists of both offline [100] and online [21] approaches. Offline approaches involve designing and executing carefully planned motion sequences to obtain sufficient motion excitations and diverse trajectory patterns, so as to construct a well-constrained calibration problem [101]. Many open toolboxes exist, and their methodologies are common. Here we choose the most influential and community-recognized toolbox Kalibr [100] for a description. During the

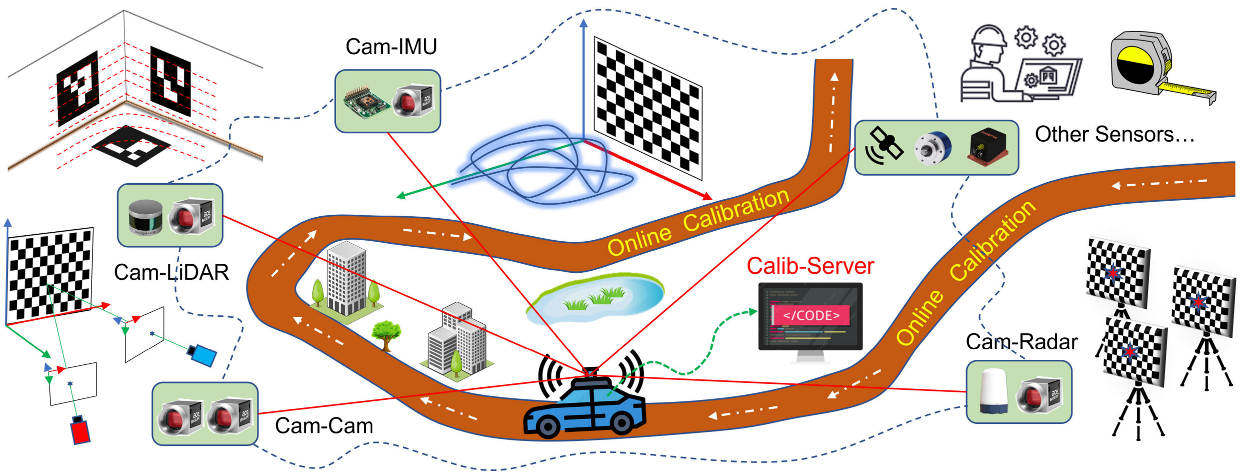


Fig. 4. A versatile calibration framework suitable for full-source sensor systems.

calibration process, the visual-inertial sensor suite is handheld to wave in front of a visual pattern, as illustrated in the upper middle of Fig. 4. The captured data from the sensor suite, combined with the constraints provided by the calibration pattern, serve as inputs for motion estimation. Kalibr utilizes a unified principled maximum-likelihood estimator that parameterizes both the transformation matrix and time-offset [101], [102]. By minimizing the overall objective errors, the Levenberg-Marquardt (LM) algorithm can estimate all the unknown parameters simultaneously. The precision can also be evaluated by the reprojection errors of visual patterns.

Online approaches optimize an initial coarse extrinsic or estimate extrinsics from scratch within real-time operations. Such methods typically rely on real-time sensor fusion algorithms and optimization techniques, widely used for miscalibrations or adapting the calibration parameters on the fly to compensate for dynamic changes and environmental variations [103]. Many systems have implemented this methodology, including VINS-Mono [21], OpenVINS [104], *etc.* From the perspective of calibration precision, offline approaches make sure to build a well-constrained optimization problem, leading to optimal and reliable outcomes, which are necessary for creating standard datasets. On the other hand, in practice, online approaches also hold significant importance in scenarios where geometric relations may experience subtle shifts over extended periods and are in need of timely adjustments.

D. Camera-LiDAR Calibration

Camera and LiDAR are two crucial modalities in ANM systems, and their fusion is important for accurate mapping [105] and semantic perceptions. As both of them have explicit observations, the no-doubt choice is matching-based calibration. Target-based matching and targetless matching are both possible [82]. Target-based matching requires calibration patterns to be both visually and structurally recognized. Geiger *et al.* [106] used a combination of 2D chessboards at different orientations and positions. With the 3D recovery of visual corner points and the target extraction from LiDAR scans, the two point-sets can be accurately registered, and the extrinsics are thus solved. Autoware [107] presents an interaction calibration toolbox, which enables users to manually align two sensors' data based on board edges. Targetless matching methods extract mutual environmental features in camera and LiDAR frames. Yuan *et al.* [80] developed a toolbox that uses image contours and LiDAR edges for alignment. For disambiguation, they choose spatially discontinuous point cloud edges for alignment and have achieved comparative accuracy against target-based methods. However, this method only performs well in structured scenarios that have rich edge features. For sparse LiDARs, a pre-mapping process that accumulates the point clouds is necessary, which could bring in additional errors. In assessing the calibration precision, methods based on 3D reconstruction are typically evaluated using the registration error of corresponding 3D points [41], and methods based on edge extraction are typically evaluated based on the alignment error after projecting LiDAR points onto the pixel plane [80]. Under proper settings, they are supposed to obtain sub-cm and sub-pixel precision respectively.

Many online and learning-based calibration methods also exist, such as RegNet [108], CalibNet [86], RGGNet [109], *etc.* Compared with visual-inertial calibration, camera-to-LiDAR calibration is exempted from motion degeneracy, thus online approaches are also possible to achieve qualified results. Whereas, to ensure the authenticity of datasets, offline and target-based methods are still the optimal choice. On the other hand, online calibration can serve as a complement for adapting to the geometrical shifts over time.

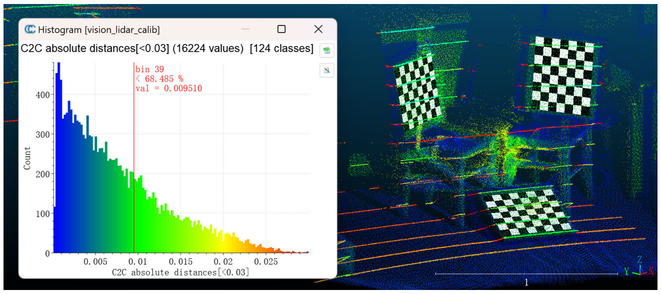
E. Camera-Radar Calibration

Radar, compared to LiDAR, has significantly lower scanning density and point resolution, making it challenging to perform feature recognition when calibrating with cameras. To address this issue, a well-recognized approach is to use composite targets, precisely combining visual markers with radar reflectors to facilitate simultaneous recognition. Since the relative positions of both markers are precisely known, extrinsic calibration can be easily achieved by projecting Radar points onto the image plane. The calibration accuracy can be assessed through reprojection errors, which are typically within a few pixels for well-aligned setups. Traditionally, a radar reflector is built from three orthogonal metal plates [110], which are designed to efficiently retroreflect the radar emissions, producing highly distinct points on radar imagery. Recent advancements have seen the emergence of compact, highly precise active radar reflectors that demonstrate improved performance, and have become the standard calibration setup of many vehicle companies [110], [111]. We have given an illustration of such a target in the bottom right of Fig. 4, where the red star denotes the active radar reflector.

While there are sensor-fusion-based methods capable of estimating the extrinsic parameters online [112], it is firmly believed that target-based calibration is indispensable. We maintain the viewpoint that fusion-based methods should ideally serve as refinement tools, supplementing the calibration process, rather than being utilized for calibration from scratch.

F. Versatile Calibration Framework

Building upon the techniques delineated above, we introduce a versatile calibration framework that is suitable for full-source sensor systems, as illustrated in Fig. 4. This framework is proficient either in conducting initial calibrations or adapting to long-term geometrical drifts. Considering the ubiquity and good precision of visual sensors in intelligent vehicles, we have designated the camera as the center of our entire calibration framework, allowing other sensors to be interconnected via the camera node. For Camera-Camera, Camera-LiDAR, Camera-IMU, and Camera-Radar calibrations, we have provided detailed explanations of their respective techniques in the preceding sub-sections. Importantly, to enhance the flexibility and precision for Camera-LiDAR calibration, we also contribute a novel toolbox that requires only three quasi-orthogonally placed visual plates [41], as illustrated in Fig. 5. Different from Geiger's method [106], we reconstruct a dense and noiseless vision 3D model, and the LiDAR point clouds are reversely registered to vision models by point-to-plane ICP. Since the vision model is highly-precise, our method is more robust to the LiDAR noise. This tool has been open-sourced to the community on our website. For all the calibration groups, we recommend using target-based methods to attain high reliability and precision. If circumstances impose limitations, then alternative targetless methods can be considered. For other navigation sensors such as GNSS, Inertial Navigatioin System (INS), and DMI, the manufacturers typically have provided precise measurement origins for them. As a result, one can directly obtain their 3D coordinates with respect to the vehicle using Computer-Aided Design (CAD) softwares or determine their mounting positions through manual external measurements. This method often yields extrinsics precision at sub-cm level, while their angular parameters can further be figured out through algorithmic adjustments. So far, the vehicle sensor system has been fully calibrated, and subsequently, a motion-based online calibration process can be performed, in order to optimize the calibration-chain to a highly consistent one. Through extensive data tuning, all the extrinsics will be refined and adjusted to their optimal values.


Histogram [vision_lidar_calib]

C2C absolute distances[<0.03] (16224 values) [124 classes]

bin 39
< 68.485 %
val = 0.009510

Fig. 5. Calibration demonstration between Camera and LiDARs (Spinning & MEMS) using the proposed calibration framework, resulting in a precision of ~9.5mm std. (evaluated by CloudCompare [114] software).

Sometimes, in case cameras are not available, it is also feasible to calibrate the sensor pairs directly. For example, there already exist mature open tools (*e.g.*, LI-Calib [81] and OpenCalib [113]) which leverage motion-based optimization to estimate the extrinsics between LiDAR-IMU pairs. Besides, for multi-LiDAR systems, since the point clouds are already precise, it is an explicit way to calibrate them by scan-to-scan or map-to-map registration. These calibration pairs make sense in specific sensor systems, while simultaneously, they can also be merged into our main chain as additional constraints.

To demonstrate the effectiveness of our framework, a complete calibration process has been conducted on our constructed robot platform. For multi-camera calibration, we employ Zhang's method [79], achieving better than 0.1 pixels reprojection error. For camera-IMU calibration, we utilize the Kalibr toolbox [101], achieving a reprojection error of sub-pixel. Importantly, for camera-LiDAR calibration, we use our proposed toolbox and verify its registration precision of ~9.5mm, as illustrated in Fig. 5. For the other sensors, their calibration parameters are accurately measured in the CAD model and post-refined by algorithm optimization. To benefit the community, we create a detailed calibration guideline and open-source it on our website.

VI. Time Synchronization

Time synchronization can be categorized into software-based and hardware-based techniques. Generally, soft-sync is ubiquitous and hardware-independent, while hard-sync can achieve higher precision and stability but necessitates hardware support. This section explores the key techniques of both categories (their properties are subsequently compared in Table III), and finally, a versatile synchronization framework suitable for all-level sensor systems is introduced.

A. Software Synchronization

1) Network Time Protocol (NTP): NTP [115] is a basic and common network protocol for time synchronization among computers and sensors. It works at the software level and does not require specific hardware except for a network interface [115]. The basic principle of NTP synchronization is achieved through a client-server model. In this model, the NTP server is configured as the time source, while sensors or terminal computers act as NTP clients, synchronizing with the server by exchanging time packets and determining the time offsets [116]. NTP can work both in wired and wireless settings, and can achieve up to sub-ms precision in local Ethernets [117].

2) Real-Time Kernel (RT-Kernel): A moderate soft-sync sensor system is subject to transfer delay, clock jitter, and data buffer time, which make the received timestamps of data unstably later than its real sampling time [118]. To bridge this gap, a real-time kernel makes sure to guarantee a response within specified time constraints, which avoids large time deviations and can lower the data latency to milliseconds and sometimes microseconds level [119]. The principle of real-time kernel mainly lies in preemption, priority-based scheduling, and high-precision timer [120]. These mechanisms organize the process's execution based on their priority levels and allow high-priority tasks to interrupt those of lower priority, ensuring time-sensitive events to be completed at first and on time. State-of-the-art real-time kernel implementations include PREEMPT_RT, RTLinux, Xenomai, RTAI, [120] *etc.* Thereinto, PREEMPT_RT is realized as a patch of Linux kernel [121], offering a simple way to achieve quasi-real-time performance with excellent compatibility with standard Linux systems. In contrast, RTLinux, Xenomai, RTAI, *etc.*, offer hardware-level real-time performance but might require deep systemic modifications and much more complex development and maintenance works.

