Chapter 28
Comparative Analysis of Monocular SLAM Algorithms Using TUM and EuRoC Benchmarks

Eldar Mingachev, Roman Lavrenov, Evgeni Magid, and Mikhail Svinin

Abstract Stable and robust path planning and movement in ground mobile robots require a combination of accuracy and low latency in their state estimation. However, state estimation algorithms must provide these qualities under the computational and power constraints of embedded hardware. Simultaneous localization and mapping (SLAM) algorithms are the best choices for state estimation in these scenarios, in addition to their ability to operate without external localization from motion capture or global positioning systems. Moreover, a single-camera setup is the most common solution for robotic platforms, which reduces our domain of interest to the specific SLAM algorithms type—monocular SLAM. Yet, it is still not clear from the existing literature, which monocular SLAM algorithms perform well under the accuracy, latency, and computational constraints of a ground mobile robot with onboard state estimation. This paper evaluates an array of the most recent publicly available monocular SLAM methods: ORB-SLAM2, DSO, and LDSO. The evaluation considers the pose estimation accuracy (alignment error, absolute trajectory error, and relative pose error) while processing the TUM Mono and EuRoC datasets on the specific hardware platform with a balanced amount of computational resources and power consumption. We present our complete results as a benchmark for the research community.

E. Mingachev · R. Lavrenov (✉) · E. Magid
Laboratory of Intelligent Robotic Systems (LIRS), Intelligent Robotics Department, Higher Institute for Information Technology and Intelligent Systems, Kazan Federal University, Kazan, Russian Federation
e-mail: lavrenov@it.kfu.ru
URL: https://kpfu.ru/erobotics

E. Mingachev
e-mail: ERMingachev@stud.kpfu.ru

E. Magid
e-mail: magid@it.kfu.ru

M. Svinin
College of Information Science and Engineering, Ritsumeikan University, Kyoto, Japan
e-mail: svinin@fc.ritsumei.ac.jp

A. Ronzhin and V. Shishlakov (eds.), *Proceedings of 15th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*, Smart Innovation, Systems and Technologies 187, https://doi.org/10.1007/978-981-15-5580-0_28

28.1 Introduction

Dissanayake, Newman, Clark, Durrant-Whyte, and Csorba [1] defined the simultaneous localization and mapping (SLAM) as an ability for an autonomous vehicle to start in an unknown location in an unknown environment and then, using relative observations only, to incrementally build a map of this environment while simultaneously using this map to compute a bounded estimate of vehicle location.

In general, simultaneous localization and mapping (SLAM) methods estimate the egomotion, the 3D motion of a camera within an environment, using only images of single or multiple cameras. Underlying all formulations is a probabilistic model that takes noisy measurements Y as input and computes an estimator X for the unknown, hidden model parameters (3D world model and camera motion). Typically, a maximum likelihood approach is used, which finds the model parameters that maximize the probability of obtaining the actual measurements [2].

SLAM is currently applied to state and pose estimation problems in a variety of domains, including autonomous vehicles, virtual and augmented reality, and robotics. Delmerico and Scaramuzza also noticed [3] that the field has reached a level of maturity such that many commercial products now utilize proprietary SLAM algorithms, and there are several open-source software packages available that offer off-the-shelf pipelines that can be deployed on an end user's platform of choice. However, the most common is a single sensor platform setup [4, 5], which is cheaper and therefore more widespread than multiple sensor platforms. This implies the separate SLAM group—monocular methods.

As Engel, Schöps, and Cremers concluded [6], one of the major benefits of monocular SLAM—and simultaneously one of the biggest challenges—comes with the inherent scale ambiguity: The scale of the world cannot be observed and drifts over time, being one of the major error sources. The advantage is that this allows switching seamlessly between differently scaled environments, such as a desk environment indoors and large-scale outdoor environments. Scaled sensors on the other hand, such as depth or stereo cameras, have a limited range at which they can provide reliable measurements and hence do not provide this flexibility.

The present study offers some comparative results on the accuracy of the most recent and popular open-source monocular SLAM methods (ORB-SLAM2 [7, 8], DSO [2] and LDSO [9]), considering the power constraints of mobile ground robots.

