SLAM&Render: A Benchmark for the Intersection Between Neural Rendering, Gaussian Splatting and SLAM

The International Journal of Robotics Research
XX(X):1–9

DOI: 10.1177/ToBeAssigned
ijr.sagepub.com/
SAGE

Samuel Cerezo[1], Gaetano Meli[12], Tomás Berriel Martins[1], Kirill Safronov[2] and Javier Civera[1]

Abstract

Models and methods originally developed for Novel View Synthesis, such as Neural Radiance Fields (NeRF) and Gaussian Splatting, are increasingly being adopted as representations in Simultaneous Localization and Mapping (SLAM). However, existing datasets fail to include specific research challenges of both fields, such as sequential processing and multi-modality in SLAM, or generalization across viewpoints and illumination changes in neural rendering. Additionally, datasets are often recorded using sensors that are handheld or mounted on drones or mobile robots, which complicates an accurate reproduction of sensor motions under different conditions. To bridge these gaps, we introduce **SLAM&Render**, a novel dataset designed to benchmark methods in the intersection between SLAM and Novel View Rendering. Recorded with a robot manipulator, it uniquely includes 40 sequences with time-synchronized RGB-D, IMU, robot kinematics and ground-truth pose streams. The dataset features five setups with consumer and industrial objects under four controlled lighting conditions, each with training and test trajectories with significant viewpoint changes. All sequences are static, with different levels of object rearrangements and occlusions. Our experimental results, obtained with several baselines from the literature, validate **SLAM&Render** as a relevant benchmark for this emerging research area. The dataset can be accessed through the following link: https://samuel-cerezo.github.io/SLAM&Render.

Keywords

Dataset, SLAM, Novel View Synthesis, Ground-truth

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a key enabler for autonomous robot navigation and scene understanding. SLAM targets the estimation of a robot's pose while simultaneously building a consistent map of its surroundings using only onboard sensors. Several high-performing multimodal Campos et al. (2021) and unimodal Teed and Deng (2021); Zhang and Singh (2014) methods are available today, although open challenges still remain Cadena et al. (2016); Macario Barros et al. (2022). SLAM differs from alternative approaches to 3D reconstruction (e.g., Structure from Motion Schonberger and Frahm (2016)) by its online, real-time and multimodal capabilities, fulfilling the requirements imposed by robotic applications. Several works Klingensmith et al. (2016); Süβ et al. (2022); Li et al. (2019) show the benefits of combining SLAM methods with robot kinematic data in the presence of uncertainties in the kinematic model and/or actuators. Robot kinematics can be of great help for SLAM, as it may provide accurate pose measurements. However, robot kinematics alone does not fully solve SLAM, as it does not address, among others, shifts in the extrinsic parameters Li et al. (2024), the estimation of high-definition maps Tang et al. (2023), cross-session relocalization and map merging Campos et al. (2021), all of which which are paramount for workspace exploration Placed et al. (2023) and lifelong representations Catalano et al. (2025).

Novel View Synthesis (NVS), seemingly distant from robot SLAM, has experienced a resurgence due to groundbreaking research in Neural Radiance Fields (NeRFs) Mildenhall et al. (2021). NeRFs have been quickly adopted as a map representation in visual Sucar et al. (2021); Zhu et al. (2022) and LiDAR Deng et al. (2023) SLAM pipelines. On the other hand, Gaussian Splatting Kerbl et al. (2023), developed as a much faster alternative to NeRFs, has had an even bigger impact in visual SLAM Matsuki et al. (2024); Keetha et al. (2024). For a comprehensive overview of how these emerging paradigms are reshaping SLAM, see the recent survey by Tosi et al. Tosi et al. (2024).

Motivated by the growing convergence of visual SLAM, Neural Rendering, and Gaussian Splatting, and by the lack of benchmarks explicitly dedicated to it, we introduce

[1]Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, 50018 Zaragoza, Spain
[2]Technology & Innovation Center, KUKA Deutschland GmbH, 86165 Augsburg, Germany

Corresponding author:
Samuel Cerezo, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, 50018 Zaragoza, Spain.
Email: samueladriancerezo@unizar.es

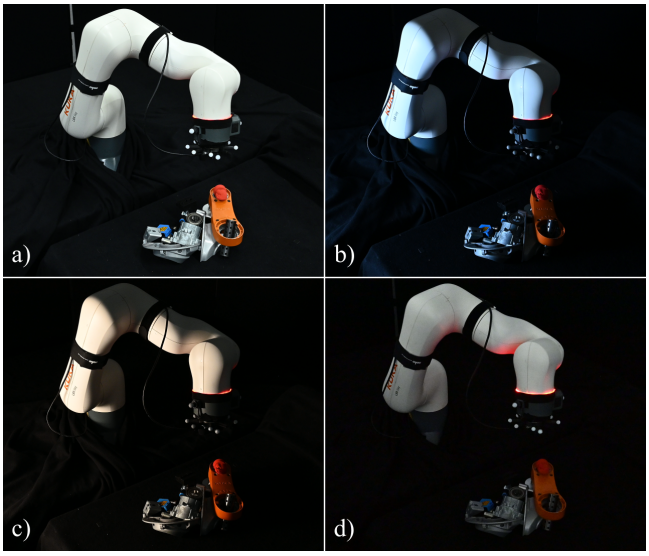

Figure 1. Illustration of the capture setup for our **SLAM&Render** dataset, recorded from a Intel RealSense at the end effector of a robotic arm that moves around a set of objects on a table. See the four different light conditions present in our dataset: a) natural, b) cold, c) warm, and d) dark

SLAM&Render. Recorded with a robot manipulator, it ensures accurate reproductions of camera motions for different lighting conditions and scene setups, in order to study their effect separately. The sequences are divided into *train* and *test*, with significant viewpoint changes between them in order to assess generalization. The dataset provides 40 static sequences with synchronized RGB-D images, IMU readings, robot kinematic data, camera parameters, and ground-truth data. It comprises five setups with consumer and industrial objects under four controlled lighting conditions, showcasing object rearrangements with varying levels of occlusion.