3) Temporal Calibration: For most low-cost and self-assembled sensors, synchronization interfaces could be unavailable. Though RT-Kernel can avoid messy timestamps, minor transmission delays (several milliseconds and more) will still exist [122]. Under such a situation, it is necessary to calibrate the time offset and compensate it on the fly. Temporal calibration can be achieved with both offline and online approaches. The methodology is similar to motion-based extrinsics calibration – parameterizing the time offset and constructing a well-constrained motion optimization problem, the best suitable delay can be determined. The calibration is usually carried out against IMU, for example, Kalibr [100] and LI-Calib [81] respectively provide offline time-offset calibration between camera-IMU and LiDAR-IMU, while VINS-Mono [21] and Fast-LIO2 [123] respectively provide online camera-IMU and LiDAR-IMU temporal calibration. Besides, there is also a simplified approach that leverages the steep points to align the motion processes of different sensors. For example, [124] uses gyro rotational series to align different IMUs, and Rawseeds [125] uses sensor odometry results to align different sensor motions. These approaches are likely to obtain sub-ms level calibration accuracy.

Note that, temporal calibration assumes that the time offset is constant among a time segment or the whole process, thus it cannot deal with random timestamp jitters. Besides, there is a high risk that ANM algorithms might break down under subpar software-only synchronization systems.

B. Hardware Synchronization

1) GNSS Timing: Except for positioning service, GNSS systems are also capable of providing highly accurate time information anywhere on Earth [126]. Crucially, each satellite contains multiple atomic clocks that contribute very precise time data to the GNSS signals, offering stability and accuracy to the level of billionths of a second. GNSS receivers decode these signals, enabling them to determine the actual time without owning and operating atomic clocks. Typically, GNSS receivers can output one-pulse-per-second (1PPS) signals to trigger the sensors or devices to reset their internal counters, and the subsequently arrived National Marine Electronics Association (NMEA) sentences will transfer the pulse-per-second (PPS) corresponding times to force the clocks up to date [41]. GNSS timing is widely recognized as the most ubiquitous, precise, and mature synchronization technique. However, its functionality is limited in GNSS-denied areas, and its precision might experience drift after prolonged signal losses [127]. In such scenarios, the concept of mimicking a GNSS-clock can offer a viable alternative [41], [127].

2) Precision Time Protocol (PTP): PTP, also known as IEEE 1588, is a protocol to achieve high-standard synchronization on specific hardware through local networks [128]. Similar to NTP, PTP uses a master-slave model, and by exchanging time packets, the clock offset can be determined and the slave clocks can thus be adjusted and synchronized. Crucially, the operations are all progressing at the hardware level [41], which can achieve sub-us and even nanoseconds precision. Many up-to-date advanced sensors have provided PTP capability, such as Livox [129] and Ouster [130] LiDARs, Dalsa [131] and Basler [132] cameras, *etc.* Especially, PTP is very good at inter-host synchronization, enabling users to distribute massive sensors in different host computers.

3) Hardware Trigger: Hardware trigger is a technique to synchronize sensors by external interruptions [41]. It also requires the hardware capabilities of target sensors [133]. Typically, the trigger signal is a set of electrical pulses, when the rising or falling edge arrives, the sensors can be controlled to start or stop sampling. For some sensors, the trigger signal does not directly control the measurement process, instead, it calls for feedback of the actual time from sensors and bridges a relationship between the controller and sensor clocks [41]. Hardware triggers can typically achieve nanoseconds synchronization precision, and its robustness could be better than PTP as no data transmission process is involved.

TABLE III
COMPARISON OF TYPICAL TIME SYNCHRONIZATION TECHNIQUES

Type	Technique	Requirements	Precision
Soft-sync	RT-Kernel	Short data packet	up to sub-ms
	NTP	Network interface	sub-ms to 10ms
	Temporal calibration	Online/offline align	up to sub-ms
Hard-sync	GNSS Timing	Open-sky, PPS	10s of ns
	Mimicked GNSS	PPS interface	sub-us to 10us
	External Trigger	GPIO interface	10s of ns
	IEEE-1588 PTP	PTP hardware	10ns to sub-us

C. Versatile Synchronization Framework

To accommodate sensors of different grades and properties, we introduce a versatile synchronization framework that is suitable for all-level sensor systems, as illustrated in Fig. 6. We broadly categorize the vehicle sensors into four types based on

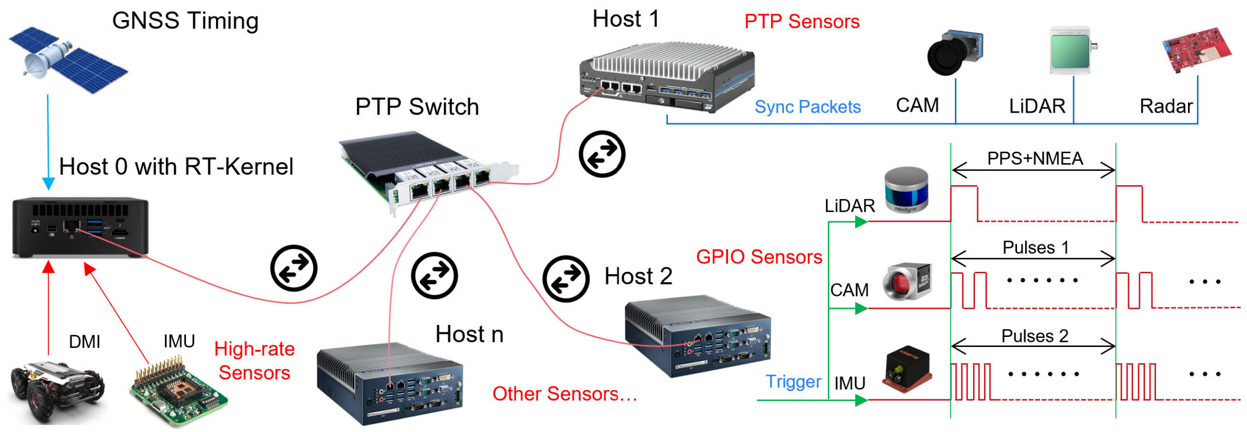


Fig. 6. A versatile synchronization framework suitable for all-level sensor systems.

their supported synchronization protocols. The first category is sensors with GPIO interfaces. The second category is sensors supporting PTP synchronization. The third category is sensors without any sync-interfaces, mainly consisting of high-rate IMUs, DMIs, and the like. The fourth category is sensors with other synchronization protocols.

Considering a system equipped with massive sensors, to prevent data loss and relax the processing load of the system, we distribute these sensors across four host machines, as illustrated in Fig. 6, labeled as Host-0, Host-1, Host-2, and Host-n. Initially, Host-0 receives the GNSS signal for satellite timing, synchronizing with UTC time at nanoseconds precision. Subsequently, Host-0 propagates its timestamps to the other host machines, namely Host-1, Host-2, and Host-n, which also support PTP, in a master-slave configuration. This ensures that all these host machines synchronize with Host-0 at sub-microsecond precision. It is worth noting that Net cards and host computers that support PTP-sync are quite common today, such as Intel i210 and i350 series. We do not recommend using NTP-only cards for development, as they would reduce the sync-precision to ms-level at most.

Next, we address the synchronization between the sensors and their respective mounting hosts. For Host-0, since the sensors themselves do not support synchronization, we use a real-time kernel to restrict data buffering and processing time. We estimate the total data delay using theoretical calculations of transmission time and algorithmic temporal calibration to compensate for it. For Host-1, since all the connected sensors support PTP, we configure Host-1 as the master clock, while the sensors are configured as slaves. It is important to note that the sync quality of PTP is also influenced by the network condition, so it is advisable to use dedicated data acquisition equipments. For Host-2, as all the attached sensors support general-purpose input/output (GPIO) interfaces, it can generate multiple pulses with exactly the same phase. Then, the sensors can achieve high-precision synchronization with the host through synchronized triggering, PPS timing, and master-slave clock alignment. Finally, Host-n will handle sensors with other sync-protocols, such as NTP and IEEE1394. So far, the entire system has been synchronized successfully, aligned with Coordinated Universal Time (UTC) time at a high precision.

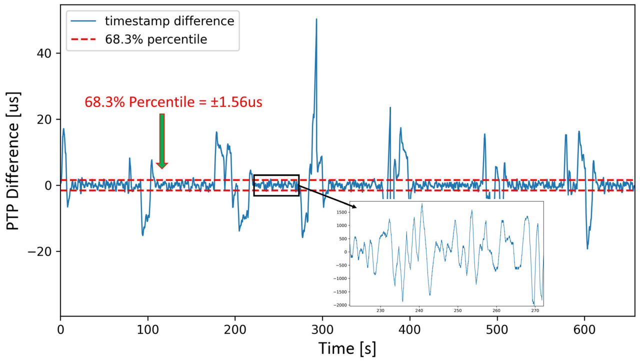


Fig. 7. Timestamp difference between two hardware-triggered cameras from two PTP-synced host computers, suggesting a precision of ~1.56us std.

If the number of sensors continues to increase, our synchronization network can also be expanded accordingly. When GNSS is unavailable, Host-0 can serve as the reference UTC clock, ensuring a high sync-precision within the local network. Note that, for non-navigation sensors that do not support any sync protocols, such as camera and LiDAR, due to their large data volume, the transmission time can be unpredictable, making it difficult to accurately recover the real data sampling time. Therefore, such sensors are never recommended for the construction of standard datasets.

To demonstrate the effectiveness of our framework, a rigorous synchronization system is implemented on our platform. We distribute all the sensors across two host computers that are synchronized through PTP. One computer is connected to the robot via a Controller Area Network (CAN) bus and runs a real-time kernel, dedicated to the acquisition of wheel encoder data. The other computer is responsible for receiving data from all the other sensors, which are fully synchronized by hardware trigger pulses. Note that, the employed cameras also support PTP protocol, which can be used for verification of the synchronization precision of our framework. The whole system is synchronized with UTC by GNSS timing. To validate the synchronization precision of the system, we intentionally distribute four cameras on two

PTP-synced host computers. This setup allows us to verify the timestamp difference between the cameras which are triggered by consistent pulse rising edges. The results, as shown in Fig. 7, indicate that the synchronization precision has achieved 1.56us within the 68.3% confidence interval (one standard deviation) based on statistical analysis. It is important to note that, since the cameras need to capture images to complete the test, this already represents the sync-quality of the system under full collection load, which also tactfully proves the robustness of our framework. To benefit the community, a detailed synchronization guideline is provided on our website.

VII. Data Collection

A. Scene Preparation

Scene preparation mainly involves the selection of collection scenes and the arrangement of environmental conditions.

Collection scenes, depending on their appearance and structural properties, can generate different difficulties in ANM datasets. Generally, the scenes can be categorized into structured and unstructured ones. A structured scene refers to a scene with clear organization and regular patterns. Such scenes often exhibit apparent motifs, geometric shapes, and specific layouts. Examples include indoor rooms, urban buildings, and road networks. In structured scenes, the textures and structures are straightforward to recognize, track, and reconstruct, making navigation and mapping less difficult. On the other hand, unstructured scenes lack explicit organization and regularity. These scenes tend to have complex shapes, randomness, and diversity. Examples include natural environments, forests, mountains, and deserts. In unstructured scenes, the features and attributes may be irregular and highly variable, making navigation and mapping relatively tough.

There are several special scenes that are known to be challenging for ANM systems, which are also the current research focus of this field:

- GNSS-denied environments, such as indoor scenes [134], undergrounds [16], thick vegetations [41], and urban canyons [8], which are likely to cause signal loss and inaccurate satellite positioning.
- Texture-less areas, such as pure-white walls [135], uniform grasslands [64], sandy beaches [136], and more, which are difficult for visual tracking.
- Degeneration areas, such as long corridors [42], tunnels [8], caves [137], and so on, which may cause misregistration for LiDAR-based methods.

Aside from the scene type, environmental conditions can also impact the suitability and complexity of the dataset. This primarily involves factors such as lighting intensity, weather conditions, seasonal effects, time frame, and dynamic objects and humans. A high-quality dataset should encompass a wide range of environmental conditions to meet the demands of comprehensive algorithm testing.