28.2 Related Work

28.2.1 Slam

Indirect methods proceed in two steps. First, the raw sensor measurements are preprocessed to generate an intermediate representation, solving part of the overall problem, such as computing the image coordinates of corresponding points. Second,

the computed intermediate values are interpreted as noisy measurements Y in a probabilistic model to estimate geometry and camera motion. Note that the first step is typically approached by extracting and matching a sparse set of keypoint—however, other options exist, like establishing correspondences in the form of dense, regularized optical flow. It can also include methods that extract and match parametric representations of other geometric primitives, such as line- or curve-segments [10, 11].

In our work, we estimated the accuracy of the recently proposed monocular visual SLAM method that tracks ORB features in real-time, ORB-SLAM [7], specifically, the latest revision—ORB-SLAM2 [8], which has the same core but an improved and optimized workflow. As an indirect method, it provides a good trade-off between accuracy and runtime.

Direct methods in contrast to indirect, which abstract images into a sparse set of feature points, skip the pre-computation step and directly use the actual sensor values, using the entire image information in order to minimize the photometric error. Therefore, as Krombach, Droeschel, and Behnke noticed [12], these methods are computationally very intense and thus much slower than indirect methods. They also often need to use GPUs to achieve real-time performance [13].

In our work, we estimate the accuracy of the recently proposed pure direct method and its latest revision with loop closure ability—DSO [2] and LDSO [9], respectively.

28.2.2 Benchmarks

Datasets

A Photometrically Calibrated Benchmark for Monocular Visual Odometry. Engel, Usenko, and Cremers [14, 15] presented a dataset for evaluating the tracking accuracy of monocular visual odometry and SLAM methods. It contains 50 real-world sequences comprising more than 100 min of video, recorded across dozens of different environments—ranging from narrow indoor corridors to wide outdoor scenes. All sequences contain mostly exploring camera motion, starting and ending in the same position. This allows evaluating the tracking accuracy via the accumulated drift from start to end, without requiring ground truth for the full sequence. In contrast to existing datasets, all sequences are photometrically calibrated which allows to reliably benchmark direct methods. The authors provided exposure times for each frame as reported by the sensor, the camera response function, and dense lens attenuation factors.

The EuRoC Micro-Aerial Vehicle Datasets. Burri et al. [16] proposed visual-inertial datasets collected onboard a *micro*-aerial vehicle. The datasets contain synchronized stereo images, IMU measurements, and accurate ground truth. The first batch of datasets facilitates the design and evaluation of visual-inertial localization algorithms on real flight data. It was collected in an industrial environment and contains millimeter accurate position ground truth from a laser tracking system. The second batch of datasets is aimed at precise 3D environment reconstruction

and was recorded in a room equipped with a motion capture system. The datasets contain 6D-pose ground truth and a detailed 3D scan of the environment. Eleven datasets are provided in total, ranging from slow flights under good visual conditions to dynamic flights with motion blur and poor illumination, enabling researchers to thoroughly test and evaluate their algorithms. All datasets contain raw sensor measurements, spatiotemporally aligned sensor data and ground truth, extrinsic and intrinsic calibrations and datasets for custom calibrations.

The EuRoC MAV dataset includes 11 indoor sequences recorded with a Skybotix stereo VI sensor from a MAV. Accurate ground truth (approx. 1 mm) is recorded using a laser tracker or a motion capture system. Compared to the TUM benchmark, the sequences in EuRoC MAV are shorter and have less variety as they only contain recordings in one machine hall and one laboratory room. Furthermore, EuRoC MAV does not include full photometric data, which is important for benchmarking direct methods.

The KITTI Vision Benchmark Suite. Geiger and Lenz [17] *developed the benchmarks* for the tasks of stereo, optical flow, visual odometry/SLAM, and 3D object detection. The used recording platform was equipped with four high-resolution video cameras, a Velodyne laser scanner, and a state-of-the-art localization system. Proposed benchmarks comprise 389 stereo and optical flow image pairs, stereo visual odometry sequences of 39.2 km length, and more than 200 k 3D object annotations captured in cluttered scenarios (up to 15 cars and 30 pedestrians are visible per image). Results from state-of-the-art algorithms reveal that methods ranking high on established datasets such as Middlebury perform below average when being moved outside the laboratory to the real world. The authors' goal was to reduce this bias by providing challenging benchmarks with novel difficulties to the computer vision community.

