IndoorMCD: A Benchmark for Low-Cost Multi-Camera SLAM in Indoor Environments

Marco Sewtz[1], Yunis Fanger[1], Xiaozhou Luo[1], Tim Bodenmüller[1] and Rudolph Triebel[1,2]

***Abstract*—Navigating mobile robots within home environments is essential for future applications, e.g. in household or within the field of elderly care. Therefore, these systems, equipped with multiple sensors, have to deal with changing environments.**

This work presents the IndoorMCD dataset that allows for benchmarking SLAM algorithms within static and changing indoor environments of various difficulties. The dataset provides synchronized and calibrated RGB-D images from a low-cost multi-camera setup, as well as additional IMU data. Further, highly accurate ground truth movement data is provided. It is the first dataset that provides static and changing environments for a multi-camera setup. Evaluations with state-of-the-art SLAM algorithms show the dataset's applicability and also present limitations of current approaches. The dataset is made available in a structured format and a utility library with example scripts is provided.

***Index Terms*—Data Sets for SLAM, Visual-Inertial SLAM, Localization, Mapping, RGB-D, Multi-Camera**

I. Introduction

ROBOTIC assistance in home environments is an emerging field of research, opening up new opportunities and applications for autonomous systems. Symbiotic human-robot collaboration and interaction are essential for the success of those ambitions. Thus, robotic systems need to operate, especially navigate, in changing environments reliably. A central element for global navigation is Simultaneous Localization and Mapping (SLAM), as it continuously updates the environmental knowledge of the robot. Although the robustness of state-of-the-art applications is progressively enhanced with each subsequent generation, most of them still rely on a single sensor. A failure of the system likely results in the total loss of localization. However, modern commercial off-the-shelf (COTS) sensors, like RGB-D cameras, are cheap, small and only require little energy. Thus, adding multiple sensors becomes feasible and increases robustness by redundancy.

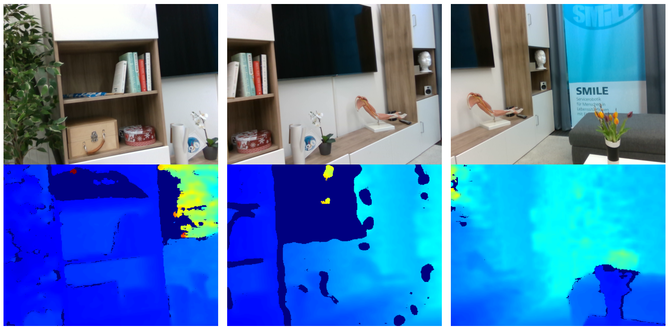


Fig. 1: Multi-camera view of a living room environment captured by commercial off-the-shelf RGB-D sensors.

By using COTS hardware, redundancy can be added while limiting the increase in cost. However, this often results in degraded sensor measurements that integrated software has to consider. For the future development of frameworks to solve the problems mentioned above, a common dataset that includes representative scenarios is crucial. While several datasets have been released as benchmarks for SLAM or other navigation systems, most of them concentrate on the single-sensor case, the use of expensive sensors like LIDAR, or are meant for evaluation in autonomous outdoor-vehicle development.

In this work, we present a dataset that aims to enable research on SLAM systems that address both robustness and redundancy using COTS sensors. It contains five different scenarios, each consisting of several runs of increasing complexity. The recordings include highly accurate ground truth estimation measured by a high-speed motion capture system (MCS). Furthermore, we show the applicability of our data by evaluating the trajectories with state-of-the-art Visual-Inertial Navigation System (VINS) and SLAM systems, as well as an in-house development for multi-camera SLAM [1]. Finally, we also provide a utility library for easy access to the data.

https://rmc.dlr.de/rm/en/staff/marco.sewtz/benchmark

We summarize our contribution as following:

- The IndoorMCD dataset containing 105 individual sequences recorded in indoor environments using multiple COTS sensors, consisting of RGB-D and Inertial Measurement Unit (IMU) modules.
- An additional high accuracy ground truth reference.
- Various scenarios with increasing complexity in their trajectories including loops, motion blur and changes in the environment.
- An extensive evaluation of renowned approaches including performance benchmarks to demonstrate our data's applicability.

Manuscript received: August, 25, 2022; Revised November, 22, 2022; Accepted December, 28, 2022.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) by means of the project SMiLE2gether (LABAY102).

[1] Marco Sewtz, Yunis Fanger, Xiaozhou Luo, Tim Bodenmüller and Rudolph Triebel are with the Institute of Robotics and Mechatronics, German Aerospace Center, Wessling, Germany {firstname.lastname}@dlr.de

[2] Rudolph Triebel is with the Technical University of Munich (TUM), Munich, Germany

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

The proposed dataset is, to our knowledge, the first dataset combining multiple sensors and high accurate ground truth for static and changing indoor environments.

II. Related Work

Along with the rising potential of vision-based algorithms, datasets containing realistic environmental conditions have been proposed to provide a reference for new approaches and a baseline for performance benchmarks of existing developments. While the number of available datasets is growing continuously, we provide an overview of the most relevant datasets including visual and inertial data in Table I.