B. Collection Policies

When collecting ANM datasets, there are certain requirements and considerations to ensure the quality and suitability of the data. Some common points are:

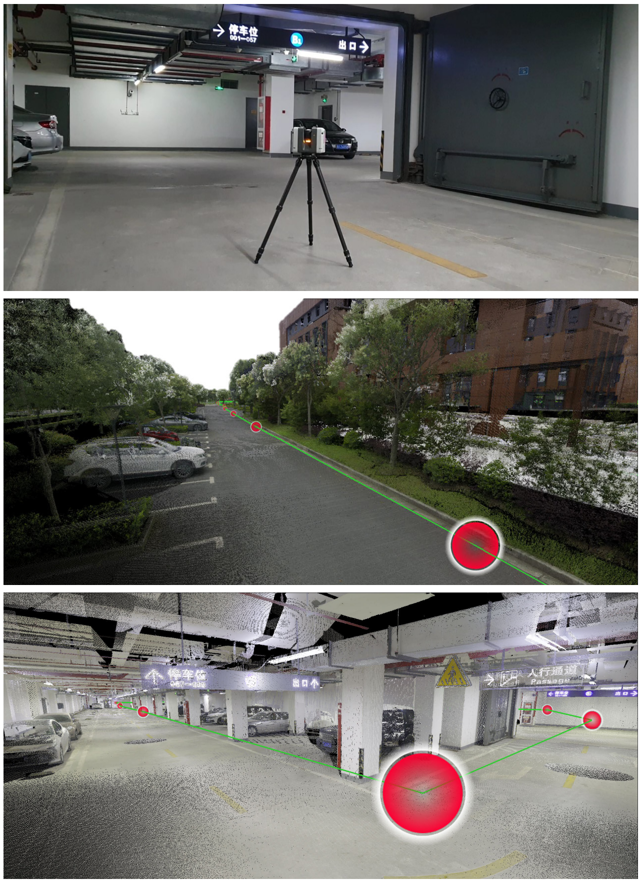


Fig. 8. Map construction by registering numerous individual scans as a whole with graph optimization.

- Completeness and Coverage: The collection path should cover the entire scene as much as possible. It is essential to ensure that the path traverses various types of areas, including open spaces, narrow passages, corners, regions of different heights, and so on, which are essential for the thorough testing of algorithms.
- Pose and Motion Diversity: The platform motion and poses along the path should exhibit diversity, including different viewpoints, rotations, translations, and accelerations. This mainly helps to test the robustness and adaptability of algorithms.
- Adequate Path Overlap: Sufficient path overlap is important for triggering the relocalization and loop closure functions of algorithms [26], which is one of the key evaluation metrics for ANM. Additionally, for mapping-centric systems, adequate overlap can improve the accuracy and consistency of the map.

VIII. Ground Truth Generation

A. Pose Ground Truth

The technique for generating GT-pose could differ regarding different collection scenarios. For simple small-scale and indoor scenes, the most effective solution is the motion capture system (MoCap) [6], [138]. It works by emitting infrared light to the reflective markers and recognizing their imaging pixel locations, thus the marker pose can be precisely solved.

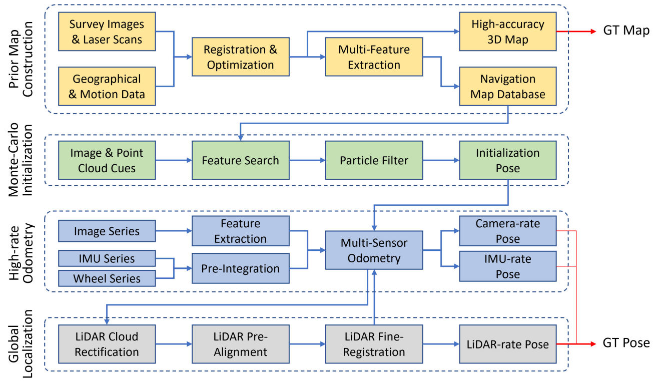


Fig. 9. Schematic graph of the proposed mapping-localization ground truth system.

But there are still drawbacks of it: MoCap cannot work well under severe occlusion or areas with infrared interference (*e.g.*, strong sunlight) [138]. For large-scale outdoor scenarios, D-GNSS could be the most effective and mature solution [4]. It can provide up to cm-level positioning accuracy, and with a dual-antenna configuration, the orientation states can also be accurately measured. However, the working requirements are also strict: the receiver should be exposed to an open sky [4], otherwise, its accuracy may seriously decline thus cannot meet the standard of ground truth. For open environments, another reliable choice is the Laser Tracker (LasTrack) system which attains up to sub-mm precision [64], [43]. However, it cannot provide orientation measurement, and occlusion-free should be ensured [43] in case of tracking losses.

The above are all high-precision but expensive techniques, some cheaper alternatives include the use of visual marker [139], ultrasound positioning [140], and SLAM with multi-sensor-fusion [8]. These methods are passable choices for testing purposes, while they are likely to suffer from low precision and incomplete trajectory coverage, making them unqualified for building high-quality ANM datasets.

B. Map Ground Truth

For GT-map reconstruction of a regular-sized environment, the ideal solution is to conduct rigorous survey works using professional terrestrial laser scanners (see Fig. 8), which make sure to provide mm-level precision in each individual scan, and can achieve cm-level accuracy in global coordinates after registration [41]. Benefiting from the stationary scan process, the point cloud is noiseless and consistent. However, when extended to city-scale environments, the time cost could be exceptionally long. In such a case, a more effective solution is to use a professional MMS (mounted by ground or aerial vehicles) for mobile collection. Such systems fuse tactical D-GNSS&INS, survey-grade LiDARs, and odometry data, capable of building maps at cm-level global-accuracy [75]. On the other hand, in cases where high-end instruments are not accessible, one can consider reconstructing the map using SLAM with moderate sensors [141] and multi-sensor fusion. However, this approach can hardly achieve a qualified accuracy for constructing standard datasets.

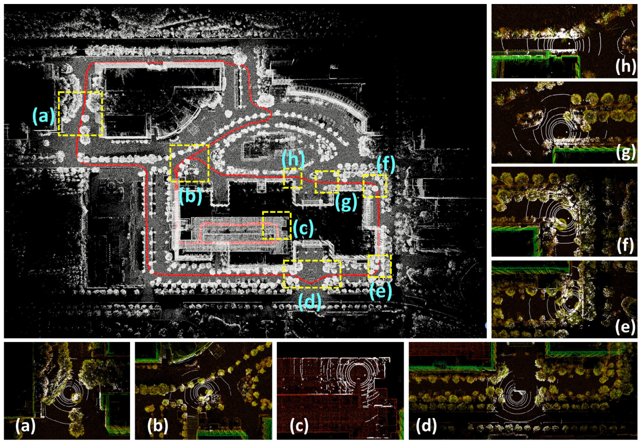


Fig. 10. GT-pose generation within the pre-build high-quality map database. The real-time LiDAR point clouds are fully undistorted for final registration.

C. Integrated Mapping-Localization Ground Truth System

To solve the challenge of generating GT-pose in large-scale complex environments, we propose a high-accuracy mapping-localization integrated ground truth system, as illustrated in Fig. 9. The methodology of this system involves creating a high-accuracy prior 3D map as a global reference, and then utilizing the information from onboard LiDAR and other sensors to achieve matching, tracking, and localization within the map. This mechanism is fundamentally similar to satellite navigation, as both aim to establish high-accuracy global references and perform reliable pose estimation within the coverage area. However, unlike satellite navigation, this system can operate in any complex environment without the need for external equipment once the map has been pre-scanned. It is also resilient to environmental obstructions or variations in scene scale. The system primarily consists of four modules: the multi-level feature map construction module, the Monte Carlo initialization and localization module, the high-frequency multi-sensor fusion odometry module, and the LiDAR point cloud rectification and registration module.

Firstly, in the map construction module, professional surveying works are conducted to collect laser point clouds, images, and geographical data all around the scene. This is typically accomplished by mobile mapping systems and high-end terrestrial laser scanners, as illustrated in Fig. 8. Subsequently, the point clouds and images from each individual frame or station are registered and stitched together to form a globally consistent high-accuracy 3D map. Meanwhile, salient vision and structural features are extracted and associated with map coordinates to support subsequent position indexing. Next, in the initialization and localization module, the real-time vehicle vision and LiDAR data are searched within the map database for key feature matching. A Monte Carlo localization model is then constructed to ensure that the initialization results converge correctly within a short period of time. Once the initialization is complete, the vehicle enters a dual-thread tracking framework that combines high-frequency local odometry and timely global localization. The odometry thread utilizes high-frequency data from vision, IMU, and encoders

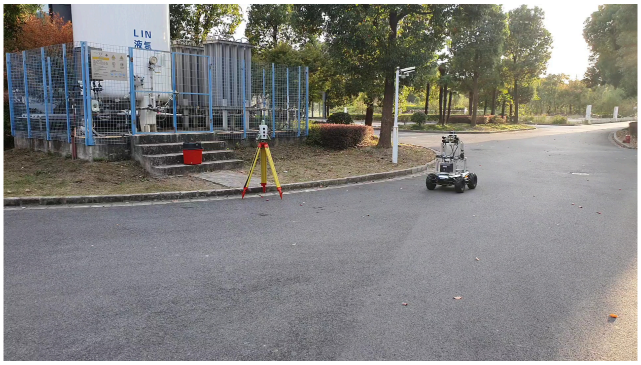


Fig. 11. Validation of the proposed mapping-localization ground truth system by a Leica MS60 (1mm accuracy) Laser Tracker.

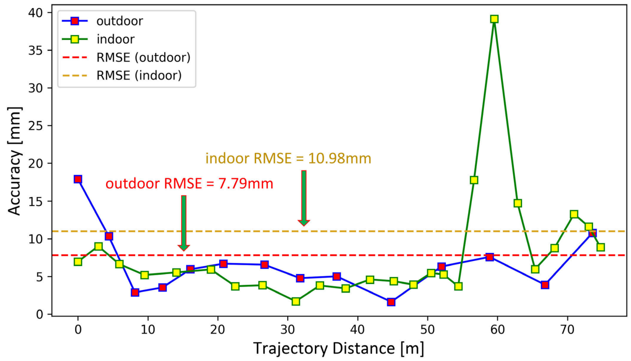


Fig. 12. Evaluation of GT-localization accuracy for both indoor and outdoor environments: 10.98mm RMSE (indoor) and 7.79mm RMSE (outdoor).

TABLE IV
COMPARISON OF GROUND TRUTH TECHNIQUES

GT	Technique	Degree	Accuracy	Requirement
Pose	MoCap	6D	sub-mm/sub-deg	Sparse sunlight
	D-GNSS	3D/6D	up to cm-level	Outdoor open-sky
	LasTrack	3D	up to sub-mm	Sparse occlusion
	SO/SLAM	3D/6D	2%-10% length	Multi-sensor fusion
	Marker, Ultrasound	3D/6D	dm-level	Accurate placement
Map	Scanner	3D	mm- to cm-level	Sufficient overlap
	MMS	3D	cm-level	Authentic sensors
	LMS	3D	2%-10% length	Multi-sensor fusion
Full	MapLoc (proposed)	3D-6D	cm-level	Avoid deserted area

to achieve rapid and smooth motion estimation. Subsequently, based on the odometry poses, by LiDAR undistortion and fine-registration with prior map (as illustrated in Fig. 10), the global localization thread makes sure to generate accurate GT-poses at LiDAR frame rate. Finally, the output GT-poses are also back-propagated to the odometry module, enabling real-time bias correction and thus boosting the ground truth much higher to the IMU frame-rate (over 100Hz).

To demonstrate the effectiveness of our methodology, we carry out a rigorous workflow in a large-scale indoor-outdoor interconnected scenario. For GT-map construction, we employ a survey-grade Leica RTC360 laser scanner and conduct an authentic survey and mapping job with professional colleagues. This scanner can acquire dense and colored point clouds with 1mm accuracy and 130m ranging radius in each individual scan. To ensure the completeness of the map, we have scanned 137 locations all around the scene. By accurate pre-alignment and fine-registration, we have achieved an overall 8mm accuracy in the final map as reported by the Leica Cyclone Register360 software. For GT-pose generation, we fully implement the proposed mapping-localization system on our robot platform, yielding complete trajectories in all short and long sequences (a sample trajectory is shown in Fig. 10). To further verify the accuracy of this system, a Leica MS60 laser tracker (1mm accuarcy) is employed to track the crystal prism mounting on top of the LiDAR center for reference positioning, as shown in Fig. 11. We choose two paths without occulusion to conduct the experiments, ensuring that MS60 can work consistently. After alignments for both trajectories, we have achieved an accuracy of 10.98mm root-mean-square error (RMSE) for the indoor test and 7.79mm RMSE for the outdoor test, as shown in Fig. 12.

