

Evaluating the Mapping Quality of Monocular SLAM Solutions for Micro Aerial Vehicles

Jie Wang ^{1,*}, Camilo Cortes ¹ and Mozhdeh Shahbazi ¹

¹ Department of Geomatics Engineering, University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4, Canada;

* Correspondence: wang4@ucalgary.ca

Received: date; Accepted: date; Published: date

Abstract: Monocular simultaneous localization and mapping (SLAM) has attracted much attention in mobile-robotics domain over the past decade 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 have 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 geometry accuracy of the global mapping. Since there are no ground-truth information of the testing environment in existing visual SLAM benchmark datasets (e.g. EuRoC, TUM and KITTI), in this study an evaluation dataset using a quadcopter and a terrestrial laser scanner is created. 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 solutions was mainly conducted on geometry 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.

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

1. Introduction

In order to explore unknown environments autonomously, mobile robotic systems such as drones 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 being studied in computer vision and mobile robotics fields for the past two decades [1]. The SLAM problem is especially challenging for Micro Aerial Vehicles (MAVs) [2] due to payload limit, power consumption, and high degrees of freedom motion. Thus, Monocular SLAM [3] solutions have attracted much attention of both commercial and research sectors due to being advantageous in terms of small size, light weight, low power consumption and cost effectiveness.

Generally speaking, a keyframe-based monocular SLAM system [4] is composed of the Visual Odometry (VO) [5][6], Global Optimization (GO) and Global Mapping (GM) parts [7]. The VO component is responsible for estimating sequential pose (position and orientation) of the camera. In VO, a small portion of the map and only the relative camera motions are computed; the global consistency between these small maps is not considered [8]. The GO component includes loop closure [9] and global pose-graph optimization [10] elements to handle the sequentially propagated errors, drifts, and global geometric consistency of the global map and camera trajectory. During loop closure, the current image is constantly compared to the previous observed image frames, and a loop is

detected when 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 position of the 3D structure points [11]. The GM refers to the three-dimensional (3D) reconstruction of the whole explored environment. In monocular SLAM systems, the environment is represented as a point clouds map. The point clouds of the environment generated by SLAM systems can either be expressed as meshed surfaces [12] or tree-based structures [13], and can even be directly used for high-level robotic applications such as path planning, navigation and obstacle avoidance [14].

Since the majority research studies of SLAM initiate from VO [15][7], there have been an abundant of studies [16][17][18][19][20] that report the quality of the camera trajectory estimation evaluated by the Absolute Trajectory Error (ATE) and the Absolute Translation Root-Mean-Square Error (AT-RMSE) metrics. However, very limited resources exist for evaluating the mapping quality of modern monocular SLAM solutions. Even though the VO component plays an important role in evaluating the states of camera, the mapping representation of the environment is equally important to improve the autonomy of the mobile robotic systems. Not mention direct mapping applications such as the planning and monitoring of construction processes and building conditions [21].

In this paper, we compare the quality of the 3D maps generated by the state-of-the-art monocular SLAM solutions implemented on a MAV. These SLAM solutions include Large-Scale Direct Monocular SLAM (LSD-SLAM) [22], ORB-SLAM [19] and Direct Sparse Odometry with Loop Closure (LDSO) [20]. These methods were selected due to their superior trajectory-estimation quality, their ability to create semi-dense or dense maps, and being equipped with global optimization components. The accuracy of the generated maps was tested using a high-fidelity point cloud generated by a terrestrial laser scanner. To the best of our knowledge, this is the first comparison of these monocular SLAM algorithms applied on MAVs in terms of the mapping quality.

The article is organized as follows. Section 2 reviews the selected monocular SLAM algorithms from the mapping perspective. Section 3 presents experiment settings and the hardware used for implementing the SLAM algorithms. Section 4 presents and discusses the results. Finally, Section 5 concludes the article.

2. Monocular SLAM Algorithms

Two main criteria are considered for selecting these solutions, first, being a full monocular SLAM solution including a global-optimization component and, second, generating dense or semi-dense 3D maps of the environment. Since GO are important to improve the mapping quality [4][23], monocular SLAM systems such as Parallel Tracking and Mapping (PTAM) [8], Semi-Direct Visual Odometry (SVO) [24] and Dense Piecewise Planar Tracking and Mapping (DPPTAM) [25] are not considered. Besides advantages of size, weight, and power characteristics, the visual SLAM system captures rich information about the environment from images. In the perspective of the way the system uses the image data, visual SLAM solutions can be categorized as either direct or indirect (i.e. feature-based) methods [26]. The indirect methods need to pre-process the raw image data to generate intermediate representations as geometric features (both the keypoint and descriptor), and then utilize the extracted features to localize the camera and reconstruct the environment. Instead, direct methods use the image pixel intensities (photometry) directly. From the mapping perspective, sparse methods conduct 3D mapping reconstruction with a selected set of independent pixels such as corners, while dense methods attempt to reconstruct all pixels in the 2D image domain or most pixels with intensity salience (e.g. high gradient). In fact, the fundamental difference between sparse and dense solutions is the usage of the geometry prior. The geometry parameters (keypoint positions) are conditionally independent in sparse methods, while dense (or semi-dense) methods make use of the connectedness of the past image region to formulate a geometry prior [26].

