

# Measuring robustness of Visual SLAM

David Prokhorov  
ksetherg@gmail.com

Dmitry Zhukov  
d.zhukov@samsung.com

Olga Barinova  
o.barinova@samsung.com

Konushin Anton  
a.konushin@samsung.com

Anna Vorontsova  
a.vorontsova@samsung.com

Samsung AI Center, Moscow

## Abstract

*Simultaneous localisation and mapping (SLAM) is an essential component of robotic systems. In this work we perform a feasibility study of RGB-D SLAM for the task of indoor robot navigation. Recent visual SLAM methods, e.g. ORBSLAM2 [9], demonstrate really impressive accuracy, but the experiments in the papers are usually conducted on just a few sequences, that makes it difficult to reason about the robustness of the methods. Another problem is that all available RGB-D datasets contain the trajectories with very complex camera motions. In this work we extensively evaluate ORBSLAM2 to better understand the state-of-the-art. First, we conduct experiments on the popular publicly available datasets for RGB-D SLAM across the conventional metrics. We perform statistical analysis of the results and find correlations between the metrics and the attributes of the trajectories. Then, we introduce a new large and diverse HomeRobot dataset where we model the motions of a simple home robot. Our dataset is created using physically-based rendering with realistic lighting and contains the scenes composed by human designers. It includes thousands of sequences, that is two orders of magnitude greater than in previous works. We find that while in many cases the accuracy of SLAM is very good, the robustness is still an issue.*

## 1 Introduction

Simultaneous localization and mapping (SLAM) is the key component of many robotics, augmented reality and 3D scanning systems. The goal of visual SLAM is to estimate the camera poses and simultaneously reconstruct the environment using the images from a camera. For the case of indoor navigation RGB-D SLAM methods that work with 3D sensors [6, 12, 9] are most suitable since most robotic platforms are equipped with RGB-D cameras. We have tried different implementations of the recent SLAM methods and have chosen state-of-the-art ORBSLAM2 [9] due to its performance and stability of the implementation <sup>1</sup>.

<sup>1</sup>We have tried out all the methods from SLAMBench [2] in our experiments, but unfortunately the provided implementa-

Benchmarking of SLAM systems for indoor applications requires a dataset of image sequences with ground truth camera poses. Usually, motion capture is used to obtain camera poses for real-world image sequences [11]. Another option is to use photo-realistic 3D rendering [4, 8] that provides accurate camera poses. The most popular accuracy metrics for SLAM are Absolute Trajectory Error (ATE) and Relative Position Error (RPE) that measure the quality of localization.

Most of the research papers on RGB-D SLAM are compared by selected sequences from real world TUM RGB-D [11] or rendered ICL-NUIM [4] benchmarks and demonstrate really impressive accuracy in these experiments. However, only a small number of sequences is usually used in the experiments.

As far as we know this paper is the first effort to perform a large-scale statistical analysis of the accuracy of visual SLAM methods. Our contributions are two-fold. First, we evaluate ORBSLAM2 across all open benchmarks and analyze the correlations between the attributes of the sequences and the accuracy of SLAM. As a byproduct, we perform a degrading experiment with turning off the loop closure component of ORBSLAM2 and surprisingly find that loop closure does not significantly influence the robustness of the system. Second, we introduce a new large and diverse HomeRobot dataset of indoor sequences consisting of over a thousand trajectories. It targets indoor robotics applications and is much larger than ICL-NUIM and the open part of InteriorNet. We evaluate ORBSLAM2 on different modifications of the new dataset and extensively analyze the influence of the speed of agent on the accuracy of localization.

The structure of the paper is: in Section 2, we overview standard metrics for measuring the quality of SLAM algorithms. In Section 3, we describe the datasets that we used. In Section 4, we describe our experiments and present the analysis of obtained results. In Section 5, we discuss the results and conclude the paper.

tions were unstable, that did not allow us to perform large-scale experiments. In the end, we switched to the original implementation of ORBSLAM2 because it could better reproduce the results reported in the original paper [9]