The TUM RGB-D dataset [2] provides a collection of synchronized color and depth data in an indoor scenario recorded in an office environment and an industrial hall. Supplemented by a ground truth reference recorded by a highly accurate MCS, it is one of the most extensively used and established benchmarks for RGB-D Visual Odometry (VO) and SLAM algorithms. Furthermore, the 7-Scenes dataset [3] focuses on realistic indoor-scenes captured by a RGB-D camera and generated ground-truth pose information. Around the same period, the KITTI benchmark suite [4] was proposed for research on vision-based navigation in autonomous driving. In addition to gray-scale mono and RGB stereo sequences, it also includes IMU information. However, the low-frequency inertial data is not synchronized with the visual information, which is mandatory for a well-designed visual-inertial (VI) benchmark. Nevertheless, KITTI has established itself well in the research community and serves as a foundation for further modifications and developments, e.g., object scene flow research [5].

Over time, the focus in the research community has shifted towards the fusion of information provided by different kinds of sensors. Most prominently, many recent datasets are designed to evaluate VO and SLAM applications by including time-synchronized high-frequency IMU measurements. The EuRoC MAV [6] and the more challenging UZH-FPV dataset [7] were recorded with a Micro Aerial Vehicle (MAV). In contrast, one can rely on benchmarks such as TUM VI [8] and OpenLORIS [9] in the case of ground-based carriers. These last four datasets are also equipped with sophisticated ground truth references, which are provided, at least partially, by MCS with an accuracy of approximately $1\mathrm{mm}$.

While the previously presented datasets only include one main viewing direction, the Field-of-View (FoV) size can be significantly expanded by deploying multiple sensing devices with differing orientations. However, most representatives of datasets that employ this approach, such as the NCLT [10] and PennCOSYVIO dataset [11], do neither include high-precision ground truth information nor a hardwired time-synchronization between IMU and the relevant sensors. Therefore, they cannot be considered as an evaluation reference for performance benchmarks between individual VO and SLAM approaches. Currently, the only dataset in the VI domain containing multiple viewing directions that fulfills the requirements for a benchmark is, besides our proposal, the M2DGR dataset [12].

Although the latter benchmark contains a sizable collection of information from different sensor types, data containing multiple viewing orientations are only available in RGB format. This is due to the original design purpose of those sensors, which has the target of achieving an omnidirectional coverage of the related sceneries. Lastly, we also want to mention RIO10 [13], an indoor visual dataset dedicated to changes in the environment – in specific different lightning conditions, object pose changes and appearance. To the best of our knowledge, there is no dataset available that contains multiple visual sensing modalities exceeding the information provided by RGB cameras and operating in dynamic indoor scenes. By supplementing multiple RGB sources with the respectively associated depth information on top of acceleration and angular data, our target is to foster research of multi-camera VO and SLAM approaches in the VI domain.

During the research process, we discovered a significant deficit of datasets for benchmarking the behavior of localization and mapping algorithms in the case of world-model alternation between static and dynamic changing objects within comparable environmental settings. While many conventional VINS and SLAM applications are based on the assumption of a static world, robust approaches must be able to deal with dynamic elements within this world. With the exception of OpenLORIS, all other datasets in Table I are recorded either in a static environment or a dynamic setting with moving objects. Although the benchmark includes static sequences and ones with dynamic moving objects by design, the world-model assumption does not change within individual scenes. Therefore, the performance differences between static and dynamic world assumptions cannot be evaluated in particular since no performance baseline can be provided for the world model within a specific scene.

Hence, we intended to establish our dataset as a benchmark for applications in home environments by providing realistic environmental conditions considering an urban housing scenario based on COTS hardware. In contrast to other established datasets, which are primarily recorded on industrial-grade and customer furnished hardware, the utilization of state-of-the-art COTS sensors allows for a rare peek into the ordinary application-related domain instead of the predominant, more or less idealized, scientific domain. Hence, algorithms have to demonstrate their practicability in real-world situations with imperfect data (e.g. motion blur) and changing environments (e.g. moved chair). However, we neglected temporary dynamic elements in our datasets, e.g. a human walking through the room, as they are more focused on permanent changes and not temporal disturbances.

In terms of emulating the kinematic behavior of typical applications, our dataset is recorded by two different carrier platforms representing either a ground-based robot or a handheld device. The latter assembly provides a total of six unlimited degrees of freedom (DoF) in contrast to the ground-based platforms utilized in benchmarks of similar quality, from which at least 3 DoF are fairly restricted in their magnitude of variability.

III. Hardware Setup

Our hardware setup for recording the dataset consists of three RGB-D Intel RealSense D435i (denoted as *left*, *front*,

TABLE I: Overview of most common datasets for visual and inertial SLAM in changing indoor environments.