The key features of the three SLAM solutions are summarized in Table 1. The LSD-SLAM is one of the most recognizable monocular SLAM solutions using direct method. The direct approach doesn't

need feature extraction and thus avoid the corresponding artifacts. The camera is localized by optimizing directly over image pixel intensities. The LSD-SLAM system employs a photometric error as well as a geometric prior thus can perform dense or semi-dense reconstructions of the environment. The ORB-SLAM is the most reliable and complete feature-based solution for monocular SLAM [27]. As an extension of PTAM, updated research progress including ORB features, a loop closure detection and pose graph optimization blocks are incorporated. Even though a directly-obtained dense map is preferred for high-level robotics applications [15][28], a method to gain a dense or semi-dense mapping from the sparsely generated point clouds is proposed in [23]. A probabilistic semi-dense mapping module is developed to work together with the ORB-SLAM. The integrated system performs a rich semi-dense reconstruction over keyframes in real-time on a conventional computer without GPU. LDSO becomes a full SLAM solution after integrating the loop closure and global map optimization into the Direct Sparse Odometry (DSO) [20]. DSO has a novel sparse and direct structure, which combines the benefits of direct methods with the flexibility of sparse approaches [26]. The VO part of the LDSO system performs optimizations to directly minimize photometric errors on a sparse set of points. Thus, unlike feature-based sparse system tracking only corners, it makes the system to track any pixels with large enough intensity gradients. It computes points on all gradient-rich areas including edges. The density of the points cloud mapping can be changed by specifying how many points to be kept in the active window, thus it is able to establish a sparse points cloud map as well as semi-dense reconstructions similar to the LSD-SLAM [26].

Table 1. Key features of the selected visual SLAM systems

System	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

3. Experiments

This section describes the experiment setups and platforms to implement the monocular SLAM solutions on a MAV. We utilized a Parrot AR Drone 2.0 [29] to collect videos of the test environment, conducted camera calibrations and then processed the test data properly for each SLAM solutions.

3.1. Platforms

The Parrot AR. Drone 2.0 is a commercial quadrotor shown in Figure 1 (a). The main highlights of the drone are its low price, robustness to crashes and high safety for indoor usage. Even though neither the hardware nor software running onboard can easily be modified, the communication and control with the quadrotor through Wi-Fi are convenient and powerful thanks to continuing academic researches and development [30][31][32]. The AR. Drone 2.0 has a front facing camera shown in Figure 1 (b) in a green rectangle that can be configured to stream 720p (1280*720) videos at 30 fps. The AR. Drone 2.0 has a master USB port with a standard USB-A connector as shown in Figure 1 (b) with a blue rectangle, which can be used for video recording purpose.



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

Being a high-tech toy, the AR. Drone 2.0 also comes with all required software to fly. Using the free available, AR.FreeFlight App [33] shown in Figure 2 provided by Parrot, it's straightforward to control the AR. Drone 2.0 using a smart phone or tablet. Moreover, the video of the front facing camera can be easily recorded and stored into a USB by one click of the right "REC" button, the "REC" will keep flashing until another click to finish the video recording.



Figure 2. The AR.FreeFlight App running on an iPhone connected with a AR. Drone 2.0. The altitude and yaw are controlled by the right button, the horizontal movement commands can be sent by tilting the phone to the respective direction by keep holding the left button.

3.2. Dataset

There are three most commonly used datasets for visual SLAM system evaluations including the EuRoC dataset [34], TUM RGB-D and monoVO dataset [18][35], and KITTI dataset [36]. The recent proposed SLAM solutions including ORB-SLAM and LDSO initiativly show the evaluation results by implementing these datasets. Among all three popular datasets, the EuRoC dataset contains 11 image sequences collected on-board by flying a MAV around a machine hall in ETH Zurich and two different vicon rooms. The datasets contain stereo images, synchronized IMU measurements, and accurate motion and structure ground-truth of cameras. Because we focused on comparing monocular SLAM solutions specifying to MAVs applications, we first tested all three selected SLAM systems by the EuRoC dataset. However, because all above datasets including EuRoC dataset doesn't provide the ground-truth map of the testing environment, we need to create our own dataset for evaluating the mapping quality of the three SLAM solutions.

