Robust SLAM Systems: Are We There Yet?

Robust SLAM Systems: Are We There Yet?

Mihai Bujanca[1], Xuesong Shi[2], Matthew Spear[1], Pengpeng Zhao[2,3], Barry Lennox[1], Mikel Luján[1]

Fig. 1: Sample frames exhibiting challenging perturbations; *e.g.* occlusions and dynamically-moving elements; frames from drone sequences containing motion blur and no reliable features; frames containing lighting differences, blur, and dynamic objects.

***Abstract*— Progress in the last decade has brought about significant improvements in the accuracy and speed of SLAM systems, broadening their mapping capabilities. Despite these advancements, long-term operation remains a major challenge, primarily due to the wide spectrum of perturbations robotic systems may encounter.**

Increasing the robustness of SLAM algorithms is an ongoing effort, however it usually addresses a specific perturbation. Generalisation of robustness across a large variety of challenging scenarios is not well-studied nor understood. This paper presents a systematic evaluation of the robustness of open-source state-of-the-art SLAM algorithms with respect to challenging conditions such as fast motion, non-uniform illumination, and dynamic scenes. The experiments are performed with perturbations present both independently of each other, as well as in combination in long-term deployment settings in unconstrained environments (*lifelong operation*).

The detailed results (approx. 20,000 experiments) along with comprehensive documentation of the benchmarking tool for integrating new datasets and evaluating SLAM algorithms not studied in this work are available at https://robustslam.github.io/evaluation.

I. Introduction

SLAM algorithms are an essential component of embodied AI systems, providing a fundamental infrastructure necessary for navigation and other high-level tasks. The progress of SLAM systems during the last three decades has been remarkable, improving both the localisation and the mapping capabilities. While initially only very small spaces such as table tops or small rooms could be mapped, today's SLAM algorithms can operate on large scales [12], [13]. Thanks to advancements in computing hardware, sensors, and machine learning, SLAM has also extended well beyond the initial landmark-based mapping, leading to dense 3D reconstruction, non-rigid 3D reconstruction, and semantic mapping. The localisation accuracy of SLAM systems has also improved dramatically: the top 40 submissions on the KITTI odometry benchmark [14] have errors below 1%.

[1] The University of Manchester, Manchester, UK
[2] Intel Labs China, Beijing, China
[3] Beihang University, Beijing, China

Thanks to these advancements, SLAM has enabled new applications and while opportunities for further improvement remain ahead, robustness is widely regarded as today's most difficult challenge [15]. We define *robustness* as the capacity of a system to avoid fatal failures either by continuously performing accurately, or by detecting and quickly recovering from soft failures. A *fatal failure* is any failure that renders a system unable to perform its duties without external intervention, and is most commonly caused by environmental perturbations such as noise, dim or bright lighting, blurred frames, as well as short or long-term scene changes (*e.g.* dynamic objects). While some use cases only require episodic or short-term operation, many applications call for long-term deployment: home maintenance, autonomous inspection of industrial facilities, and so on. In the context of robot navigation, we refer to such long-term operation as *Lifelong SLAM*. Given the current capabilities and performance described above, we believe that the success of Lifelong SLAM is primarily dependent on the capacity of a system to be generally robust with respect to perturbations which may not be known a priori.

Previous efforts in evaluating the robustness of multiple SLAM systems have focused on specific types of perturbations [3], [16], often limited to a specific sensing modality [17], [18], without considering whether building in resilience against specific perturbations may incur any trade-offs with respect to other challenging factors or measuring the general robustness of the system. Our work addresses this gap by introducing an evaluation methodology for assessing the robustness of SLAM solutions supporting various sensing modalities and degrees of freedom, in the presence of a variety of perturbations evaluated independently as well as in combination. We demonstrate the validity of our approach by performing an extensive evaluation of 6 SLAM systems (Table II) on 6 datasets (Table I) across 3 computing platforms, in both episodic and long-term operation settings. Figure 1 contains a selection of frames with occlusions and dynamically-moving elements, illumination changes in real and synthetic scenes, frames from drone sequences containing motion blur and no reliable features, lifelong operation

Name	Sensors	Perturbations	Platform	Year
OpenLORIS [1]	RGB-D, Stereo, LiDAR IMU, Wheel odometry	Dynamic movement, sensor degradation, changed viewpoints and objects, illumination	Ground robot	2020
BONN Dynamic [2]	RGB-D	Dynamic movement	Handheld	2019
ETHI [3]	RGB-D	Illumination changes	Synthetic, handheld	2017
EuRoC MAV [4]	Stereo, IMU	Baseline dataset; Motion blur	Drone	2016
ICL-NUIM [5]	RGB-D	Baseline dataset	Synthetic	2014
TUM RGB-D [6]	RGB-D	Baseline dataset; Dynamic objects,	Handheld	2012

TABLE I: Datasets used in the evaluation.

