2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Are We Ready for Service Robots? The OpenLORIS-Scene Datasets for Lifelong SLAM

Xuesong Shi*[1], Dongjiang Li*[2,4], Pengpeng Zhao[1,5], Qinbin Tian[2,4], Yuxin Tian[2,5], Qiwei Long[2,4], Chunhao Zhu[2,4], Jingwei Song[2,4], Fei Qiao[2], Le Song[3], Yangquan Guo[3], Zhigang Wang[1], Yimin Zhang[1], Baoxing Qin[3], Wei Yang[4], Fangshi Wang[4], Rosa H. M. Chan[6] and Qi She[1]

Abstract— **Service robots should be able to operate autonomously in dynamic and daily changing environments over an extended period of time. While Simultaneous Localization And Mapping (SLAM) is one of the most fundamental problems for robotic autonomy, most existing SLAM works are evaluated with data sequences that are recorded in a short period of time. In real-world deployment, there can be out-of-sight scene changes caused by both natural factors and human activities. For example, in home scenarios, most objects may be movable, replaceable or deformable, and the visual features of the same place may be significantly different in some successive days. Such out-of-sight dynamics pose great challenges to the robustness of pose estimation, and hence a robot's long-term deployment and operation. To differentiate the forementioned problem from the conventional works which are usually evaluated in a static setting in a single run, the term *lifelong SLAM* is used here to address SLAM problems in an ever-changing environment over a long period of time. To accelerate lifelong SLAM research, we release the OpenLORIS-Scene datasets. The data are collected in real-world indoor scenes, for multiple times in each place to include scene changes in real life. We also design benchmarking metrics for lifelong SLAM, with which the robustness and accuracy of pose estimation are evaluated separately. The datasets and benchmark are available online at lifelong-robotic-vision.github.io/dataset/scene.**

I. Introduction

The capability of continuous self localization is fundamental to autonomous service robots. Visual Simultaneous Localization and Mapping (SLAM) has been proposed and studied for decades in robotics and computer vision. There have been a number of open source SLAM systems with careful designs and heavily optimized implementations. Do they suffice for deployment in real-world robots? We claim there is still a gap, coming from the fact that most SLAM systems are designed and evaluated for a single operation. That is, a robot moves through a region, large or small, with a fresh start. Real-world service robots, on the contrary, usually need to operate at a region day after day, with the requirement of reusing a persistent map in each operation to retain spatial knowledge and coordinate consistency. This requirement is more than saving the map and loading it for the next operation. The scene changes in real life and other uncontrolled factors in a long-term deployment bring considerable challenges to SLAM algorithms.

In this work, we use the term *lifelong SLAM* to describe the SLAM problem in long-term robot deployments. *For a robot that needs to operate around a particular region over an extended period of time, the capability of lifelong SLAM aims to build and maintain a persistent map of this region and to continuously locate the robot itself in the map during its operations.* To this end, the map must be reused in different operations, even if there are changes in the environment.

We summarize the major source of algorithmic challenges for lifelong SLAM as following:

- *Changed viewpoints* - the robot may see the same objects or scene from different directions.
- *Changed things* - objects and other things may have been changed when the robot re-enters a previously observed area.
- *Changed illumination* - the illumination may change dramatically.
- *Dynamic objects* - there may be moving or deforming objects in the scene.
- *Degraded sensors* - there may be unpredictable sensor noises and out-of-calibrations due to mechanical stress, temperature change, dirty or wet lens, *etc.*

While each of these challenges has been more or less addressed in existing works, there is a lack of public datasets and benchmarks to unify the efforts towards building practical lifelong SLAM systems. Therefore, we introduce the OpenLORIS-Scene datasets, which are particularly built for the research of lifelong SLAM for service robots. The data are collected with commodity sensors carried by a wheeled robot in typical indoor environments as shown in Fig. 1. Ground-truth robot poses are provided based on either a Motion Capture System (MCS) or a high-accuracy LiDAR. The major distinctions of our datasets are:

- The data are from real-world scenes with people in it.
- There are multiple data sequences for each scene, which include not only changes in illumination and viewpoints, but also scene changes caused by human activities in their real life.

*Equal contribution.
[1]Intel Labs China, Beijing, 100190 China.
[2]Department of Electronic Engineering and BNRist, Tsinghua University, Beijing, 100084 China.
[3]Gaussian Robotics, Shanghai, 201203 China.
[4]Beijing Jiaotong University, Beijing, 100044 China.
[5]Beihang University, Beijing, 100191 China.
[6]City University of Hong Kong, Hong Kong, China.
Corresponding authors: xuesong.shi@intel.com, qiaofei@tsinghua.edu.cn.