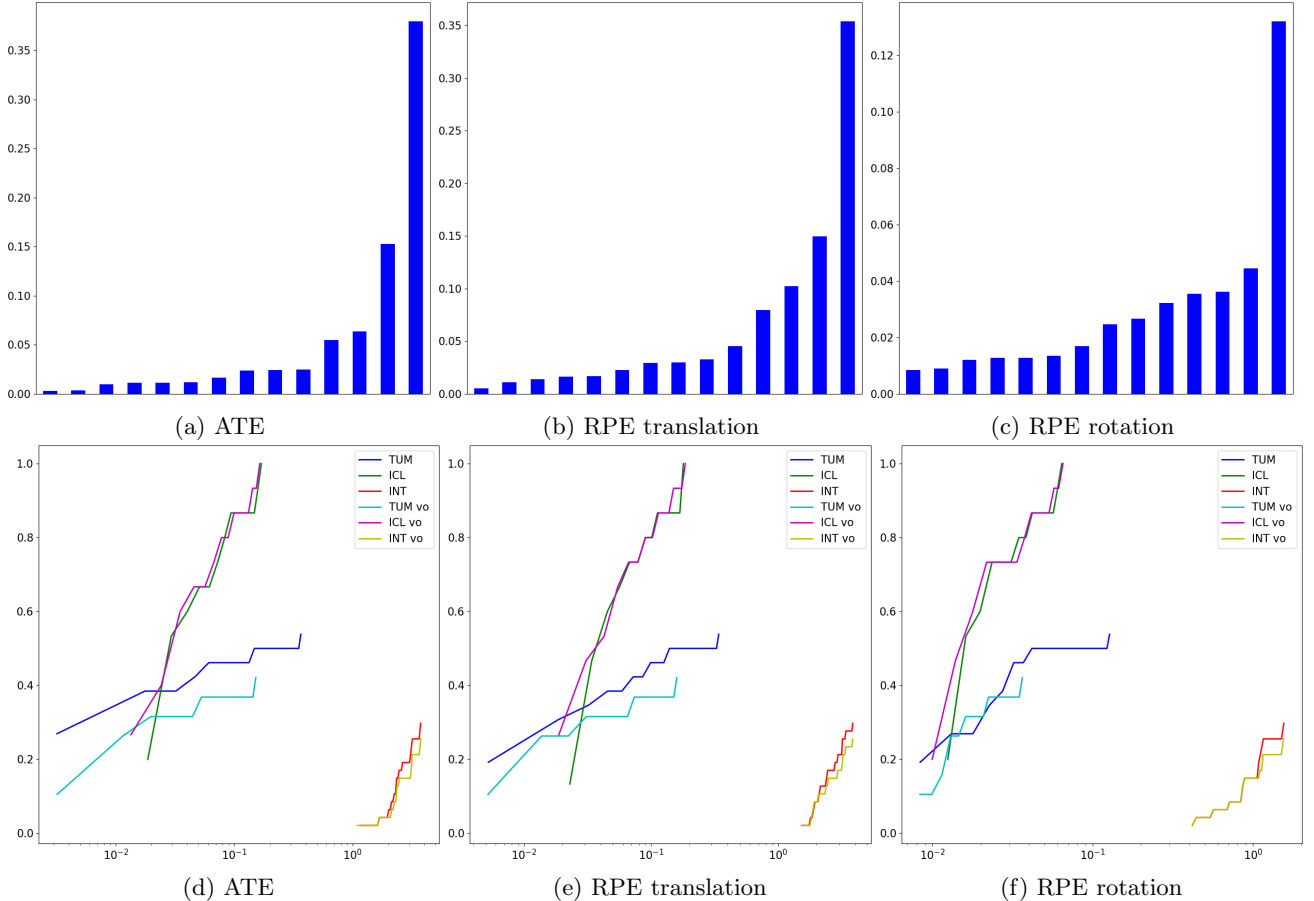


Figure 1: We have tested ORBSLAM2 with and without loop closure on 3 open datasets: ICL-NUIM [4], TUM RGB-D [11] and InteriorNet dataset [8]. Above are bar plots and cumulative plots in logarithmic scale. (a - c) - bar plots showing the sorted values of the metrics for all trajectories from TUM RGB-D where ORBSLAM2 successfully initialized and did not lose tracking. (d) - Average Trajectory Error (ATE), (e) - Relative Pose Error (RPE) translation, (f) - Relative Pose Error (RPE) rotation. One can see, that the simplest dataset is ICL-NUIM, and the hardest is InteriorNet. Below we perform analysis of the attributes of these datasets to find the reasons of such performance gap.

