



MAPPING QUALITY EVALUATION OF MONOCULAR SLAM SOLUTIONS FOR MICRO AERIAL VEHICLES

J. Wang^{1,*}, M. Shahbazi¹

¹ Dept. of Geomatics Engineering, University of Calgary, University Drive NW, Calgary, T2N 1N4 Canada - (wang4, mozhdeh.shahbazi)@ucalgary.ca

Commission II

KEY WORDS: Monocular SLAM, Mapping quality, Points cloud, Micro Aerial Vehicles, Navigation

ABSTRACT:

Monocular simultaneous localization and mapping (SLAM) attracted much attention in the mobile-robotics domain over the past decades along with the advancements of small-format, consumer-grade digital cameras. This is especially the case for micro air vehicles (MAV) due to their payload and power limitations. The quality of global 3D reconstruction by SLAM solutions is a critical factor in occupancy-grid mapping, obstacle avoidance, and map representation. Although several benchmarks have been created in the past to evaluate the quality of vision-based localization and trajectory-estimation, the quality of mapping products has been rarely studied. This paper evaluates the quality of three state-of-the-art open-source monocular SLAM solutions including LSD-SLAM, ORB-SLAM, and LDSO in terms of the geometric accuracy of the global mapping. Since there is no ground-truth information of the testing environment in existing visual SLAM benchmark datasets (e.g., EuRoC, TUM, and KITTI), an evaluation dataset using a quadcopter and a terrestrial laser scanner is created in this work. The dataset is composed of the image data extracted from the recorded videos by flying a drone in the test environment and the high-fidelity point clouds of the test area acquired by a terrestrial laser scanner as the ground truth reference. The mapping quality evaluation of the three SLAM algorithms was mainly conducted on geometric accuracy comparisons by calculating the deviation distance between each SLAM-derived point clouds and the laser-scanned reference. The mapping quality was also discussed with respect to their noise levels as well as further applications.

1. INTRODUCTION

In order to explore unknown environments autonomously, mobile robotics systems such as unmanned aerial vehicles (UAVs) must be equipped with the ability to build a consistent map of the environment while localizing themselves with the aid of onboard sensors. This problem, known as simultaneous localization and mapping (SLAM), has been actively studied in computer vision and mobile robotics fields for the past two decades (Grisetti et al., 2010). The SLAM problem is especially challenging for micro aerial vehicles (MAVs) (Kumar and Michael, 2017) due to payload limitation, power consumption, and high degrees-of-freedom motion. Thus, solutions based on monocular SLAM have attracted much attention of both commercial and research sectors due to being advantageous in terms of small size, lightweight, low power consumption, and cost-effectiveness.

Generally speaking, a keyframe-based monocular SLAM system (Younes et al., 2017) is composed of the visual odometry (VO) (Scaramuzza and Fraundorfer, 2011), global optimization (GO) and global mapping (GM) parts (Taketomi et al., 2017). The VO component is responsible for estimating sequential ego poses (position and orientation) of the camera. In VO, small portions of the map and only the relative camera motions are computed; the global consistency between these portions is not considered (Klein and Murray, 2007). The GO component includes loop closure (Williams et al., 2009) and global pose-graph optimization (Dubbelman and Browning, 2015) elements to handle the sequentially propagated errors, drifts, and global geometric consistencies of the global map and camera trajectory. During loop closure, the current image is compared to the

previously observed image frames; a loop is detected when an appropriate amount of correspondence is identified. Thus, the accumulative error of the camera trajectory can be eliminated. The global pose-graph optimization uses a graph to minimize the accumulated errors of a large environment. The graph consists of the camera poses as nodes that are connected by edges representing camera motions. Instead of pose-graph optimization, multi-view bundle adjustment (BA) can be used to minimize not only the camera motions but also the position of the three-dimensional (3D) structure points (Krombach et al., 2017). The GM component refers to the consistent 3D mapping of the whole explored environment. In monocular SLAM, the environment map is represented as a point cloud. The point clouds of the environment generated by SLAM approaches can either be represented as meshed surfaces (Sadat et al., 2014) or tree-based structures (Hornung et al., 2013). These generated maps are primarily used for high-level applications such as obstacle avoidance, path planning, and navigation in the closed-loop robotics architecture (Liu, 2016). Since the majority SLAM researches initiate from VO (Taketomi et al., 2017; Yang et al., 2017), there are an abundant number of studies that report the quality of camera trajectory estimations (Bodin et al., 2018; Delmerico and Scaramuzza, 2018; Gao et al., 2018; Mur-Artal and Tardos, 2017; Sturm et al., 2012). However, minimal resources exist for evaluating the mapping quality of modern monocular SLAM solutions. Even though the VO component plays an essential role in evaluating states of the camera, the mapping representation of the environment is equally important in mobile robotic systems and other applications such as the planning and monitoring of construction process and building conditions (Kim et al., 2018).