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

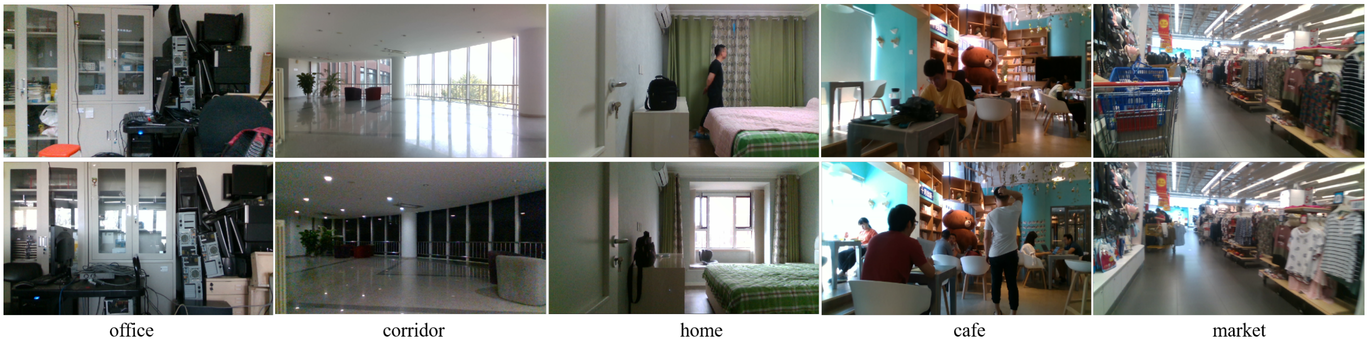

Fig. 1. Examples of color images in the OpenLORIS-Scene datasets. The upper and lower images in each column show approximately the same place in different data sequences, but the scene had been changed.

- There is a rich combination of sensors including RGB-D, stereo fisheyes, inertial measurement units (IMUs), wheel odometry and LiDAR, which can facilitate comparison between algorithms with different types of inputs.

This work also proposes new metrics to evaluate lifelong SLAM algorithms. As we believe the robustness of localization should be the most important concern, we use correct rates to explicitly evaluate it, as opposed to existing benchmarks where robustness is partially implied by the accuracy metrics.

II. Related Works

The adjective of *lifelong* has been used in SLAM-related works to emphasis either or both of the two capabilities: robustness against scene changes, and scalability in the long run. A survey of both directions can be found in [1].

Most SLAM works evaluate their algorithms on one or more public datasets to justify their effectiveness in certain aspects. The most well-used datasets include TUM RGB-D [2], EuRoC MAV [3] and KITTI [4]. A recent contribution is the TUM VI benchmark [5], where aligned visual and IMU data are provided. One of the major distinctions of those datasets is their sensor types. While there is favor of RGB-D data source in recent SLAM algorithm research for dense scene reconstruction, there is a lack of dataset with both RGB-D and IMU data. Our dataset provides aligned RGB-D-IMU data, along with odometry data which are widely used in the industry but often lack in public datasets.

Synthesized datasets are also used for SLAM evaluation. Recent progress in random scene generation and photo-realistic rendering [6][7] makes it theoretically possible to synthesize scene changes for lifelong SLAM, but it would be difficult to model realistic changes as in natural lives.

For real-world scene changes, the COLD database [8] provides visual data of several scenes with variations caused by weather, illumination, and human activities. It is the most related work with ours in the principle of data collection, though with different sensor setups. Object-level variations can also be found in the change detection datasets [9], but it is not designed for SLAM and does not provide ground-truth camera poses.

Recently there are efforts towards unified SLAM benchmarking and automatic parameter tuning [10][11], our work contributes to this direction by introducing new data and performance metrics.

III. OpenLORIS-Scene Datasets

The OpenLORIS-Scene datasets are designed to be a testbed of real-world practicality of lifelong SLAM algorithms for service robots. Therefore, the major principle is to make the data as close to service robots scenarios as possible. Commercial wheeled robot models equipped with commodity sensors are used to collect data in typical indoor scenes with people in it, as shown in Fig. 1. Rich types of data are provided to enable comparison of methods with different kind of inputs, as listed in Table I. All the data are calibrated and synchronized.