Algorithm	Type	Sensors	Processing	Year
OpenVINS [7]	Sparse	Stereo, IMU	CPU	2020
ORB-SLAM3 [8]	Sparse	RGB-D, Stereo, Monocular, IMU	CPU	2020
FullFusion [9]	Dense, non-rigid, semantic	RGB-D	GPU	2019
ReFusion [2]	Dense	RGB-D	GPU	2019
ORB-SLAM2 [10]	Sparse	RGB-D, Stereo, Monocular	CPU	2016
ElasticFusion [11]	Dense	RGB-D	GPU	2015

TABLE II: SLAM systems evaluated.

challenges, colour frames containing lighting differences, blur, and dynamic objects. The accompanying video shows a qualitative comparison of 4 algorithms running on a sequence with dynamic elements.

II. Related Work

While the problem of robustness has been acknowledged since the early days of SLAM [15], [19], [20], it remains one of the most significant challenges. We briefly review the literature on SLAM robustness with respect to perturbations relevant to our work.

Illumination changes may occur due to natural (*e.g.* varying sunlight) or artificial causes (*e.g.* blinking lightbulbs), translating into sudden changes in image brightness, either locally or globally. A large number of works rely on brightness constancy for mapping [11], [21]–[23], and may be negatively affected by such changes. Methods to improve robustness to illumination changes include active exposure control [24]–[26], binary local descriptors for brightness normalization such as *Census transform* [27], [28], while other works developed illumination-invariant metrics to register images [29], [30]. A detailed evaluation of the performance of direct methods under such perturbations is presented in [3], whose dataset we adopt.

Dynamic elements are one of the most widely encountered type of perturbation: virtually all settings where SLAM is employed, from home robots to autonomous vehicles to augmented reality are bound to feature movement. Over the years, a number of solutions have been proposed [31]. Given that the static part of a scene provides the most reliable information for computing the camera pose, many approaches to dynamic SLAM attempt to segment the input into static and dynamic parts. Methods include the use of optical flow [32]–[34], geometric constraints [35], alignment residuals [2], and semantic information [9], [36]–[40].

Fast camera movement on robots and drones often results in motion blur, hindering both feature detection and direct alignment, methods widely employed by SLAM systems. [41]–[43] use frame deblurring to ensure reliable features can be identified; FLaME [44] proposes to use low quality but high frequency depth estimation to aid obstacle avoidance in drone flight.

Lifelong SLAM and long-term localisation are long-standing problems [45]–[48]. In the past year, new benchmarks challenging the state-of-the-art have appeared [1], [49], and promising results (usually based on detecting learned features) have been proposed for localisation [50]–[52] as well as SLAM [53]. We reuse the dataset and metrics of the Lifelong Robotic Vision challenge in our evaluation [1].

We aim for a comprehensive evaluation of the robustness of SLAM systems, but recognise that other factors, such as weather [54], [55] or limited visibility [56] are also of practical importance. Our methodology should help evaluate such perturbations in the future, as well as other factors (e.g. 3D reconstruction, semantic labelling).

III. Methodology

A. Evaluation workflow

We design our pipeline to support single and multi-sequence inputs and use the latter for Lifelong SLAM evaluation. Our evaluation pipeline adopts and extends the tools for trajectory alignment, visualisation, and metric computation provided by the open-source SLAMBench framework[1] [57], [58]. The software containing routines for configuring and initialising each system, streaming data into the algorithm and collecting outputs (estimated pose and monitoring the state of the system) will be made public. Importantly, preparing an algorithm for evaluation only involves writing a thin wrapper around each algorithm and does not require modifying the code of the system.

To assess the accuracy of each algorithm, we use the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) introduced in the TUM RGB-D [6] dataset. In contrast to most evaluation procedures where the alignment and computation of the metrics is done only using the final trajectory, we continuously monitor the ATE and RPE by realigning the trajectories in $SE(3)$ using Umeyama's method [59] and measuring the errors every time the SLAM system outputs a new pose. To prevent algorithms from dropping frames, new data is sent *after* the previous frame finished processing.

Figure 5 shows spikes in error correlated with the input data causing them, allowing us to deduce the particular sensitivities of individual algorithms, as well as to identify scenes

[1]Code available at https://github.com/pamela-project/slambench

which are generally challenging for SLAM algorithms. Since performing these routines for every frame can be expensive and could affect measurements, we use existing mechanisms in SLAMBench to report execution times and resource usage by the SLAM algorithms independently of evaluation and trajectory alignment.

B. Lifelong SLAM

Evaluating Lifelong SLAM entails simulating common long-term operation scenarios. Each algorithm is fed multiple sequences captured in the same environment, with aspects such as initial position, time of day, lighting, and so on, varying across sequences.

In addition to computing the per-sequence ATE and RPE, the metric Correct Rate of Tracking (CRT) is adopted [1]. Environmental perturbations may cause SLAM algorithms to lose tracking. The ATE may be unevenly affected by a loss of tracking; e.g. losing tracking in the late stages of a sequence could have a significant impact on the Mean ATE. On the other hand, the CRT metric measures the ratio of correct tracking time with respect to the whole time span of the data. Correctness of each estimated pose can be determined with user-specified thresholds of ATE and other per-frame metrics. By combining the two metrics, we can better capture the overall performance, tracking failures, as well as the time spent correctly tracking the camera pose.