All data is available online under CC-BY 4.0 license at our website. Additional information, software tools, and videos for visual inspection of the data are also included.

2 Related Work

Publicly available datasets have played a crucial role in advancing research in SLAM and NVS. Table 1 provides an overview of the most relevant datasets in this field. Unless otherwise stated, the following datasets *do not* feature (i) robot kinematic data, (ii) full control of lighting conditions, (iii) reproducible sensor motions with high accuracy, and (iv) object rearrangements across sequences.

To our knowledge, SLAM&Render is the first dataset providing time-synchronized multi-sensor and kinematic data, object occlusions and rearrangements in a fully controlled recording setup (see Table 1).

2.1 Datasets for SLAM

The EuRoC dataset Burri et al. (2016) provides visual–inertial data from a micro aerial vehicle organized into two batches: industrial hall sequences for SLAM evaluation, and indoor room sequences for 3D reconstruction. Several challenges, such as varying natural light conditions and dynamic scenes, are included. The TUM VI dataset Schubert et al. (2018) is designed for evaluating odometry algorithms. It contains RGB images, photometric calibration, synchronized IMU data, and ground-truth camera poses, while depth images are not included. The OpenLORIS-Scene dataset Shi et al. (2020) consists of visual–inertial, LiDAR and odometry data (i.e., encoder readings) from a mobile robot to benchmark SLAM and scene understanding in everyday scenarios. The Newer College Ramezani et al. (2020) and the Oxford Spires Tao et al. (2025) datasets provide visual-inertial, and LiDAR data from handheld devices. Both include time-synchronized data, calibration parameters for all the involved sensors, loop closures, and cluttered sequences observed from varying viewpoints. The Replica dataset Straub et al. (2019) includes 18 photorealistic indoor multi-scale reconstructions with dense meshes, rich textures, and reflective surfaces.

2.2 Datasets for Novel View Synthesis

Despite the popularity of NeRFs, most datasets used for evaluation sample frames along a single camera trajectory, often lacking the diversity and complexity of real-world settings. DTU Jensen et al. (2014) was introduced to evaluate multi-view stereo techniques using a robotic setup for object 3D reconstruction tasks. It features systematic light control with a light scanner, however it addresses a very specific use case. In the same application field, Tanks and Temples Knapitsch et al. (2017) was captured in realistic indoor and outdoor settings using an industrial laser scanner for ground-truth collection. Recently, Mip-NeRF360 Barron et al. (2022) introduced a dataset for high-quality NVS. It captured camera trajectories around a central object in indoor and outdoor scenarios with detailed backgrounds. ScanNet++ Yeshwanth et al. (2023) benchmarks 3D scene understanding methods by combining 3D laser scans, high-quality images, and RGB-D streams with rich semantic annotations and independent camera trajectories for NVS.

Within datasets for robotics applications, the YCB Object and Model Set Calli et al. (2015) provides RGB-D images of several object manipulation sequences. In contrast, JIGSAWS Gao et al. (2014) dataset collects time-synchronized stereo images from an endoscopic camera, along with kinematic data from a surgical robot. However, it solely addresses applications in the field of surgery activities.

Unlike prior works, our dataset uniquely benchmarks SLAM, Neural Rendering, and Gaussian Splatting methods with 40 time-synchronized sequences featuring multimodal data, including RGB-D images, IMU readings, ground-truth, and robot kinematic data. Recorded with a robot manipulator it allows for accurately reproducing hand-eye camera motions, which are split in train and test trajectories. The five setups contain consumer and industrial objects under four controlled lighting conditions, which can generally have a negative impact on SLAM and Novel View Synthesis Ye et al. (2024); Martin-Brualla et al. (2021).

3 Notation

3.1 Convention for rigid transformations

A rigid transformation between two reference frames A and B is defined as $\mathbf{T}_A^B = \left[\mathbf{R}_A^B,\, \mathbf{p}_A^B;\, \mathbf{0}_{1\times 3},\, 1\right] \in SE\,(3)$,

Table 1. Overview of commonly used datasets for SLAM and NVS, including available sensor modalities (RGB, Depth, IMU, Encoders), ground-truth camera pose (GT), illumination variations (Ill.), scene rearrangements (Rearr.), and time synchronization (TS).

Year	Dataset	Use	RGB	Depth	IMU	Enc.	GT	Ill.	Rearr.	TS.
2014	DTU Jensen et al. (2014)	NVS	✓	✓	✗	✗	✓	✓	✗	✗
2014	JIGSAWS Gao et al. (2014)	Motion Model.	✓	✗	✗	✓	✗	✗	✗	✗
2015	YCB object and Model set Calli et al. (2015)	Robotic Manip.	✓	✓	✗	✗	✓	✗	✗	✗
2016	EuRoC Burri et al. (2016)	SLAM	✓	✗	✓	✗	✓	✓	✗	✓
2017	Tanks and Temples Knapitsch et al. (2017)	NVS	✓	✓	✗	✗	✗	✗	✗	✗
2018	TUM VI Schubert et al. (2018)	Odom.	✓	✗	✓	✗	(✓)	✗	✗	✗
2019	OpenLORIS-Scene Shi et al. (2020)	SLAM	✓	✓	✓	✓	✓	✗	✗	✓
2019	Replica Straub et al. (2019)	SLAM & NVS	✓	✓	✓	✗	✓	✗	✗	✓
2020	Newer College Ramezani et al. (2020)	SLAM & NVS	✓	✗	✓	✗	✓	✗	✗	✓
2021	Mip-NeRF360 Barron et al. (2022)	NVS	✓	✗	✗	✗	✗	✗	✗	✗
2023	ScanNet++ Yeshwanth et al. (2023)	SLAM & NVS	✓	✓	✗	✗	✓	✗	✗	✓
2025	Oxford Spires Tao et al. (2025)	SLAM & NVS	✓	✓	✓	✗	✓	✗	✗	✓
2025	**SLAM&Render**	SLAM & NVS	✓	✓	✓	✓	✓	✓	✓	✓