## 2 Localization accuracy

The methods for measuring accuracy of SLAM systems have evolved over a long period [7]. For now there is a number of metrics used in research community and sometimes their definitions in papers are misleading. Thus, below we recall the definitions of most popular ATE and RPE metrics.

The motion of a rigid body can be expressed as a sequence in  $SE_3$  which gives the transformation from the world to the body frame for each timestamp.  $SE_3$  is the group of all rigid transformations in  $\mathbb{R}^3$ . It is semi-direct product  $SO_3 \times \mathbb{R}^3$ , where  $SO_3$  is the group of all rotations and  $t \in \mathbb{R}^3$  is the 3D Euclidean space. Therefore it has 6 degrees of freedom: 3 for rotation and 3 for translation. The  $SE_3$  could be expressed as a

pair  $(R, t)$  where  $R \in SO_3$  and  $t \in \mathbb{R}^3$ .<sup>2</sup> Consequently the action of  $SE_3$  on vectors  $x \in \mathbb{R}^3$  is a combination of rotation and translation:

$$x' = Rx + t$$

Homogeneous coordinates are used to provide a matrix representation of the group. So it could be expressed as a  $4 \times 4$  matrix:

$$M = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

where  $R \in SO_3$  and  $t \in \mathbb{R}^3$ . It should be noticed that  $\forall R \in SO_3$  is  $R^{-1} = R^T$ . Note that orthogonal transformation is isometric. These advices will help

<sup>2</sup>More information about  $SE_3$  you can find in [3] and [1]

you to build an effective algorithm metrics calculation algorithm.

Next, we need three operators to define the metrics: let  $M \in SE_3$ , then

$$\begin{aligned} \text{trans}(M) &:= t \\ \text{rot}(M) &:= R \\ \angle R &:= \arccos\left(\frac{\text{tr}(R) - 1}{2}\right) \end{aligned}$$

For the evaluation, we assume there is a sequence of spatial poses from the estimated trajectory  $P_i \in SE_3$  and from the ground truth trajectory  $Q_i \in SE_3$  in arbitrary world coordinate system. For simplicity, we assume that the both sequences are time-synchronized, and both have length  $n$ . In practice, these two sequences have different sampling rates, lengths, potentially missing data and in some cases it is necessary to use relative poses, so that an additional data association and pre-processing is required.

## 2.1 Absolute trajectory error(ATE)

The global consistency can be evaluated by comparing the absolute distances between estimated and ground truth trajectory. As both trajectories can be specified in arbitrary coordinate frames, they first need to be aligned. This can be achieved by using the Horn method [5], which finds the rigid-body transformation  $S$ . Let us define absolute trajectory error matrix at time  $i$  as:

$$E_i := Q_i^{-1} S P_i$$

The ATE is defined as the root mean square error from error matrices:

$$\text{ATE}_{rmse} = \left( \frac{1}{n} \sum_{i=1}^n \|\text{trans}(E_i)\|^2 \right)^{\frac{1}{2}}$$

Usually, mean or median values are computed. Actually, absolute trajectory error is the average deviation from ground truth trajectory per frame.

## 2.2 Relative pose error(RPE)

The relative pose error measures the local accuracy of the trajectory over a fixed time interval  $\Delta$ . Therefore, the relative pose error corresponds to the drift of the trajectory which is in particular useful for the evaluation of visual odometry systems. Let us define the relative pose error matrix at time step  $i$  as:

$$F_i^\Delta := (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta})$$

from a sequence of  $n$  camera poses we obtain  $m = n - \Delta$  individual relative pose error matrices along the sequence. The RPE is usually divided into translation

Table 1: Characteristics of different datasets. M.vel.p.f - mean velocity per frame; m.ang.v.p.f - mean angular velocity per frame; m.frames - mean frames.