IV. Experiments

A. Experimental setup

We perform the experiments on the following three hardware platforms (*Workstation*, *Laptop*, and *Jetson*), running under 64-bit Ubuntu 18.04 OS:

The workstation is a desktop with 32 GB of RAM, a 14-core Intel Core i9-9940X chip (3.30GHz), and an Nvidia TITAN RTX GPU with 24GB VRAM and 4608 CUDA cores. The laptop is a Lenovo ThinkPad P53 with 16GB of RAM, a 6-core Intel Core i7-9850H (2.60GHz), and an Nvidia Quadro RTX 3000 with 1920 CUDA cores and 6GB of VRAM. The Jetson is an Nvidia Jetson Xavier AGX. This is a platform commonly used in ground robots, featuring a 8-core ARMv8.2 64-bit CPU (2.25GHz), 16 GB of RAM, and a 512-core Nvidia Volta GPU. The device is set up to deliver the maximum performance, with a peak power use of 30W.

To control for any differences not inherent to the algorithms, we ensure that, wherever possible, on each platform any common dependencies undertaking significant computational tasks, such as *OpenCV* or *g2o*, are fixed to the same version across all the SLAM systems evaluated. We use *gcc 7* for compilation across all algorithms and platforms, and CUDA 10.2 for GPU-based implementations. The DVFS of the processing cores (Turboboost) and GPU (adaptive clocking) are disabled. All the build processes have been modified to use the highest levels of compiler optimisation. The hyperparameters of SLAM systems are configured following the recommendations of the original papers/repositories, if available, or otherwise using the default settings.

Using the appropriate input modalities provided by each dataset (Table I), we evaluate 6 open-source SLAM systems selected to cover a diversity of designs with respect to input modalities and map representations.

OpenVINS [7] is a stereo visual-inertial SLAM system which uses an Extended Kalman Filter to fuse visual odometry with inertial measurements.

ORB-SLAM2 [10] is a popular real-time SLAM system based on sparse ORB features. It incorporates RGB-D, monocular and stereoscopic input modalities.

ORB-SLAM3 [8] is a recently released SLAM system developed on top of ORB-SLAM2 which introduces a multiple map system and visual-inertial odometry to improve robustness.

ElasticFusion [11] provides a globally-consistent dense RGB-D reconstruction approach that does not require a pose graph and represents the map using fused surfels [60].

FullFusion [9] is a framework for semantic reconstruction of dynamic scenes. FullFusion leverages semantic information to separate RGB-D inputs into a static and a dynamic frame. A modified implementation of KinectFusion is used to compute the pose and reconstruct a semantically labelled model of the static scene elements.

ReFusion [2] is a dense RGB-D 3D reconstruction method which exploits residuals obtained after the registration of input data with the reconstructed model to identify and filter out dynamic elements in the scene.

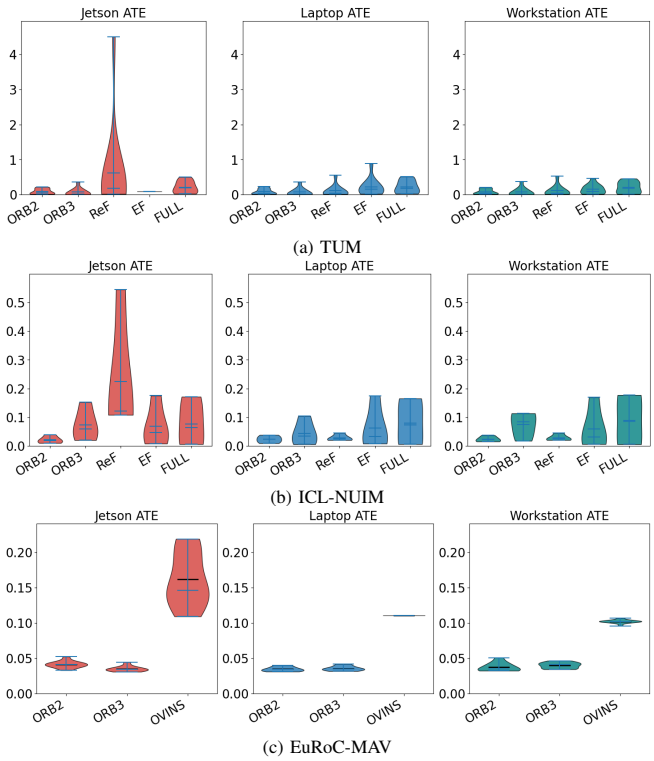
The datasets have been chosen to cover a wide range of conditions common in SLAM applications:

- Camera motion / hardware platform: ground robot, aerial vehicle, handheld sensor, linear motion (in synthetic scenes).
- Scene type: synthetic, indoor, outdoor, empty corridors, busy market or cafe.
- Lighting: sudden exposure changes, daylight, night, continuously changing local and global illumination, flashlights.
- Movement: varying levels of movement, both rigid and non-rigid. All combinations of static and moving camera with static and moving scenes.
- Sensors: RGB-D, Stereo cameras, IMU, Wheel odometry, Sequences featuring sensor degradation.

An experiment refers to the execution combining one SLAM system (Table II) and one sequence of a given dataset (Table I). Each experiment is performed 10 times on each of the 3 platforms generating a total of approximately 20000 data points.