Dataset	Environ.	Platform	Cameras	IMU	Scene mode	Ground truth	Accuracy
NCLT [10]	In-/outdoors	Segway	6 RGB 1600×1200 @ 5Hz	1 3DM-GX3-45 3-axis acc./gyro @ 100Hz	Dynamic	Fused GNSS/ IMU/Laser pose @ 150Hz	$\leq$ 10cm
EuRoC MAV [6]	Indoors	MAV	1 stereo gray-scale 2 × 752×480 @ 20Hz	1 ADIS16488 3-axis acc./gyro @ 200Hz	Static	Laser tracker pose @ 20Hz, **MCS @ 100Hz**	**$\leq$ 1mm (MCS)**
PennCOSYVIO [11]	In-/outdoors	Handheld	4 RGB (rolling shutter) 1920×1080 @ 30Hz, 1 stereo gray-scale 2 × 752×480 @ 20Hz, 1 fisheye gray-scale 640×480 @ 30Hz	1 ADIS16488 3-axis acc./gyro @ 200Hz, 2 Tango 3-axis acc. @ 128Hz 3-axis gyro @ 100Hz	Dynamic	Fiducial markers pose @ 30Hz	$\leq$ 15cm
TUM VI [8]	In-/outdoors	Handheld	1 stereo gray-scale 2 × 1024×1024 @ 20Hz	1 BMI160 3-axis acc./gyro @ 200Hz	Static	Partial MCS pose @ 120Hz	**$\leq$ 1mm**
UZH-FPV [7]	In/-outdoors	MAV	1 stereo gray-scale 2 × 640×480 @ 30Hz 1 event camera 346×260 @ 50Hz + events	1 MPU-9250 3-axis acc./gyro/ magn. @ 500Hz, 1 3-axis acc./gyro @ 1000Hz	Static	Laser tracker pose @ 20Hz	**$\leq$ 1mm**
OpenLORIS [9]	Indoors	Ground robot	1 RGB-D (rolling shutter) 848×480 @ 30Hz, 1 stereo fisheye RGB 2 × 848×480 @ 30Hz	2 BMI055 3-axis acc. @ 250Hz 3-axis gyro @ 400Hz	**Static or Dynamic**	Laser tracker pose @ 40Hz, **MCS pose @ 240Hz**	$\leq$ 3cm (Laser), **$\leq$ 1mm (MCS)**
M2DGR [12]	In-/outdoors	Ground robot	6 fish-eye RGB 1280×1024 @ 15Hz, 1 infrared camera 640×512 @ 25Hz, 1 event camera 640×480 @ 15Hz + events, 1 RGB-D (rolling shutter) 640×480 @ 15Hz	1 Handsfree A9 3-axis acc./gyro/ magn. @ 150Hz, 1 BMI055 3-axis acc./gyro @ 200Hz	Dynamic	GNSS pose @ 100Hz, **Laser tracker pose @ 100Hz**, **MCS pose @ 50Hz**	$\leq$ 2cm (GNSS), **$\leq$ 1mm (Laser, MCS)**
7-Scenes [3]	Indoors	Handheld	1 RGB-D 640×480 @ 30Hz	None	Static	Visual Pose Tracking[2]	$\leq$ 2cm
RIO10 [13]	Indoors	Handheld, synthetic	1 RGB 540×960[1], 1 synthetic depth 540×960[1]	None	**Dynamic**	Visual Pose Tracking[2]	$\leq$ 10cm
IndoorMCD (Ours)	Indoors	**Handheld, ground robot**	3 RGB-D (rolling shutter) 640×480 @ 15Hz	3 BMI055 3-axis acc. @ 250Hz 3-axis gyro @ 400Hz	**Static and Dynamic**	**MCS pose @ 100Hz**	**$\leq$ 1mm**

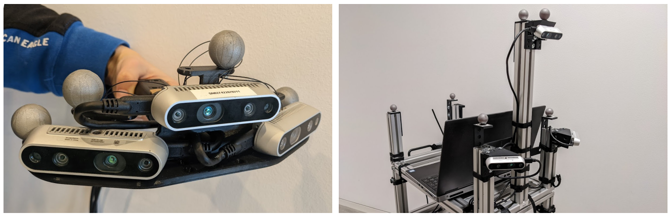
[1] Frame-rate unknown for this dataset. [2] Ground-truth accuracy is unknown and information is based on error metric.

right) in two different configurations.

The first one is a handheld camera device (HCD) which offers 6 DoF and can be easily moved around in the scene. The second one is a robotic platform mock-up called Marvin, which simulates the movement of wheel-based systems. Both are displayed in Figure 2.

A. Sensor Carriers

1) HCD: This device, as depicted in Figure 2a, integrates all sensors in a compact configuration. The small form-factor allows simple and uncomplicated use by the operator and enables mobile manipulation. While the center camera has an overlapping FoV with both outward-facing cameras, the sensors *left* and *right* do not share a common view. Hence the configuration can be used in algorithms that require visual overlap as well as systems that merely need a known rigid transform. Further, this platform offers a hardware synchronization of all camera modules.



(a) The handheld camera device used for capturing motion with six degrees of freedom.

(b) The robotic mock-up platform Marvin used for simulating motion of wheel-based systems.

Fig. 2: The used hardware devices for this dataset.

2) Marvin: The used mock-up, as seen in Figure 2b, simulates the movement of a wheel-based robotic system. This reduces the motion to only 3 DoF, in particular x, y and θ. The design is intended to mimic the view of sensors equipped on real assistant systems like Rollin' Justin [14] or the motorized wheelchair EDAN [15]. Due to this fact, the sensors may be blocked by obstacles when closely approaching objects. Furthermore, the configuration of the outer cameras is comparable to the integration in the HCD. However, the center camera is tilted down and raised to offer an improved view of desktops or tables.

B. Sensors

The Intel RealSense D435i consists of a RGB camera, two infrared cameras for depth estimation and an Inertial Measurement Unit.

The image processing of the two infrared cameras is performed internally, and the resulting depth image is pixel-aligned to the color image. Furthermore, a pattern projector operating in the infrared range is integrated to enhance the depth estimation even in textureless environments. The cameras are operated 15Hz with a resolution of 640×480 pixels.