A. Robots and Sensors

To enable monocular, stereo, RGB-D and visual-inertial SLAM algorithms, two camera devices are used for data collection: a RealSense D435i providing RGB-D images and IMU measurements, and a RealSense T265 tracking module providing stereo fisheye images and IMU measurements. The IMU data are hardware synchronized with images from the same device. The cameras are mounted on a customized Segway Deliverybot S1 robot (for all the scenes except `market`) or a Gaussian Scrubber 75 robot (for the `market` scene), front-facing, at the height of about one meter. The resolution of RGB-D images are chosen to maximize the field of view (FOV), and for the best depth quality [12]. We provide not only aligned depth data as in other RGB-D datasets, but also raw depth images since they have a larger FOV which could benefit depth-based SLAM algorithms.

Wheel encoder-based odometry data are also provided, as they are widely available in wheeled robots. The odometry data from the Segway robot are fused from wheel encoders and a chassis IMU by proprietary filtering algorithms along with the robot.

To provide ground-truth trajectories, the Segway robot equips markers of an OptiTrack MCS and a Hokuyo UTM-30LX LiDAR, all near the cameras. The Gaussian robot equips a RoboSense RS-LiDAR-16.

TABLE I

DATA TYPES IN THE OPENLORIS-SCENE DATASETS

Device	Data	FPS	Resolution	FOV
D435i	color	30	848x480	H:69 V:42 D:77
D435i	depth	30	848x480	H:91 V:65 D:100
D435i	aligned depth[a]	30	848x480	H:69 V:42 D:77
D435i	accel	250	-	-
D435i	gyro	400	-	-
T265	fisheye1	30	848x800	D:163
T265	fisheye2	30	848x800	D:163
T265	accel	62.5	-	-
T265	gyro	200	-	-
base	odometry	20[b]	-	-
LiDAR	laser scan	40	1080	H:270[c]

[a] Depth images aligned to color images for per-pixel correspondence.

[b] This value is different for the `market` scene: 40.

[c] This value is different for the `market` scene: H:360 V:30.

TABLE II

TOOLS USED FOR EXTRINSIC CALIBRATION

Sensors		Tool
D435i	T265	Kalibr [13] github.com/ethz-asl/kalibr
MCS	D435i	robot_cal_tools github.com/Jmeyer1292/robot_cal_tools
MCS	T265	basalt [5] gitlab.com/VladyslavUsenko/basalt
LiDAR	D435i T265	LaserCamCal [14] github.com/MegviiRobot/CamLaserCalibraTool
odometer	D435i T265	proprietary

D435i refers to its color camera; T265 refers to its left fisheye camera.

B. Calibration

The intrinsics and intra-device extrinsics of cameras and IMUs are from factory calibration. Other extrinsics are calibrated with the tools listed in Table II. Redundant calibrations are made for quality evaluation. Each non-camera sensor (MCS, LiDAR and odometer) is calibrated against both cameras, so that the extrinsics between the two cameras can be deduced, which is then compared with their extrinsics directly calibrated with Kalibr [13]. The resulted errors are all below 1cm in translation and 2° in rotation, except for odometry calibration whose translation error is 7cm.

C. Synchronization

Images and IMU measurements from the same RealSense device are hardware synchronized. Software synchronization is performed for each data sequence between data from different devices, including RealSense D435i, RealSense T265, LiDAR, MCS and odometer. For each of those devices, its trajectory can be obtained either via a SLAM algorithm or directly from the measurements. Those per-device trajectories are then synchronized by finding the optimal time offsets to minimize the RMSE of absolute trajectory errors (ATEs). The ATEs of each per-device trajectory are calculated against the trajectory of MCS for the scene of `office`, and T265 for others, as the two provide poses in highest rates.

To mitigate the affection by SLAM and measurement noises, we generated a controlled piece of data at the beginning of each data sequence by pushing the robot back and forth for several times in a static and feature rich area, and used only this piece of data for synchronization.

The synchronization quality is evaluated by the consistency of resulted optimal time offsets. From our experiments, the standard deviation of offsets ranges from 1.7 ms (MCS to T265) to 7.4 ms (odometry to T265), with a positive correlation with the measurement cycle of each sensor. We think the results are acceptable for our scenarios, yet better synchronization methods can be discussed. One inherent drawback of the ATE minimization method is that systematic errors can be introduced if the scale of each estimated trajectory differs, which is frequently observed in the data. We mitigate this effect by using back-and-forth trajectories instead of move-and-stop ones, and also by carefully selecting a period of data when all trajectories can be best matched.