A comparison of different GT techniques has been shown in Table IV, depicting their data degree, theoretical accuracy, and operation requirements. Among these solutions, our proposed system can provide ultimate cm-level mapping and localization accuracy with the slightest requirements (avoid deserted areas, where D-GNSS is the optimal choice). We recommend researchers comprehensively consider the collection environments, financial cost, desired accuracy, and GT degree for the determination of ground truth techniques in dataset creation.

IX. SYNTHETIC DATASET PRODUCTION

As illustrated in previous sections, the collection of real-world ANM datasets [4] is known to be complicated and time-consuming, making the data difficult to collect frequently or to meet the timely customization demands of users. To circumvent these issues, synthetic datasets that simulate real-world vehicles, sensors, motions, and environments have been proposed for testing purposes [142]. By computerized techniques, the environmental factors, such as weather, illumination, and dynamic objects can be fully controlled, and the motion patterns and trajectories can be arbitrarily customized within the same environmental settings [143], which can support variable-controlled and closed-loop simulators [144].

Depending on whether part of the data is seeded from the real world, synthetic approaches [145] can be categorized into full-simulation and semi-simulation ones. Section III has already described a general construction pipeline of synthetic datasets, ranging from platform construction to ground truth generation. This section will delve deeper into the inherent methodology from a technical perspective, with some milestone works analyzed to serve as good lessons.

A. Full-simulation Synthesis

In full-simulation synthesis, the entirety of the dataset is artificially generated, devoid of any real-world data seeding. This approach empowers researchers to design and create datasets from scratch, tailoring them precisely to the needs of experiments. The cornerstone of this approach lies in the creation of virtual 3D environments and rendering of realistic data. Full-simulation synthesis can generally be categorized into three technical lines: synthesis with 3D creator and renderer, synthesis with vehicle simulator, and synthesis with 3D computer game.

1) Synthesis with 3D Creator and Renderer: Synthesis with 3D creator and renderer involves modelling a virtual environment and simulating realistic data within it. Off-the-shelf 3D creators and renderers include OpenGL [146], Unreal Engine [147], PovRay [148], Unity [149], *etc*. Many popular synthetic datasets were constructed with this methodology, including SYNTHIA [142], SceneNet [150], Replica [151], InteriorNet [143], and more. SYNTHIA created a virtual city using Unity3D platform and simulated a moving car within it. SceneNet built virtual environments directly from CAD designing, and the collection was simulated by Chrono physics engine [152] and NVIDIA ray tracer. Replica also leveraged diverse indoor CAD models and simulated vehicle exploration within it. InteriorNet imported large-scale 3D furniture designs from Kujiale, and dedicated simulator and renderer were developed to provide realistic data simulation. For such implementations, the quality of scene models plays a crucial role towards high-fidelity rendering, and meticulous design of sensor models are also required for realistic data simulation.

2) Synthesis with Vehicle Simulator: In recent years, with the rapid development of autonomous vehicles, there has been a growing demand for simulation data, leading to the emergence of various open-source vehicle simulators. Such simulators can provide mature and full-stack data simulations, including environment construction, dynamic objects, real-time manipulation, comprehensive sensors modelling, and ground truth generation, greatly improving production efficiency of datasets. Among the off-the-shelf simulators, CARLA [144] and AirSim [153] are most extensively used for dataset creation. CARLA is a simulator focused on road scenarios, arising many popular datasets such as V2X-Sim [154], CarlaScenes [155], and SHIFT [60]. These datasets offer a wide array of sensor types, including camera, LiDAR, GNSS, and IMU, along with accurate trajectories, maps, and semantic ground truth, supporting various tasks for on-road vehicles. AirSim, on the other hand, is a simulator suitable for both aerial and ground vehicles. Compared to CARLA, it provides more challenging motion patterns. TartanAir [38] could be the most representative dataset constructed using this simulator. It creates a large number of extreme environmental conditions in AirSim and provides a rich array of data types, including depth images, optical flow, occupancy maps, and more. There are also other simulators like LGSVL [156] and Gazebo [157] that can achieve similar functionalities, enabling mature closed-loop simulations as well. In summary, public vehicle simulators have offered significant convenience for data production and frequent testing of algorithms. However, since they are also built on 3D platforms like Unreal Engine and Unity, there is still considerable room for future improvement in terms of image rendering quality.

3) Synthesis with 3D Computer Game: Open-source off-the-shelf simulators often suffer from lower image rendering quality, which significantly affects the authenticity of the generated datasets. In contrast, using meticulously crafted and high-budget 3D computer games for data collection ensures a high level of realism. In such case, the challenge lies in the fact that the source code and game content are typically not open to the public. To address this problem, Richter et al. [158] proposed a novel data simulation methodology based on the GTA-V game engine, yielding the large-scale VIPER vision dataset [158]. They injected a middleware between the game and the graphics library to intercept rendering commands. By integrating software updates, bytecode rewriting, and bytecode analysis techniques, they achieved scene object recognition, coordinate extraction, and data association, enabling real-time generation of ground truth such as scene structures, camera trajectories, and semantic annotations. However, due to the restriction of the game engine, only monocular camera sequences can be provided, lacking other commonly used vehicle sensors like LiDAR and IMU, which limits its application range for testing of diverse algorithms.

B. Semi-simulation Synthesis

Semi-simulation synthesis leverages real-world data seeding for synthetic dataset generation, resulting in a hybrid approach that capitalizes on the benefits of both the virtual and real world. This blending allows for a substantial increase in data realism, which can significantly enhance the applicability of datasets. At present, semi-simulation is typically achieved with four approaches: trajectory cloning, scene layout cloning, scene model cloning, and neural scene/sensor simulation.

1) Trajectory Cloning: Trajectory cloning entails recording real-world motions to capture datasets within virtual environments. Handa *et al.* [159] and Antonini *et al.* [160] present two exemplary implementations of this approach, yielding the popular ICL-NUIM and BlackBird datasets, respectively. Handa employed Kintinuous SLAM [161] to generate real-world trajectories on several handheld RGB-D sequences. These paths were then transposed into two virtual indoor environments, where realistic image sequences were captured through the Pov-Ray [148] render engine. Antonini recorded several aggressive UAV flights using an accurate motion capture system. Subsequently, these trajectories were replayed with FlightGoggles [162] engine to generate sequences across multiple scenarios. By cloning real-world trajectories, such simulation can attain a high realism at motion pattern level, greatly improving the authenticity of synthetic datasets.

2) Scene Layout Cloning: Scene layout cloning involves constructing virtual environments by faithfully replicating the real-world scene layout, encompassing structure, assets, dynamic objects, and more. Representative implementations include Virtual KITTI [163] and Virtual KITTI2 [164]. These two datasets selected 5 sequences from the original KITTI dataset [4] for extracting real-world environmental elements. Subsequently, leveraging the Unity engine [149], twin virtual environments were constructed, and diverse weather and environmental conditions are simulated to generate new synthetic sequences. Through this approach, the characteristics of the original environment can be preserved to some extent, allowing for variable-controlled scene manipulations and facilitating comprehensive testing of algorithms.

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3360273

3) Scene Model Cloning: Scene model cloning involves scanning real-world 3D environments and simulating new sequences within the twin scenarios. Representative implementations include HM3D [165] dataset and CMU-Exploration [166] platform. HM3D used Matterport to generate thousands of high-resolution indoor 3D scans, and these models were then imported into Habitat Sim to serve as virtual environments. In contrast to HM3D, CMU-Exploration was specifically designed to accommodate larger-scale environments, encompassing both indoor and outdoor spaces. It can support more various types of scene models, including photo-realistic models from Matterport3D and CMU-Reacon, as well as professional survey models of point cloud environmental maps. Then by configuring motion trajectories within the simulators, numerous synthetic sequences can be generated. In comparison to cloning motion trajectories and scene layouts, replicating scene models allows for more detailed incorporation of real-world elements. However, on the other hand, due to the inherent imperfections of scene models and the unfamiliarity with new viewpoints, achieving highly realistic image rendering has become a significant technical challenge.

4) Neural Scene/Sensor Simulation: To overcome the limitations of conventional rendering methods in realistically reproducing real-world environmental conditions and achieving high-fidelity synthesis from unfamiliar viewpoints, a recent breakthrough called Neural Radiance Fields (NeRF) [167] has been proposed and widely explored. NeRF is an advanced 3D modeling and novel view synthesis technique based on neural networks. As illustrated the principle in Fig. 13, it represents a scene using a fully-connected deep network, whose input is a single continuous 5D coordinate (spatial location (x, y, z) and viewing direction (θ, ϕ)) and whose output is the volume density and view-dependent emitted radiance at that spatial location. By using volume rendering techniques, a novel synthesized image can be composited by querying along camera rays to integrate the related values. As the rendering function is differentiable, by minimizing the residual between synthesized and ground truth observed images, a neural scene representation can be trained and optimized, and accurate lighting conditions can be rendered. Based on this underlying mechanism, many neural scene/sensor simulators and datasets have been proposed, including READ [168], MARS [169], UniSim [61], NVIDIA Drive Sim [62], and more, which achieve not only highly realistic vision renderings but also multi-sensor simulation. Up to this point, Neural scene/sensor simulation can be regarded as the approach closest to real-world settings. However, due to the difficulty in scene manipulation, they also present greater challenges when it comes to achieving closed-loop simulation.

Generally, full-simulation synthesis is much easier to accomplish, while semi-simulation can provide outstanding data realism (see Fig. 14 for a comparison). During the early stage of algorithm development, conducting rapid feasibility tests is quite essential. Therefore, considering the developmental cost, it is reasonable and widely encouraged to use full-simulation synthesis for data production. However, as the algorithm progresses to the testing and elevation phases, excessive reliance on full-simulation dataset should be avoided. Researchers are supposed to comprehensively incorporate high-fidelity semi-simulation data and real-world data to ensure the correctness of model training and performance assessment.

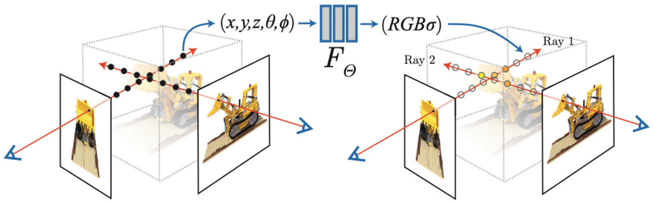


Fig. 13. An overview of the NeRF scene representation [167].

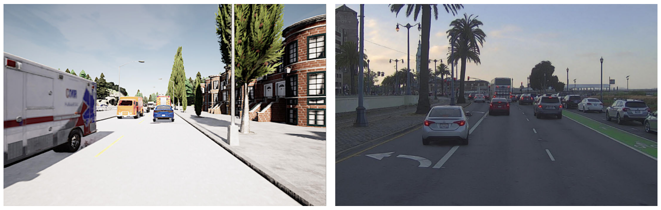


Fig. 14. A comparison of data realism between SHIFT (left) and UniSim (right), which are produced by full-simulation (CARLA) and semi-simulation (Neural) respectively. The artifact of the left image can be clearly recognized, while it is fairly hard to distinguish the right image from real world.

X. Benchmark Criteria

A. Pose Criteria

For pose (position and orientation) evaluation, the most widely used criteria could be the two proposed in the TUM RGB-D Benchmark [5]: Relative Pose Error (RPE), and Absolute Trajectory Error (ATE).