The Inertial Measurement Unit has a triaxial 12-bit linear acceleration and a triaxial 16-bit angular velocity module. The accelerometer is operated at 250Hz and the gyroscope at 400Hz. In our dataset, we provide the single data streams and a fused stream that interpolates the acceleration readings between the gyroscope measurements.

The carriers are equipped with a trigger synchronization circuit. The *front* camera is used as trigger commander and the *left* and *right* cameras are configured as receivers. Although this introduces a slight delay on the trigger for the receiving devices, our results with existing algorithms showed that this offset is negligible in practice.

C. Ground Truth

For all except the real indoor scenario we obtained a highly accurate ground truth estimation using a Vicon MX T40 motion capture tracking system. The recording devices are equipped with several reflective markers, which can be monitored by six infrared cameras hanging from the ceiling. The alignment configuration of the tracking system is individually adapted for each scene to obtain the best and at-all-time continuous estimation of the current pose. The system operates at 100Hz.

The Vicon cameras emit infrared light at the same wavelength as the RealSense pattern projector. However, as the pattern is projected statically and only small dots are visible, we did not measure any interference of the pattern with the tracking system.

IV. Calibration

A. Cameras

The pinhole camera model is used to calibrate the intrinsic parameters of the sensors, which can be obtained using different views of a checkerboard target for each sensor [16]. These parameters consist of the focal-lengths f_x and f_y, the principal point (c_x, c_y) and the skew k_{skew}. The depth image is aligned to the color image on the hardware side of the RealSense devices result in a pixel-to-pixel correspondence in the images. In addition, the Brown-Conrady [17] model can be applied to remove distortion from the color image.

We provide the parameters of the pinhole as well as the Brown-Conrady model in our dataset.

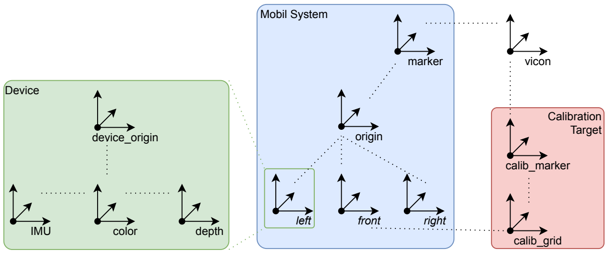


Fig. 3: Illustration of different frames in the dataset including markers and calibration utilities. All individual device origins are calibrated to the overall system's origin.

B. IMUs

The calibration procedure for IMU model estimation is defined by Intel for the RealSense devices [18]. Therefore, each device is orientated in six directions. Several thousand samples are acquired for each, and the parameters are finally optimized over the available set of data. The accelerometer parameters consist of the scale factor $\vec{s} = [s_x, s_y, s_z]^T$, the bias $\vec{b} = [b_x, b_y, b_z]^T$ and the axis alignment $c_{xy}, c_{yx}, c_{xz}, c_{zx}, c_{yz}, c_{zy}$. The intrinsics for the gyroscope include the bias values $\vec{\omega} = [\omega_x, \omega_y, \omega_z]^T$.

C. Extrinsics

The handling of extrinsic calibrations is organized on two levels. At first, all sensors of one RealSense device are handled on the device level, where the color sensor is set as the origin of each device. Therefore, the IMU is calibrated with respect to this sensor. As the depth stream provides a pixel-to-pixel alignment, the resulting displacement is zero.

On the system level, each device is also calibrated using the color sensor. Here, we make use of the fact that the *front* camera overlaps with both the *left* and the *right* camera. Multiple images of a checkerboard calibration target with distinctive origin are captured for estimating the relative pose transform from the *front* camera to the respective target camera. For each image, the correspondences between the checkerboard corners on the calibration target and the projected pixel coordinates are mapped and the transform is estimated by minimizing the reprojection error using Levenberg-Marquardt optimization [19].

For calibrating the Vicon system to the origin of the overall system, the same calibration target as before is used. In addition, several reflective markers are placed on the checkerboard and registered manually to its origin. Afterward, the transform of the *front* camera to the checkerboard and the transform of the markers in the tracking system is estimated and used for aligning the tracking markers to the system origin.

All frames and transforms are illustrated in Figure 3.

D. Time Domains

Within the dataset, different time domains are present as depicted in Figure 4. Each device has its own clock source, which is used for timestamps on the sensor measurements of each device. The timestamping of IMU readings is $\pm50\mu\text{s}$, which leads to tolerance of roughly 2% when operating the

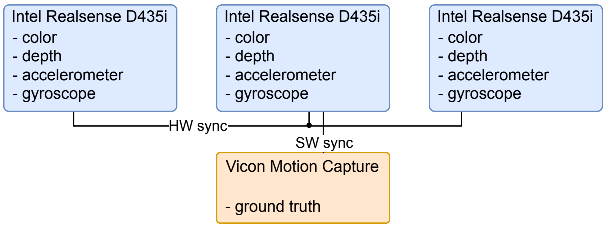


Fig. 4: Overview of the time domains in this dataset. Each RealSense has its own clock and the sensors are triggered device-central. In case the hardware synchronization is present, the trigger signals of the images are synced. The Vicon system is software synchronized.

gyroscope at $400\mathrm{Hz}$. Therefore, the temporal offset of IMU readings and image capturing on a specific RealSense device can be neglected. Image acquisition is hardware triggered, and the color and depth streams are temporal synced.