Given the large number of experiments and the necessity to differentiate by platform and perturbation, the paper contains only a subset of the results and metrics. Full data is available on the website[2]. We adopt the following strategy to present aggregated data under each setting: for each sequence, we compute the median of the translational ATE-RMSE over the 10 runs, normalised by the metric length of the sequence to ensure equal weighting across sequences. Note that the

[2] https://robustslam.github.io/evaluation



(a) TUM

(b) ICL-NUIM

(c) EuRoC-MAV

Fig. 3: Baseline performance results.

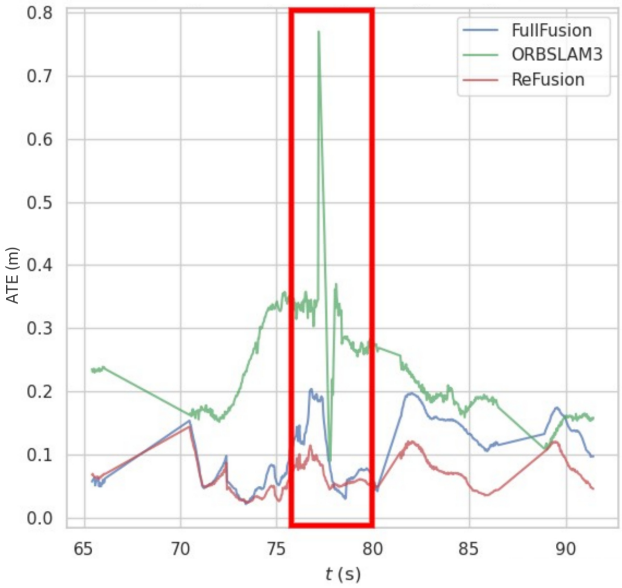


Fig. 4: ReFusion, ORB-SLAM3 and FullFusion executing the *moving_nonobstructing_box* sequence of the Bonn dataset. The red rectangle highlights the period of time when a person enters the scene, moves a box and leaves.

aggregate plots may not always be representative of the performance on individual sequences.

B. Results

Baseline performance – We evaluate the trajectory estimation accuracy of each SLAM system on selected sequences of widely-adopted datasets where no significant perturbations are present. The RGB-D based SLAM systems are evaluated with 12 sequences from the TUM *freiburg1* and *freiburg2* datasets [6] and the 4 sequences of the ICL-NUIM *living room* dataset [5]. Our results (Figures 3-a and 3-b) are consistent with the existing literature. ORB-SLAM2, ORB-SLAM3 and ElasticFusion are accurate within 1% on all sequences and no individual runs exceeded 3% error. FullFusion and ReFusion maintained their ATE below 3% on most runs, but scored worse than the aforementioned systems (with few exceptions). ORB-SLAM3 is the most accurate in this baseline setting, with ORB-SLAM2 closely after.

SLAM systems supporting stereo and visual-inertial SLAM are evaluated on the 7 *easy* and *medium* sequences of the EuRoC-MAV dataset. Figure 3-c shows all 3 SLAM systems have similar accuracies and performed within a 0.5% error margin. Unexpectedly, on some of the *machine hall* sequences, ORB-SLAM3, using the stereo VIO mode, performed slightly worse than ORB-SLAM2 in stereo mode.

Illumination changes – We use the ETH Illumination dataset to analyse the resilience of SLAM systems using 3 real and 10 synthetic RGB-D sequences. The dataset features multiple types of illumination change: local, global, local and global, and flashlight. The real sequences are captured with handheld Kinect v1 sensor, in an environment closely resembling the TUM RGB-D setting. The synthetic scenes are adapted from the ICL-NUIM dataset. Thanks to the illumination invariance of ORB features, both ORB-SLAM2 and ORB-SLAM3 appear to be unaffected by any type of illumination change, obtaining similar scores to the baseline TUM and ICL-NUIM. In contrast, ElasticFusion and ReFusion use photometric errors which assume constant illumination, leading to high error rates. Figure 5 highlights the effects of changes in illumination on ReFusion.

Dynamic elements — We use the 24 dynamic sequences in the Bonn RGB-D Dataset. All scenes were captured in the same space and include people handling objects such as boxes and balloons. Varied levels of movement are present, ranging from mostly static scenes to complete occlusion of the background by moving objects for extended periods.

Having been published together with the Bonn dataset, ReFusion performs best in the presence of dynamic elements. Nonetheless, ORB-SLAM2 and ORB-SLAM3 perform more accurately than ReFusion on scenes with negligible movement or when the dynamic elements are untextured, relying mostly on background keypoints, and are able to recover when dynamic objects briefly enter and leave the frame, but fail under severe motion. Figure 4 highlights the moderate increase in error in ReFusion and FullFusion compared to a significant spike in error for ORB-SLAM3. FullFusion's segmentation module relies on semantics to remove dynamic objects from frames. As such, FullFusion performs well only when recognized classes are present in the scene (*person*), but is highly sensitive to any other movement, often experiencing failures (*balloon, box* sequences), unlike more general algorithms such as ReFusion. As expected, due to using all