* Corresponding author

In this paper, we compare the geometric quality of 3D maps generated by state-of-the-art monocular SLAM algorithms implemented on a low-cost MAV. These algorithms are selected due to their superior trajectory-estimation quality, their variations in ranges of sparse, semi-dense or dense mapping, and being equipped with global optimization components. These SLAM methods include LSD-SLAM (Engel et al., 2014), ORB-SLAM (Mur-Artal and Tardos, 2017), and LDSO (Gao et al., 2018). Because there is no ground-truth information of the testing environment in existing visual SLAM benchmarks (e.g., EuRoC, TUM, and KITTI), in this work, an evaluation dataset using a quadcopter and a terrestrial laser scanner is created. The dataset is composed of images extracted from videos captured by flying a MAV in the test environment and a high-fidelity point cloud of the test area acquired by a terrestrial laser scanner. The mapping quality of these three SLAM algorithms is evaluated by the deviation distances between each SLAM-derived point cloud and the laser-scanned reference point cloud. These point clouds are also compared with respect to their noise level and 3D reconstruction ability. The implementation details of the LSD-SLAM, ORB-SLAM, and LDSO algorithms for mapping quality evaluation in this work are made publicly available at the following link: <https://github.com/jwangjie/Mapping-ARDrone>.

2. MONOCULAR SLAM ALGORITHMS

The key features of three selected SLAM algorithms are summarized in Table 1. LSD-SLAM is one of the most recognizable monocular SLAM algorithms implementing the direct method (as opposed to a feature-based solution). The direct method doesn't need feature extraction and thus is adaptable to environments with less distinctive features. The camera is localized by optimizing directly over image pixel intensities. LSD-SLAM employs a photometric error as well as a geometric prior; thus it can perform dense or semi-dense reconstructions of the environment.

Approach	LSD-SLAM	ORB-SLAM	LDSO
Points density	Semi Dense	Sparse	Sparse - Semi Dense
Method	Direct and Dense	Indirect and Sparse	Direct and Sparse
Global optimization	Yes	Yes	Yes
Loop Closure	Yes	Yes	Yes

Table 1. Key features summary of the three selected visual monocular SLAM algorithms

ORB-SLAM is one of the most reliable and complete feature-based algorithms for monocular SLAM. Originated from PTAM (Klein and Murray, 2007), ORB-SLAM benefits from updated research progress incorporating ORB features, loop closure detections, and pose graph optimization blocks. Because a dense or semi-dense map is preferred for high-level robotics applications (Faessler et al., 2016), methods to gain dense or semi-dense maps from sparsely-generated point clouds are useful for ORB-SLAM (Mur-Artal and Tardos, 2015). Mur-Artal et al. developed a probabilistic semi-dense mapping module to work together with ORB-SLAM. The integrated approach performs semi-densification over keyframes in real-time on a CPU to obtain semi-dense maps. Therefore, the ORB-SLAM algorithm is also applicable to high-level robotics applications if it can generate high-accuracy sparse point clouds.

LDSO is a full SLAM algorithm that integrates loop closure and global map optimization into direct sparse odometry (DSO) (Engel et al., 2018). DSO has a novel sparse and direct structure, which combines the benefits of direct methods with the flexibility of sparse approaches. The VO component of LDSO performs optimizations to minimize photometric errors on a sparse set of points directly. Unlike conventional feature-based approaches, it tracks any pixels with large-enough intensity gradients. It computes points on all gradient-rich areas including edges. The density of the point clouds can be changed by defining the gradient threshold numbers. As a result, LDSO can establish maps with density ranging from spare to semi-dense.

3. EXPERIMENTS

This section describes the platform and algorithms setups to implement the monocular SLAM methods on a MAV. We utilized a Parrot AR Drone 2.0 (Parrot Drones SAS., n.d.) to collect videos of the test environment, conducted camera calibrations, and processed each test dataset and SLAM algorithms properly for implementations.

3.1 Platform

A Parrot AR Drone 2.0 was used to collect videos of the test environments. The Parrot AR. Drone 2.0 is a commercial quadrotor shown in Figure 1a. The MAV has highlights of low price, robustness to crashes, and high safety for indoor usage. Because neither hardware nor software running onboard can be modified easily, communication and control were realized by Wi-Fi connections. The MAV has a front-facing camera shown in Figure 1b that can be configured to capture and stream videos with 1280*720 resolution at 30 fps. The camera was calibrated using a standard checkerboard marker. In order to remove the impact of substantial radial lens distortions caused by the wide-angle lens of the camera, images extracted from collect videos were rectified before being fed to each SLAM algorithm.



Figure 1. The Parrot AR. Drone 2.0: (a) with the hull; (b) the front-facing camera

3.2 Point Clouds Mapping

There are three most commonly used datasets for visual SLAM system evaluations including the EuRoC dataset (Burri et al., 2016), TUM RGB-D and monoVO dataset (Sturm et al., 2012)(Engel et al., 2016), and KITTI dataset (Geiger et al., 2013). However, all the above datasets don't include ground-truth point clouds of the test scenes. For our mapping quality evaluation, a reference point cloud map of our test environment was generated using a terrestrial laser scanner as ground truth shown in Figure 2. Videos of the same test scene were recorded with 1280*720 resolution at 30 fps by flying the drone shown in Figure 1. Because the three SLAM algorithms have different configurations, proper modifications of each algorithm package

are made to implement with our custom dataset. For example, the ORB-SLAM package closes the point clouds window immediately when the pipeline finishes running; it is impossible to view and save the generated point clouds afterward. Modifications of the original codes are conducted to run our test dataset and to save generated point cloud files for conducting accuracy evaluations.