We recorded videos of our lab by flying the drone as well as scanned the same area by using a terrestrial laser scanner. The videos were recorded as 1280*720 resolution at 30 fps by the front facing camera of the AR. Drone 2.0. The laser scanner point clouds were created by a FARO Focus3D X. Due to different data format requirement for implementing the three SLAM algorithms, we prepared

datasets from our recorded videos for each respectively. Moreover, the ORB-SLAM package closes the mapping window immediately when the pipeline finishes running, and it doesn't allow to view and save the generated point clouds afterwards. Some modifications of the original codes to run our customized image dataset as well as to save the point clouds file for conducting comparisons are conducted.

3.2.1. LSD-SLAM

When the LSD-SLAM was proposed, the only available dataset among above three was the TUM RGB-D dataset created by the authors. The images were first extracted from recorded videos at each frame sequentially, then as explained in [37], the package can be executed by using the "dataset_slam" command. The generated point clouds map can be saved by keep pressing "p" for a few seconds when the "lslam_viewer" window is selected. The file will be saved as "pc.ply" in the "lslam_viewer" folder shown in Figure 3 a.

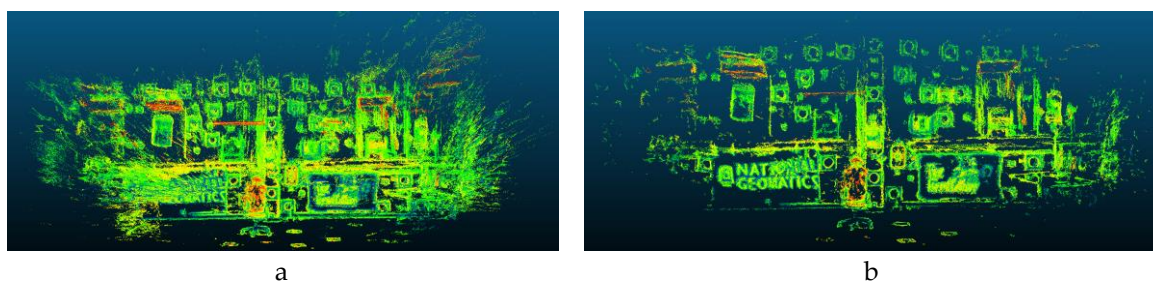


Figure 3. Point clouds derived by LSD-SLAM: (a) raw data; (b) SOR filtered data

3.2.2. ORB-SLAM

It's very straightforward to evaluate ORB-SLAM algorithm by using the commands provided by the author [38] for all three popular datasets. However, in order to execute the pipeline with our collected image data, we prepared our data as same format as the EuRoC dataset. The images are sorted sequentially and stored in the "data" folder, a "data.csv" are created containing the timestamp and image filename information, and a "sensor.yaml" was modified specifying the camera calibration parameters. The camera calibration file and image timestamp file also need to be modified and placed inside the "Monocular" folder following the software instruction [38]. The codes of several files including "System.cc", "System.h" and "CMakeLists.txt" were modified in order to save the point clouds as a file [39]. A new file "mono_ardrone.cc" was also created based on "mono_tum.cc" to create "mono_ardrone.exe" as an executable file for our dataset. A file will be saved as "pointcloud.pcd" in the "ORB_SLAM2" folder shown in Figure 4.

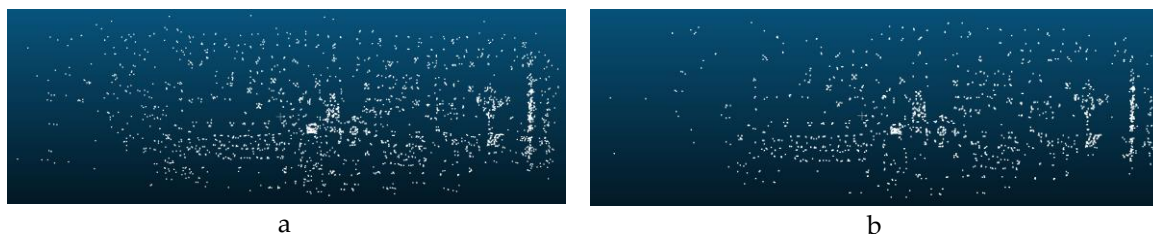


Figure 4. Point clouds derived by ORB-SLAM: (a) raw data; (b) SOR filtered data

3.2.3. LDSO

Similar to ORB-SLAM, the LDSO package can be easily evaluated by using the provided commands [40] for all three popular datasets. By creating our image dataset as same format as the EuRoC dataset, the LDSO can be executed with our dataset using the EuRoC dataset command after modifying the camera calibration parameters of "EUROC.txt". When the program finishes running,

shut down the “Pangolin” window will automatically save the points cloud map as a “pointcloud.ply” file shown in Figure 5.