D. Scenes and Sequences

There are five scenes in the current datasets. For each scene, there are 2-7 data sequences recorded at different times. The sequences are manually selected and clipped from much more recordings to form a concise benchmark including most major challenges in lifelong SLAM. Some of the scene changes were deliberately influenced by the authors to maximize the difference between sequences, but all the manual changes were those that would likely to happen given a longer time. (e.g. relocated table and sofa in corridor)

- `Office`: 7 sequences in a university office with benches and cubicles. This scene is controlled: in `office-1` the robot walked along a U-shape route; in `office-2` the scene is unchanged but the route is reversed, so the cameras observe from opposite views; `office-3` is a turn-around that could connect `office-1` and `office-2` (to potentially be used to align sub-maps); in `office-4` and `office-5` the illumination is different from previous sequences; in `office-6` there are object changes; and `office-7` further introduced dynamic objects (persons).
- `Corridor`: 5 sequences in a long corridor with a lobby in the middle and the above office at one end. Apart from the well-known challenges in feature-poor long corridors, additional difficulties come from the high contrast between the corridor and the window at daytime, and extremely low light at night. Between sequences, there are not only illumination changes, but also moved furniture, which could make re-localization and loop closure a tough task. And the largeness of the scene would magnify the inconsistency of maps from different sequences if the algorithm fails to align them.
- `Home`: 5 sequences in a two bedroom apartment. There are lots of scene differences between sequences, such as changed sheets and curtains, moved sofa and chairs, and people moving around.
- `Cafe`: 2 sequences in an open café. There are different people and different things in each sequence.
- `Market`: 3 sequences in an open supermarket. Each trajectory is a long loop (150-220 meters). There are people moving around in the scene. The goods on some shelves have been changed between sequences.

There are 22 sequences in total. The accumulated length of the data is 2244 seconds.

E. Ground-truth

For each scene, ground-truth robot poses in a persistent map are provided for all sequences. For the office scene they are obtained from an MCS which wholly covers all the sequences, with a persistent coordinate system. The MCS-based ground-truth is in a rate of 240 Hz, with outliers removed. For other scenes, a 2D laser SLAM method is employed to generate ground-truth poses. A full map is built for each scene, and the robot is localized in the map with each frame of laser scan in the sequences. For the scene of corridor and cafe, a variant of hector_mapping [15] is used for map construction and localization. For home and market, another laser-based SLAM system combined with multi-sensor fusion is used to avoid from mismatching. The initial pose estimation of each sequence is manually assigned, and the output is manually verified to be correct. A comparison between laser-based ground-truth and MCS-based ground-truth is made with the in-office part of corridor data, which gives an ATE of 3 cm.

IV. Benchmark Metrics

Like most existing SLAM benchmarks, we mainly evaluate the quality of camera trajectory estimated by the SLAM algorithms. We adopt the same definition of Absolute trajectory error (ATE) and Relative pose error (RPE) as in the TUM RGB-D benchmark [2] to evaluate the accuracy of pose estimation for each frame. However, estimation failures or wrong (mismatched) poses are more severe than inaccuracies, and they may occur more commonly in lifelong SLAM due to scene changes. Therefore, we design separate metrics to evaluate the correctness and accuracy respectively.

A. Robustness Metrics

Correctness. For each pose estimate p_k at time t_k, given the ground-truth pose at that time, we assess the correctness of the estimate by its ATE and absolute orientation error (AOE):

$$c^{\varepsilon,\phi}(p_k) = \begin{cases} 1, & \text{if ATE}(p_k) \leq \varepsilon \text{ and AOE}(p_k) \leq \phi \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

Correct Rate (CR) and Correct Rate of Tracking (CR-T). While correctness evaluates a single pose estimate, the overall robustness metric over one or more data sequences can be defined as the correct rate over the whole time span of data. For a sequence from $t_{\min}$ to $t_{\max}$, given an estimated trajectory $\{t_k, p_k\}_{k=0,\ldots,N}$, define

$$\mathrm{CR}^{\varepsilon,\phi} = \frac{\sum_{k=0}^{N}\left(\min(t_{k+1}-t_k,\delta)\cdot c^{\varepsilon,\phi}(p_k)\right)}{t_{\max}-t_{\min}}, \qquad (2)$$

$$\mathrm{CR}^{\varepsilon,\phi}\text{-T} = \frac{\sum_{k=0}^{N}\left(\min(t_{k+1}-t_k,\delta)\cdot c^{\varepsilon,\phi}(p_k)\right)}{t_{\max}-t_0}, \qquad (3)$$

where $t_{N+1} \doteq t_{\max}$, δ is a parameter to determine how long a correct pose estimation is valid for. Note that in $\mathrm{CR}^{\varepsilon,\phi}$-T the time for re-localization and algorithm initialization ($t_0 - t_{\min}$) is excluded, since tracking is not functioning during that time. In practice, the ATE threshold ε and AOE threshold ϕ should be set according to the area of the scene and the expected drift of the SLAM algorithm. δ should be set larger than the normal cycle of pose estimation, and much smaller than the time span of data sequence. For common room or building size data, we would suggest to set ε to meter-size and δ around one second.