The proposed dataset was recorded in an outdoor environment with two stereo cameras and a fast-moving car as a carrier. It provides low-frequency IMU information, which is, however, not time-synchronized with the camera images, and a GPS/INS-based ground truth with an accuracy below 10 cm, which is not so crucial for researches on computer vision algorithms in general, but critical for SLAM benchmark purposes.

28.3 Benchmark Comparisons

A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. Delmerico and Scaramuzza [3] presented the paper, which evaluates an array of publicly available VIO pipelines (MSCKF, OKVIS, ROVIO, VINS-Mono, SVO + MSF, and SVO + GTSAM) on different hardware configurations, including several single-board computer systems that are typically found on flying robots. The evaluation considers the pose estimation accuracy, per-frame processing time, CPU, and memory load while processing the EuRoC datasets, which contain six degree-of-freedom (6DoF) trajectories typical of flying robots.

Table 28.1 Hardware platform specifications

CPU	Intel Core i7-7700HQ, 2800 MHz
RAM	16 GB, DDR4, 2400 MHz
Weight	2.56 kg
Battery	70 Wh Li-Ion
Power consumption	80 W (avg. load)

The conducted survey of the state estimation performance strictly focuses on VO methods usage in flying robotic platforms. The primary goal of the study is to benchmark VO methods performance on hardware and trajectories that are representative of state estimation for a flying robot with limited onboard computing power. In contrast, the current study mainly focuses on the trajectory estimation accuracy of SLAM methods for ground robots.

28.4 Experiments

28.4.1 Hardware

This study focuses on the use of SLAM methods with ground robots, which implies a restriction on energy consumption and the absence of significant constraints on the weight of the platform.

The hardware platform that we consider is the *HP Omen 15-ce057ur* laptop with the technical specifications briefly described in Table 28.1.

28.4.2 Datasets

Considering all the advantages and disadvantages of the approaches reviewed in 2.2, we decided to use the TUM and EuRoC dataset formats for benchmark purposes.

The TUM-Mono dataset [15] is a monocular dataset that consists of 50 indoor and outdoor sequences. It provides photometric camera calibration, but no full ground truth camera trajectories. The camera always returns to the starting point in all sequences, making this dataset very suitable for evaluating accumulated drifts of VO systems.

The EuRoC MAV dataset [16] consists of eleven visual-inertial sequences recorded with two monocular cameras onboard a micro-aerial vehicle while it was manually piloted around three different indoor environments. Within each environment, the sequences increase qualitatively in difficulty as the sequence number increases. For example, Machine Hall 01 is ‘easy,’ while Machine Hall 05 is a more

challenging sequence in the same environment, introducing things like faster motions, poor illumination, etc.

28.4.3 Metrics

TUM. To evaluate the TUM benchmark results, we used the metrics proposed by Engel, Usenko, and Cremers [15] in the corresponding study.

For the evaluation of the metrics, we used the tracked positions $p_1 \dots p_n \in \mathbb{R}^3$ of frames 1 to n and the frame indices $S \subset [1, n]$, $E \subset [1, n]$ for the start and end segments for which aligned ground truth positions $\hat{p} \in \mathbb{R}^3$ are provided. It is important that both S and E contained sufficient poses in a non-degenerate configuration to well-constrain the alignment—hence the loopy motion patterns at the beginning and end of each dataset sequence.

First, we aligned the tracked trajectory with both the start and end segments independently and computed two relative transformations:

$$T_s^{gt} := \underset{T \in \mathrm{Sim}(3)}{\arg\min} \sum_{i \in S} \left(T p_i - \hat{p}_i \right)^2, \tag{28.1}$$

$$T_e^{gt} := \underset{T \in \mathrm{Sim}(3)}{\arg\min} \sum_{i \in E} \left(T p_i - \hat{p}_i \right)^2. \tag{28.2}$$

Then we computed the accumulated drift as:

$$T_{\mathrm{drift}} = T_e^{\mathrm{gt}} \left(T_s^{\mathrm{gt}} \right)^{-1} \in \mathrm{Sim}(3). \tag{28.3}$$

from which we explicitly computed the scale drift:

$$e_s := \mathrm{scale}(T_{\mathrm{drift}}), \tag{28.4}$$

the rotation drift:

$$e_r := \mathrm{rotation}(T_{\mathrm{drift}}), \tag{28.5}$$

and the translation drift:

$$e_t := \| \mathrm{translation}(T_{\mathrm{drift}}) \|. \tag{28.6}$$

Furthermore, we calculated the combined error measure defined by the authors, the alignment error, which equally takes into account the error caused by scale, rotation, and translation drift over the full trajectory. It is the translational RMSE between the tracked trajectories when aligned to the start and end segments:

$$e_{\text{align}} := \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\| T_s^{\text{gt}} p_i - T_e^{\text{gt}} p_i \right\|_2^2}. \tag{28.7}$$