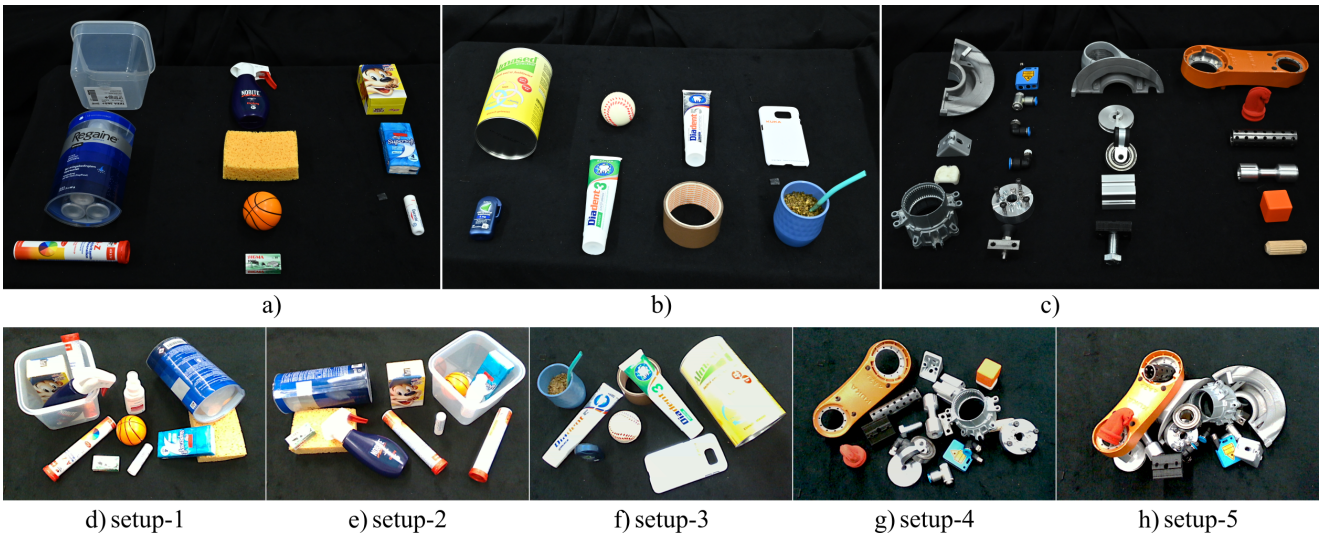

Figure 2. Illustration of the objects included in **SLAM&Render**: a-b) supermarket goods, c) industrial goods, d) – h) object arrangements of the five setups.

where $\mathbf{R}_A^B \in SO(3)$ and $\mathbf{p}_A^B \in \mathbb{R}^3$ denote the rotation and translation of A with respect to B, respectively. For any point $\mathbf{p}^A \in \mathbb{R}^3$, its coordinates in B are $\tilde{\mathbf{p}}^B = \mathbf{T}_A^B \tilde{\mathbf{p}}^A$, where $\tilde{\mathbf{p}}^A$ and $\tilde{\mathbf{p}}^B$ are the homogeneous vectors of $\mathbf{p}^A$ and $\mathbf{p}^B$, respectively.

3.2 *Forward kinematics*

A manipulator consists of a sequence of rigid bodies (*links*) connected through kinematic *joints*. Let $\boldsymbol{x}(t) \in \mathbb{R}^m$ be the task variable to be controlled (e.g., end-effector pose) and $\boldsymbol{q}(t) \in \mathbb{R}^n$ the vector of the variables describing the configuration (i.e, joint positions) of a given n-DOFs robot, with $n \geq m$. The relation between $\boldsymbol{q}$ and $\boldsymbol{x}$ is given by the forward kinematics equation

$$\boldsymbol{x}(t) = f(\boldsymbol{q}(t)), \tag{1}$$

where $f(\cdot)$ is, in general, a nonlinear function allowing for the computation of the task variables starting from the robot configuration.

4 The SLAM&Render Dataset

The proposed dataset comprises 40 static sequences that include both inertial and RGB-D data. These were recorded using an Intel RealSense D435i camera, mounted in an eye-to-hand configuration on a KUKA LBR iisy 3 R760 manipulator. The robot has been employed as a precise and reliable sensor carrier, ensuring consistent and reproducible camera trajectories. Additionally, kinematic data were collected from the manipulator. In particular, the dataset contains the robot configurations $\boldsymbol{q}(t)$ along the motion and the corresponding flange poses computed using (1). An external Motion Capture System (MCS) was used to obtain the ground-truth camera poses. All sequences were recorded in an environment where the objects were placed on a black background, surrounded by black panels featuring pictures that display complex and colorful patterns (see Sec. 4.1). Moreover, the dataset contains *train* and *test* camera trajectories recorded under the following four different lighting conditions:

- *natural* – Natural light of the environment
- *cold* – Cold artificial light
- *warm* – Warm artificial light

- *dark* – No light.

This enables our dataset to be used with classical SLAM approaches as well as newer Gaussian Splatting- and NeRF-based paradigms, which are particularly sensitive to lighting conditions and may require proper training to perform effectively. Therefore, this dataset feature ensures that SLAM algorithms are tested and trained in environments that reflect real-world variability, ultimately evaluating their performance and reliability in practical applications. Although the KUKA LBR iisy 3 R760 manipulator has a repeatability parameter* of ± 0.1 mm, we release the joint configurations and flange poses for all the sequences belonging to train and test trajectories. As a result, the camera pose estimation conducted by any SLAM algorithm on this dataset remains unaffected by potential variations in joint configurations. Furthermore, with smooth robot motions, the camera performs loop closures in both train and test trajectories (see Fig. 3). The sequences include scenes with both opaque and transparent objects, reflective and non-reflective materials, as well as varying occlusions resulting from the rearrangement of objects. The dataset features the following five distinct scenes (see Fig. 2):

- *setup-1* and *setup-2*: supermarket goods, including transparent objects;
- *setup-3*: supermarket goods with only opaque objects;
- *setup-4*: industrial objects;
- *setup-5*: industrial objects in a cluttered arrangement.