Correctness Score of Re-localization (CS-R). As tracking and re-localization are often implemented with different methods in common SLAM pipelines, they should be evaluated separately. The correctness of re-localization can be decided by the same ATE threshold as in CR. But besides correctness, we would also like to know how much time it takes to re-localize. Therefore, we define a score of re-localization as

$$\mathrm{C}^{\varepsilon,\phi}\mathrm{S}^{\tau}\text{-R} = e^{-(t_0-t_{\min})/\tau}\cdot c^{\varepsilon,\phi}(p_0) \qquad (4)$$

where τ is a scaling factor. Note that for an immediate correct re-localization with $t_0 = t_{\min}$, there will be CS-R = 1. The score drops with the time for re-localization increases. For normal evaluation cases we would suggest to set $\tau = 60$s.

B. Accuracy Metrics

To evaluate the accuracy of pose estimation without affected by incorrect results, we suggest to use statistics of ATE and RPE over one or more trajectories with only correct estimations. For example, $\mathrm{C}^{0.1}$-RPE RMSE is the root mean square error of RPE of correct pose estimates selected by an ATE threshold of 0.1 meter.

V. Experiments

The OpenLORIS-Scene datasets and the proposed metrics are tested with open-source SLAM algorithms. The algorithms are chosen to cover most data types listed in Table I, and to represent a diverse set of SLAM techniques. ORB-SLAM2 is a feature-based SLAM algorithm [16]. It can optimize poses with absolute scale by using either stereo features or depth measurements. DSO, on the contrary, tracks the camera's states with a fully direct probabilistic model [17]. DS-SLAM improves over ORB-SLAM2 by removing features on moving objects [18]. VINS-Mono provides robust pose estimates with absolute scale by fusing pre-integrated IMU measurements and feature observations [19]. InfiniTAM is a dense SLAM system based on point cloud matching with an iterative closed point (ICP) algorithm [20]. ElasticFusion combines the merits of dense reconstruction and globally consistent mapping by using a deformable model [21].

A. Per-sequence Evaluation

Method. First we test each data sequence separately, as done in most existing works. For each algorithm, the ground-truth trajectory are transformed into the target frame of pose estimation, for example, the color sensor of D435i for ORB-SLAM2 with RGB-D input. Then the estimated trajectory are aligned with the ground-truth using the method of Horn. For