EuRoC. In contrast to the previous dataset format, the EuRoC includes full ground truth camera trajectories, which allows the use of the existing metrics and does not imply a novel error measurement approach. To evaluate the EuRoC benchmark results, we used the metrics proposed by Sturm, Engelhard, Endres, Burgard, and Cremers [18].

Absolute Trajectory Error. Due to the monocular SLAM nature, the estimated trajectory alignment and scale are subjective and each test result will not match directly with the other results and even the ground truth. This is a common mathematical problem for computer vision algorithms. Shinji Umeyama [19] proposed a closed-form solution of the similarity transformation parameter estimation, using the singular value decomposition.

The absolute trajectory error (ATE) measures global consistency by comparing the absolute distances between the estimated and the ground truth trajectory. As both trajectories can be specified in arbitrary coordinate frames, they first need to be aligned using the solution described previously, which finds the rigid-body transformation T that maps the estimated trajectory $P_{[1:n]}$ onto the ground truth trajectory $Q_{[1:n]}$. Given this transformation, we computed the absolute trajectory error at time step i as:

$$F_i := Q_i^{-1}\text{TP}_i. \tag{28.8}$$

Then we evaluated the root-mean-squared error over all time indices of the translational components:

$$\text{RMSE}(F_{1:n}) := \left(\frac{1}{n}\sum_{i=1}^{n}\|\text{translation}(F_i)\|^2\right)^{1/2}, \tag{28.9}$$

where translation(F_i) refers to the translational components of the absolute trajectory error F_i.

Relative Pose Error. The relative pose error (RPE) measures the local accuracy of the trajectory over a fixed time interval Δ. In other words, it measures the difference in the relative motion between pairs of poses, which is useful for estimating the drift. Sturm et al. [18] defined the relative pose error at time step i as:

$$E_i := \left(Q_i^{-1} Q_{i+\Delta}\right)^{-1}\left(P_i^{-1} P_{i+\Delta}\right), \tag{28.10}$$

where $P_{[1:n]}$ is the estimated trajectory and $Q_{[1:n]}$ is the ground truth trajectory.

From a sequence of n camera poses, we obtain individual relative pose errors along the sequence as $m = n - \Delta$. From these errors, as authors proposed, we computed the

root-mean-squared error (RMSE) over all time indices of the translational component as:

$$\mathrm{RMSE}(E_{1:n}, \Delta) := \left(\frac{1}{m} \sum_{i=1}^{m} \|\mathrm{translation}(E_i)\|^2 \right)^{1/2}, \tag{28.11}$$

where $\mathrm{translation}(E_i)$ refers to the translational components of the relative pose error E_i.

To evaluate the global error of a trajectory, we averaged an error over all possible time intervals Δ:

$$\mathrm{RMSE}(E_{1:n}) := \frac{1}{n} \sum_{\Delta=1}^{n} \mathrm{RMSE}(E_{1:n}, \Delta). \tag{28.12}$$

Overall, the RPE metric provides an elegant way to combine rotational and translational errors into a single measure, while the ATE only considers the translational errors. As a result, the RPE is always slightly larger than the ATE (or equal if there is no rotational error). However, rotational errors typically also manifest themselves in wrong translations and are thus indirectly also captured by the ATE. From a practical perspective, the ATE has an intuitive visualization, which facilitates visual inspection. Nevertheless, as the authors noted, the two metrics are strongly correlated.

28.5 Evaluation

TUM. We compared ORB-SLAM2, DSO, and LDSO on the TUM-Mono dataset and evaluated the metrics described in 3.3 TUM section over all dataset sequences, each of which was run 10 times forward and backward to account for non-deterministic behavior.