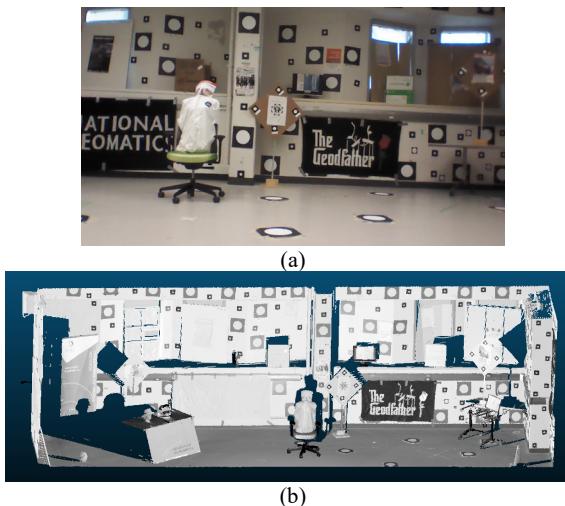
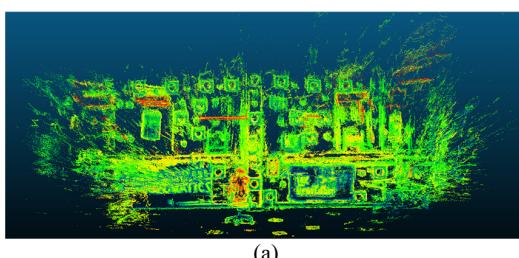


Figure 2. Test scene: (a) one image frame; (b) reference point cloud captured by a terrestrial laser scanner FARO Focus3D X

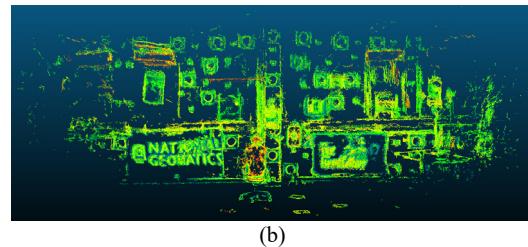
After several round implementations of the three SLAM algorithms with our test images, a uniform test dataset was created. Our custom dataset follows the format of the EuRoC dataset. Images extracted from the recorded video at each frame were named sequentially in the “data” folder; a “data.csv” was created containing the timestamp and image filename information; the “data” folder and “data.csv” file were stored in the same folder.

3.3 LSD-SLAM

The LSD-SLAM algorithm provides implementation with the TUM RGB-D dataset among the three datasets. For implementing with our dataset, the LSD-SLAM method was executed using the “dataset slam” command. The generated point cloud map was saved by entering “p” for a few seconds continuously when the “lsd_slam_viewer” window was selected. The point cloud map was saved as a “pc.ply” file in the “lsd_slam_viewer” folder shown in Figure 3a.



(a)



(b)

Figure 3. Semi-Dense point clouds generated by LSD-SLAM: (a) raw data; (b) noise-filtered data

3.4 ORB-SLAM

It is straightforward to evaluate the ORB-SLAM algorithm by commands provided by the author (Gaud, 2017) for all three datasets. In order to be implemented with our dataset, the camera calibration file and image timestamp file were modified to specify the AR. Drone camera image information and were stored in the “Monocular” folder following instructions in (Gaud, 2017). The codes of “System.cc”, “System.h” and “CMakeLists.txt” files were modified in order to output generated point clouds as a file (Jeroen Zijlmans, 2017). A new “mono_ar drone_pcl.cc” file was created to generate an executable file to execute our dataset. A point cloud map was saved as a “pointcloud.pcd” file in the “ORB_SLAM2” folder shown in Figure 4a.



(a)



(b)

Figure 4. Sparse point clouds derived by ORB-SLAM: (a) raw data; (b) noise-filtered data

3.5 LDSO

Similar to ORB-SLAM, the LDSO algorithm can be tested by provided commands (Gao and Demmel, 2018) with all three datasets. After modifying the camera calibration parameters in the “EUROC.txt” file, the LDSO was applied with the custom dataset using the same command for the EuRoC dataset. When the package finished running, the generated points cloud map was automatically saved as the “pointcloud.ply” file by shutting down the “Pangolin” window shown in Figure 5a.

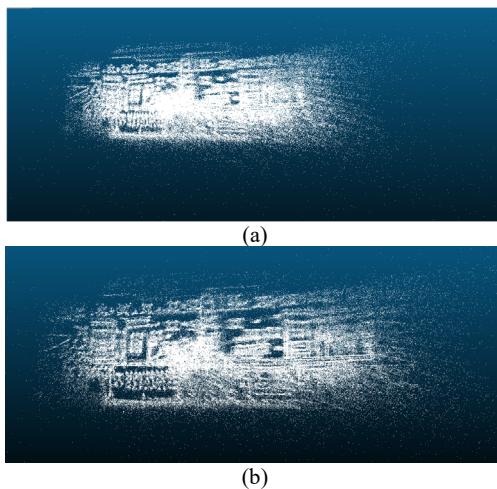


Figure 5. Semi-dense point clouds derived by LDSO: (a) raw data; (b) noise-filtered data

4. EVALUATION RESULTS

In this section, the mapping quality of LSD-SLAM, ORB-SLAM, and LDSO are evaluated by conducting cloud-to-cloud (C2C) comparisons of each extracted point cloud with the laser-scanned reference respectively. The outliers of each point cloud are filtered first to avoid possible following point cloud registration failures. A coarse registration between each SLAM-derived and the reference point cloud is performed using a point-pair based alignment method. The point cloud registration is refined by iterative closest point (ICP) processing on the coarse registration results. The main reason for conducting C2C registration is that monocular SLAM algorithms are initiated in arbitrary coordinate systems, and due to the lack of external measurements, e.g., inertial measurements, maps generated by these SLAM methods are not represented in the true scale.