RPE investigates the local motion accuracy across a fixed time interval Δ. Define the estimated poses as $\boldsymbol{P}_1, ..., \boldsymbol{P}_n \in SE(3)$ and the ground truth poses as $\boldsymbol{Q}_1, ..., \boldsymbol{Q}_n \in SE(3)$, then the relative pose error at time step i over period Δ could be defined as:

$$\boldsymbol{E}_i := (\boldsymbol{Q}_i^{-1}\boldsymbol{Q}_{i+\Delta})^{-1}(\boldsymbol{P}_i^{-1}\boldsymbol{P}_{i+\Delta}). \tag{1}$$

Considering a sequence of n camera poses, there could be (m=n–Δ) individual *RPEs*. To measure the global performance, *RPE* metric is proposed to compute the (*RMSE*) over all possible time intervals, and typically only the translational part is taken into account:

$$RMSE(\boldsymbol{E}_{1:n}, \Delta) := \left(\frac{1}{m}\sum_{i=1}^{m}||trans(\boldsymbol{E}_i)||^2\right)^{1/2}. \tag{2}$$

Note that due to the amplification of the Euclidean norm, *RMSE* could be sensitive to outliers. So, if desired, it is also reasonable to evaluate the mean or median errors as an alternative. Additionally, the selection of time interval Δ could vary with different systems and conditions. For instance, Δ can be set to 1 to examine real-time performance, resulting as per frame drift. As for systems estimating states based on a batch of frames, performing local optimization, or focusing on long-range navigation, it is not necessary nor reasonable to count on each individual frame. However, it is also not appropriate to set Δ to n directly, because it penalizes early rotational errors much more than those that occurred near the end. Instead, it is recommended to average the *RMSEs* over a wide range of time intervals. In practice, to avoid excessive calculation complexity, a simplification that counts on a fixed number of sample intervals could also make sense [4].

ATE measures the global accuracy by comparing the estimated trajectory against the ground truth to get absolute distances. As the two trajectories could lie in different coordinates, an alignment via a rigid-body transformation $\boldsymbol{S}$ that maps the estimated poses $\boldsymbol{P}_{1:n}$ onto the ground truth poses $\boldsymbol{Q}_{1:n}$ is required in advance. Then the *ATE* at time step i could be computed as:

$$\boldsymbol{F}_i := \boldsymbol{Q}_i^{-1}\boldsymbol{S}\boldsymbol{P}_i. \tag{3}$$

Similar to *RPE*, *ATE* is also proposed to compute the *RMSE* over all time indices of the translational part:

$$RMSE(\boldsymbol{F}_{1:n}) := \left(\frac{1}{n}\sum_{i=1}^{n}||trans(\boldsymbol{F}_i)||^2\right)^{1/2}. \tag{4}$$

For ease of expression and calculation, in the above definitions, only translational parts of the poses are taken into account. However, as biased rotations can also result in drifted translations, evaluating only the translational components can still provide insight into the overall performance of the algorithm, covering both position and orientation aspects. In practice, compared to *RPE*, *ATE* has an more intuitive error visualization on the whole trajectories, which is benefit for inspecting the accidental position of the algorithms.

Building upon the aforementioned criteria, over the years, there have been various extensions and complementary approaches that contribute to more comprehensive evaluations. For example, KITTI extends *RPE* to include rotational errors as well for a thorough assessment [4]; TartanAir defines the Success Rate (SR) metric as the ratio of non-lost frames [38], which is particularly useful under the context that both *RPE* and *ATE* have not accounted failure instances inside.

B. Map Criteria

The evaluation criteria for 3D reconstruction mainly originated from the Middlebury dataset [170]. Denoting the ground truth as G and the reconstruction as R, there are mainly two metrics: reconstruction accuracy, and completeness.

The accuracy measures how close R is to G by computing the distances between the corresponding points, which can be determined by the nearest match [171]. If the models are in other formats like triangle meshes, the vertices could be used for comparison. One issue could be encountered in case G is incomplete, resulting in the nearest reference points falling on the boundary or at a distant part. In such a case, a hole-filled model G' is recovered (see Fig. 15), and the points matched to the repaired region will be discounted in calculation [170].

The completeness investigates how much of G is modeled by R. In contrast to the accuracy metric that compares R against G, the completeness metric evaluates the distances from G to R. Essentially, if the distance exceeds a specified threshold, we can conclude that there is no corresponding point of G on R. Consequently, such kinds of points can be logically inferred as "not reconstructed", as illustrated in Fig. 15.

TABLE V
QUALITY COMPARISON OF OUR DESIGNED DATASET USING THE PROPOSED METHODOLOGY WITH SEVERAL SOTA DATASETS

Dataset/Metrics	Spatial-Calib ≈ prec.[a]	Time-Sync ≈ prec.[b]	GT-Pose ≈ accur.	GT-Map ≈ accur.
Rawseeds [125]	-[c]	5ms	sub-dm	-
KITTI [4]	sub-dm [106]	5ms	sub-dm	-
KAIST Urban [8]	sub-dm [173]	-	dm-level	dm-level
OpenLORIS [174]	sub-cm	ms-level	few% dist.	-
M2DGR(out) [43]	-	10ms	sub-dm	-
NewerCollege [175]	-	-	5cm	1cm
Ours	sub-cm	us-level	2cm	1cm

[a] Calibration precision between Camera and LiDAR.
[b] Synchronization precision among Camera, IMU, and LiDAR sensors.
[c] Data not reported or difficult to ascertain.

TABLE VI
NAVIGATION ASSESSMENT OF SOTA ANM ALGORITHMS AND GNSS AGAINST TRAJECTORY GROUND TRUTH

Sequence	0805-01		0805-02		0806-04	
Method/Metric	RPE/%	ATE/m	RPE/%	ATE/m	RPE/%	ATE/m
VINS-Mono	1.573	3.523	7.411	10.773	3.798	5.435
ORB-SLAM3	4.543	3.653	11.696	15.017	7.534	9.334
LOAM	2.029	1.022	2.501	3.565	10.254	15.645
Fast-LIO2	1.482	1.754	2.770	8.117	3.914	6.833
LVI-SAM	1.553	1.119	1.402	3.900	3.465	2.998
GNSS (outdoor)	N/A	N/A	12.202	13.471	5.970	8.203

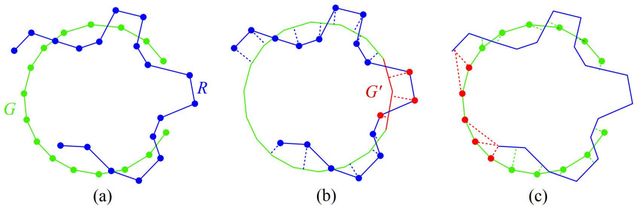


Fig. 15. Evaluation policy while G is incomplete [170]. (a) the two models. (b) the matches with hole-filled area will not be included. (c) those matched distances run beyond the threshold will be regarded as "not reconstructed".

At present, regarding pose and map assessments, the principles outlined above have already gained extensive recognition and adoption, proving their suitability and correctness. When creating new datasets, one can readily draw upon these established criteria and the available open-source tools [172], [114]. Nevertheless, depending on the complexity and specific focus, it is also encouraged to propose novel metrics, methods, and tools for a more comprehensive evaluation.

XI. DATASET AND BENCHMARK DEMONSTRATION

A. The ParkingLot Dataset

To demonstrate the effectiveness of the proposed methodology, we design a high-quality dataset in an indoor-outdoor connected scenario based on our constructed robot platform (see Fig. 3 for platform design and coordinates). Comprehensive data types are provided, including stereo RGB and Gray vision sequences, spinning and MEMS LiDAR sequences, industrial and consumer-grade IMU series, wheel odometry, and GNSS

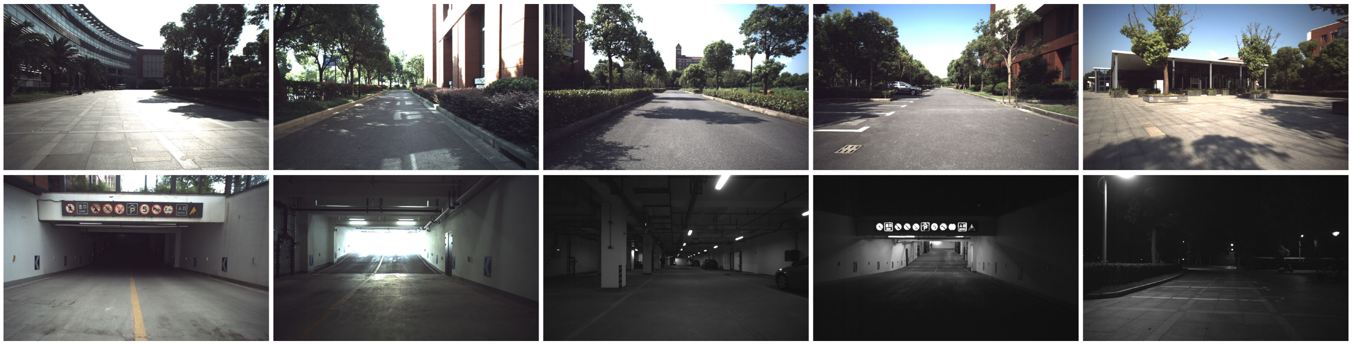


Fig. 16. Sample images of the indoor-outdoor connected ParkingLot dataset.

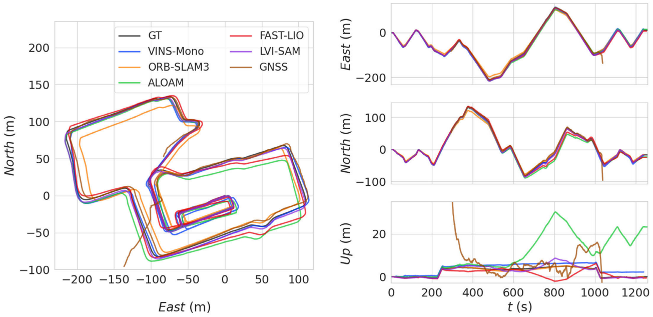


Fig. 17. Navigation results of SOTA ANM algorithms and GNSS against GT trajectory (0806-04). GNSS drifts quickly at outdoor-indoor connected areas and performs badly in lush vegetated areas (large z-axis errors), which also proves the versatility and significance of our Map-Loc ground truth system.

series (sensor specifications listed in Table II). The spatial calibration and time synchronization processes are illustrated and demonstrated in Section V and Section VI. Various urban features are covered inside, including high-rise buildings, wide and narrow roads, lush vegetations, open parking lots, underground parking lots, cafeteria, *etc.*, as shown in Fig. 16. To completely duplicate the scenario, we traverse 8 sequences of 8.27km in total, covering short and long trajectories, day-time and night-time illuminations, loop closures, and sharp turns, leading to a thorough and challenging benchmark. To facilitate comprehensive algorithm assessments, We provide both trajectory and 3D map ground truth (the GT-map and a sample trajectory are shown in Fig. 10), which were generated using the proposed Map-Loc GT system in Section VIII. In Table V, we compare the key metrics of our dataset with several SOTA multi-sensory datasets. The results indicate that our dataset possesses a comprehensive quality evaluation, and its precision/accuracy of spatial calibration, time synchronization, and ground truth all reach industry-leading levels.

TABLE VII
MAPPING ASSESSMENT OF SOTA ANM ALGORITHMS AGAINST 3D-MAP GROUND TRUTH

Sequence	0806-02		0806-04	
Method/Metric	Accuracy/m[a]	Complete/%	Accuracy/m	Complete/%
LOAM	0.675	82.351	3.095	57.969
Fast-LIO2	0.593	69.339	0.888	71.091
LVI-SAM	0.275	86.950	0.451	88.366

[a] We adopt one standard deviation (std.) here to represent the accuracy metric.

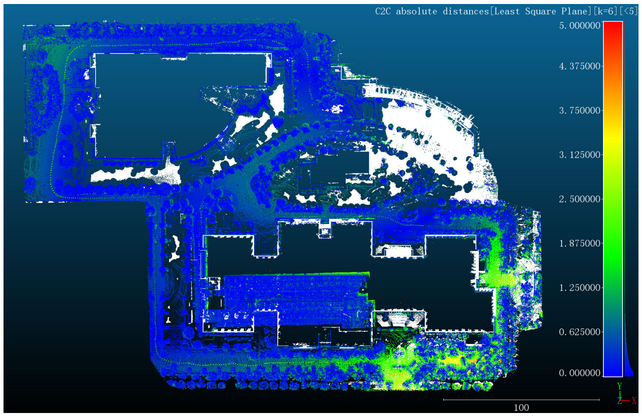


Fig. 18. LVI-SAM mapping evaluation (0806-04). We set the completeness threshold to 2m, so that points in GT map with >2m matching distance (white part) will be regarded as incomplete (either caused by >2m mapping error or lack of coverage), which in practice means should to be further complemented to meet the required qualifications. The accuracy metric is computed by the point-to-plane registration error between the LVI-SAM map and GT map, with all the distances considered (rather than the <2m parts, to comprehensively indicate the mapping performance).