Figure 28.1 shows RMSE when aligning the evaluated trajectory start and end segments with the provided ground truth trajectory segments (two first plots) along

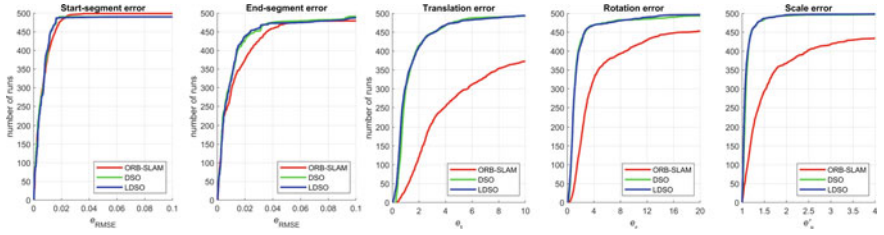

Fig. 28.1 Start and end segments e_{RMSE} and accumulated translational (e_t), rotational (e_r) and scale (e_s) drifts

with the cumulative error plots—accumulated translational, rotational, and scale drifts (the rest). The figure depicts the number of runs whose errors are below the corresponding x-values (thus, closer to left top is better). Note the difference in the order of magnitude—the RMSE within start and end segments is roughly 100 times smaller than the alignment RMSE.

Engel et al. [15] concluded that, due to ground truth origin and the similarity of the experimental results, almost all of the alignment errors originate from accumulated drift, and not from noise in the ground truth. As we can see, the results are very similar for all of the evaluated methods. This means that we can use these metrics for any benchmark with a ground truth of any accuracy as a reference.

Figure 28.2 shows the tracking accuracy when playing sequences only forwards compared to the results obtained when playing them only backward and the combined results (forwards and backward)—switching between predominantly forward motion and predominantly backward motion.

As we can see, the direct methods, DSO and LDSO, are largely unaffected by the motion bias, while the indirect, ORB-SLAM2, performs significantly better for backward motion.

Figure 28.3 shows the color-coded alignment error in all the sequences as defined in the 3.3 TUM segment.

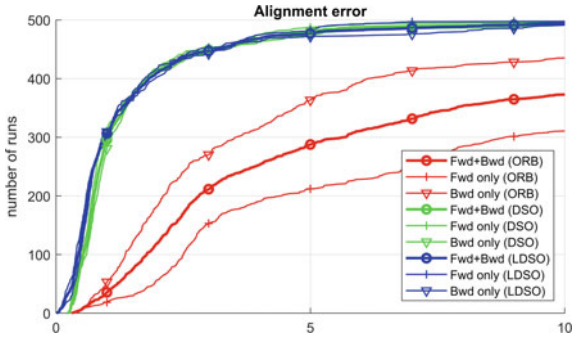

Fig. 28.2 Dataset motion bias

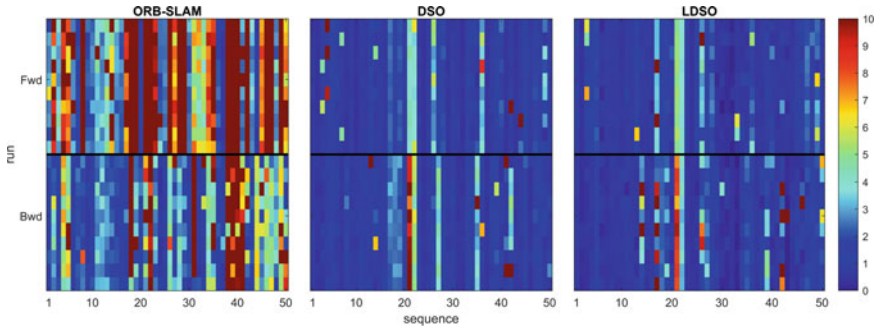

Fig. 28.3 Alignment error e_{align} for each TUM-Mono dataset sequence

Figures 28.1 and 28.3 also prove that comparing to DSO, the novel approach of corner features detection of LDSO does not reduce the VO accuracy of the original system and even slightly improves the overall accuracy due to the loop closure capabilities.

EuRoC. We compared ORB-SLAM2, DSO, and LDSO on the EuRoC MAV dataset and evaluated the metrics described in 3.3 EuRoC section over all dataset sequences for each of the two camera streams which were interpreted as separate dataset sequences with the same ground truth (see '0.0' and '0.1' notations in figures below). All of the sequences were run 10 times to account for non-deterministic behavior.

Figures 28.4, 28.5, and 28.6 show the evaluated absolute translation error e_{ate} and relative pose error e_{rpe} for the DSO, LDSO, and ORB-SLAM2 methods, respectively.