We additionally release chessboard images for intrinsic and extrinsic calibration. The calibration procedure and resulting parameters are detailed in Sec. 4.5. Tools for data association, ROS2 bag data extraction, and synchronization can be found on our website.

4.1 Data acquisition

All RGB-D data were recorded at the optimal depth† (848×480) and full frame rate (30 Hz) of the Intel RealSense D435i camera. The IMU data were acquired at a rate of 210 Hz, capturing synchronized readings from the gyroscope and accelerometer. All the dataset sequences were collected through a system based on Ubuntu 22.04 and ROS2 Humble. We used the software library `Intel RealSense SDK 2.0`‡ and its ROS2 wrapper§ to record the camera data. While the depth and color images are initially unaligned, as they are captured by two distinct sensors, we used the built-in feature of the Intel ROS2 wrapper to register the depth images to the color images. This is achieved by projecting the depth images to 3D and then back-projecting them into the color camera view. The joint angles of the robot manipulator were collected using ROS2 at a frequency of 25 Hz.

To record ground-truth data, we employed an OptiTrack MCS consisting of twelve PrimeX 13W cameras installed around the robot cell. These cameras track the position and orientation of passive markers at a frequency of 120 Hz. The markers are mounted on a circular structure between the camera and the robot flange (see Fig. 4). This structure was printed using an industrial-grade 3D printer, ensuring high accuracy while tracking the markers. Additionally, we have added two markers on the flange and one marker on the camera itself to enhance the asymmetry of the marker setup.

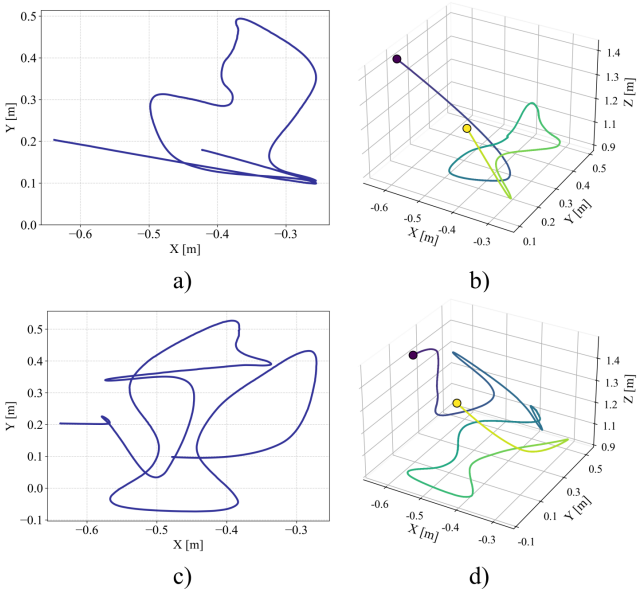

Figure 3. Our **SLAM&Render** dataset contains train and test camera trajectories. From top left to bottom right: a) 2D representation of train trajectory, b) 3D representation of train trajectory, c) 2D representation of test trajectory, and d) 3D representation of test trajectory. Blue and yellow points represent start and end points of the trajectories, respectively.

To minimize reflections from the surrounding environment, non-reflective black panels were installed around the perimeter of the robot cell. Pictures featuring complex and colorful patterns have been placed on the black panels to avoid oversimplifying the novel view synthesis of NeRF-based algorithms. Further, the metal table supporting the robot manipulator was covered with a black cloth. This recording setup was chosen to ensure high-quality, accurate ground-truth data, as environmental reflections and lighting variations within the robot cell degraded the ground-truth data in early recordings. The cold and warm lighting setups were obtained thanks to a Neewer 660 (50W) PRO LED lamp.

4.2 Dataset structure

Inspired by Sturm et al. (2012), our dataset is organized into sequences that share the same structure across all setups. For the sake of simplicity, consider the following *setup-1* structure:

```
setup-1
├── natural/
│   └── ...
├── warm/
│   └── ...
└── cold/
    └── ...
```

*The repeatability parameter gives a measure of the manipulator's ability to return to a previously reached position. The smaller the value, the higher the repeatability.

†https://dev.intelrealsense.com/docs/tuning-depth-cameras-for-best-performance

‡https://www.intelrealsense.com/sdk-2/

§https://dev.intelrealsense.com/docs/ros2-wrapper

```
├── dark/
│   └── ...
├── intrinsics.yaml
└── extrinsics.yaml
```

Each lighting condition has a *train* and *test* sequence, which are organized equally. For example, the structure of the *train* sequence is reported below.

```
train
├── rgb/
│   ├── 1739371345.939813614.png
│   ├── 1739371346.193981361.png
│   └── ...
├── depth/
│   ├── 1739371345.939813614.png
│   ├── 1739371346.193981361.png
│   └── ...
├── robot_data/
│   ├── joint_positions.txt
│   └── flange_poses.txt
├── associations.txt
├── imu.txt
├── groundtruth_raw.csv
└── groundtruth.txt
```

These folders and files contain the following data:

- `rgb/`: timestamped color images (`PNG` format, 3 channels, 8 bits per channel);
- `depth/`: timestamped depth images (PNG format, 1 channel, 16 bits per channel, distance scaled by a factor of 1000). The depth images are already aligned with the RGB sensor;
- `associations.txt`: list of all the temporal associations between the RGB and depth images (format: `rgb_timestamp rgb_file depth_timestamp depth_file`);
- `imu.txt`: timestamped accelerometer and gyroscope readings (format: `timestamp ax ay az gx gy gz`);
- `groundtruth.txt`: timestamped ground truth of the camera poses w.r.t world reference frame, expressed as a sequence of quaternions and translation vectors (format: `timestamp qx qy qz qw tx ty tz`);
- `groundtruth_raw.csv`: MCS ground-truth raw data of flange marker ring trajectory;
- `extrinsics.yaml`: transformation matrices of the reference frames involved in the data collection process (see Sec. 4.3);
- `instrinsics.yaml`: intrinsic calibration parameters of the camera;
- `robot_data/flange_poses.txt`: timestamped poses of the robot flange. Each pose is expressed w.r.t. the robot base as a quaternion and translation vector (format: `timestamp qx qy qz qw tx ty tz`);
- `robot_data/joint_positions.txt`: timestamped angles of each robot joint (format: `timestamp q1 q2 q3 q4 q5 q6`).