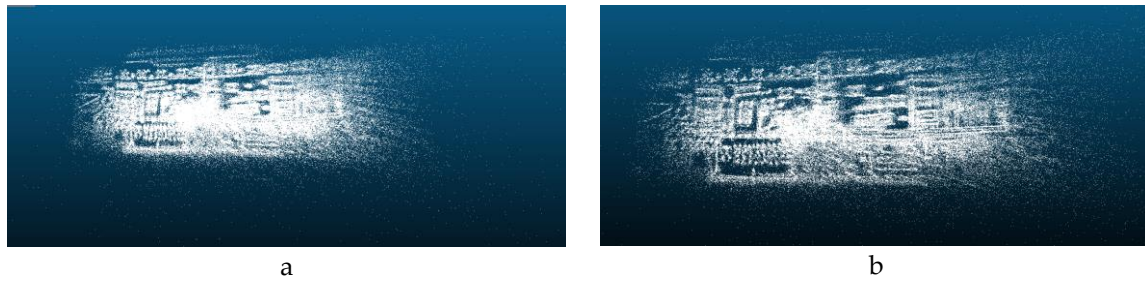


Figure 5. Point clouds derived by LDSO: (a) raw data; (b) SOR filtered data

3.2.4. Laser point clouds map

The reference ground-truth map of the testing area was acquired using a FARO Focus3D X terrestrial laser scanner with $\pm 2\text{mm}$ resolution shown in Figure 6. The SLAM-derived point clouds are to be compared against the reference respectively for the mapping quality evaluations in the next section.

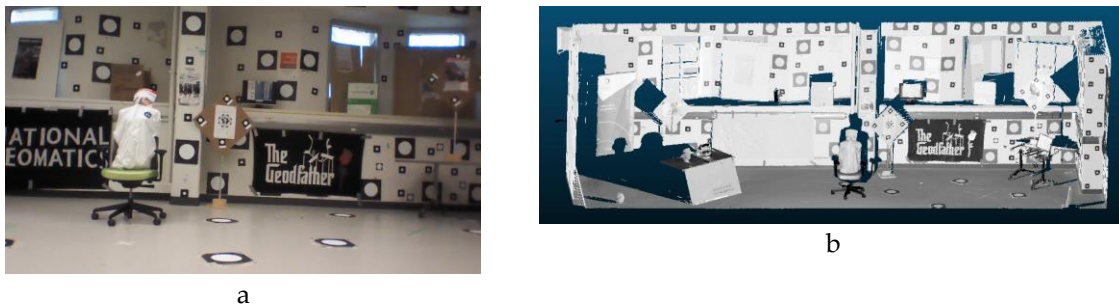


Figure 6. High-accuracy ($\pm 2\text{mm}$) points cloud map of the testing area obtained by a terrestrial laser scanner FARO Focus 3D.

4. Evaluation

Point cloud to point cloud (C2C) comparison is to determine the similarity of the two given point clouds. In this section, the mapping quality of LSD-SLAM, ORB-SLAM and LDSO are evaluated by conducting C2C comparison of the extracted point clouds with the reference respectively. To conduct the C2C comparison, the outliers of each point clouds are filtered first to avoid possible following point cloud registration failures. Then, point-pair based alignment between the SLAM-derived point clouds and the reference point cloud is performed as the coarse registration process. Since there are no inertial measurements integrated in these monocular SLAM solutions, the scales of the point clouds are different from the true scale. Therefore, the scale of the SLAM-derived point clouds is also adjusted to be as same as the scale of the reference point clouds during the coarse registration process. The registration results were refined by iterative closest point (ICP) processing on the coarse registration results.

Then, the C2C deviation distances between each SLAM derived point clouds and the reference point clouds are calculated on the fine registration results using four C2C distance computation methods (nearest neighbor, least square plane, 2D1/2 triangulation and quadric) in the CloudCompare [41]. The distance computation results are compared and the method has the smallest standard deviation (least square plane method) is selected to evaluate the SLAM derived point clouds accuracy [21]. The whole evaluation process comprises the raw point clouds outlier filtering, C2C registration, C2C deviation distance calculation and results analysis.