As we can see, all methods perform quite well on this dataset in terms of an absolute translation error. Still, ORB-SLAM2 comparing to direct methods has some advantage here—it withstands the most difficult sequences successfully—MH_{05},

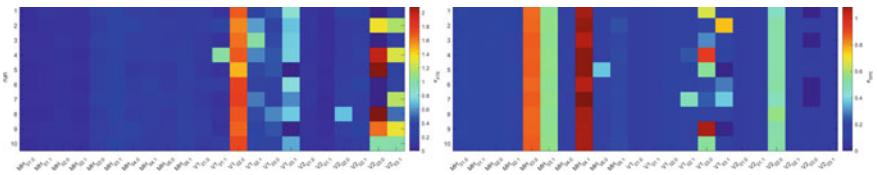

Fig. 28.4 DSO evaluation results. Absolute translation error e_{ate} (m) (left) and relative pose error e_{rpe} (m/s) (right) for each EuRoC dataset sequence and camera stream

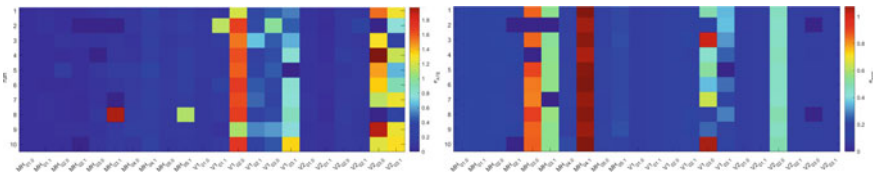

Fig. 28.5 LDSO evaluation results. Absolute translation error e_{ate} (m) (left) and relative pose error e_{rpe} (m/s) (right) for each EuRoC dataset sequence and camera stream

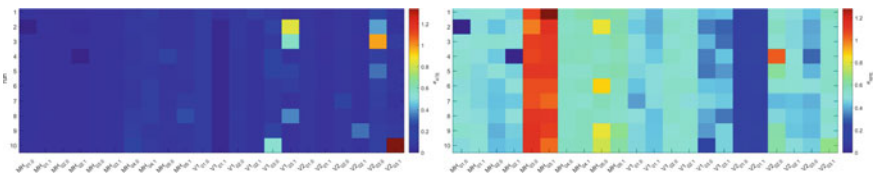

Fig. 28.6 ORB-SLAM2 evaluation results. Absolute translation error e_{ate} (m) (left) and relative pose error e_{rpe} (m/s) (right) for each EuRoC dataset sequence and camera stream

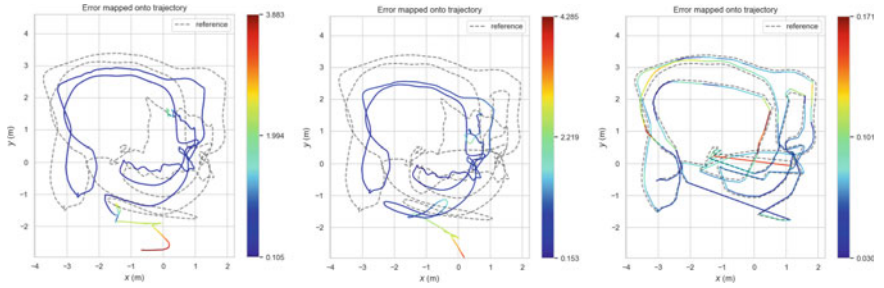

Fig. 28.7 Absolute translation error e_{ate} mapped onto estimated trajectory for each method: DSO (left), LDSO (middle) and ORB-SLAM2 (right) respectively

$V1_{03,}$ and $V2_{03}$—the ones with the lack of light and much more rotational movements than translational.

In terms of relative pose error, purely direct methods generally perform much better than the indirect one, but it is still difficult for them to overcome the harsh environment. Both of the direct methods fail on sequence $V2_{03}$, but on most of the other sequences, comparing to DSO, LDSO significantly improves the camera tracking accuracy, again due to the LDSO loop closure capability.

Qualitative comparison. To show some qualitative results, examples of the reconstructed maps for the most difficult $V2_{03}$ sequence for each of the methods along with the ground truth are shown in Fig. 28.7.

The figure shows that the overall trajectory pattern is distinguishable, but, as we can see, DSO and LDSO path was distorted at the end of the sequence. This has led to issues in alignment and scale, which are also clearly visible.