In scenarios where the hardware synchronization between the devices is available, the trigger of the image sensors is derived from the commanding camera. In all cases, this is the *front* camera. However, the clocks will not be synced, leading to different timestamps on the images. Exploiting the fact that the images are triggered simultaneously and that the offset between the trigger points is negligible, the clock offset can be estimated by the offset of the color images.

In the non-synced scenarios, the synchronization of the time domains between the devices is not possible without evaluating the trajectory.

The remaining time domain is the Vicon tracking system for ground truth estimation. Thereby, a calibration target is positioned in the view of the *front* camera and tracked by the Vicon system. The target is then slowly moved in the view of the camera. Afterward, the motion is estimated, and the temporal offset is determined by minimizing the absolute pose error (APE). This approach is based on the proposed calibration of Sturm et al. and we refer to their publication [2] for in-depth explanation.

E. Ground Truth

Accurate and continuous information of the actual pose is crucial for investigating the performance of navigation algorithms. Therefore close attention is paid to calibrating the Vicon system before every scenario recording.

The procedure is provided by the manufacturer. It involves operating a calibration stick which is moved in the area of operation and extensively observed by the cameras to create correspondences between individual views. Once enough samples are received, the system calibrates itself by performing optimization for low reprojection error.

V. Dataset

A. Calibration Sequences

These sequences contain the calibration runs used in Section IV. They contain the raw data without any further processing.

TABLE II: Overview of each scenario's (S) specific properties and number of runs (R), as well as if hardware sync has been enabled and if ground truth is available. Scenarios 0-4 have been captured in created environments in our lab, the last one is recorded in an actual apartment.

S	#R	Environment	Device	Sync	GT
0	19	kitchen, office, living-room	HCD	✓	✓
1	28	kitchen, office, living-room	HCD		✓
2	20	2 rooms: kitchen, living-room	HCD	✓	✓
3	15	2 office desktops	HCD	✓	✓
4	15	kitchen, office, living-room	Marvin		✓
5	10	actual apartment	HCD	✓	

B. Recorded Scenarios

Several scenarios have been recorded in varying setups. Three different environments are created in our labs, including a kitchen, an office area and a living room, which provide a broad set of visual inputs for algorithms. Temporary walls and a door are used to create different room layouts between the scenarios with a total available area of $6.50\mathrm{m} \times 4.50\mathrm{m}$. An exemplary subset of views is shown in Figure 5.

The kitchen consists of a counter including an oven, a fridge, several electronic appliances and commonplace items like apples, cucumbers, or a scale. Most of the structures are static and do not offer a lot of textures. The office area contains depending on the scenario either one or two desktops, including computer monitors, keyboards and a office chair. Further commodities like pens, scissors, or markers are added, which frequently change their position. The living room offers a sofa, including a coffee table, multiple plants and a television shelf. Furniture, as well as the appearance of objects, change over time to simulate human presence. Finally, we also provide a scenario captured in an actual apartment's living room. This room offers a sofa, a television, a fish tank, multiple bookshelves, plants and other common furniture objects. While this scenario does not offer a ground truth, we included it as a proof-of-concept whether proposed systems perform in real environments. An overview is provided in Table II. For measuring the impact of synchronization, scenario 0 and 1 are recorded with and without hardware synchronization in the same environment.

We took care that each run within a specific scenario increments the complexity of the trajectory. At first, they only contain a small number of rotations and single translations. The static-world assumption, meaning no dynamics in the perceived data, is held true. With progressing runs, the trajectories increase in length and amount of movement and ultimately include loops and revisits of previously explored areas. Final runs add changes in the environment that can be observed when places are viewed multiple times. The changes can be seen in Figure 6.

C. Utilities

Additionally to the datasets, we also provide a library for reading the data. It is able to parse the dataset and load the sensor measurements on-demand into the computer memory with a low footprint. Meta-information like extrinsic and

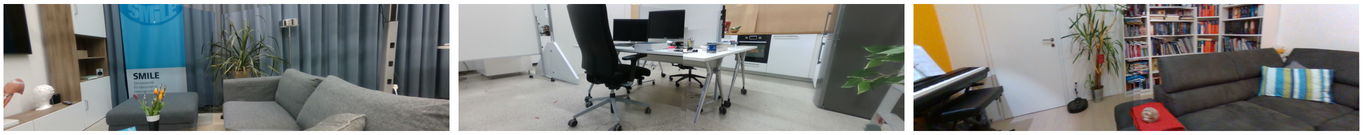


Fig. 5: Stitched panoramic images of views in the dataset. The image on the left-hand side shows the living room as seen in scenarios 0, 1 and 4, in the middle scenario 3 in the office, and on the right-hand side the actual apartment.

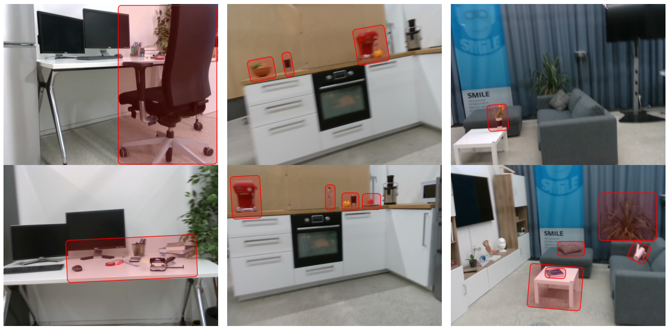


Fig. 6: The dataset captured several changes in the environment during each run. Objects like chairs, the table or the coffee machine are moved around in the scene, smaller objects like books are moved or completely removed, plants have a different appearance over the course of time.