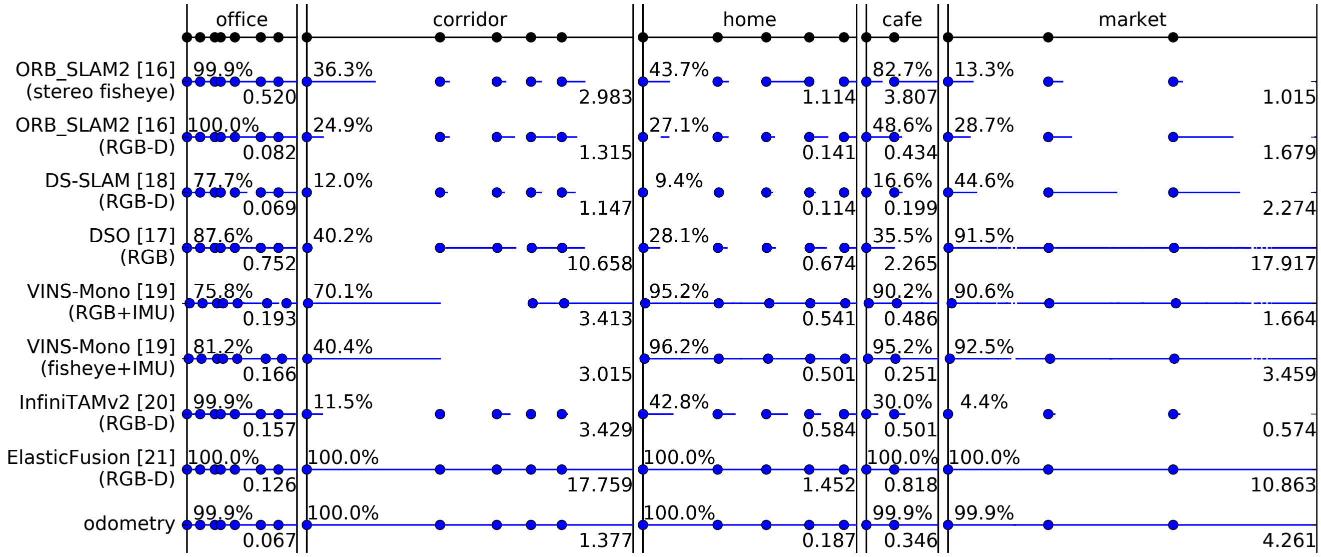

Fig. 2. Per-sequence testing results with the OpenLORIS-Scene datasets. Each black dot on the top line represents the start of one data sequence. For each algorithm, blue dots indicate successful initialization, and blue lines indicate successful tracking. The percentage value on the top left of each scene is average CR$^{\infty}$, larger means more robust. The float value on the bottom right is average ATE RMSE, smaller means more accurate.

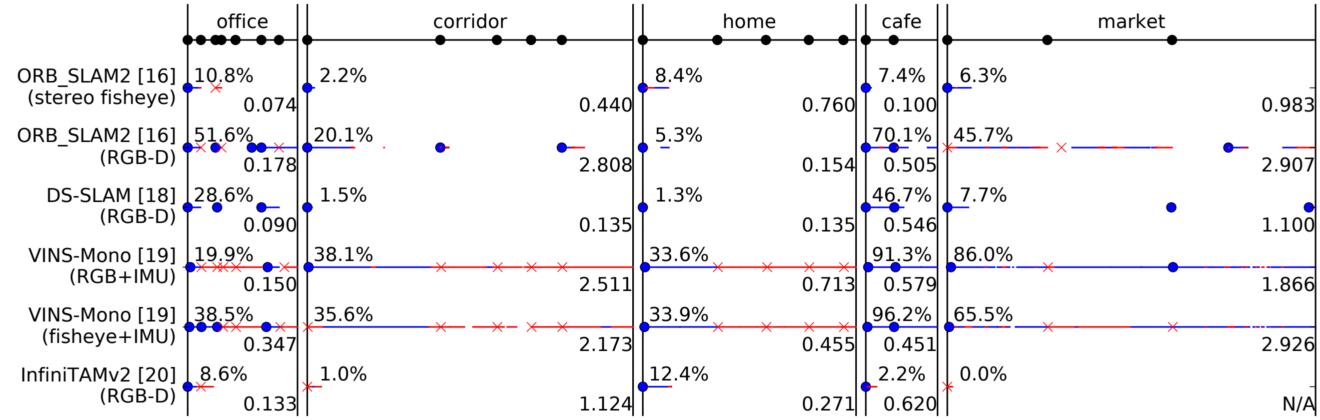

Fig. 3. Lifelong SLAM testing results with the OpenLORIS-Scene datasets. For each algorithm, blue dots indicate successful initialization or correct re-localization, while red crosses are incorrect re-localization. Line segments in blue and red indicate correct and incorrect pose estimation, respectively. The percentage value on the top left of each scene is average Correct Rate (CR$^{\varepsilon,\phi}$ as in Eq. (2)). The float value on the bottom right is average C$^{\varepsilon,\phi}$-ATE RMSE. The ATE threshold ε is 1 m for office, 3 m for home and cafe, and 5 m for corridor and market. The AOE threshold ϕ is 30° for all scenes.

DSO, an optimal scaling factor is calculated with Umeyama's method [22]. Then ATE of each matched pose is calculated and their RMSE over each sequence is reported. The only difference in our ATE calculation process from conventional ones is that we interpolate the ground-truth trajectory to let each estimated pose get an exact match on the timeline, as opposed to matching the closest ground-truth pose. The reason is that our laser-based ground-truth trajectories have a lower rate than MCS-based ones.