4.1. Outlier Points Filtering

In order to get a cleaner point cloud, a Statistical Outlier Removal (SOR) filter is applied. For each point, the average distances to its k nearest neighbors are computed. By assuming that the resulted distribution is Gaussian, points whose mean distances are outside a given interval can be considered as outliers and trimmed from the point clouds. The numbers of raw and filtered points are list in Table 2. The raw and filtered point clouds derived by LSD-SLAM, ORB-SLAM and LDSO are shown in Figure 3, 4 and 5 respectively.

SLAM solutions	Points number			
	Raw PC	Filtered PC	Trimmed PC	Noise Points %
Laser scanner	453,090	NA	NA	NA
LSD-SLAM	1,169,856	560,735	557,055	52.38%
ORB-SLAM	3,402	3,131	3099	8.91%
LDSO	248,341	215,482	201,873	18.71%

Table 2. Points number of the raw, filtered and trimmed point clouds.

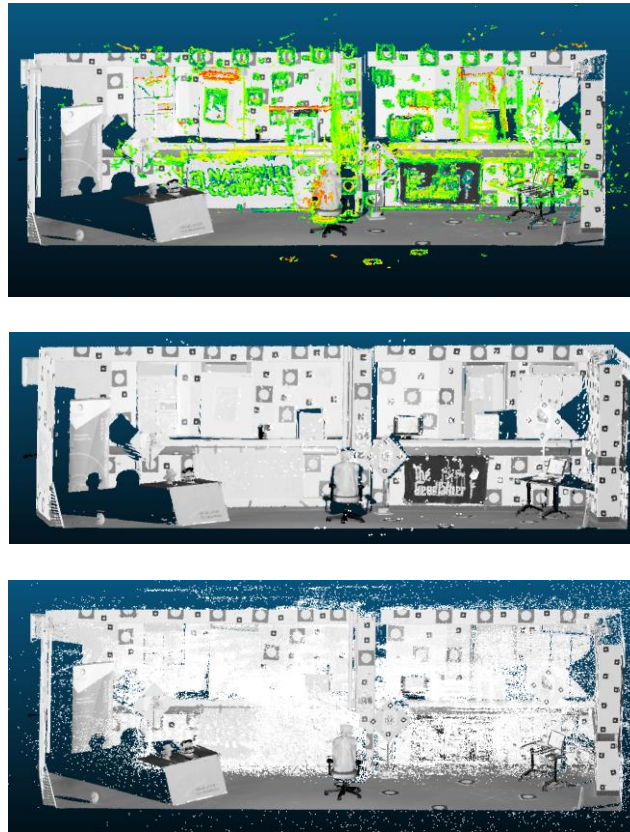
4.2. C2C Registration

A proper C2C registration between the SLAM derived and laser scanner reference point clouds is essential to evaluate the mapping accuracy. It is conducted by manual point-pair based alignment and automatic ICP based registration. The point-pair based alignment is used to coarsely initialize the registration. After the two clouds are roughly registered, the two clouds can be further fine registered using the ICP algorithm. Due to the fact that the SLAM derived point clouds and laser scanner reference point cloud have a very low overlap and different scale, thus point-pair based registration initialization is conducted first. During the point-pair based alignment process, at least three equivalent point pairs in both point clouds need is picked manually using the “Align” tool in CloudCompare. The transformation matrix and root-mean-square error (RMSE) are calculated based on selected points, and the scale of the align point cloud is also adjusted to the scale of the reference point cloud.

A fine registration is 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 two point sets can be denoted 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 $E(R, T) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - Ry_i - T\|^2$. In the ICP fine registration process, the registration process can either be stopped after a maximum number of iterations, or as soon as the error (RMSE) difference becomes lower than a given threshold. The transformation matrix and RMSE of the point-pair alignment and ICP registration are listed in Table 3. The C2C registration results between each SLAM derived and laser scanner reference point clouds are shown in Figure 7.