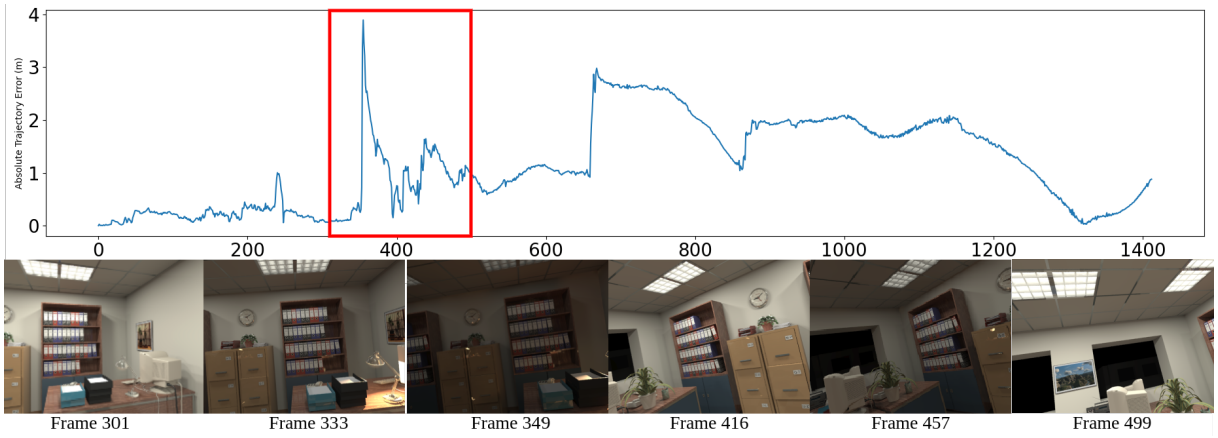


Fig. 5: Single run of ReFusion on the *syn2* sequence with both local and global illumination changes. Significant spikes in error occur during changes in illumination (top). Bottom: frames corresponding to the highlighted area.

the data in the frame and assuming only camera movement, ElasticFusion is severely affected, with a noticeable drop in accuracy occurring as soon as a dynamic object enters the scene, without subsequent recovery, and fails entirely on the highly dynamic scenes.

Lifelong SLAM — OpenLORIS-Scene is a comprehensive dataset featuring a total of 22 sequences captured in 5 common environments (office, corridor, home, cafe, and market) at different times of the day using commercial service robots. Compared to most SLAM datasets which often present tightly controlled scenarios, OpenLORIS contains realistic settings for service robots, and a wide variety of challenging factors: occlusions, dynamic motion, featureless areas, and lighting changes.

OpenLORIS is the most challenging of the datasets. Figure 7 illustrates ATE for a subset of the sequences evaluated. Figure 8 illustrates the CRT metric for all the sequences. Thus we can observe, for example, that for the sequence cafe2 although the ATE may be less than 1 meter for ReFusion, ElasticFusion and FullFusion, their CRT illustrates significant portions of frames where the error was larger than 3 meters. ORB-SLAM2 and ORB-SLAM3 are severely affected in textureless environments. In particular, most of the *corridor* and *home* sequences disproportionately affect sparse algorithms. ReFusion performed well in the presence of dynamic objects as long as they moved in a consistent fashion. However on the *market* sequences, where persons often moved and stopped, artefacts were produced in the reconstruction, impacting the pose estimation accuracy. FullFusion performs well when it is able to recognise dynamic objects, but tends to drift whenever unknown objects enter the scene.

C. Other Observations

In assessing the robustness of a SLAM system, one should consider not only variation across perturbations, but also matters of portability, setup, ease of use, consistency, and operation in previously untested environments.

Setup and execution — ORB-SLAM2 and ORB-SLAM3 produced hard crashes (segfault) more than 10% of the time

	EF	FULL	ReF	OVINS	OS2	OS3
Jetson	40	25	0.2	10	5	5
Laptop	50	30	10	17.5	5	10
Workstation	50	150	15	20	10	12

TABLE III: Average frame rate for each SLAM algorithm.

across all platforms, requiring frequent restarts. Additionally, their reliance on old dependencies made it hard to identify working versions across all algorithms. Discrepancies in performance across platforms may relate to different versions of these dependencies. OpenVINS is highly sensitive to correct initial parameters, which may not always be available in deployment. We were not able to find working hyperparameters for the OpenLORIS dataset. FullFusion attempts to compute dynamic masks whether or not there are any dynamic objects in the scene, resulting in slightly lower accuracy as well as up to 80% lower frame rate on compute-constrained platforms compared to disabling masks on sequences known to be static. ReFusion sees a drastic drop in frame rate to 0.2 FPS on the *Jetson* from 10-15 FPS on *Laptop* and *Workstation*.

Consistency — While overall we have found no major discrepancies between the results on each platform, ORB-SLAM2 and ORB-SLAM3 were the least consistent across runs and across platforms, with some variability observed in other SLAM systems (see Figure 6). At the other end, OpenVINS performed almost identically across all runs for any sequence on a given platform.

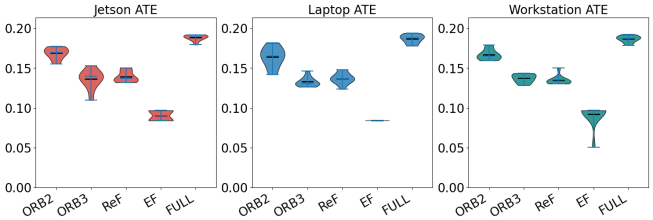


Fig. 6: Consistency evaluation using 30 runs on (*fr1_360*).