B. SOTA Algorithms Benchmarking

To demonstrate the versatility of the constructed dataset, we comprehensively test SOTA ANM algorithms of different sensor modalities on representative sequences, including indoor-only (0805-01), outdoor-only (0806-02), indoor-outdoor connected (0806-04), and night-time scene conditions (0805-02). Specifically, we choose VINS-Mono (mono-IMU) [21] and ORB-SLAM3 (stereo-IMU) [45] for testing of Visual-Inertial navigation, choose LOAM [23] and Fast-LIO2 [123] for testing of LiDAR and LiDAR-Inertial navigation and mapping, and choose LVI-SAM [24] for testing of Visual-LiDAR-Inertial navigation and mapping. We follow the benchmark criteria from Section XI for performance assessments, and the evaluation results and visualization are shown in Table VI, Table VII, Fig. 17, and Fig. 18. The successful workflow demonstrates that our methodology can construct full-task ANM datasets and enables comprehensive performance assessments for algorithm benchmarking.

XII. Challenges and Future Directions

A. Challenges

1) Data Collection and Annotation Costs: Gathering and annotating ANM datasets are expensive in terms of hardware and human resources. One has to procure mobile vehicles and mainstream sensors for platform integration, as well as high-end instruments for GT generation, which are huge financial burdens for individual researchers and smaller institutions. Besides, data collection and post-processing are known to demand significant human labor for conducting field experiments and annotations. These factors are very likely to hinder the large-scale expansion of standard datasets.

2) Accuracy Evaluation for Ground Truths: Datasets with deficient GT accuracy may cause bias assessments and should not be included in the compliant dataset repository. Although we have analyzed the theoretical accuracy of existing ground truth techniques in Section VIII, and experimentally demonstrated the accuracy of our proposed Map-Loc GT system, their actual performances are still largely dependant on the on-site operations and configurations, with potential significant variations in accuracy across different scenarios. For example, the accuracy of D-GNSS can significantly decrease in complex environments such as bridges, tunnels, and urban canyons; besides, when using terrestrial scanner for GT-map, its accuracy could also decrease if the scan-overlaps are not enough. Given the inherently high accuracy of the GT itself, evaluating it will demand much higher level measurement techniques in specific environments. This undoubtedly presents a notable challenge in both equipment and technology aspects.

3) Data Realism of Synthetic Datasets: Despite their crucial role, synthetic datasets are currently suffering from a lack of realism in terms of sensor models, motion patterns, and especially image quality. For datasets generated by full simulation, the image renderings exhibit distinctly artificial, which are not realistic enough to substitute real-world datasets. On the other hand, semi-simulation, despite its considerable potential to produce highly-realistic image renderings, remains in an nascent stage and faces a shortage of readily available open-source materials. As a whole, due to the technical intricacies and developmental costs, attaining mature near-true simulators will demand huge research efforts.

B. Future Directions

1) High-quality Challenging Datasets: Future development of ANM datasets should continue to emphasize challenging scenarios that push the boundaries of algorithms. This entails creating datasets featuring dense dynamic objects, adverse weather conditions, intricate scene structures, and situations where conventional sensors encounter limitations. Such challenges will push researchers to devise solutions that can excel in the most demanding real-world settings.

2) Automatic Data Annotation System: Developing automated data annotation systems will greatly accelerate the construction process for high-quality datasets. Such systems should support a diverse range of data types, such as vision semantics, LiDAR semantics, object tracking, and more. These data will enhance the application range of datasets.

3) Specialized Sensor Suites and Hardware: General-purpose hardware and sensors commonly encounter issues of integration and exhibit suboptimal performance when adapted for field experiments. Therefore, it is advisable to explore the development of specialized sensor suites that not only offer comprehensive modalities but also with a reduced cost. Additionally, there is a demand for novel industrial computers equipped with ample bandwidth and protocol support to guarantee reliable data collection and precise time synchronization.

4) Near-true and Closed-loop Simulator: A high-fidelity simulator plays a pivotal role in bridging the gap between the slow production of real-world data and the urgent demand for comprehensive datasets. Besides, it effectively addresses the scarcity of corner cases in real-life data collection. To further advance in this field, future endeavors should be focused on elevating data realism, with the ultimate goal of attaining near-true closed-loop simulations within the virtual domain.

XIII. Conclusion

This paper proposes a full-stack methodology for construction of standard ANM datasets. Specifically, the introduced versatile calibration and synchronization frameworks, along with the proposed integrated Map-Loc ground truth system, address the long-standing challenges within the field.

We have utilized the proposed methodology to construct a multi-sensor robot platform and curated a high-quality dataset. Through rigorous experimental validation, the dataset achieves sub-cm spatial calibration precision, us-level time synchronization precision, 2cm accuracy for pose ground truth, and 1cm accuracy for 3D map ground truth. By comparing our dataset quality with other state-of-the-arts, we have achieved industry-leading levels across all the key metrics. This indicates that our methodology can significantly reduce the dataset construction threshold, thereby accelerate breakthroughs in ANM field.

At the same time, we are also considering a crucial question: what quality standard should a dataset meet to be considered satisfactory? Currently, setting a strict high threshold may seem overly ambitious. However, based on the methodology proposed in this paper, even without employing optimal hardware configurations (such as assembly without CAD models, using hardware with merely NTP protocol, or utilizing less-accurate mobile mapping systems), achieving calibration precision of better than 2cm, synchronization precision in the sub-ms range, and accuracy of pose and map ground truth at sub-dm level are still very straightforward. These are the quality expectations we currently have for standard ANM datasets.

Given the inherent challenge for individuals or single institutions to build a grand comprehensive dataset on their own, we believe the future flourishment of ANM data society should be achieved by coordination and crowdsourcing. To achieve this goal mainly involves three stages. The first stage involves establishing a full-stack methodology for constructing standard datasets, which has been addressed in this paper. The second stage is the defination of quality standards for datasets by authoritative organizations. Though we have set preliminary standards based on the lower hardware specifications or the absence of certain protocols when using our methodology, further iterations and evaluations based on algorithm performance are still necessary for objective refinements. Ultimately, authoritative organizations such as IEEE, ITSS, and RAS will be responsible for setting the final standards. The third stage involves exploring a crowdsourcing data collection framework and establishing a cloud platform for data-sharing. This framework will encompass modules of task distribution and management, data quality control, data security and privacy protection, data integration and processing, data storage and openness, *etc*. In subsequent research, we will continue to explore the technological framework and systems for crowdsourced data collection. Additionally, we encourage more authors to present their perspectives on this topic to expedite the flourishment of public data repository.

REFERENCES

[1] L. Chen, Y. Li, C. Huang, B. Li, Y. Xing, D. Tian, L. Li, Z. Hu, X. Na, Z. Li, S. Teng, C. Lv, J. Wang, D. Cao, N. Zheng, and F.-Y. Wang, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1046–1056, 2023.

[2] Y. Tian, X. Li, H. Zhang, C. Zhao, B. Li, X. Wang, and F.-Y. Wang, "Vistagpt: Generative parallel transformers for vehicles with intelligent systems for transport automation," *IEEE Trans. Intell. Veh.*, 2023.

[3] K. Liu, L. Chen, L. Li, H. Ren, and F.-Y. Wang, "Metamining: Mining in the metaverse," *IEEE Trans. Syst. Man Cybern. -Syst*, vol. 53, no. 6, pp. 3858–3867, 2023.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, pp. 3354–3361, 2012.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 573–580, 2012.

[6] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[7] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.

[8] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 642–657, 2019.

[9] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, 2019.

[10] Y. Li, Z. Li, S. Teng, Y. Zhang, Y. Zhou, Y. Zhu, D. Cao, B. Tian, Y. Ai, Z. Xuanyuan, *et al.*, "AutoMine: An unmanned mine dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Patt. Recognit.*, pp. 21308–21317, 2022.

[11] Y. Kang, H. Yin, and C. Berger, "Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 171–185, 2019.

[12] K. Takeyama, T. Machida, Y. Kojima, and N. Kubo, "Improvement of dead reckoning in urban areas through integration of low-cost multisensors," *IEEE Trans. Intell. Veh.*, vol. 2, no. 4, pp. 278–287, 2017.

[13] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 3946–3952, 2008.

[14] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[15] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 9572–9582, 2019.

[16] Z. Gong, J. Li, Z. Luo, C. Wen, C. Wang, and J. Zelek, "Mapping and semantic modeling of underground parking lots using a backpack LiDAR system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 734–746, 2021.

[17] B.-S. Cho, W.-s. Moon, W.-J. Seo, and K.-R. Baek, "A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding," *J. Mech. Sci. Technol.*, vol. 25, pp. 2907–2917, 2011.

[18] A. Chilian and H. Hirschmüller, "Stereo camera based navigation of mobile robots on rough terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4571–4576, 2009.

[19] D. Droeschel, M. Schwarz, and S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner," *Robot. Auton. Syst.*, vol. 88, pp. 104–115, 2017.

[20] J. Lin and F. Zhang, "R^{3}LIVE: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 10672–10678, 2022.

[21] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[22] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 965–972, 2018.

[23] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot. Sci. Syst.*, vol. 2, pp. 1–9, 2014.

[24] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 5692–5698, 2021.

[25] A. S. Huang, M. Antone, E. Olson, L. Fletcher, D. Moore, S. Teller, and J. Leonard, "A high-rate, heterogeneous data set from the DARPA urban challenge," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1595–1601, 2010.

[26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[27] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, pp. 834–849, 2014.

[28] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and LiDAR dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2016.

[29] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "KAIST multi-spectral day/night data set for autonomous and assisted driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 934–948, 2018.

[30] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1680–1687, IEEE, 2018.

[31] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017.

[32] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, 2016.

[33] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, 2017.

[34] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," *arXiv preprint arXiv:1607.02555*, 2016.

[35] R. Song, X. Li, X. Zhao, M. Liu, J. Zhou, and F.-Y. Wang, "Identifying critical test scenarios for lane keeping assistance system using analytic hierarchy process and hierarchical clustering," *IEEE Trans. Intell. Veh.*, 2023.

[36] X. Li, H. Duan, B. Liu, X. Wang, and F.-Y. Wang, "A novel framework to generate synthetic video for foreground detection in highway surveillance scenarios," *IEEE Trans. Intell. Transp. Syst.*, 2023.

[37] I. Ali, A. Durmush, O. Suominen, J. Yli-Hietanen, S. Peltonen, J. Collin, and A. Gotchev, "FinnForest dataset: A forest landscape for visual SLAM," *Robot. Auton. Syst.*, vol. 132, p. 103610, 2020.

[38] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "TartanAir: A dataset to push the limits of visual SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4909–4916, 2020.

[39] X. Li, K. Wang, X. Gu, F. Deng, and F.-Y. Wang, "Paralleleye pipeline: An effective method to synthesize images for improving the visual intelligence of intelligent vehicles," *IEEE Trans. Syst. Man Cybern. -Syst*, 2023.

[40] X. Li, Y. Wang, L. Yan, K. Wang, F. Deng, and F.-Y. Wang, "Paralleleye-cs: A new dataset of synthetic images for testing the visual intelligence of intelligent vehicles," *IEEE Trans. Veh. Tech.*, vol. 68, no. 10, pp. 9619–9631, 2019.

[41] Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Xu, F. Chen, B. Goossens, P. Z. Sun, H. Yu, C. Liu, *et al.*, "BotanicGarden: A high-quality dataset for robot navigation in unstructured natural environments," *IEEE Robot. Automat. Lett.*, 2024.

[42] L. Gao, Y. Liang, J. Yang, S. Wu, C. Wang, J. Chen, and L. Kneip, "Vector: A versatile event-centric benchmark for multi-sensor slam," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8217–8224, 2022.

[43] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2266–2273, 2021.

[44] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "Radiate: A radar dataset for automotive perception in bad weather," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 1–7, IEEE, 2021.

[45] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[46] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4758–4765, 2018.

[47] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, pp. 6243–6252, 2017.

[48] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4530–4537, 2019.

[49] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 994–1001, 2018.

[50] Z. Hong, Y. Petillot, and S. Wang, "RadarSLAM: Radar based large-scale SLAM in all weathers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 5164–5170, 2020.