Point Clouds	C2C Registration				
	Point-pair alignment		ICP registration		
	Transformation matrix	RMSE	Transformation matrix	RSME	
LSD-SLAM	$\begin{bmatrix} 0.976 & 0.260 & -0.020 & 0.667 \\ -0.235 & 0.910 & 0.371 & -3.879 \\ 0.113 & -0.354 & 0.940 & -4.777 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$	0.043	$\begin{bmatrix} 1.000 & 0.014 & -0.002 & 0.110 \\ -0.014 & 1.000 & 0.027 & 0.027 \\ 0.002 & -0.027 & 1.000 & -0.168 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$	0.164	
ORB-SLAM	$\begin{bmatrix} 0.920 & 0.279 & 0.080 & 0.069 \\ -0.285 & 0.919 & 0.069 & -1.187 \\ -0.056 & -0.089 & 0.959 & -4.326 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$	0.003	$\begin{bmatrix} 0.969 & 0.018 & 0.016 & 0.000 \\ -0.017 & 0.967 & -0.060 & -0.089 \\ -0.017 & 0.060 & 0.967 & 0.166 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$	0.094	
LDSO	$\begin{bmatrix} 1.113 & -0.178 & 0.169 & -2.751 \\ 0.239 & 0.956 & -0.571 & 2.094 \\ -0.052 & 0.593 & 0.971 & 7.674 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$	0.066	$\begin{bmatrix} 0.972 & 0.233 & -0.012 & -0.245 \\ -0.167 & 0.728 & 0.665 & -3.789 \\ 0.164 & -0.645 & 0.747 & -0.275 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$	0.288	

Table 3. Transformation matrix and RMSE of the point-pair alignment and ICP registration.

**Figure 7.** The C2C registration results between each SLAM derived and laser scanner reference point clouds.

4.3. C2C Deviation Distance Calculation

The most basic C2C distance computation approach is the nearest neighbor distance method [42]. The nearest neighbor distance method searches the nearest point in the reference for each point in the compared 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 [42]. This is especially true if the reference cloud has a low density, which is the case for the LSD-SLAM derived point clouds who has a higher point density compared to the reference point clouds as 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 modelling methods, least square plane, 2D1/2 triangulation and quadric were used for the C2C deviation distance calculations. These modeling methods estimate the least-square best fitting plane (plane, triangulation and quadric respectively) that goes through the nearest point and its neighbors either on a given number of neighbors or inside a defined spherical radius of the reference cloud. The mean distance and standard deviation of each local modeling methods can also be calculated using above equations. These results are listed in Table 4.

Local points model	Mean distance	Standard deviation
LSD-SLAM derived point clouds		
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-SLAM derived point clouds		
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 derived point clouds		
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 4. The mean distance and standard deviation results by nearest neighbour, least square plane, 2D1/2 triangulation and quadric C2C distance calculations.

4.3. Results analysis

In Table 4, It can be seen that the least square plane method provides deviation distance calculation results with the smallest standard deviation compared to 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 scanner reference point clouds similar to [21]. The C2C deviation distance results calculated by least square plane method presented in color scale mode are shown in Figure 8.

In Table 4, it can be concluded that the overall geomatic accuracy of point clouds derived by the SLAM solutions evaluated by the mean distance ranked from high to low is ORB-SLAM, LSD-SLAM and LDSO. This can also be clearly viewed in Figure 8.

Besides the geomatic accuracy, the noise level of each SLAM derived point clouds is evaluated. The remaining outliers of filtered point clouds are further cleaned by direct trimming based on visual check. The noise points percentages are calculated by the 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 noisier and ORB-SLAM is quite noise-free. The noise level of the derived point clouds may relate to the sparse level of the corresponding SLAM solutions.

Beside above quantitative analysis, several direct observations can be noticed in Figure 8. Due to the high sparsity of ORD-SLAM derived point clouds, it can't be directly applied for robotic navigations even its map accuracy is high. A probabilistic semi-dense mapping module proposed in [23] working together with the ORB-SLAM to gain a dense or semi-dense map may provide a good solution. The LSD-SLAM, on the other hand, provides a semi-dense map with medium geomatic accuracy, which is much applicable to navigation applications.