D. Summary of Results

Table IV provides a summary of the results presented. The second to fourth columns show the dense SLAM systems. For the accuracy on baseline datasets (TUM, ICL-NUIM, EuRoC-MAV), no SLAM system is classified as

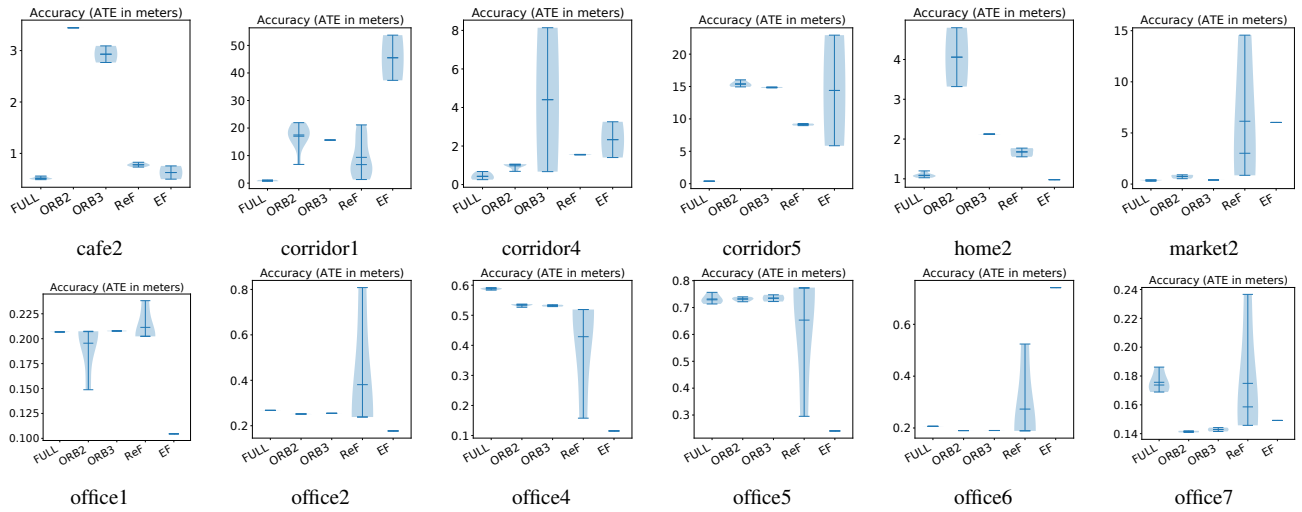


Fig. 7: OpenLORIS – accuracy results for a subset of scenes.

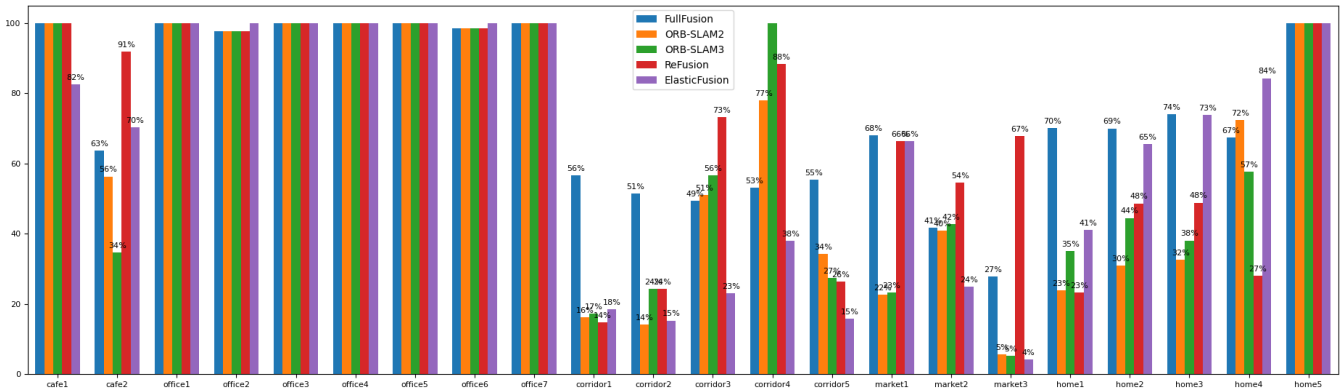


Fig. 8: *Correct Rate of Tracking* displayed as percentage of frames within an absolute error threshold on OpenLORIS sequences. The threshold values suggested in the dataset: $MAX\_ATE = 1m$ for *office*, $3m$ for *home* and *cafe*, and $5m$ for *corridor* and *market* sequences.

Excellent because, although their ATEs show good accuracy, they all have sequences where accuracy problems occur. ElasticFusion is fast and accurate when no perturbations are present but generally not robust due to the photometric error assuming fixed coefficients for the RGB channels. ReFusion is not very accurate on the baseline datasets and is sensitive to illumination changes due to the photometric error similar to ElasticFusion, however it is robust to dynamic objects. FullFusion is robust to illumination because it only uses depth data for mapping, but this can be a disadvantage in structureless areas. FullFusion has proven sensitive to unrecognized dynamic objects.