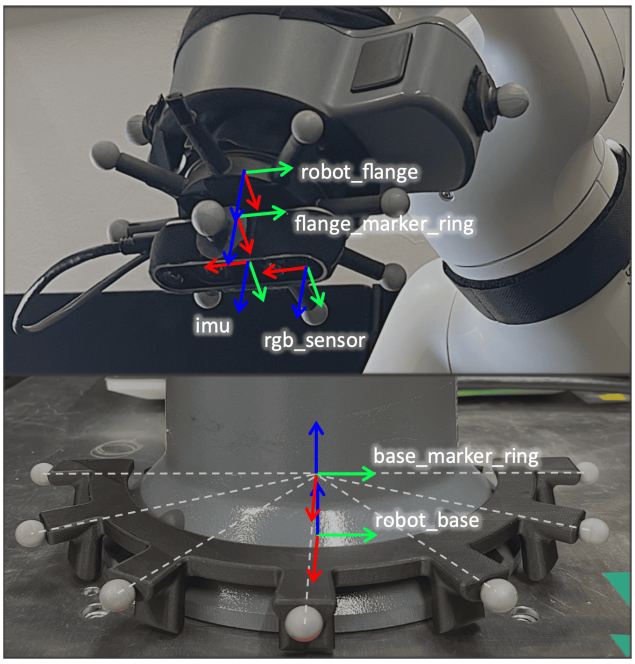

Figure 4. Illustration of the reference frames involved in the data collection.

The accelerometer data are provided in $\mathrm{m/s^2}$, the gyroscope data in $\mathrm{rad/s}$, and the ground truth and flange positions in mm. All sequences are also available as ROS2 bag files.

4.3 Reference frames

The reference frames shown in Fig. 4 were used in the data collection process to align the data coming from the camera, the robot and the MCS. These reference frames are described below.

- `robot_flange`: defined at the center of the real robot flange;
- `flange_marker_ring`: used by the MCS to locate the robot flange;
- `rgb_sensor`: optical center of the RGB sensor;
- `imu`: IMU sensor pose;
- `base_marker_ring`: used by the MCS to locate the robot base;
- `robot_base`: defined at the center of the real robot base.

4.4 Motion Capture System (MCS)

Calibration. To collect ground-truth data, twelve Optitrack PrimeX 13W cameras were used. These cameras were carefully positioned to focus on the robot flange from various angles, minimizing occlusion caused by the robot's movements. The cameras were calibrated using the Motive software[¶] provided by Optitrack. This process involves waving a calibration wand, throughout the area to be captured. After the wanding process, Motive compares the 2D results from the individual cameras to reconstruct a 3D capture volume. The calibration process resulted in a 3D

[¶] https://optitrack.com/software/motive/

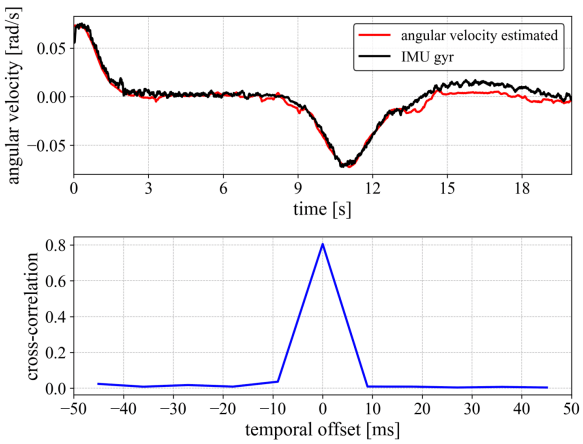

Figure 5. Validation of time-synchronized data by comparing angular velocity estimates from the MCS and the IMU gyroscope measurements.

Table 2. Intrinsic camera parameters used in our dataset. The parameters include focal lengths (fx,fy), principal point (cx,cy), and the distortion coefficients.

Parameters	fx [px]	fy [px]	cx [px]	cy [px]	distortion coefficients
factory	605.2	605.1	425.7	246.0	0 0 0 0 0
estimated	599.6	599.2	426.6	246.3	0.1 -0.4 0 0 0.3

error mean of 0.4 mm, representing the convergence error between the twelve cameras.

Synchronization. The data gathered from the MCS are initially unsynchronized. To ensure temporal alignment, for each sequence we retrieved the *capture starting time* provided by OptiTrack. It was used to compute the Unix Epoch timestamp for each pose reported in the provided ground-truth file. Following the approach presented in Bonarini et al. (2006), the synchronization was validated by comparing the angular velocity estimated from the MCS with the gyroscope measurements from the IMU using cross-correlation (see Fig. 5). The maximum at zero offset confirms correct alignment.

4.5 Camera calibration

To estimate the hand-eye transformation (i.e., the transformation from the robot flange to the camera projection plane), more than 200 images containing the calibration plate were captured, and the corresponding robot pose for each image was saved. Subsequently, the calibration method presented in Shiu and Ahmad (1989) was applied. The calibration plate (a chessboard with squares of 30×30 mm) was detected by applying a custom detection method (see Fig. 6). For a more accurate calibration, not only the extrinsic parameters (hand-eye transformation) were estimated, but also the intrinsic parameters. Our experiments have proven that the use of the estimated intrinsic parameters resulted in more accurate extrinsic parameters. The intrinsic parameters of the camera (focal length, principal point and camera distortion coefficients) are shown in Table 2. The extrinsic parameters are shown in the Table 3.