[51] Y. Cheng, M. Jiang, J. Zhu, and Y. Liu, "Are we ready for unmanned surface vehicles in inland waterways? the USVInland multisensor dataset and benchmark," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3964–3970, 2021.

[52] X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F.-Y. Wang, "From features engineering to scenarios engineering for trustworthy ai: I&i, c&c, and v&v," *IEEE Intell. Syst.*, vol. 37, no. 4, pp. 18–26, 2022.

[53] Z. Zhu, Y. Chen, Z. Wu, C. Hou, Y. Shi, C. Li, P. Li, H. Zhao, and G. Zhou, "LATITUDE: Robotic global localization with truncated dynamic low-pass filter in city-scale NeRF," in *Proc. Int. Conf. Robot. Automat.*, pp. 8326–8332, 2023.

[54] J. Deng, X. Chen, S. Xia, Z. Sun, G. Liu, W. Yu, and L. Pei, "NeRF-LOAM: Neural implicit representation for large-scale incremental LiDAR odometry and mapping," *arXiv preprint arXiv:2303.10709*, 2023.

[55] X. Li and F.-Y. Wang, "Scenarios engineering: Enabling trustworthy and effective ai for autonomous vehicles," *IEEE Trans. Intell. Veh.*, 2023.

[56] K. Burnett, D. J. Yoon, Y. Wu, A. Z. Li, H. Zhang, S. Lu, J. Qian, W.-K. Tseng, A. Lambert, K. Y. Leung, *et al.*, "Boreas: A multi-season autonomous driving dataset," *Int. J. Robot. Res.*, vol. 42, no. 1-2, pp. 33–42, 2023.

[57] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3292–3310, 2022.

[58] X. Li, R. Song, J. Fan, M. Liu, and F.-Y. Wang, "Development and testing of advanced driver assistance systems through scenario-based system engineering," *IEEE Trans. Intell. Veh.*, 2023.

[59] L. Guo, C. Shan, T. Shi, X. Li, and F.-Y. Wang, "A vectorized representation model for trajectory prediction of intelligent vehicles in challenging scenarios," *IEEE Trans. Intell. Veh.*, 2023.

[60] T. Sun, M. Segu, J. Postels, Y. Wang, L. Van Gool, B. Schiele, F. Tombari, and F. Yu, "SHIFT: A synthetic driving dataset for continuous multi-task domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Patt. Recognit.*, pp. 21371–21382, 2022.

[61] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "UniSim: A neural closed-loop sensor simulator," in *Proc. IEEE/CVF Conf. Comput. Vis. Patt. Recognit.*, pp. 1389–1399, 2023.

[62] *Nvidia Drive Sim*. Accessed: Aug. 21, 2023 [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/simulation/.

[63] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3D perception," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2032–2039, 2018.

[64] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset," in *Proc. Int. Conf. Robot. Automat.*, pp. 6713–6719, 2019.

[65] J. Fan, Y. Ou, X. Li, C. Zhou, and Z. Hou, "Structured light vision based pipeline tracking and 3d reconstruction method for underwater vehicle," *IEEE Trans. Intell. Veh.*, 2023.

[66] J. Li, H. Duan, X. Li, and Y. Huang, "Dvl-aided in-motion coarse alignment for underwater vehicles with latitude uncertainty," *IEEE Trans. Veh. Tech.*, 2023.

[67] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1935–1942, 2015.

[68] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 141–148, 2015.

[69] L. Chen, L. Sun, T. Yang, L. Fan, K. Huang, and Z. Xuanyuan, "RGB-T SLAM: A flexible SLAM framework by combining appearance and thermal information," in *Proc. Int. Conf. Robot. Automat.*, pp. 5682–5687, 2017.

[70] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, *et al.*, "Event-based vision: A survey," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2020.

[71] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, 2021.

[72] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[73] T. Raj, F. Hanim Hashim, A. Baseri Huddin, M. F. Ibrahim, and A. Hussain, "A survey on LiDAR scanning mechanisms," *Electronics*, vol. 9, no. 5, p. 741, 2020.

[74] Y. Li and J. Ibanez-Guzman, "LiDAR for autonomous driving: The principles, challenges, and trends for automotive LiDAR and perception systems," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, 2020.

[75] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias, "Review of mobile mapping and surveying technologies," *Measurement*, vol. 46, no. 7, pp. 2127–2145, 2013.

[76] K. Burnett, Y. Wu, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot, "Are we ready for radar to replace LiDAR in all-weather mapping and localization?," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10328–10335, 2022.

[77] A. Kramer, K. Harlow, C. Williams, and C. Heckman, "ColoRadar: The direct 3D millimeter wave radar dataset," *Int. J. Robot. Res.*, vol. 41, no. 4, pp. 351–360, 2022.

[78] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[79] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[80] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7517–7524, 2021.

[81] J. Lv, J. Xu, K. Hu, Y. Liu, and X. Zuo, "Targetless calibration of LiDAR-IMU system based on continuous-time batch estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 9968–9975, 2020.

[82] G. Yan, Z. Liu, C. Wang, C. Shi, P. Wei, X. Cai, T. Ma, Z. Liu, Z. Zhong, Y. Liu, *et al.*, "OpenCalib: A multi-sensor calibration toolbox for autonomous driving," *Software Impacts*, vol. 14, p. 100393, 2022.

[83] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.

[84] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-LiDAR calibration: A targetless and structureless approach," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1556–1563, 2020.

[85] O. Bogdan, V. Eckstein, F. Rameau, and J.-C. Bazin, "DeepCalib: A deep learning approach for automatic intrinsic calibration of wide field-of-view cameras," in *Proc. Eur. Conf. Vis. Media Prod.*, pp. 1–10, 2018.

[86] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1110–1117, 2018.

[87] S. Wu, A. Hadachi, D. Vivet, and Y. Prabhakar, “This is the way: Sensors auto-calibration approach based on deep learning for self-driving cars,” *IEEE Sens. J.*, vol. 21, no. 24, pp. 27779–27788, 2021.

[88] W. Wang, S. Nobuhara, R. Nakamura, and K. Sakurada, “Soic: Semantic online initialization and calibration for LiDAR and camera,” *arXiv preprint arXiv:2003.04260*, 2020.

[89] Z. Liu, H. Tang, S. Zhu, and S. Han, “SemAlign: Annotation-free camera-LiDAR calibration with semantic alignment loss,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 8845–8851, 2021.

[90] C.-K. Chang, J. Zhao, and L. Itti, “DeepVP: Deep learning for vanishing point detection on 1 million street view images,” in *Proc. Int. Conf. Robot. Automat.*, pp. 4496–4503, 2018.

[91] Y. Almalioglu, M. Turan, M. R. U. Saputra, P. P. de Gusmão, A. Markham, and N. Trigoni, “SelfVIO: Self-supervised deep monocular visual–inertial odometry and depth estimation,” *Neural Netw.*, vol. 150, pp. 119–136, 2022.

[92] M. Manafifard, “A review on camera calibration in soccer videos,” *Multimed. Tools Appl.*, pp. 1–32, 2023.

[93] R. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” *IEEE J. Robot. Automat.*, vol. 3, no. 4, pp. 323–344, 1987.

[94] S. De Ma, “A self-calibration technique for active vision systems,” *IEEE Tran. Robot. Automat.*, vol. 12, no. 1, pp. 114–120, 1996.

[95] O. D. Faugeras, Q. T. Luong, and S. J. Maybank, “Camera self-calibration: Theory and experiments,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 321–334, 1992.

[96] B. Triggs, “Autocalibration and the absolute quadric,” in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, pp. 609–614, 1997.

[97] W. Li, T. Gee, H. Friedrich, and P. Delmas, “A practical comparison between Zhang’s and Tsai’s calibration approaches,” in *Proc. 29th Int. Conf. Image Vis. Comput.*, pp. 166–171, 2014.

[98] J. Kannala and S. S. Brandt, “A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, 2006.

[99] Y. Yang, P. Geneva, and G. Huang, “Multi-visual-inertial system: Analysis, calibration and estimation,” *arXiv preprint arXiv:2308.05303*, 2023.

[100] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, “Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes,” in *Proc. Int. Conf. Robot. Automat.*, pp. 4304–4311, 2016.

[101] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1280–1286, 2013.

[102] P. Furgale, T. D. Barfoot, and G. Sibley, “Continuous-time batch estimation using temporal basis functions,” in *Proc. Int. Conf. Robot. Automat.*, pp. 2088–2095, 2012.

[103] Y. Yang, P. Geneva, X. Zuo, and G. Huang, “Online self-calibration for visual-inertial navigation: Models, analysis, and degeneracy,” *IEEE Trans. Robot.*, 2023.

[104] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, “OpenVINS: A research platform for visual-inertial estimation,” in *Proc. Int. Conf. Robot. Automat.*, pp. 4666–4672, 2020.

[105] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, “SemanticFusion: Dense 3D semantic mapping with convolutional neural networks,” in *Proc. Int. Conf. Robot. Automat.*, pp. 4628–4635, 2017.

[106] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” in *Proc. Int. Conf. Robot. Automat.*, pp. 3936–3943, 2012.

[107] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, “An open approach to autonomous vehicles,” *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[108] N. Schneider, F. Piewak, C. Stiller, and U. Franke, “RegNet: Multimodal sensor registration using deep neural networks,” in *Proc. IEEE Intell. Vehicles Symp.*, pp. 1803–1810, 2017.

[109] K. Yuan, Z. Guo, and Z. J. Wang, “RGGNet: Tolerance aware LiDAR-camera online calibration with geometric deep learning and generative model,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6956–6963, 2020.

[110] L. Marty, P. Gerbaud, and F. Christophe, “From passive to active radar reflectors and beyond,” in *Proc. IEEE Conf. Antenna Meas. Appl.*, pp. 605–607, 2021.

[111] *Image-Engineering*. Accessed: Aug. 21, 2023 [Online]. Available: https://www.image-engineering.de/news/newsletters/1133-aligning-radar-and-visual-cameras/.

[112] E. Wise, Q. Cheng, and J. Kelly, “Spatiotemporal calibration of 3D mm-Wavelength Radar-camera pairs,” *arXiv preprint arXiv:2211.01871*, 2022.

[113] G. Yan, J. Pi, C. Wang, X. Cai, and Y. Li, “An extrinsic calibration method of a 3D-LiDAR and a pose sensor for autonomous driving,” *arXiv preprint arXiv:2209.07694*, 2022.

[114] *CloudCompare*. Accessed: Aug. 21, 2023 [Online]. Available: https://www.cloudcompare.org/.

[115] D. L. Mills, “Internet time synchronization: The network time protocol,” *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, 1991.

[116] A. Harrison and P. Newman, “TICSync: Knowing when things happened,” in *Proc. Int. Conf. Robot. Automat.*, pp. 356–363, 2011.

[117] *Network Time Protocol*. Accessed: Aug. 21, 2023 [Online]. Available: https://en.wikipedia.org/wiki/Network_Time_Protocol/.

[118] E. Olson, “A passive solution to the sensor synchronization problem,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1059–1064, 2010.

[119] C. S. V. Gutiérrez, L. U. S. Juan, I. Z. Ugarte, and V. M. Vilches, “Real-time Linux communications: An evaluation of the linux communication stack for real-time robotic applications,” *arXiv preprint arXiv:1808.10821*, 2018.

[120] F. Reghenzani, G. Massari, and W. Fornaciari, “The real-time Linux kernel: A survey on preempt_rt,” *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–36, 2019.

[121] C. S. V. Gutiérrez, L. U. S. Juan, I. Z. Ugarte, and V. M. Vilches, “Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications,” *arXiv preprint arXiv:1809.02595*, 2018.

[122] T. Qin and S. Shen, “Online temporal calibration for monocular visual-inertial systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 3662–3669, 2018.

[123] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “FAST-LIO2: Fast direct LiDAR-inertial odometry,” *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, 2022.

[124] M. Faizullin, A. Kornilova, A. Akhmetyanov, and G. Ferrer, “Twist-n-Sync: Software clock synchronization with microseconds accuracy using MEMS-Gyroscopes,” *Sensors*, vol. 21, no. 1, p. 68, 2020.

[125] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei, “Rawseeds ground truth collection systems for indoor self-localization and mapping,” *Auton. Robot.*, vol. 27, pp. 353–371, 2009.