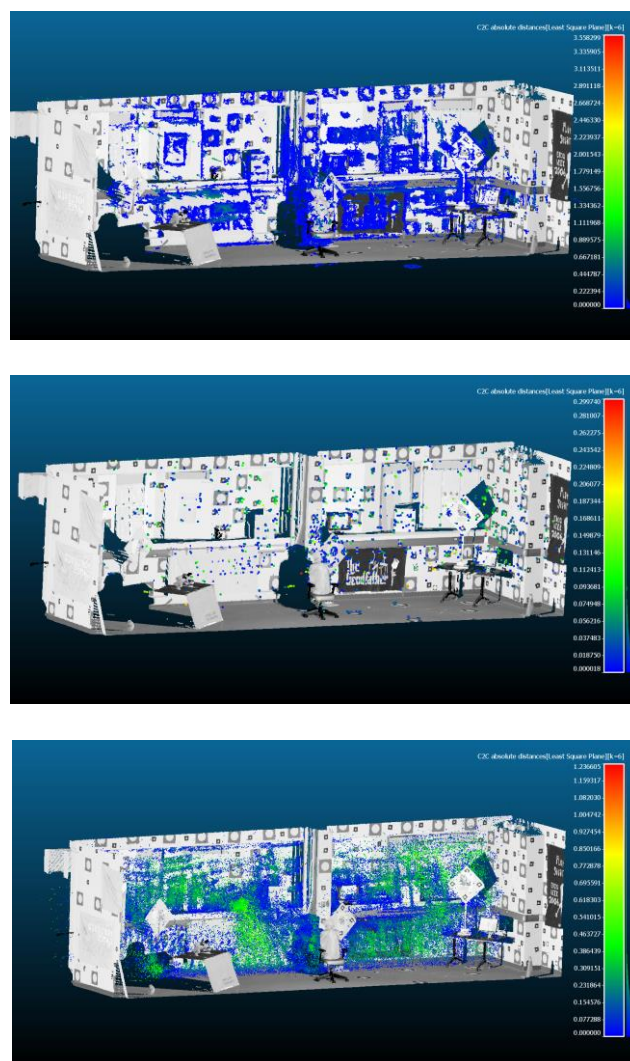


Figure 8. The C2C registration results between each SLAM derived and laser scanner reference point clouds.

In Figure 8, both LSD-SLAM and ORB-SLAM provide decent results in 3D reconstruction perspective. It's obvious they generate points for the chair, table and boards in front of the wall. Depth information are included in the generate point clouds map by LSD-SLAM and ORB-SLAM. While, LDSO provide the worst 3D reconstruction of the testing area. The derived point clouds are generated in 2D manner instead of 3D in our testing environment. It seems LDSO fails to integrate depth information into derived point clouds map. Thus, LDSO may not suitable for 3D reconstruction of the spatial environment.

5. Conclusions

In this paper, the quality of 3D point clouds maps generated by monocular SLAM solutions including LSD-SLAM, ORB-SLAM and LDSO is compared. A custom evaluation dataset is created comprising images extracted from the videos recorded by flying a MAV in the test environment and a high-fidelity point clouds of the same area acquired by a terrestrial laser scanner as the ground truth reference. After processing outlier points filtering of the SLAM derived point clouds and registering these point clouds and reference properly, the quality of the SLAM derived point clouds maps is mainly evaluated on their geometric accuracy by calculating the deviation distance between each SLAM-derived point clouds and the laser scanner reference. In our MAV indoor environment 3D reconstruction test, it concludes that the ORB-SLAM provides the highest accuracy, LSD-SLAM generates medium accuracy and LDSO gives worse geometric accuracy point clouds map. The noise level of LSD-SLAM is high, the LDSO is lower and ORB-SLAM is the lowest. With respect to future applications such as navigation and 3D reconstruction, LSD-SLAM is the only possible solution. The ORB-SLAM and LDSO fails because of low points density and poor spatial depth information representation of the derived point clouds respectively.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y.”, please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: “This research received no external funding” or “This research was funded by NAME OF FUNDER, grant number XXX” and “The APC was funded by XXX”. Check carefully that the details given are accurate and use the standard spelling of funding agency names at <https://search.crossref.org/funding>, any errors may affect your future funding.

Acknowledgments: The authors would like to thank Dr. Xiang Gao with the Computer Vision Group, Technical University of Munich who provided the software support for point clouds extraction function of LDSO.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Grisetti, G.; Kummerle, R.; Stachniss, C.; Burgard, W. A Tutorial on Graph-Based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2010**, *2*, 31–43.
2. Kumar, V.; Michael, N. Opportunities and challenges with autonomous micro aerial vehicles. *Int. J. Rob. Res.* **2012**, *31*, 1279–1291.
3. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067.
4. Younes, G.; Asmar, D.; Shammas, E.; Zelek, J. Keyframe-based monocular SLAM: design, survey, and future directions. *Rob. Auton. Syst.* **2017**, *98*, 67–88.
5. Nister, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.; IEEE; Vol. 1, pp. 652–659.
6. Scaramuzza, D.; Fraundorfer, F. Visual Odometry [Tutorial]. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92.
7. Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 16.
8. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR; IEEE, 2007; pp. 1–10.
9. Williams, B.; Cummins, M.; Newman, P.; Reid, I.; Tardós, J. A comparison of loop closing techniques in monocular SLAM. *Rob. Auton. Syst.* **2009**, *57*, 1188–1197.
10. Dubbelman, G.; Browning, B. COP-SLAM: Closed-Form Online Pose-Chain Optimization for Visual SLAM. *IEEE Trans. Robot.* **2015**, *31*, 1194–1213.
11. Krombach, N. Combining Feature-based and Direct Methods for Semi-dense Real-time Visual SLAM from Stereo Cameras, University of Bonn, 2016.
12. Sadat, S.A.; Chutskoff, K.; Jungic, D.; Wawerla, J.; Vaughan, R. Feature-rich path planning for robust