To compare the accuracy of the calibration using both factory and estimated intrinsic parameters, the set of calibration images was divided into several batches (10 batches with 20 images in each), and the same calibration process was applied to each batch. The accuracy of the calibration was assessed using the metrics proposed in Enebuse et al. (2022), which include translation error in millimeters (*Trans.* [mm]), rotation error in degrees (*Rot.* [deg.]), and unitless relative transformation error (*Rel. transf. error* [a.u.]). Table 4 shows the mean, standard deviation, and standard error of the calibration metrics. The accuracy of the hand-eye transformation, calculated using the estimated intrinsic parameters, showed more accurate results, with reduced reprojection, translation, rotation, and relative transformation errors.

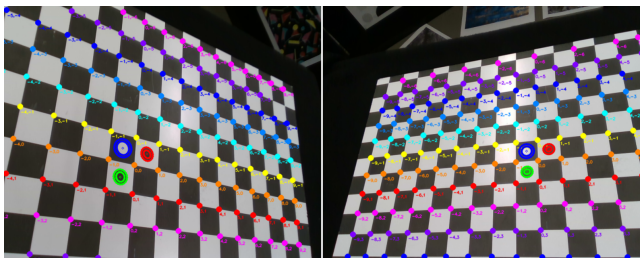

Figure 6. Corners and fiducial markers detected within the calibration procedure.

Table 3. Extrinsic parameters (hand–eye transformation): translation [mm] and unit quaternion rotation.

Used intrinsics	tx	ty	tz	qx	qy	qz	qw
factory	0.5895	32.5419	46.0707	0.0060	0.0041	-0.7035	0.7106
estimated	-0.2852	32.7894	46.1513	0.0063	0.0054	-0.7037	0.7104

Table 4. Mean, standard deviation and standard error of the calibration metrics. The last two rows show the results using all images with the estimated and factory intrinsic parameters.

	Reproj. error	Hand-eye		
	[px]	Trans. [mm]	Rot. [deg.]	Rel. transf. err. [a.u.]
Mean	0.36	2.38	0.33	2.90
Std. dev.	0.04	0.41	0.08	0.66
Std. error	0.01	0.13	0.02	0.21
All images (estim. intrinsics)	**0.38**	**2.31**	**0.31**	**2.69**
All images (factory intrinsics)	0.61	2.37	0.33	2.77

5 Experiments

To evaluate the dataset's suitability for benchmarking methods at the intersection of SLAM and Novel View Synthesis, we present results from several baselines.

5.1 Metrics

We assess camera tracking performance using the Root Mean Square Error of the Absolute Trajectory Error (ATE RMSE ↓). Following Kerbl et al. (2023) to evaluate NVS, we report Peak Signal-to-Noise Ratio (PSNR ↑), Structural Similarity Index Measure (SSIM ↑), and Learned Perceptual Image Patch Similarity (LPIPS ↓). PSNR captures pixel-level differences, while SSIM reflects perceptual similarity in structure, luminance, and contrast Wang et al. (2004). Finally, LPIPS measures perceptual fidelity using deep feature distances Zhang et al. (2018).

Table 5. Classic evaluation of Gaussian Splatting (gsplat) and FeatSplat (fsplat) reporting PSNR [↑], LPIPS [↓], and SSIM [↑].

	natural			warm			dark		
	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
gsplat-7k	17.32	0.445	0.694	17.67	0.497	0.609	23.77	0.464	0.678
fsplat-7k	17.40	0.445	0.700	17.75	0.499	0.610	23.43	0.467	0.687
gsplat-21k	18.43	0.411	0.724	18.75	0.468	0.637	24.99	0.443	0.708
fsplat-21k	**19.45**	**0.406**	**0.743**	**20.01**	**0.461**	**0.656**	**25.13**	**0.444**	**0.714**

Table 6. Evaluation on independent test trajectory of Gaussian Splatting (gsplat) and FeatSplat (fsplat) reporting PSNR [↑], LPIPS [↓], and SSIM [↑].

	natural			warm			dark		
	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
gsplat-7k	14.53	0.493	0.608	**15.45**	0.543	0.527	21.82	0.499	**0.596**
fsplat-7k	**14.73**	0.488	**0.623**	15.42	0.541	**0.540**	21.73	0.498	0.630
gsplat-21k	14.42	**0.478**	0.606	15.35	**0.531**	0.524	21.53	0.492	0.600
fsplat-21k	14.71	**0.478**	0.614	15.29	0.532	0.535	**21.91**	**0.490**	0.627

5.2 Results

Novel View Synthesis (NVS). To highlight the value of independent test trajectories, we evaluated two state-of-the-art baselines: Gaussian Splatting (GSPLAT) Kerbl et al. (2023) and FeatSplat (FSPLAT) Martins and Civera (2024), the latter designed to mitigate viewpoint overfitting. In this experiment, *setup-1*, *setup-3*, and *setup-4* were used under natural, warm, and dark lighting conditions. Baselines were initialized with random point clouds and a black background. GSPLAT used four spherical-harmonics bands, and FSPLAT used 32-dimensional latent vectors and a multilayer perceptron with one hidden layer of 64 units. They were optimized for a total of 21k iterations, and we report metrics both at 7k and 21k iterations. The baselines were optimized on the respective *train* trajectories of each setup, skipping one every ten frames. We evaluate both with the common approach of using skipped frames from the *train* trajectory Mildenhall et al. (2021); Kerbl et al. (2023) (see Table 5) and using the independently recorded *test* trajectory of each setup (see Table 6).

The common evaluation indicates an improvement in rendering performance, for both baselines, through the complete optimization process (see Table 5). Both baselines achieve their best performance under *dark* lighting conditions, which is probably due to using black as initial background color. Nevertheless, the test trajectory at **SLAM&Render** reveals that both baselines overfit to the training trajectory. Note how the rendering quality does not improve beyond the 7k iteration and even slightly declines in Table 6. Qualitative analysis in Fig. 7 also showcases the contrast between the high quality rendering of validation viewpoints and the artifacts generated in test viewpoints. This result validates the need for datasets with independent trajectories to evaluate novel view synthesis methods and shows that the classic approach of sampling evaluation frames from the training trajectory does not inform about trajectory overfitting.