Dataset	m.vel.p.f.	m.ang.v.p.f.	m.frames
TUM	0.009	2.39	1424
ICL-NUIM	0.007	1.29	1160
InteriorNet	0.054	0.96	1000
HR Ds. skip 0	0.006	1.45	1000
HR Ds. skip 1	0.013	1.47	1000
HR Ds. skip 2	0.019	1.49	1000
HR Ds. skip 3	0.025	1.51	1000

and rotation components. Similar to the absolute trajectory error, we propose to evaluate the root mean squared error over all time indices for RPE translation error:

$$\text{RPE}_{trans}^{i,\Delta} = \left( \frac{1}{m} \sum_{i=1}^m \|\text{trans}(F_i)\|^2 \right)^{\frac{1}{2}}$$

As for rotation component we use mean error approach:

$$\text{RPE}_{rot}^{i,\Delta} = \frac{1}{m} \sum_{i=1}^m \angle(\text{rot}(F_i^\Delta))$$

For the evaluation of SLAM systems, it makes sense to average over all possible pairs in both translation and rotation component.

## 3 Datasets

**TUM RGB-D Benchmark Dataset** [11] is a large dataset containing RGB-D data and ground-truth camera poses. TUM dataset contains the RGB and Depth images of Microsoft Kinect sensor along the ground-truth trajectory of the sensor. The data was recorded at full frame rate (30 Hz) and sensor resolution  $640 \times 480$ . The ground-truth trajectory was obtained from a high-accuracy motion-capture system with eight high-speed tracking cameras (100 Hz). TUM dataset contains both a typical office environment (fr1,  $6 \times 6m^2$ ) and a large industrial hall (fr2,  $10 \times 12m^2$ ). The benchmark contains 27 sequences.

**ICL-NUIM RGB-D Benchmark Dataset** [4] is a synthetic dataset that consists of images obtained from camera in raytraced 3D models in POV-Ray for two different scenes: the living room and the office room. Images corresponding to different camera poses on a given trajectory are stored in typical RGB-D frame pairs while the camera poses are represented in  $SE_3$  format. All data is recorded at 30Hz speed. The benchmark contains 11 sequences.

**InteriorNet Photo-realistic Indoor Scenes Dataset** [8] is an end-to-end pipeline to render photo-realistic images and associated ground truth data.