Result. The results are visualized in Fig. 2, with blue line segments indicating successful localization and blank otherwise. The success rate indicated by CR$^{\infty}$, and accuracy indicated by ATE RMSE are calculated for each sequence. On the figure only statistics over each scene are shown, where CR$^{\infty}$ and ATE RMSE are averaged weighted by the time span of each sequence and the count of pose estimates, respectively. All the algorithms can track successfully most of the time in `office`, but other scenes are challenging. For example, most algorithms tend to lost in `corridor` because of the featureless walls and low light, yet VINS-Mono can fully track some of the sequences in this scene. Note that VINS-Mono fails to initialize in some low-light sequences in `corridor`. Nevertheless, VINS-Mono shows the best robustness among the tested algorithms.

The wheel odometry data in OpenLORIS-Scene are evaluated along with SLAM algorithms in Fig. 2. It can be seen that our odometry data provides reliable tracking results even in large scenes. We think that odometry should not be neglected by practical SLAM algorithm designers for service robots.

Metrics discussion. If we compare between the CR^{∞} from DS-SLAM and ORB-SLAM2 with the same inputs, the former tends to lost more often since it uses less features to localize, but it succeeds in `market` which are highly dynamic. If we also note their ATEs, it can be found a consistent negative correlation between the two similar algorithms' ATE and CR^{∞}. The reason is that the longer an algorithm tracked, the more error is likely to accumulate. It implies that evaluating algorithms purely by ATE could be misleading. On the other hand, considering only CR^{∞} could also be misleading. For example, DSO results high CR^{∞} in `corridor` and `market`, but the estimated trajectories are actually erroneous. Its CR would be much lower if we set a proper ATE threshold.

B. Lifelong SLAM Evaluation

Method. To test whether an algorithms could continuously localize in changed scenes, we feed it sequences of the same scene one by one. There may be a significant view change when switching to the next sequence. The algorithm could either wait for a successful re-localization (e.g. ORB-SLAM2), or start with a fresh map and then try to align it with the old map by loop closing (e.g. VINS-Mono). DSO and ElasticFusion are excluded from this test since the implementation we use does not support re-localization. For ORB-SLAM2 RGB-D, we use a revised version with a few engineering improvements but no algorithmic changes. For each scene, we align the estimated trajectory of the first sequence to the ground-truth, and using the resulted transformation matrix to transform all the estimated trajectories of this scene, then compare them with the ground-truth.

Result. The results are shown in Fig. 3, with red cross and line segment indicating incorrect pose estimates, judged by an ATE threshold of 1/3/5 meters for small/medium/large scenes and an AOE threshold of 30°. It shows that re-localization is challenging. For example, most algorithms completely fail to re-localize in the 2nd-5th sequences of `home`.

Metrics discussion. From the results we see that the metrics are imperfect. For example, for `corridor` and `market`, some algorithms get an incorrect initial localization for the first sequence, which is technically unsound. The reason is that large drifts have been accumulated over the long trajectories, and after aligning the full trajectory to the ground-truth, its initial part has a large error. It suggests that we should set even larger ATE thresholds for large scenes, and that further refinement of the accuracy judgement method should be discussed. Besides the false alarm in initial and final parts of `corridor-1` and `market-1`, the metrics succeeds to recognize incorrect localization, and gives meaningful statistics.

TABLE III

RE-LOCALIZATION SCORES WITH CONTROLLED CHANGING FACTORS

Data: office-	**1,2**	**2,4**	**2,5**	**1,6**	**2,7**
Key factor	**viewpt.**	**illum.**	**low light**	**objects**	**people**
ORB (stereo)	0	0	0	0.742	0.995
ORB (RGB-D)	0	0.764	0	0.716	0.997
DS-SLAM	0	0	0	0.994	0.996
VINS (color)	0	0	0	0.837	0
VINS (fisheye)	0	0	0	0	0
InfiniTAMv2	0	0	0	0	0

The values are $C^{0.3,\infty}S^{60}$-R as defined in Eq. (4)

Factor analysis. Correct re-localization is rare in Fig. 3 partly because we have deliberately selected the most challenging sequences for the released data. In most scenes, the challenge comes from mixed factors including changed viewpoints, changed illumination, changed things and dynamic objects. The `office` data are designed to disentangle those factors. We conduct another set of tests with specified sequence pairs in `office`. The two sequences in each pair have one key different factors, as described in Section III.D. The re-localization scores are listed in Table III. The results suggest that changed viewpoints and illumination are most difficult to deal with. The former is expected as natural scenes are likely to generate different visual and geometric features from different viewpoints. The latter might be mitigated by carefully tuning algorithms and devices. We expect that deep learning based features and semantic information should be able to help address both problems.