TABLE III: Properties of SLAM systems used for evaluation.

SLAM-system	Type	RGB	IMU	Depth
VINS-Mono	feature-based	✓	✓	
ORBSLAM2	feature-based	✓		✓
ORBSLAM3	feature-based	✓	✓	✓
MROSLAM	feature-based	✓		✓
DSO	direct	✓		

intrinsic calibration and online interpolation of data points are also available. This shall ease access to our data. Furthermore, we provide sample scripts to generate *bag* files to be used within the Robot Operating System (ROS).

VI. Evaluation

To assess the suitability of this dataset for benchmarking, we evaluate it with state-of-the-art SLAM systems. As examples for feature-based methods, we deploy VINS-Mono [20], ORBSLAM2 [21], ORBSLAM3 [22] and our in-house developed multi-camera approach MROSLAM [1]. Hereby, VINS-Mono processes both IMU and camera data while ORBSLAM2 and MROSLAM purely rely on RGB-D information. ORBSLAM3 incorporates color, depth and inertial data. As a representative of direct visual SLAM methods, we also deploy DSO [23] on the dataset. In this case, it requires only monocular RGB camera images as input. The general properties of the deployed SLAM algorithms are summarized in Table III.

All SLAM applications are configured using the calibration information provided within the dataset (see Section IV) but use the respective systems default parameters otherwise. For each run, three separate instances of these applications are deployed simultaneously to process the data provided by each of the devices. In order to evaluate the dataset's applicability, we assessed our selection of renowned algorithms both in a quantitative and qualitative scope. For the first one, we recorded how many of the devices reach the end of a run without losing tracking at any point or outright failing. The results are presented in Table IV, where scenarios 1-3 were recorded using the HCD and scenario 4 using Marvin. Since scenario 5 does not include ground truth trajectories, we do not consider it here. Therein, only devices where the respective SLAM instance ran for at least 90% of the ground truth trajectory's duration without losing tracking are declared as successful.

At a closer look, it is noteworthy that the multi-camera approach achieved the best results among the purely vision-based approaches. By utilizing information from all devices with different orientations at the same time, a robust construct with multiple redundancies is established, which results in the reduction of potential loss-of-tracking. Especially in comparison to ORBSLAM2, on which MROSLAM is primarily based, the rate of total failure is reduced by a factor of 2.5 in scenario 0 or 2.0 in total. Nevertheless, VINS-Mono already provides a very robust approach which only failed in situations where the sensor's view was blocked and the LoT was not resolvable. Lastly mentioning ORBSLAM3, the performance on tracking seems to be less stable compared to ORBSLAM2. We assume that the extensions focus primarily on the accuracy of the trajectory estimation (as shown later) accepting small deficits in the robustness.

Furthermore, a qualitative assessment is performed using the evo evaluation package [24]. It allows to align the pose estimates of the SLAM systems with ground truth information and the computation of performance measuring metrics from them.

To illustrate the usefulness of having multiple cameras on a single system, we determine the worst-performing device from each scenario. To compare the performance of each instance, we choose the relative pose error (RPE) as our metric. We assume that the utilization of multiple sensors has an measurable impact on local tracking as more data is available. In contrast, global estimation accuracy, in case of continuous tracking, is depending on the selected backend optimization strategy. Therefore we expect the improvements by the used sensor configuration to be observable in short-term domain and neglect APE evaluation. The respective mean and maximum RPE scores for each scenario are presented in Table V.

It is noteworthy that even though the SLAM algorithms occasionally have a high peak error, the mean errors are often reasonably small. This suggests that there were only temporary losses in tracking, which could be recognized and avoided by taking the output of other SLAM instances into account. The utilization of multiple devices has a beneficial

TABLE IV: Quantitative tracking evaluation for each algorithm. The table illustrates how many instances per run did not loose tracking. A value of 18% for 2 devices reads *In 18% of all runs in this scenario, two instances did not loose tracking*.

SLAM-system	**scenario 0**				**scenario 1**				**scenario 2**			
successful devices	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
VINS-Mono	0%	0%	18%	82%	0%	0%	7%	93%	0%	6%	6%	88%
ORBSLAM2	81%	5%	14%	0%	28%	14%	34%	24%	25%	60%	15%	0%
ORBSLAM3	77%	9%	9%	5%	90%	7%	3%	0%	95%	0%	0%	5%
MROSLAM(*)	32%			68%	21%			79%	8%			92%
DSO	58%	18%	18%	6%	79%	18%	0%	3%	75%	5%	10%	10%
SLAM-system	**scenario 3**				**scenario 4**				**Total**			
successful devices	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
VINS-Mono	0%	0%	8%	92%	0%	0%	7%	93%	0%	1%	12	87%
ORBSLAM2	0%	6%	31%	63%	0%	0%	27%	73%	30%	18%	25%	27%
ORBSLAM3	0%	0%	25%	75%	0%	0%	47%	53%	61%	4%	14%	21%
MROSLAM(*)	0%			100%	0%			100%	14%			86%
DSO	86%	7%	0%	7%	72%	14%	14%	0%	75%	12%	9%	4%

(*) MROSLAM is a multi-camera approach. There is not differentiation between single instances.