Camera tracking. To illustrate the advantage on using kinematic data, three approaches based on Gaussian Splatting SLAM Matsuki et al. (2024) (MonoGS) were evaluated. In the first experiment, we used the default configuration for MonoGS, while in the second one we integrated the robot kinematic data to describe the camera

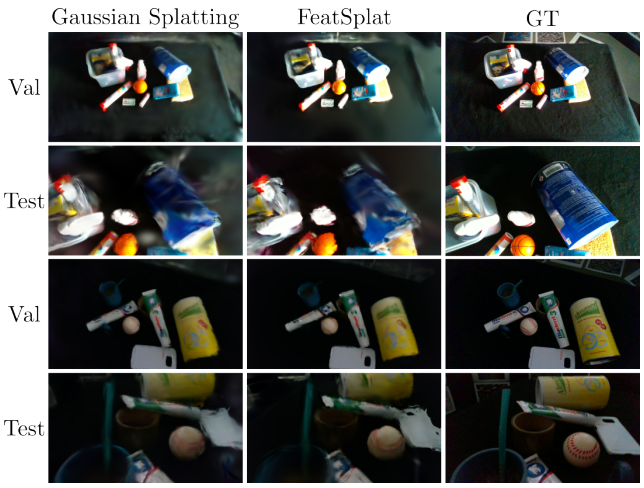

Figure 7. Comparison of NVS for validation and test viewpoints. Although both models perform well when evaluated on validation viewpoints of the training trajectory, the independent test trajectory reveals overfitting.

Table 7. ATE RMSE [↓ cm] across multiple sequences using three algorithm configurations: MonoGS, Kinematics and MonoGS with Kinematic data as seed.

Setup	Ill.	Traj.	MonoGS	Kinematics	MonoGS (Kinem. seed)
setup-1	natural	train	3.3	0.5	2.0
	natural	test	3.9	0.4	2.1
	cold	test	2.4	0.3	2.2
	warm	test	2.1	0.3	1.4
	dark	test	3.1	0.4	1.9
setup-3	natural	train	2.6	0.5	1.1
	natural	test	3.7	0.4	1.7
	dark	train	0.6	0.5	0.6
	dark	test	3.8	0.3	1.6
setup-4	natural	train	0.9	0.5	0.7
	cold	train	2.4	0.5	1.0
	warm	train	1.6	0.5	0.8
	dark	train	1.0	0.5	0.9

motion. The evaluation of the second approach is referred to as the *Kinematics* experiment. In the third experiment, robot kinematics were incorporated as an initial seed for camera pose estimation within MonoGS. This is referred to as *Kinem. seed* experiment. Following the evaluation protocol in Matsuki et al. (2024), the ATE RMSE for keyframes is reported in Table 7. For this experiment, *setup-1*, *setup-3*, and *setup-4* were used under different light conditions while performing *train* and *test* trajectories. In all cases, it is evident that incorporating kinematic data improves camera tracking performance allowing for drift corrections. However, the results also show that the naive approach of using robot kinematics as the initial seed of MonoGS decreases its accuracy. This result motivates the use of our **SLAM&Render** dataset for research on multimodal fusion within NeRF and Gaussian Splatting SLAM. Fig. 8 illustrates the camera tracking results for *setup-4* under cold lighting condition. The trajectory using MonoGS is shown in Fig. 8a, while Fig. 8b shows the result using kinematic data as initial seed.

6 Conclusions

In this work, we introduced **SLAM&Render**, a dataset for benchmarking methods that combine SLAM with Neural Rendering and Gaussian Splatting. Our dataset

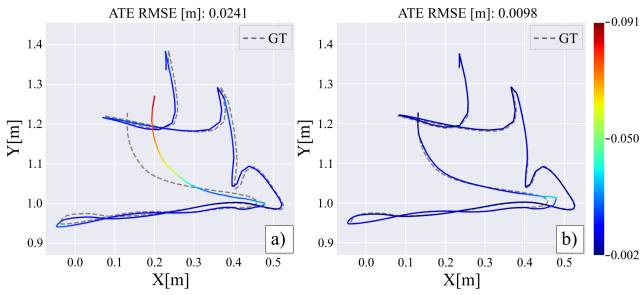

Figure 8. Camera tracking result for *setup-4* under cold light using a) MonoGS and b) MonoGS with kinematic data as initial seed. Dashed line represent the ground-truth while colored line represent the camera trajectory. Incorporating kinematic data allows drift correction in camera tracking.

includes 40 sequences of synchronized RGB and depth images, IMU readings, robot kinematic data, intrinsic and extrinsic camera parameters, and ground-truth pose data across diverse scenes and conditions. **SLAM&Render** addresses research challenges such as generalization across reproducible trajectories and robustness to scene and illumination changes. By releasing robot kinematic data, it also enables the assessment of novel SLAM strategies when applied to robot manipulators. It aims to drive future research in learning-based and hybrid methods requiring accurate localization and multi-sensor data fusion, ultimately enhancing the deployment of systems that combine perception, mapping, and rendering. Future extensions of the dataset may include dynamic scenes and a wider range of object categories, with priorities shaped by community feedback and usage patterns. These enhancements will strengthen robustness and generalization tests while maintaining the dataset's core focus.

7 Data access methods

The datasets can be downloaded from https://samuel-cerezo.github.io/SLAM&Render. Furthermore, some example tools to load and plot the data in Python are provided and can be used as a starting point for implementations in other programming languages.

8 Acknowledgments

The authors would like to thank the Technology & Innovation Center of KUKA Deutschland GmbH (Augsburg, Germany) for their support and resources provided during the course of this research.

References

Barron JT, Mildenhall B, Verbin D, Srinivasan PP and Hedman P (2022) Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR* .

Bonarini A, Burgard W, Fontana G, Matteucci M, Sorrenti DG, Tardos JD et al. (2006) Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets. In: *In proceedings of IROS*, volume 6. p. 93.

Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW and Siegwart R (2016) The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research* 35(10): 1157–1163.

Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I and Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics* 32(6): 1309–1332.

Calli B, Singh A, Walsman A, Srinivasa S, Abbeel P and Dollar AM (2015) The ycb object and model set: Towards common benchmarks for manipulation research. In: *2015 international conference on advanced robotics (ICAR)*. IEEE, pp. 510–517.