VI. CONCLUSION

This paper introduces the OpenLORIS-Scene datasets and metrics for benchmarking lifelong SLAM algorithms. The datasets capture scene changes caused by day-night shifts and human activities, as well as viewpoint changes, moving people, poor illumination, and blur. We found these factors challenging enough to existing SLAM systems. New metrics are proposed to evaluate the localization robustness and accuracy separately. With the datasets and metrics, we hope to help identify shortcomings of SLAM algorithms and to encourage new designs with more robust localization capabilities, such as by introducing high-level scene understanding capabilities. The datasets can also be a testbed of the maturity for real-world deployment of future SLAM algorithms for service robots.

Beyond SLAM, the data may also facilitate a broader scope of long-term scene understanding research for service robots. With proper annotation, it could serve as a benchmark of incremental learning algorithms [23][24], enabling robots to keep learning new tasks. It would also be interesting to explore spatio-temporal modeling [25] with the data.

ACKNOWLEDGEMENT

The authors would like to thank Yusen Qin, Dongyan Zhai and Jin Wang for customizing the Segway robot and lending it for this project, and performing odometer-camera calibration. Thank Yijia He for providing LiDAR-camera calibration tools and guidance. Thank Phillip Schmidt, Chuan Chen, Hon Pong Ho and Yu Meng for the technical support of RealSense cameras. Thank Mihai Bujanca and Bruno Bodin for helping integrate OpenLORIS-Scene into SLAMBench. Thank Yinyao Zhang, Long Shi and Zeyuan Dong for helping collect and maintain the data. Thank all the anonymous participants in data collection.

References

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

[2] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[3] 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*, vol. 35, no. 10, pp. 1157–1163, 2016.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1680–1687.

[6] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: 5M photorealistic images of synthetic indoor trajectories with ground truth," *arXiv preprint arXiv:1612.05079*, 2016.

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

[8] A. Pronobis and B. Caputo, "COLD: COsy Localization Database," *International Journal of Robotics Research (IJRR)*, vol. 28, no. 5, pp. 588–594, May 2009. [Online]. Available: http://www.pronobis.pro/publications/pronobis2009ijrr

[9] M. Fehr, F. Furrer, I. Dryanovski, J. Sturm, I. Gilitschenski, R. Siegwart, and C. Cadena, "TSDF-based change detection for consistent long-term dense reconstruction and dynamic object discovery," in *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2017, pp. 5237–5244.

[10] B. Bodin, H. Wagstaff, S. Saeedi, L. Nardi, E. Vespa, J. Mayer, A. Nisbet, M. Luján, S. Furber, A. Davison, P. Kelly, and M. O'Boyle, "SLAMBench2: Multi-objective head-to-head benchmarking for visual SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2018.

[11] Y. Zhao, S. Xu, S. Bu, H. Jiang, and P. Han, "GSLAM: A general SLAM framework and benchmark," *arXiv:1902.07995*, 2019.

[12] A. Grunnet-Jepsen, J. N. Sweetser, and J. Woodfill. Best known methods for tuning Intel® RealSense™ depth cameras D415 and D435. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/BKMs_Tuning_RealSense_D4xx_Cam.pdf

[13] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1280–1286.

[14] Qilong Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, Sep. 2004, pp. 2301–2306 vol.3.

[15] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[16] 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, Oct 2017.

[17] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, March 2018.

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

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

[20] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S. Torr, and D. W. Murray, "Very high frame rate volumetric integration of depth images on mobile device," *IEEE Transactions on Visualization and Computer Graphics (Proceedings International Symposium on Mixed and Augmented Reality 2015*, vol. 22, no. 11, 2015.

[21] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016. [Online]. Available: https://doi.org/10.1177/0278364916669237

[22] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 04, pp. 376–380, apr 1991.

[23] Q. She, F. Feng, X. Hao, Q. Yang, C. Lan, V. Lomonaco, X. Shi, Z. Wang, Y. Guo, Y. Zhang *et al.*, "OpenLORIS-Object: A robotic vision dataset and benchmark for lifelong deep learning," *arXiv preprint arXiv:1911.06487*, 2019.

[24] F. Feng, R. H. Chan, X. Shi, Y. Zhang, and Q. She, "Challenges in task incremental learning for assistive robotics," *IEEE Access*, 2019.

[25] T. Krajník, M. Kulich, L. Mudrová, R. Ambrus, and T. Duckett, "Where's Waldo at time t? Using spatio-temporal models for mobile robot search," in *International Conference on Robotics and Automation (ICRA)*, 2015.