The C2C deviation distances between each SLAM derived point cloud and the reference point cloud are calculated on the fine registration results using four distance computation methods (nearest neighbor, least square plane, 2D1/2 triangulation, and quadric). The distance computation results are compared and the method has the smallest standard deviation (least square plane method) is selected to evaluate the point cloud accuracy (Ahmad Fuad et al., 2018). The whole evaluation process comprises the raw point cloud outlier filtering, C2C registration, C2C deviation distance calculation, and results analysis. These processes are performed in the CloudCompare open-source software (Girardeau-Montaut, 2015).

4.1 Outlier Points Filtering

In order to get cleaner point clouds, a statistical outlier removal (SOR) filtering process was applied. For each point, the average distances to its k nearest neighbors were computed. By assuming that the distribution of inliers is Gaussian, points whose distances were outside a 3-sigma interval from the mean were considered as outliers and removed from the point clouds. Median filtering was applied afterward to remove the additional noise. The numbers of points in each raw and filtered point cloud were listed in Table 2.

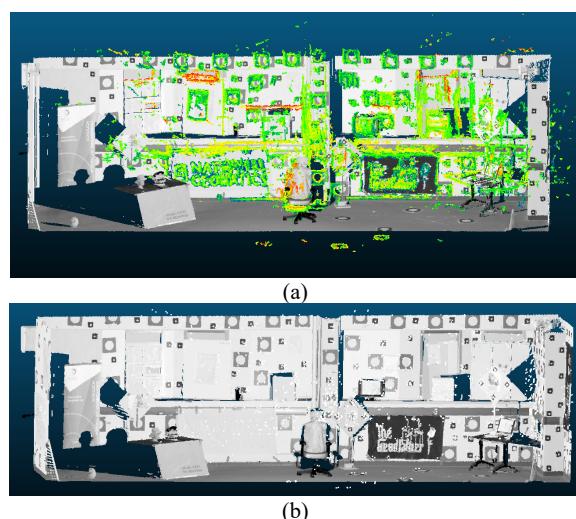
Mapping Solutions	Number of points		
	Raw PC	Filtered PC	Outliers %
Laser scanner	453,090	NA	NA
LSD-SLAM	1,169,856	560,735	52.38%
ORB-SLAM2	3,402	3,131	8.91%
LDSO	248,341	215,482	18.71%

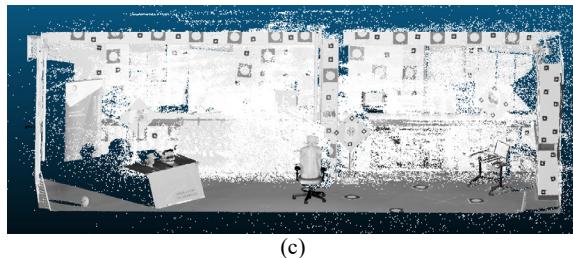
Table 2. Number of points in raw and filtered point clouds

4.2 C2C Registration

A proper C2C registration between the SLAM derived and laser-scanned reference point clouds is essential to evaluate the mapping accuracy. It was conducted by a manual point-pair based alignment and automatic ICP based registration (Pottmann et al., 2004). The point-pair based alignment was applied to initialize the registration coarsely. When the two clouds are roughly registered, they can be further fine registered by an ICP method. Because the SLAM derived point clouds and laser-scanned reference point cloud had a very low overlap with different scales, point-pair based registration initialization was conducted first. During the point-pair based alignment process, at least three equivalent point pairs in both point clouds were picked manually using the “Align” tool in CloudCompare. The transformation matrix and root-mean-square error (RMSE) were automatically calculated based on selected points, and the scale of the align point cloud was also adjusted to the scale of the reference point cloud.

A fine registration was then conducted on the roughly aligned point clouds using the “Fine Registration” tool in CloudCompare. During the automatic ICP based registration process, the ICP algorithm takes the two point clouds as inputs and calculates the rigid transformation (rotation matrix R and translation vector T) that best registers the two point clouds. If the two point sets (n points) express as $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$ respectively, then ICP iteratively calculate R and T that minimize the sum of the squared error $E(R, T) = \frac{1}{n} \sum_{i=1}^n \|x_i - Ry_i - T\|^2$. In the ICP fine registration, the process can either be stopped after a defined maximum iteration number or when the error (RMSE) difference becomes lower than a given threshold. The C2C registration results between each SLAM derived and laser-scanned reference point cloud are shown in Figure 6.