The fifth to seventh columns present the sparse SLAM systems. OpenVINS is the most consistent across runs and across platforms, and attains high accuracy on drone sequences, but is strictly visual-inertial and could not be tested on datasets without IMU data. ORB-SLAM2 and ORB-SLAM3 cover the broadest variety of input modalities. Consistent with the published ORB-SLAM3 [8] results, but using different datasets, we have found that the addition of a VIO mode over ORB-SLAM2 and the multi-map merging scheme improves robustness against temporary tracking loss (usually caused by dynamic objects or fast movement).

We use 5 categories to qualitatively describe the results. *Excellent* means almost perfect - the system performs similarly with or without perturbations. This is only awarded in Illumination where FullFusion does not use color data and ORB-SLAM performs well even in low light. The category *Very Good* covers the vast majority of perturbations and does not fail even when perturbations are significant. Nonetheless, a SLAM system may fail when encountering severe perturbations for a long period. If short term failures are encountered, it is often able to recover. The *Good* category captures mostly robust outcomes. A good example is FullFusion which has robustness against a set number of classes, but may fail when encountering unknown classes. The category *Acceptable* captures SLAM systems with some robustness, which can deal with perturbations for a short amount of time. For example, ORB-SLAM3 can recover well in the presence of dynamic objects, if they are encountered for a couple of frames or occupy only a small portion of the frame, but will fail otherwise.

V. Conclusions

This paper has presented a systematic evaluation of the robustness of 6 open-source state-of-the-art SLAM algorithms with respect to challenging conditions, such as fast

	ElasticFusion	FullFusion	ReFusion	OpenVINS	ORB-SLAM2	ORB-SLAM3
Baseline Accuracy	Very good	Good	Good	Very good	Very good	Very good
Illumination	Not robust	Excellent*	Not robust	No data	Excellent	Excellent
Dynamic	Not robust	Good	Very good	No data	Not robust	Acceptable
Fast	No data	No data	No data	Very good	Acceptable	Very good
OpenLORIS (Combined)	Not robust	Acceptable	Acceptable	No data	Acceptable	Good

*FullFusion is not impacted by illumination changes as it does not use color information.

TABLE IV: Overall robustness – A qualitative summary of the experiments.

motion, non-uniform illumination, and dynamic scenes. The experiments have covered 6 datasets across 3 computing platforms, in both episodic and long-term operation settings. Thus, this evaluation is the most comprehensive study of the robustness of SLAM systems to date. By including the Nvidia Jetson Xavier platform, we also consider constraints associated with deployments on systems embedded within robots.

Overall, we have found that ORB-SLAM3 provides the best balance between baseline accuracy, illumination and fast changes, support for dynamic environments and Lifelong scenarios, although its FPS is below 15 (5 FPS on Jetson). Considering the three dense SLAM systems, FullFusion provides the best balance, but reaches 30 FPS only on the laptop and workstation (Jetson 25 FPS). ElasticFusion offers between 40-50 FPS processing on the three platforms, but its robustness falls below the other SLAM systems.

Finally, the sparse SLAM systems have proved more robust than the dense ones, probably because there are fewer data points which can negatively impact pose estimation. We consider that combining sparse tracking with dense 3D reconstruction will help systems build expressive representations while maintaining high robustness.

ACKNOWLEDGMENTS

This research is supported by the EPSRC, grant RAIN Hub EP/R026084/1. Mikel Luján is supported by an Arm/RAEng Research Chair Award and a Royal Society Wolfson Fellowship. Thanks to Patrick Geneva for assisting with experiments on OpenVINS. Thanks to all researchers who provided the datasets.

REFERENCES

[1] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song *et al.*, "Are we ready for service robots? the openloris-scene datasets for lifelong slam," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3139–3145.

[2] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss, "Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7855–7862.

[3] S. Park, T. Schöps, and M. Pollefeys, "Illumination change robustness in direct visual slam," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[4] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract

[5] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014, pp. 1524–1531.

[6] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012.

[7] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[8] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *arXiv preprint arXiv:2007.11898*, 2020.

[9] M. Bujanca, M. Luján, and B. Lennox, "Fullfusion: A framework for semantic reconstruction of dynamic scenes," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

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

[11] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "Elasticfusion: Dense slam without a pose graph," *Proc. Robotics: Science and Systems, Rome, Italy*, 2015.

[12] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the atlas framework," *The International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, 2004.

[13] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization." in *Robotics: Science and Systems, vol. 1*, 2015.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

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

[16] O. Roesler and V. P. Ravindranath, "Evaluation of slam algorithms for highly dynamic environments," in *Robot 2019: Fourth Iberian Robotics Conference*, M. F. Silva, J. Luís Lima, L. P. Reis, A. Sanfeliu, and D. Tardioli, Eds. Cham: Springer International Publishing, 2020, pp. 28–36.

[17] J. Lomps, A. Lind, and A. Hadachi, "Evaluation of the robustness of visual slam methods in different environments," *arXiv preprint arXiv:2009.05427*, 2020.

[18] D. Prokhorov, D. Zhukov, O. Barinova, K. Anton, and A. Vorontsova, "Measuring robustness of visual slam," in *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019, pp. 1–6.