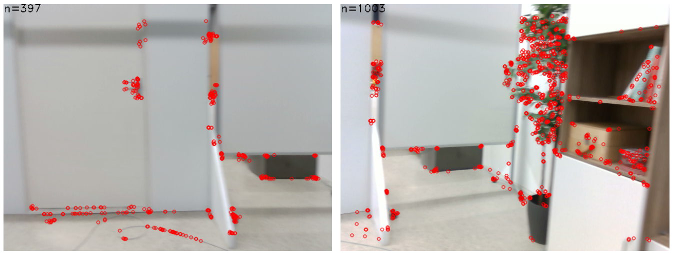


Fig. 7: ORBSLAM2 detected keypoints for two views in a low-texture environment at the same time. The left image shows significant less landmarks (n=397) which could lead to degraded estimation performance or loss of tracking compared to an adjacent camera view (n=1007). Utilizing both views at the same time would further increase the number of available landmarks for tracking and improve accuracy and robustness.

effect on the mean error since the results for MROSLAM rank as one of the lowest in our evaluation. Figure 7 shows the detected keypoints of ORBSLAM2 of two adjacent views. While relying only on a single input, the left image may not provide enough suitable landmarks and tracking will be lost. MROSLAM can use both and is more robust in low-texture cases. However, its maximum error measures are relatively high, indicating even more significant outliers produced in the fusion process which adds constant drift to the estimation as later seen in Figure 8. The more recent ORBSLAM3 occasionally outperforms the multi-camera approach, showing the progress since the introduction of ORBSLAM2 and the derived MROSLAM.

In addition, we also provide representative examples of the pose estimates for the employed SLAM systems compared to the ground truth trajectories in Figure 8. These results show that the visual-inertial system performs better than the purely visual systems in general. Especially during fast rotational movements, the additional information from the IMU leads to significantly better tracking result. Moreover, feature detecting systems perform better than the DSO algorithm, which uses a direct approach. However, the multi-camera MROSLAM suffers a constant drift as it does not implement loop-closure functionality on multiple sensors.

Finally, we evaluate the occurred loss of tracking. We manually examined the frame series in which tracking failure occurred. A significant amount of frames show motion blur or offer only few visual features which can be used for the estimation process. Figure 9 illustrates four individual selected events. They include motion blur and low-textured views offering only limited visual clues for the algorithms. Noteworthy, these defects are frequently observable at the same time. Regardless of the either using a direct approach or relying on features, all algorithms have reduced performance in these situations. However, due to it's multi-sensor nature, MROSLAM is able to recover tracking most of the times.

In summary, this evaluation demonstrates the validity of our dataset as a benchmark for evaluating SLAM systems but also shows the problems of state-of-the-art approaches with motion blur and low-texture environments. Particularly the feature-based visual-inertial system performed well. It also highlights the advantages which multi-camera SLAM approaches could provide. Even though a single device may have poor performance or lose tracking temporarily, others may be more accurate and therefore able to keep the entire system from losing localization.

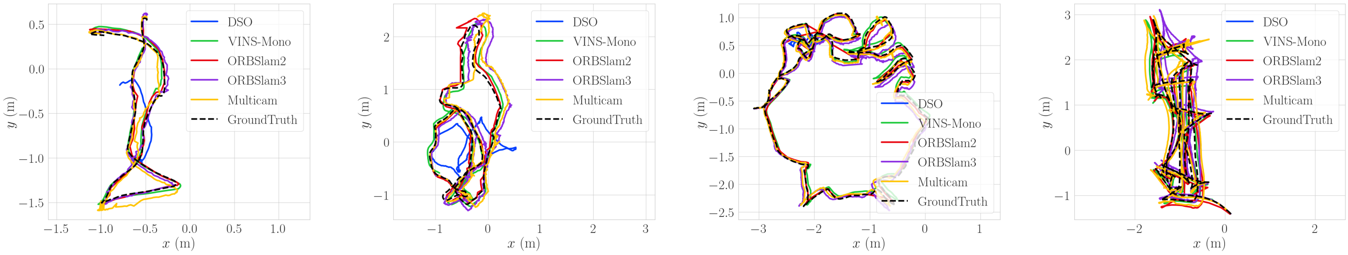
VII. Conclusion

This paper presents a novel dataset for the benchmark of SLAM systems in home environments. It mainly focuses on COTS hardware to decrease the costs for sensor setups while providing multiple similar devices to promote robustness. The environments shown represent common areas for service robotics as office, kitchen and living room settings, where static scenarios as well as ones with changes of objects can be observed. High accurate ground truth information obtained through a motion capture system accompanies the recorded data for evaluation of novel systems.

Finally, we analyzed the proposed data using diverse selections of state-of-the-art SLAM systems to prove its applicability. Furthermore, the outcome showed that these algorithms have difficulty tracking under the influence of motion blur, obstructed view, or in an environment of textureless surroundings. Multi-sensor approaches like MROSLAM however promise less loss of tracking and a better performance regarding local pose estimation. Nevertheless, it still has high outliers

TABLE V: Mean and maximum RPE of the worst performing SLAM instance in a scenario.