[126] W. Lewandowski and E. Arias, “GNSS times and UTC,” *Metrologia*, vol. 48, no. 4, pp. S219–S224, 2011.

[127] M. Faizullin, A. Kornilova, and G. Ferrer, “Open-source lidar time synchronization system by mimicking GNSS-clock,” in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas. Control Commun.*, pp. 1–5, 2022.

[128] F. Girela-López, J. López-Jiménez, M. Jiménez-López, R. Rodríguez, E. Ros, and J. Díaz, “IEEE 1588 high accuracy default profile: Applications and challenges,” *IEEE Access*, vol. 8, pp. 45211–45220, 2020.

[129] *Livox*. Accessed: Aug. 21, 2023 [Online]. Available: https://www.livoxtech.com/avia/.

[130] *Ouster*. Accessed: Aug. 21, 2023 [Online]. Available: https://ouster.com/products/scanning-lidar/os2-sensor/.

[131] *Dalsa*. Accessed: Aug. 21, 2023 [Online]. Available: https://www.teledynedalsa.com/en/products/imaging/cameras/gige-cameras/.

[132] *Basler*. Accessed: Aug. 21, 2023 [Online]. Available: https://docs.baslerweb.com/basler-ace/.

[133] S. Liu, B. Yu, Y. Liu, K. Zhang, Y. Qiao, T. Y. Li, J. Tang, and Y. Zhu, “Brief industry paper: The matter of time — a general and efficient system for precise sensor synchronization in robotic computing,” in *Proc. IEEE RTAS Conf.*, pp. 413–416, 2021.

[134] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, “The Hilti SLAM challenge dataset,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7518–7525, 2022.

[135] S. Yang, Y. Song, M. Kaess, and S. Scherer, “Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1222–1229, 2016.

[136] R. A. Hewitt, E. Boukas, M. Azkarate, M. Pagnamenta, J. A. Marshall, A. Gasteratos, and G. Visentin, “The Katwijk beach planetary rover dataset,” *Int. J. Robot. Res.*, vol. 37, no. 1, pp. 3–12, 2018.

[137] J. G. Rogers, J. M. Gregory, J. Fink, and E. Stump, “Test your SLAM! the SubT-Tunnel dataset and metric for mappingg,” in *Proc. Int. Conf. Robot. Automat.*, pp. 955–961, 2020.

[138] Y. Fu, J. Zhang, L. Zhou, Y. Liu, M. Qin, H. Zhao, and W. Tao, “Passive binocular optical motion capture technology under complex illumination,” *J. Shanghai Jiaotong Univ., Sci.*, pp. 1–11, 2023.

[139] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. Int. Conf. Robot. Automat.*, pp. 3400–3407, 2011.

[140] *Marvelmind*. Accessed: Aug. 21, 2023 [Online]. Available: https://marvelmind.com/.

[141] S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother, and D. Kondermann, "When can we use KinectFusion for ground truth acquisition," in *Proc. Workshop Color-Depth Camera Fusion Robot.*, vol. 2, p. 3, 2012.

[142] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, pp. 3234–3243, 2016.

[143] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," *arXiv preprint arXiv:1809.00716*, 2018.

[144] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot. Learn.*, pp. 1–16, 2017.

[145] Z. Song, Z. He, X. Li, Q. Ma, R. Ming, Z. Mao, H. Pei, L. Peng, J. Hu, D. Yao, *et al.*, "Synthetic datasets for autonomous driving: A survey," *arXiv preprint arXiv:2304.12205*, 2023.

[146] *OpenGL*. Accessed: Aug. 21, 2023 [Online]. Available: https://www.opengl.org/.

[147] *Unreal Engine*. Accessed: Aug. 21, 2023 [Online]. Available: https://www.unrealengine.com/.

[148] *Pov-Ray*. Accessed: Aug. 21, 2023 [Online]. Available: http://www.povray.org/.

[149] *Unity*. Accessed: Aug. 21, 2023 [Online]. Available: https://unity.com/.

[150] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet RGB-D: Can 5M synthetic images beat generic ImageNet pre-training on indoor segmentation?," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2678–2687, 2017.

[151] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, *et al.*, "The Replica dataset: A digital Replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[152] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," in *High Performance Computing in Science and Engineering: Second International Conference*, pp. 19–49, Springer, 2016.

[153] *Marvelmind*. Accessed: Aug. 21, 2023 [Online]. Available: https://microsoft.github.io/AirSim/.

[154] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2X-Sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10914–10921, 2022.

[155] A. Kloukiniotis, A. Papandreou, C. Anagnostopoulos, A. Lalos, P. Kapsalas, D.-V. Nguyen, and K. Moustakas, "CarlaScenes: A synthetic dataset for odometry in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Patt. Recognit.*, pp. 4520–4528, 2022.

[156] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE Int. Conf. Intell. Transp. Syst.*, pp. 1–6, IEEE, 2020.

[157] *Gazebo*. Accessed: Aug. 21, 2023 [Online]. Available: https://gazebosim.org/.

[158] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2213–2222, 2017.

[159] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. Int. Conf. Robot. Automat.*, pp. 1524–1531, 2014.

[160] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The blackbird UAV dataset," *Int. J. Robot. Res.*, vol. 39, no. 10-11, pp. 1346–1364, 2020.

[161] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *Proc. Int. Conf. Robot. Automat.*, pp. 5724–5731, 2013.

[162] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *2019 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 6941–6948, IEEE, 2019.

[163] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, pp. 4340–4349, 2016.

[164] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," *arXiv preprint arXiv:2001.10773*, 2020.

[165] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, *et al.*, "Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI," *arXiv preprint arXiv:2109.08238*, 2021.

[166] C. Cao, H. Zhu, F. Yang, Y. Xia, H. Choset, J. Oh, and J. Zhang, "Autonomous exploration development environment and the planning algorithms," in *Proc. Int. Conf. Robot. Automat.*, pp. 8921–8928, 2022.

[167] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[168] Z. Li, L. Li, and J. Zhu, "READ: Large-scale neural scene rendering for autonomous driving," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, pp. 1522–1529, 2023.

[169] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, *et al.*, "MARS: An instance-aware, modular and realistic simulator for autonomous driving," *arXiv preprint arXiv:2307.15058*, 2023.

[170] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, vol. 1, pp. 519–528, 2006.

[171] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, 1994.

[172] *EVO*. Accessed: Aug. 21, 2023 [Online]. Available: https://github.com/MichaelGrupp/evo/.

[173] J. Jeong, Y. Cho, and A. Kim, "The road is enough! extrinsic calibration of non-overlapping stereo camera and lidar using road information," *IEEE Robot. Automat. Lett.*, vol. 4, no. 3, pp. 2831–2838, 2019.

[174] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, *et al.*, "Are we ready for service robots? the openloris-scene datasets for lifelong slam," in *2020 IEEE Int. Conf. Robot. Automat.*, pp. 3139–3145, IEEE, 2020.

[175] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld LiDAR, inertial and vision with ground truth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4353–4360, 2020.

Yuanzhi Liu received the B.S. degree in Department of Measurement and Control from Harbin Institute of Technology, China, in 2017. Now he is pursuing the Ph.D. degree in Electronic Information with Shanghai Jiao Tong University. He is also a guest researcher in image processing and interpolation (IPI) group at Ghent University.

His research interests fall mainly on Autonomous Robots – including SLAM, mobile mapping, high-precision navigation, sensor calibration and synchronization, and construction of standard datasets.

Yujia Fu is currently a research engineer at Alibaba, Hangzhou, China. He received the B.S. degree in Dalian University of Technology in 2019 and the Master's degree in Shanghai Jiao Tong University in 2023. His research interests include robotics, cross-modal & metric learning, and autonomous driving. He is also a research assistant in Shanghai Jiao Tong University since 2024.

Minghui Qin is currently a research engineer at Horizon Robotics, Beijing, China. He received the B.S. degree and the Master's degree in the Department of Instrument Science and Engineering from Shanghai Jiao Tong University in 2020 and 2023. His research interests include robotics, computer vision, and autonomous driving.

Yufeng Xu received the BS. degree in Nankai University in 2021. From September 2021 to now, he is pursuing his Master's degree with School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include multi-sensor fusion positioning and autonomous driving.

Bin Cui received his B.E. degree in measurement and control technology and instruments from Tianjin University, Tianjin, China in 2016 and his M.E. degree in instrumentation science and technology from the China Academy of Space Technology, Beijing, China in 2019. He is currently working toward his Ph.D. degree in instrumentation science and technology with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.

Kunhua Liu received her Ph.D. degree from Shandong University of Science and Technology. She was a post-doc at School of Computer Science and Engineering, Sun Yat-sen University. Currently, she is an associate professor at the School of Mechanical and Automotive Engineering, Qingdao University of Technology. Her interests include autonomous driving, computing vision, and robotics.

Fengdong Chen received the Ph.D. degree in the Department of Computer Science and Engineering from Harbin Institute of Technology, China, in 2009. Since 2000, he served as an associate professor of the Department of Instrument Science and Engineering, Harbin Institute of Technology. His research interests include computer vision and precision instruments. He is the author of 20 articles and more than 10 inventions. He has been awarded the Invention Award from the government of the Ministry of Defence in 2016.

Wei Tao (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in Instrument science and technology from Harbin Institute of Technology, Heilongjiang Province, China, in 1997,1999 and 2003. From 2003 to 2018, she was a Research Assistant and Associate Professor with Shanghai Jiao Tong University. She became a Professor with Shanghai Jiao Tong University in 2018. She is the author of three books, more than 100 articles, and more than 40 inventions. Her research interests include opto-electronic measurement technology and application, methods and algorithms in the vision measurement process, and laser sensors and measurement instruments.

Dr. Tao became an IEEE member of Instrument in 2004 and she is now also an OSA member. She was awarded the Invention Award from the government of Shanghai and the China Instrument and Control Society in 2007 and 2009.

Michiel Vlaminck received his PhD degree in Computer Science Engineering from Ghent University in 2020. Since then, he is working as a postdoctoral researcher at the UAV Research Centre of Ghent University. He is currently working on the topic of 3D scene reconstruction using active depth sensors. His research focuses on applications in the domains of augmented reality, autonomous robots and UAV.

Bart Goossens (Member, IEEE) received the M.S. degree in computer science and the Ph.D. degree in engineering from Ghent University, in 2006 and 2010, respectively.

He is currently a Professor of digital image processing with the Image Processing and Interpretation Research Group, Department of Telecommunications and Information Processing. His research interests include medical image reconstruction (CT and magnetic resonance image), noise modeling and estimation, and medical image quality assessment. He currently serves as an Associate Editor for IEEE Transactions on Image Processing.

Poly Z.H. Sun (Member, IEEE) is currently a Research Assistant with the Department of Industrial Engineering, Shanghai Jiao Tong University, Shanghai, China.

He has authored/co-authored over 40 research papers (20+ IEEE Transactions papers) in top-tier refereed international journals and conferences , such as IEEE Transactions on Engineering Management, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Intelligent Vehicles, IEEE Transactions on Computational Social Systems.

His current research interests include human-machine interaction, intelligent transportation systems, human factors in driving, neuroergonomics, brain and cognitive status under maneuvers, physiological parameter measurement, bio-inspired vision mechanism, operations research, and their applications in human-in-the-loop transportation systems. He also has been serving as a reviewer for several top-tier international journals, and a session chair for several conferences such as *2022 25th IEEE International Conference on Intelligent Transportation Systems*, *2021 IEEE International Conference on Industrial Engineering and Engineering Management*.

Hui Zhao (Member, IEEE) received the Ph.D. degree in the Department of Instrument Engineering from Harbin Institute of Technology, China, in 1996. From 1989 to 2000, he served as an Associate Professor of the Precision Instrument Laboratory, Harbin Institute of Technology. Since 2000, he has been a Professor with the Department of Instrument Science and Engineering, Shanghai Jiao Tong University. His research interests include novel sensors and vision measurement methods. He is the author of three books, more than 150 articles, and more than 50 inventions.

Dr. Zhao has been a Member of SPIE since 2002. He became an IEEE Member in 2012. He serves as the Vice Chair of the Precision Mechanism Federation of China Instrument and Control Society, and the Vice Chair of the Mechanical Quantity Measurement Instrument Federation of China Instrument and Control Society. He is a reviewer of IEEE Transactions on Instrument & Measurement, *Sensors*, *Measurement*, *IEEE Sensors Journal*, *Optik*, and several other journals.