[19] J. B. Folkesson and H. I. Christensen, "Robust slam," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 722–727, 2004.

[20] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 4372–4378.

[21] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 2100–2106.

[22] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 15–22.

[23] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[24] Z. Zhang, C. Forster, and D. Scaramuzza, "Active exposure control for robust visual odometry in hdr environments," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3894–3901.

[25] I. Shim, T.-H. Oh, J.-Y. Lee, J. Choi, D.-G. Choi, and I. S. Kweon, "Gradient-based camera exposure control for outdoor mobile platforms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1569–1583, 2018.

[26] P. Kim, B. Coltin, O. Alexandrov, and H. J. Kim, "Robust visual localization in changing lighting conditions," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5447–5452.

[27] H. Alismail, B. Browning, and S. Lucey, "Direct visual odometry using bit-planes," *arXiv preprint arXiv:1604.00990*, 2016.

[28] H. Alismail, M. Kaess, B. Browning, and S. Lucey, "Direct visual odometry in low light using binary descriptors," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 444–451, 2016.

[29] G. Pascoe, W. Maddern, M. Tanner, P. Piniés, and P. Newman, "Nid-slam: Robust monocular slam using normalised information distance," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1435–1444.

[30] F. Wang and B. C. Vemuri, "Non-rigid multi-modal image registration using cross-cumulative residual entropy," *International journal of computer vision*, vol. 74, no. 2, pp. 201–215, 2007.

[31] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual slam and structure from motion in dynamic environments: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–36, 2018.

[32] M. Derome, A. Plyer, M. Sanfourche, and G. L. Besnerais, "Moving object detection in real-time using stereo from a mobile platform," *Unmanned Systems*, vol. 3, no. 04, pp. 253–266, 2015.

[33] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[34] J. Cheng, Y. Sun, and M. Q.-H. Meng, "Improving monocular visual slam in dynamic environments: an optical-flow-based approach," *Advanced Robotics*, vol. 33, no. 12, pp. 576–589, 2019.

[35] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular slam in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 209–218.

[36] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[37] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[38] C. Sheng, S. Pan, W. Gao, Y. Tan, and T. Zhao, "Dynamic-dso: Direct sparse odometry using objects semantic information for dynamic environments," *Applied Sciences*, vol. 10, no. 4, p. 1467, 2020.

[39] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, "Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *Robotics and Autonomous Systems*, vol. 117, pp. 1–16, 2019.

[40] X. Mu, B. He, X. Zhang, T. Yan, X. Chen, and R. Dong, "Visual navigation features selection algorithm based on instance segmentation in dynamic environment," *IEEE Access*, vol. 8, pp. 465–473, 2019.

[41] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello, "A visual odometry framework robust to motion blur," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 2250–2257.

[42] H. S. Lee, J. Kwon, and K. M. Lee, "Simultaneous localization, mapping and deblurring," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 1203–1210.

[43] J. Mustaniemi, J. Kannala, S. Särkkä, J. Matas, and J. Heikkilä, "Fast motion deblurring for feature detection and matching using inertial measurements," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3068–3073.

[44] W. N. Greene and N. Roy, "Flame: Fast lightweight mesh estimation using variational smoothing on delaunay graphs," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 4696–4704.

[45] G. D. Tipaldi, D. Meyer-Delius, and W. Burgard, "Lifelong localization in changing environments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1662–1678, 2013.

[46] H. Johannsson, "Toward lifelong visual localization and mapping," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.

[47] H. Kretzschmar, G. Grisetti, and C. Stachniss, "Lifelong map learning for graph-based slam in static environments," *KI-Künstliche Intelligenz*, vol. 24, no. 3, pp. 199–206, 2010.

[48] E. Einhorn and H.-M. Gross, "Generic ndt mapping in dynamic environments and its application for lifelong slam," *Robotics and Autonomous Systems*, vol. 69, pp. 28–39, 2015.

[49] "The long term visulization challenge," https://www.visuallocalization.net.

[50] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4938–4947.

[51] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[52] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7199–7209.

[53] D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei, and F. Qiao, "Dxslam: A robust and efficient visual slam system with deep features," *arXiv preprint arXiv:2008.05416*, 2020.

[54] H. Porav, T. Bruls, and P. Newman, "I can see clearly now: Image restoration via de-raining," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7087–7093.

[55] ——, "Don't worry about the weather: Unsupervised condition-dependent domain adaptation," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 33–40.

[56] A. Kim and R. M. Eustice, "Real-time visual slam for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[57] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Luján, S. Furber, A. J. Davison *et al.*, "Slambench2: Multi-objective head-to-head benchmarking for visual slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3637–3644.

[58] M. Bujanca, P. Gafton, S. Saeedi, A. Nisbet, B. Bodin, F. O'Boyle Michael, A. J. Davison, G. Riley, B. Lennox, M. Luján, and S. Furber, "SLAMBench 3.0: Systematic automated reproducible evaluation of SLAM systems for robot vision challenges and scene understanding," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6351–6358.

[59] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.

[60] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross, "Surfels: Surface elements as rendering primitives," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 335–342.