(c)

Figure 6. The C2C registration results between each SLAM-derived and laser-scanned reference point clouds: (a) LSD-SLAM; (b) ORB-SLAM; (c) LDSO

4.3 C2C Deviation Distance Calculation

The most basic C2C distance computation approach is the nearest neighbor distance method (Moghaddame-Jafari, 2016). The nearest neighbor distance method searches the nearest point in the reference for each point in the compared point cloud and then calculates their Euclidean distances (d_i). The mean distance (x_m) and standard deviation (σ) of the compared point clouds (n points) are calculated by $x_m = \sum_{i=1}^n d_i / n$ and $\sigma = \sqrt{\sum_{i=1}^n (d_i - x_m)^2} / n$ respectively. The issue of the nearest neighbor distance method for C2C distance calculation is that the nearest neighbor is not necessarily the actual nearest point on the surface represented by the cloud (Moghaddame-Jafari, 2016). This is especially true if the reference cloud has a low density, which is the case for the LSD-SLAM derived point cloud who has a higher point density compared to the reference point cloud shown in Table 2. In this case, C2C distance methods computing a local model around the nearest point to approximate the real surface and get a better estimation of the 'real' distance should be used. Three local modeling methods, least square plane, 2D1/2 triangulation, and quadric were used for the C2C deviation distance calculations. These three modeling methods estimate the best-fitting plane (plane, triangulation, and quadric respectively) that goes through the nearest point and its neighbors. The mean distance and standard deviation of each local modeling method were calculated using the above equations. These results are listed in Table 3.

Local points model	Mean distance (m)	Standard deviation (m)
LSD-SLAM point cloud		
Nearest neighbour	0.0772	0.0961
Least square plane	0.0675	0.0856
2D1/2 triangulation	0.0766	0.0964
Quadric	0.0692	0.0876
ORB-SLAM2 point cloud		
Nearest neighbour	0.0447	0.0494
Least square plane	0.0381	0.0429
2D1/2 triangulation	0.0440	0.0500
Quadric	0.0391	0.0435
LDSO point cloud		
Nearest neighbour	0.1571	0.1152
Least square plane	0.1308	0.1075
2D1/2 triangulation	0.1568	0.1155
Quadric	0.1356	0.1103

Table 3. Distance calculations between the SLAM-derived point cloud and the reference laser-scanned reference

4.4 Results Analysis

The evaluation results of the SLAM derived point cloud maps are analyzed from the perspective of geometric accuracy, 3D reconstruction ability, and noise and spare level.

4.4.1 Geometric Accuracy: In Table 3, It can conclude that the least square plane method provides deviation distance calculation results with the smallest standard deviation compared to the other three methods. Thus, the least square plane method is selected as the C2C distance computation method for deviation calculations between the SLAM derived point clouds and the laser-scanned reference point cloud similar to (Ahmad Fuad et al., 2018). The C2C deviation distance results calculated by the least square plane method presented in color scale mode are shown in Figure 7. In Table 3, it can conclude that the overall geometric accuracy of SLAM derived point cloud maps ranking from high to low is ORB-SLAM, LSD-SLAM, and LDSO. The accuracy results can also be viewed in Figure 7.

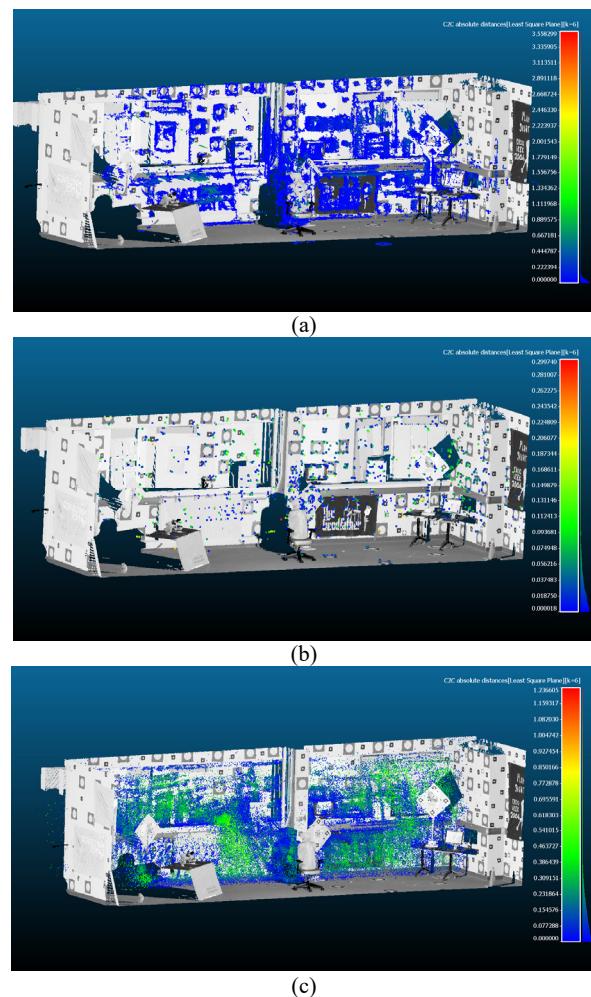


Figure 7. The C2C distances between the SLAM derived and laser-scanned reference point clouds in color scale mode: a) LSD-SLAM, b) ORB-SLAM; c) LDSO

4.4.2 3D Reconstruction: In Figure 7, LSD-SLAM and ORB-SLAM provide decent results of the 3D reconstruction. Points of foreground objects, e.g., chair, table and standing boards, were properly generated. Correct spatial information was included in their generated point cloud maps shown in Figure 7a and 7b respectively. LDSO provides the worst 3D reconstruction of the test environment shown in Figure 7c. In fact, the derived point cloud map was generated in a 2D plane instead of 3D structure; all foreground objects were neglected. It can be seen that LDSO failed to integrate spatial information into the generated point clouds map.