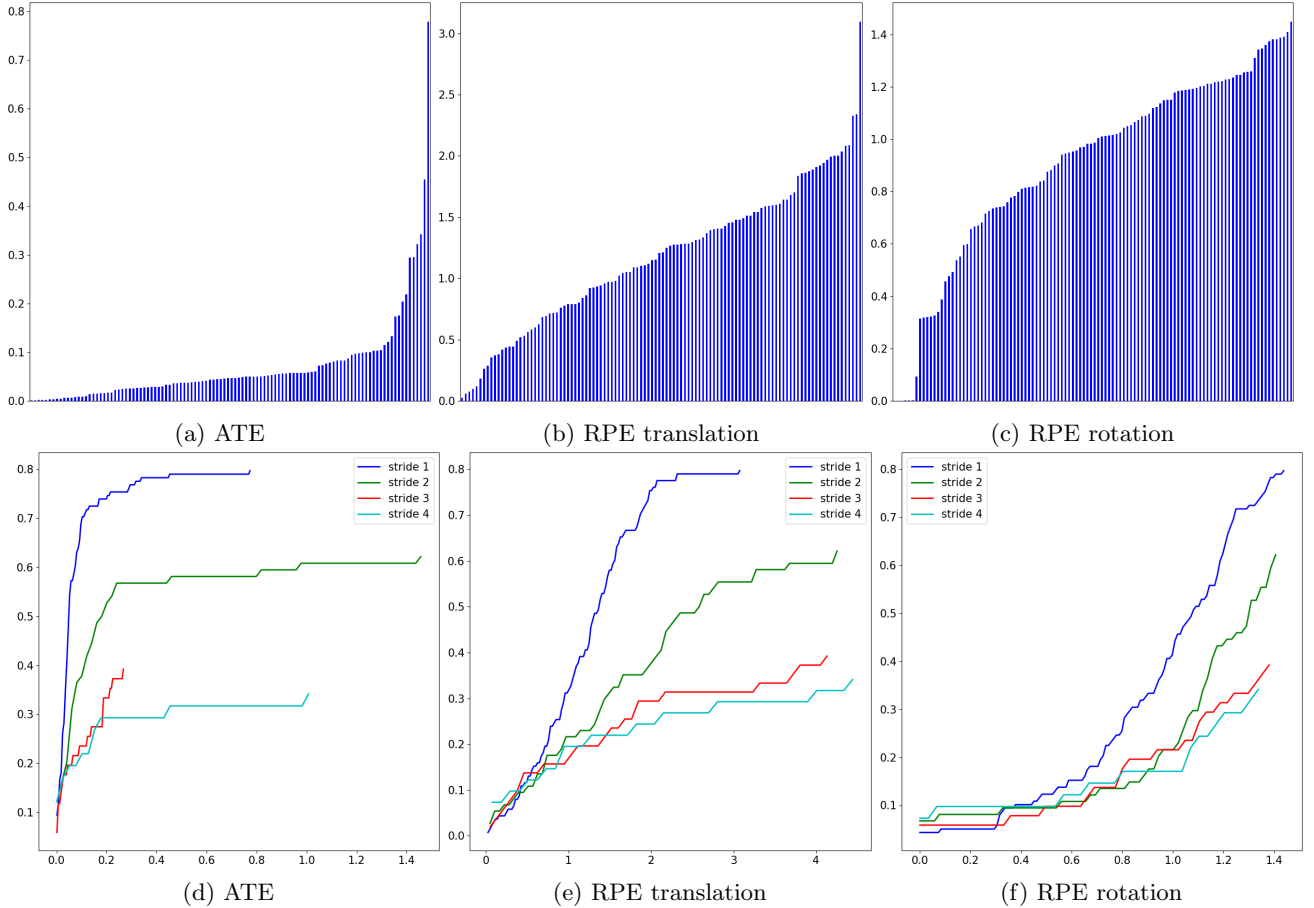


Figure 2: Above are the bar plots and cumulative distributions of the metrics for ORBSLAM2 on the new HomeRobot dataset. (a-c) - bar plots show the sorted values of the metrics for all trajectories from HomeRobot dataset where ORBSLAM2 successfully initialized and did not lose tracking. (d) - Average Trajectory Error (ATE), (e) - Relative Pose Error (RPE) translation, (f) - Relative Pose Error (RPE) rotation. We perform experiments on 4 versions of the dataset with different speed of agent’s movement. One can see the strong negative correlation between the velocity of the agent and the performance of ORBSLAM2.

We created multiple trajectories for each layout by varying the combination of linear velocity, angular velocity and trajectory types. Images support 640x480 resolution, at 25 Hz, resulting in 1,000 images per trajectory. Depth is obtained as the Euclidean distance of the ray intersection; and noisy depth is generated by simulating a real Kinect mechanism. The publicly available part of the dataset contains 40 sequences.

**HomeRobot Dataset** is a novel dataset that we have generated for the experiments. It contains long sequences sampled using a large 3D model repository for indoor scenes SUNCG [10]. We have generated randomized trajectories of the moving agent using the occupancy grids provided by the 3D models. The trajectories emulate the movements of a home robot with a camera mounted at a fixed height above the floor. Since the probability of failure of ORBSLAM2 is higher for long sequences, we have sliced the trajectories into

overlapping chunks of fixed length (we have chosen 1000 frames as a chunk length similar to the other RGB-D datasets) in order to enable more fair evaluation. We used our custom rendering engine with ambient occlusion technique that allowed us to get high quality images. The effects of the noise in depth estimation are modelled similarly to the ICL-NUIM. We have generated the sequences of HomeRobot dataset such that the speed of the agent is approximately the same as in TUM RGB-D. Overall, the new benchmark contains around 1000 sequences.

## 4 Experiments and analysis

**Experiments on the open benchmarks.** First, we proved our metrics measuring methods, we compared our results on popular trajectories with results from [9]. The results of the comparison are shown in

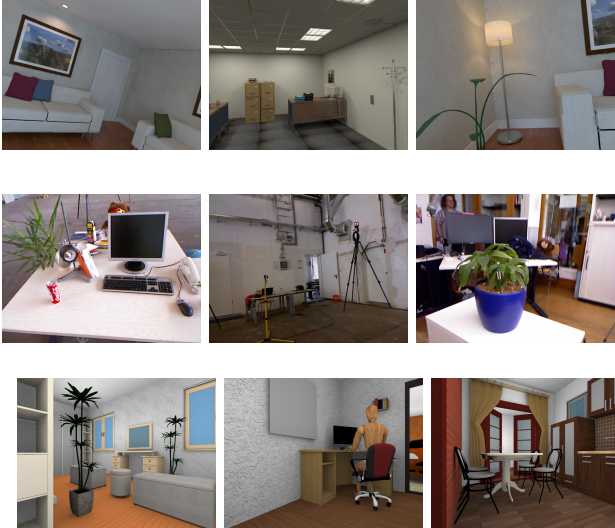


Figure 3: Examples of images and trajectories from ICL-NUIM, TUM and HomeRobot datasets.