- navigation of MAVs with Mono-SLAM. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA); IEEE, 2014; pp. 3870–3875.
13. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots* **2013**, *34*, 189–206.
14. Liu, M. Robotic Online Path Planning on Point Cloud. *IEEE Trans. Cybern.* **2016**, *46*, 1217–1228.
15. Yang, Z.; Gao, F.; Shen, S. Real-time monocular dense mapping on aerial robots using visual-inertial fusion. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA); IEEE, 2017; pp. 4552–4559.
16. Bodin, B.; Wagstaff, H.; Saecdi, S.; Nardi, L.; Vespa, E.; Mawer, J.; Nisbet, A.; Lujan, M.; Furber, S.; Davison, A.J.; et al. SLAMBench2: Multi-Objective Head-to-Head Benchmarking for Visual SLAM. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA); IEEE, 2018; pp. 1–8.
17. Delmerico, J.; Scaramuzza, D. A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA); IEEE, 2018; pp. 2502–2509.
18. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems; IEEE, 2012; pp. 573–580.
19. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262.
20. Gao, X.; Wang, R.; Demmel, N.; Cremers, D. LDSO: Direct Sparse Odometry with Loop Closure. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); IEEE, 2018; pp. 2198–2204.
21. Ahmad Fuad, N.; Yusoff, A.R.; Ismail, Z.; Majid, Z. 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.* **2018**, *XLII-4/W9*, 11–21.
22. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Proceedings of the Computer Vision -- ECCV 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, 2014; pp. 834–849.
23. Mur-Artal, R.; Tardos, J. Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM. In Proceedings of the Robotics: Science and Systems XI; Robotics: Science and Systems Foundation, 2015.
24. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.* **2017**, *33*, 249–265.
25. Concha, A.; Civera, J. DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); IEEE, 2015; pp. 5686–5693.
26. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625.
27. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163.
28. Faessler, M.; Fontana, F.; Forster, C.; Mueggler, E.; Pizzoli, M.; Scaramuzza, D. Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle. *J. F. Robot.* **2016**, *33*, 431–450.
29. Parrot Drones SAS. Quadcopter AR Drone 2.0 Elite Edition | Parrot Store Official Available online: <https://www.parrot.com/ca/drones/parrot-ardrone-20-elite-edition> (accessed on Sep 10, 2019).
30. Monajjemi, M. AutonomyLab/ardrone_autonomy Available online: https://github.com/AutonomyLab/ardrone_autonomy.
31. Engel, J.; Sturm, J.; Cremers, D. Camera-based navigation of a low-cost quadcopter. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2012; pp. 2815–2821.
32. Engel, J. Autonomous Camera-Based Navigation of a Quadrotor, Technical University of Munich, 2011.
33. Parrot Drones How to download the AR FreeFlight 2 app? | Parrot Official Available online: <https://www.parrot.com/global/support/products/parrot-ar-drone-2/how-download-ar-freeflight-2-app> (accessed on Oct 15, 2018).
34. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Rob. Res.* **2016**, *35*, 1157–1163.
35. Engel, J.; Usenko, V.; Cremers, D. A Photometrically Calibrated Benchmark For Monocular Visual Odometry. *arXiv* **2016**.
36. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Rob. Res.* **2013**,

32, 1231–1237.

37. Engel, J. tum-vision/lst_slam Available online: https://github.com/tum-vision/lst_slam.
38. Gaud, A. ayushgaud/ORB_SLAM2 Available online: https://github.com/ayushgaud/ORB_SLAM2.
39. Jeroen Zijlmans ORB-Slam2: Implementation on my Ubuntu 16.04 with ROS Kinect Available online: <https://medium.com/@j.zijlmans/orb-slam-2052515bd84c> (accessed on Feb 12, 2019).
40. Gao, X.; Demmel, N. tum-vision/LDSO Available online: <https://github.com/tum-vision/LDSO>.
41. Girardeau-Montaut, D. CloudCompare-Open Source Project Available online: <https://github.com/CloudCompare/CloudCompare>.
42. Moghaddame-Jafari, B. DEFLECTION MEASUREMENT THROUGH 3D POINT CLOUD ANALYSIS, 2016.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).