In order to identify the poor 3D reconstruction performance of LDSO is due to the AR. Drone camera or the algorithm itself, further tests were conducted. The LDSO was tested with the EuRoC vicon room 201 dataset. By trimming the right wall off for a cleaner appearance view, it's obvious that a high-quality 3D reconstruction point cloud of the test scene was generated shown in Figure 8.

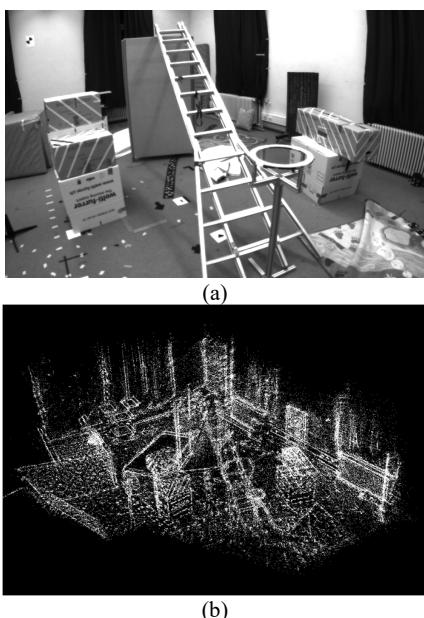


Figure 8. Semi-Dense point cloud generated by LDSO with EuRoC vicon room 201: (a) test scene; (b) point cloud map

Because there is no ground truth reference in the EuRoC dataset, the geometric accuracy of the point cloud in Figure 8b was evaluated by comparing it with a point cloud generated by running ORB-SLAM with the EuRoC vicon room 201 dataset. Because ORB-SLAM has the best overall geometric accuracy among the evaluated three SLAM method, this comparison can reflect the geometric accuracy of the LDSO derived point cloud. The deviation distance calculated by the least square plane method between the LDSO and ORB-SLAM derived point cloud is shown in Table 4 and Figure 9. The deviation distance results between the ORB-SLAM and the laser-scanned point cloud were also listed in Table 4 as a comparison reference.

Point Clouds	Mean distance (m)	Standard deviation (m)
LDSO and ORB-SLAM	0.0069	0.0079
ORB-SLAM and Laser reference	0.0381	0.0429

Table 4. Deviation distances comparison between LDSO and ORB-SLAM, and the laser reference.

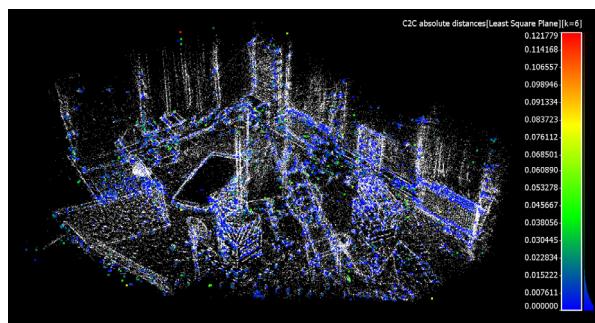
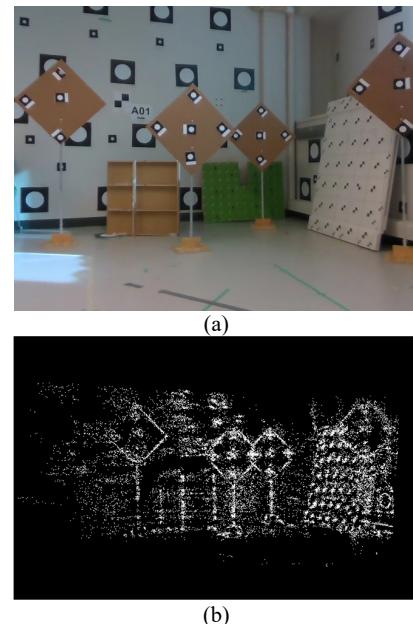
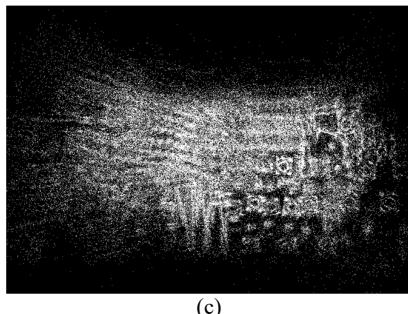


Figure 9. The C2C distances between the LDSO and ORB-SLAM derived point clouds in color scale mode.

In Figure 8 and 9, it can be seen that the point cloud map generated by LDSO with the EuRoC dataset has a solid 3D reconstruction ability and high geometric accuracy. The LDSO 3D reconstruction was tested with another environment shown in Figure 10a. Two videos were collected by moving an Intel RealSense D435 camera and flying the AR. Drone around the test scene; two datasets were created with recorded videos respectively; point cloud maps were then generated by running LDSO with the two datasets respectively shown in Figure 10b and 10c. In Figure 10b, the point cloud map generated with D435 camera images achieves a high-quality 3D reconstruction of the test scene; position differences of standing boards and wall corners were clearly shown. While a 2D point cloud map similar to Figure 5 generated with AR. Drone camera images is shown in Figure 10c.