Campos C, Elvira R, Rodríguez JJG, Montiel JM and Tardós JD (2021) Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics* 37(6): 1874–1890.

Catalano I, Zumaya CC, Placed JA, Civera J, Bessa WM and Peña-Queralta J (2025) 3d scene graphs in robotics: A unified representation bridging geometry, semantics, and action. *TechRxiv* .

Deng J, Wu Q, Chen X, Xia S, Sun Z, Liu G, Yu W and Pei L (2023) Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 8218–8227.

Enebuse I, Ibrahim BKSMK, Foo M, Matharu RS and Ahmed H (2022) Accuracy evaluation of hand-eye calibration techniques for vision-guided robots. *PLOS ONE* 17(10): 1–26.

Gao Y, Vedula SS, Reiley CE, Ahmidi N, Varadarajan B, Lin HC, Tao L, Zappella L, Béjar B, Yuh DD et al. (2014) Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In: *MICCAI workshop: M2cai*, volume 3. p. 3.

Jensen R, Dahl A, Vogiatzis G, Tola E and Aanæs H (2014) Large scale multi-view stereopsis evaluation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 406–413.

Keetha N, Karhade J, Jatavallabhula KM, Yang G, Scherer S, Ramanan D and Luiten J (2024) Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 21357–21366.

Kerbl B, Kopanas G, Leimkühler T and Drettakis G (2023) 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42(4): 139–1.

Klingensmith M, Sirinivasa SS and Kaess M (2016) Articulated robot motion for simultaneous localization and mapping (arm-slam). *IEEE robotics and automation letters* 1(2): 1156–1163.

Knapitsch A, Park J, Zhou QY and Koltun V (2017) Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics* 36(4).

Li J, Ito A and Maeda Y (2019) A slam-integrated kinematic calibration method for industrial manipulators with rgb-d cameras. In: *2019 19th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, pp. 686–689.

Li L, Zhou X, Hu Z and Xu Y (2024) Automatic robot hand-eye calibration enabled by visual sensing and motion compensation. *Journal of Intelligent & Robotic Systems* 110: 90.

Macario Barros A, Michel M, Moline Y, Corre G and Carrel F (2022) A comprehensive survey of visual slam algorithms.

Robotics 11(1): 24.

Martin-Brualla R, Radwan N, Sajjadi MS, Barron JT, Dosovitskiy A and Duckworth D (2021) Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 7210–7219.

Martins TB and Civera J (2024) Feature splatting for better novel view synthesis with low overlap. In: *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA, p. 19.

Matsuki H, Murai R, Kelly PH and Davison AJ (2024) Gaussian splatting slam. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 18039–18048.

Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R and Ng R (2021) Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* 65(1): 99–106.

Placed JA, Strader J, Carrillo H, Atanasov N, Indelman V, Carlone L and Castellanos JA (2023) A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics* 39(3): 1686–1705.

Ramezani M, Wang Y, Camurri M, Wisth D, Mattamala M and Fallon M (2020) The newer college dataset: Handheld lidar, inertial and vision with ground truth. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4353–4360.

Schonberger JL and Frahm JM (2016) Structure-from-motion revisited. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4104–4113.

Schubert D, Goll T, Demmel N, Usenko V, Stückler J and Cremers D (2018) The tum vi benchmark for evaluating visual-inertial odometry. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1680–1687.

Shi X, Li D, Zhao P, Tian Q, Tian Y, Long Q, Zhu C, Song J, Qiao F, Song L et al. (2020) Are we ready for service robots? the openloris-scene datasets for lifelong slam. In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 3139–3145.

Shiu Y and Ahmad S (1989) Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb. *IEEE Transactions on Robotics and Automation* 5(1): 16–29.

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

Sturm J, Engelhard N, Endres F, Burgard W and Cremers D (2012) A benchmark for the evaluation of rgb-d slam systems. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 573–580.

Süβ J, Schnaubelt M and Von Stryk O (2022) Multi-cam arm-slam: Robust multi-modal state estimation using truncated signed distance functions for mobile rescue robots. In: *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, pp. 293–299.

Sucar E, Liu S, Ortiz J and Davison AJ (2021) imap: Implicit mapping and positioning in real-time. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 6229–6238.

Tang X, Jiang K, Yang M, Liu Z, Jia P, Wijaya B, Wen T, Cui L and Yang D (2023) High-definition maps construction based on visual sensor: A comprehensive survey. *IEEE Transactions on Intelligent Vehicles* .

Tao Y, Muñoz-Bañón MÁ, Zhang L, Wang J, Fu LFT and Fallon M (2025) The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. *International Journal of Robotics Research* .

Teed Z and Deng J (2021) Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In: *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34. pp. 16558–16569.

Tosi F, Zhang Y, Gong Z, Sandström E, Mattoccia S, Oswald MR and Poggi M (2024) How nerfs and 3d gaussian splatting are reshaping slam: a survey. *arXiv preprint arXiv:2402.13255* 4: 1.

Wang Z, Bovik A, Sheikh H and Simoncelli E (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4): 600–612. DOI: 10.1109/TIP.2003.819861.

Ye S, Dong ZH, Hu Y, Wen YH and Liu YJ (2024) Gaussian in the dark: Real-time view synthesis from inconsistent dark images using gaussian splatting. In: *Computer Graphics Forum*, volume 43. Wiley Online Library, p. e15213.

Yeshwanth C, Liu YC, Nießner M and Dai A (2023) Scannet++: A high-fidelity dataset of 3d indoor scenes. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12–22.

Zhang J and Singh S (2014) Loam: Lidar odometry and mapping in real-time. In: *Robotics: Science and Systems (RSS)*. p. 9.

Zhang R, Isola P, Efros AA, Shechtman E and Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. p. 14.

Zhu Z, Peng S, Larsson V, Xu W, Bao H, Cui Z, Oswald MR and Pollefeys M (2022) Nice-slam: Neural implicit scalable encoding for slam. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12786–12796.