Table 2: Comparison of our results with original results from [9] in terms of ATE.

Traj. name	Our ATE	Results from [9]
TUM fr1 xyz	0.010m	0.010m
TUM fr1 desk	0.016m	0.016m
TUM fr2 xyz	0.004m	0.004m
ICL of. tr1	0.053m	0.051m

the Table 2. Second, we perform extensive evaluation of ORBSLAM2 on the open benchmarks. We use the following evaluation protocol. Since ORBSLAM2 is randomized, each sequence in the dataset is run 10 times to find the median value for each metric. We build cumulative plots, which show the total number of trajectories, expressed in percentage, that have the metric value less than or equal to a series of thresholds.

The experiments have shown that there is a threshold value of the metrics indicating that the algorithm has failed on trajectory. There is a noticeable gap between the values of the metrics Figure 1. We have visualized and inspected the estimated trajectories and confirmed the existence of complexity gap. So there is a threshold value of metrics, corresponding to a dataset, from which ORBSLAM2 did not work properly. We assume that the threshold value depends on the dataset parameters (Table 1), such as velocity and angular velocity of agent. This assumption is confirmed by our degrading experiment on our new dataset Figure 2.

**Cumulative distribution analysis.** Figure 1 shows the cumulative plots for ICL-NUIM, TUM RGB-D, and InteriorNet datasets. Easy to see that ICL-

NUIM is simple dataset for ORBSLAM2, as InteriorNet is the hardest. The loop detection and relocalization did not give significant gain in accuracy Figure 1. The effect is visible only in hard cases, where method works badly with or without loop closure.

**Experiments of the HomeRobot dataset** We have used the same protocol for evaluation of ORBSLAM2 on the new HomeRobot dataset. We varied the speed of the agent by skipping frames in the sequences and built separate cumulative plots for each value of the stride parameter. Figure 2 shows the resulting cumulative plots for ATE, RPE rotation and RPE translation. Thus higher velocity of the agent results in lower accuracy of ORBSLAM2, due to reduction of the overlap between the consecutive frames.

## 5 Discussion and future work

In this paper we analyzed robustness of modern RGB-D SLAM methods on a large scale. We analyzed cumulative distributions of ATE and RPE metrics for renowned ORBSLAM2 system across popular open benchmarks including more than a hundred trajectories in total. Our experiments show that while accuracy on some sequences is high, the robustness of SLAM to changing conditions is an overlooked issue.

We also introduce a new HomeRobot dataset that is designed for indoor robot navigation tasks. The dataset and the source code for reproducing our experiments will be publicly available at the time of publication. We also plan to generate larger datasets of synthetic data for training and validation of the learning-based visual SLAM methods. We believe our results would be of value as a framework for future research and developing new methods for visual SLAM.

## References

- [1] Jose-Luis Blanco. A tutorial on se (3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep*, 3, 2010.
- [2] Bruno Bodin, Harry Wagstaff, Sajad Saecdi, Luigi Nardi, Emanuele Vespa, John Mawer, Andy Nisbet, Mikel Luján, Steve Furber, Andrew J Davison, et al. Slam-bench2: Multi-objective head-to-head benchmarking for visual slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.
- [3] Ethan Eade. Lie groups for 2d and 3d transformations. URL <http://ethaneade.com/lie.pdf>, revised Dec, 2013.
- [4] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Robotics and automation (ICRA), 2014 IEEE international conference on*, pages 1524–1531. IEEE, 2014.
- [5] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.
- [6] Christian Kerl, Jorg Stuckler, and Daniel Cremers. Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In *Proceedings of the IEEE international conference on computer vision*, pages 2264–2272, 2015.
- [7] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. On measuring the accuracy of slam algorithms. *Autonomous Robots*, 27(4):387, 2009.
- [8] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716*, 2018.
- [9] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
- [10] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [11] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.
- [12] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elastic-fusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.