(b)



(c)

Figure 10. Semi-Dense point clouds generated by LDSO: (a) test scene; (b) with Intel RealSense D435 camera images; (c) with AR. Drone camera images.

4.4.3 Noise and Sparse Level: The noise level of each SLAM derived point clouds were evaluated in Table 2. The remaining outliers of filtered point clouds were further cleaned by direct trimming based on visual checks. The noise points percentages were calculated by dividing the raw points number with the cleaned points number. It can conclude that the LSD-SLAM derived point clouds map is noisiest, LDSO is nosier, and ORB-SLAM is quite noise-free.

The noise level of the SLAM derived point clouds may relate to the sparse level. Due to the high sparsity of ORD-SLAM derived point clouds, it can't be directly applied for robotic navigation tasks even its map accuracy is high. A probabilistic semi-dense mapping module proposed in (Mur-Artal and Tardos, 2015) working together with the ORB-SLAM to gain a semi-dense map provides a good solution. The LSD-SLAM, on the other hand, provides a semi-dense map with medium geometric accuracy, which is more applicable to navigation applications.

4.4.4 Discussion: In our indoor 3D reconstruction tests with a low-cost MAV, it can conclude that the ORB-SLAM algorithm provides the highest accuracy and noise-free point clouds map, but the point cloud is too sparse to be directly applied to many further applications. LSD-SLAM generates medium accuracy and noisiest point clouds, but, instead, the map is dense enough to embed critical information of the environment, specifically along edges. LDSO produces the worst accuracy and medium noisy point cloud map, and it fails to reconstruct 3D structures. However, further tests of LDSO with global shutter camera image datasets show an outstanding geometric accuracy and 3D reconstruction ability.

5. CONCLUSION

In this paper, the quality of 3D point cloud maps generated by monocular SLAM algorithms including LSD-SLAM, ORB-SLAM, and LDSO is compared. A custom evaluation dataset is created comprising images extracted from videos recorded by flying a MAV in the test environment and a ground-truth reference point cloud of the same area acquired by a terrestrial laser scanner. After processing outlier filtering of the SLAM derived point clouds and registering these point clouds with the reference properly, the quality of the SLAM derived point clouds maps are evaluated with their geometric accuracy by calculating the deviation distance between each SLAM-derived point cloud and the laser-scanned reference.

In our MAV indoor environment 3D reconstruction tests, it concludes that the ORB-SLAM provides the highest accuracy,

LSD-SLAM generates medium accuracy, and LDSO gives worse geometric accuracy point cloud map. The noise level of LSD-SLAM is high, the LDSO is lower, and ORB-SLAM is almost noise-free. For future applications such as navigation and 3D reconstruction, LSD-SLAM is the only possible solution. The ORB-SLAM and LDSO derived point clouds fail because of low points density and poor spatial depth information representation respectively. Additional tests also show high accuracy and 3D reconstruction ability of LDSO with global shutter camera images. Thus, LDSO is not suitable for 3D reconstructions tasks using MAVs with low cost rolling shutter cameras onboard.

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Camilo Cortes, who helped with capturing the laser-scanned points cloud, from the Autonomous Mapping Lab, University of Calgary. The authors also would like to thank Dr. Xiang Gao, who provided the software support for point cloud extraction of LDSO, from the Computer Vision Group, Technical University of Munich.

REFERENCES

- Ahmad Fuad, N., Yusoff, A.R., Ismail, Z., Majid, Z., 2018. Comparing the Performance of Point Cloud Registration Methods for Landslide Monitoring Using Mobile Laser Scanning Data. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XLII-4/W9, 11–21. <https://doi.org/10.5194/isprs-archives-XLII-4-W9-11-2018>
- Bodin, B., Wagstaff, H., Saecdi, S., Nardi, L., Vespa, E., Mawer, J., Nisbet, A., Lujan, M., Furber, S., Davison, A.J., Kelly, P.H.J., O Boyle, M.F.P., 2018. SLAMBench2: Multi-Objective Head-to-Head Benchmarking for Visual SLAM, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–8. <https://doi.org/10.1109/ICRA.2018.8460558>
- Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R., 2016. The EuRoC micro aerial vehicle datasets. *Int. J. Rob. Res.* 35, 1157–1163. <https://doi.org/10.1177/0278364915620033>
- Delmerico, J., Scaramuzza, D., 2018. A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2502–2509. <https://doi.org/10.1109/ICRA.2018.8460664>
- Dubbelman, G., Browning, B., 2015. COP-SLAM: Closed-Form Online Pose-Chain Optimization for Visual SLAM. *IEEE Trans. Robot.* 31, 1194–1213. <https://doi.org/10.1109/TRO.2015.2473455>
- Engel, J., Koltun, V., Cremers, D., 2018. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 611–625. <https://doi.org/10.1109/TPAMI.2017.2658577>
- Engel, J., Schöps, T., Cremers, D., 2014. LSD-SLAM: Large-Scale Direct Monocular SLAM. *Springer; Cham*, pp. 834–849. https://doi.org/10.1007/978-3-319-10605-2_54
- Engel, J., Usenko, V., Cremers, D., 2016. A Photometrically Calibrated Benchmark For Monocular Visual Odometry. arXiv Preprint. [arXiv1607.02555](https://arxiv.org/abs/1607.02555).

- Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., Scaramuzza, D., 2016. Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle. *J. F. Robot.* 33, 431–450. <https://doi.org/10.1002/rob.21581>
- Gao, X., Demmel, N., 2018. tum-vision/LDSO [WWW Document]. GitHub. URL <https://github.com/tum-vision/LDSO>
- Gao, X., Wang, R., Demmel, N., Cremers, D., 2018. LDSO: Direct Sparse Odometry with Loop Closure, in: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2198–2204. <https://doi.org/10.1109/IROS.2018.8593376>
- Gaud, A., 2017. ayushgaud/ORB_SLAM2 [WWW Document]. GitHub. URL https://github.com/ayushgaud/ORB_SLAM2
- Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. *Int. J. Rob. Res.* 32, 1231–1237. <https://doi.org/10.1177/0278364913491297>
- Girardeau-Montaut, D., 2015. CloudCompare: 3D point cloud and mesh processing software. Webpage: <http://www.cloudcompare.org>.
- Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W., 2010. A Tutorial on Graph-Based SLAM. *IEEE Intell. Transp. Syst. Mag.* 2, 31–43. <https://doi.org/10.1109/MITS.2010.939925>
- Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots* 34, 189–206. <https://doi.org/10.1007/s10514-012-9321-0>
- Jeroen Zijlmans, 2017. ORB-Slam2: Implementation on my Ubuntu 16.04 with ROS Kinect [WWW Document]. Medium. URL <https://medium.com/@j.zijlmans/orb-slam-2052515bd84c> (accessed 2.12.19).
- Kim, P., Chen, J., Cho, Y.K., 2018. SLAM-driven robotic mapping and registration of 3D point clouds. *Autom. Constr.* <https://doi.org/10.1016/j.autcon.2018.01.009>
- Klein, G., Murray, D., 2007. Parallel tracking and mapping for small AR workspaces, in: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*. IEEE, pp. 1–10. <https://doi.org/10.1109/ISMAR.2007.4538852>
- Krombach, N., Droeschel, D., Behnke, S., 2017. Combining feature-based and direct methods for semi-dense real-time stereo visual odometry, in: *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-3-319-48036-7_62
- Kumar, V., Michael, N., 2017. Opportunities and challenges with autonomous micro aerial vehicles, in: *Springer Tracts in Advanced Robotics*. https://doi.org/10.1007/978-3-319-29363-9_3
- Liu, M., 2016. Robotic Online Path Planning on Point Cloud. *IEEE Trans. Cybern.* 46, 1217–1228. <https://doi.org/10.1109/TCYB.2015.2430526>
- Moghaddame-Jafari, B., 2016. Deflection Measurement through 3D Point Cloud Analysis. George Mason University.
- Mur-Artal, R., Tardos, J., 2015. Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM, in: *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation. <https://doi.org/10.15607/RSS.2015.XI.041>
- Mur-Artal, R., Tardos, J.D., 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* 33, 1255–1262. <https://doi.org/10.1109/TRO.2017.2705103>
- Parrot Drones SAS., n.d. Quadcopter AR Drone 2.0 Elite Edition | Parrot Store Official [WWW Document]. URL <https://www.parrot.com/ca/drones/parrot-ardrone-20-elite-edition> (accessed 9.10.19).
- Pottmann, H., Leopoldsdorfer, S., Hofer, M., 2004. Registration without ICP. *Comput. Vis. Image Underst.* 95, 54–71. <https://doi.org/10.1016/j.cviu.2004.04.002>
- Sadat, S.A., Chutskoff, K., Jungic, D., Wawerla, J., Vaughan, R., 2014. Feature-rich path planning for robust navigation of MAVs with Mono-SLAM, in: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3870–3875. <https://doi.org/10.1109/ICRA.2014.6907420>
- Scaramuzza, D., Fraundorfer, F., 2011. Tutorial: Visual odometry. *IEEE Robot. Autom. Mag.* <https://doi.org/10.1109/MRA.2011.943233>
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., 2012. A benchmark for the evaluation of RGB-D SLAM systems, in: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 573–580. <https://doi.org/10.1109/IROS.2012.6385773>
- Taketomi, T., Uchiyama, H., Ikeda, S., 2017. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* 9, 16. <https://doi.org/10.1186/s41074-017-0027-2>
- Williams, B., Cummins, M., Newman, P., Reid, I., Tardós, J., 2009. A comparison of loop closing techniques in monocular SLAM. *Rob. Auton. Syst.* 57, 1188–1197. <https://doi.org/10.1016/j.robot.2009.06.010>
- Yang, Z., Gao, F., Shen, S., 2017. Real-time monocular dense mapping on aerial robots using visual-inertial fusion, in: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4552–4559. <https://doi.org/10.1109/ICRA.2017.7989529>
- Younes, G., Asmar, D., Shammas, E., Zelek, J., 2017. Keyframe-based monocular SLAM: design, survey, and future directions. *Rob. Auton. Syst.* 98, 67–88. <https://doi.org/10.1016/j.robot.2017.09.010>