SLAM-system	**scenario 0**		**scenario 1**		**scenario 2**		**scenario 3**		**scenario 4**	
RPE	**mean**	**max**	**mean**	**max**	**mean**	**max**	**mean**	**max**	**mean**	**max**
VINS-Mono	0.112672	1.586197	0.126820	1.451341	0.0492078	**0.470913**	0.074746	0.619127	0.029876	0.617029
ORBSLAM2	0.245182	5.421701	0.153198	4.642067	0.181427	5.828487	0.159613	2.434654	0.120661	1.731525
ORBSLAM3	0.087262	5.432609	0.087736	3.990802	**0.035891**	4.908698	**0.023212**	3.492762	0.010758	4.931409
MROSLAM	**0.031168**	3.673559	**0.017651**	6.055320	0.042194	6.195327	0.058612	6.435569	**0.010456**	1.028161
DSO	0.118443	**0.573500**	0.072375	**0.359914**	0.064864	0.704010	0.069510	**0.354366**	0.081397	**0.277259**



(a) Scenario 0 run 5.

(b) Scenario 1 run 22.

(c) Scenario 3 run 11.

(d) Scenario 4 run 10.

Fig. 8: Ground truth reference and estimated trajectories. The four runs have been manually selected out of the total 105 as they show the rare case of all methods not loosing tracking. The trajectories show the *front* instance for single-camera after final optimization or the fused pose for MROSLAM which does not have a final processing step or a loop-closure detection.

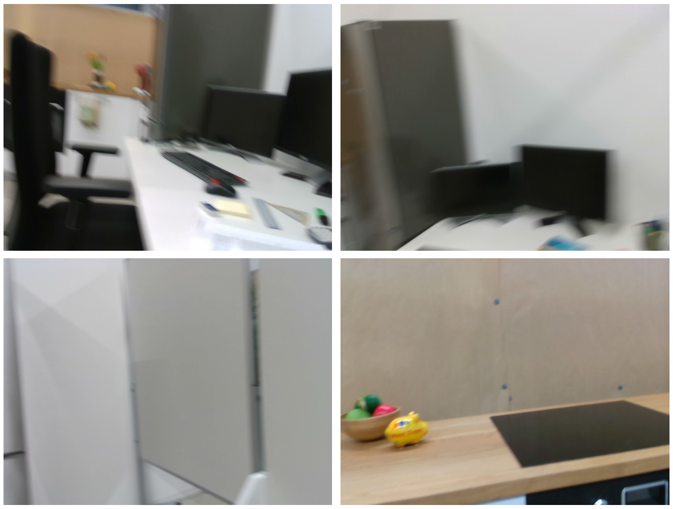


Fig. 9: Examples for views when a loss of tracking occurred. The majority of images is affected by motion blur (upper) or include few visual features (lower) for landmark detection.

which show the necessity of more research on adequate fusion strategies in the multi-sensor scenario.

We, therefore, hope that this dataset contributes to robust yet low-cost robots in home environments.

References

[1] M. Sewtz et al., "Robust approaches for localization on multi-camera systems in dynamic environments," in *2021 7th International Conference on Automation, Robotics and Applications*. IEEE, 2021.

[2] J. Sturm et al., "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE IROS*, 2012.

[3] B. Glocker et al., "Real-time rgb-d camera relocalization," in *International Symposium on Mixed and Augmented Reality*. IEEE, October 2013.

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

[5] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE CVPR*, 2015.

[6] M. Burri et al., "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, 2016.

[7] J. Delmerico et al., "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *2019 IEEE ICRA*, 2019.

[8] D.Schubert et al., "The tum vi benchmark for evaluating visual-inertial odometry," in *2018 IEEE IROS*, 2018.

[9] Shi et al., "Are we ready for service robots? the openloris-scene datasets for lifelong slam," in *2020 IEEE ICRA*, 2020.

[10] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, 2016.

[11] B. Pfrommer et al., "Penncosyvio: A challenging visual inertial odometry benchmark," in *2017 IEEE ICRA*, 2017.

[12] J. Yin et al., "M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, 2022.

[13] J. Wald et al., "Beyond controlled environments: 3d camera re-localization in changing indoor scenes," in *European Conference on Computer Vision*, 2020.

[14] C. Borst et al., "Rollin'justin-mobile platform with variable base," in *2009 IEEE ICRA*. IEEE, 2009.

[15] J. Vogel et al., "Edan: An emg-controlled daily assistant to help people with physical disabilities," in *2020 IEEE/RSJ IROS*. IEEE, 2020.

[16] K. H. Strobl and G. Hirzinger, "More accurate pinhole camera calibration with imperfect planar target," in *2011 IEEE ICCV*, 2011.

[17] A. E. Conrady, "Decentred Lens-Systems," *Monthly Notices of the Royal Astronomical Society*, vol. 79, no. 5, 1919. [Online]. Available: https://doi.org/10.1093/mnras/79.5.384

[18] D. J. Mirota and J. J. Scaife, *Intel(R) RealSense(TM) Depth Camera D435i IMU Calibration*.

[19] K. Madsen, H. Nielsen, and O. Tingleff, *Methods for Non-Linear Least Squares Problems (2nd ed.)*, 2004.

[20] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, 2018.

[21] R. Mur-Artal et al., "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, 2015.

[22] C. Campos et al., "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam."

[23] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," in *arXiv:1607.02565*, July 2016.

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