The presented situation is a common case for all of the reviewed methods due to the specific movement patterns and the lack of light in some scenes. Yet, ORB-SLAM2 performed much better than the others did.

28.6 Further Work

The current study presents a fixed set of the most popular and recent open-source monocular SLAM methods. However, there is still a number of methods such as LSD-SLAM [6] and SVO [20], which are older, but well-proven, and DynaSLAM [21], which adds new capabilities to the ORB-SLAM2 [8] system. This benchmark comparison will be expanded by the mentioned methods in the future, which will be useful for understanding the full market picture of monocular SLAM methods.

28.7 Conclusion

In this work, we have conducted a survey of the state estimation performance of publicly available real-time monocular SLAM algorithms. In evaluating these algorithms, our goal was to benchmark the accuracy of estimated trajectories. The results presented in Sect. 28.3 suggest that pure direct methods like DSO and LDSO have better accuracy only while having full photometric data *available,* and however, in terms of the overall accuracy, ORB-SLAM2 has an edge.

The accuracy and robustness can be improved with additional computation resources, but it all comes to the capabilities of the robotic platform—the more computational resources are required, the more power units and weight the platform should carry. In our research, we selected the specific hardware platform with a balanced amount of computational resources and power consumption. We hope that the results and conclusions presented in this paper may help members of the research community in finding appropriate compromises for their robot systems.

Acknowledgements The reported study was funded by the RFBR according to the research project No. 19-58-70002 and research grant of Kazan Federal University.

References

1. Dissanayake, M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. Robot. Autom. **17**(3), 229–241 (2001)
2. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2017)
3. Delmerico, J., Scaramuzza, D.: A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In: IEEE 2018 International Conference on Robotics and Automation, pp. 2502–2509 (2018)
4. Buyval, A., Afanasyev, I., Magid, E.: Comparative analysis of ROS-based monocular SLAM methods for indoor navigation. In: 2016 Ninth International Conference on Machine Vision, ICMV, pp. 103411K (2017)
5. Bokovoy, A., Yakovlev, K.: Enhancing semi-dense monocular vSLAM used for multi-rotor UAV navigation in indoor environment by fusing IMU data. In: The 2018 International Conference on Artificial Life and Robotics, pp. 391–394 (2018)
6. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: European Conference on Computer Vision, pp. 834–849 (2014)
7. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
8. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017)
9. Gao, X., Wang, R., Demmel, N., Cremers, D.: LDSO: Direct sparse odometry with loop closure. In: International Conference on Intelligent Robots and Systems, pp. 2198–2204 (2018)
10. Zakiev, A., Lavrenov, R., Magid, E., Svinin, M., Matsuno, F.: Partially unknown environment exploration algorithm for a mobile robot. J. Adv. Res. Dyn. Control Syst. **11**(8), 1743–1753 (2019)

11. Alishev, N., Lavrenov, R., Hsia, K. H., Su, K. L., Magid, E.: Network failure detection and autonomous return algorithms for a crawler mobile robot navigation. In: 11th International Conference on Developments in eSystems Engineering, pp. 169–174 (2018)
12. Krombach, N., Droeschel, D., Behnke, S.: Combining feature-based and direct methods for semi-dense real-time stereo visual odometry. In: International Conference on Intelligent Autonomous Systems, pp. 855–868 (2016)
13. Ramil, S., Lavrenov, R., Tsoy, T., Svinin, M., Magid, E.: Real-time video server implementation for a mobile robot. In: 2018 11th International Conference on Developments in eSystems Engineering, pp. 180–185 (2018)
14. Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J., Cremers, D.: The TUM VI benchmark for evaluating visual-inertial odometry. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1680–1687 (2018)
15. Engel, J., Usenko, V., Cremers, D.: A photometrically calibrated benchmark for monocular visual odometry. arXiv preprint arXiv:1607.02555 (2016)
16. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Siegwart, R.: The EuRoC micro aerial vehicle datasets. Int. J. Robot. Res. **35**(10), 1157–1163 (2016)
17. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361 (2012)
18. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 573–580(2012)
19. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Trans. Pattern Anal. Mach. Intell. **4**, 376–380 (1991)
20. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: IEEE International Conference on Robotics and Automation, pp. 15–22 (2014)
21. Bescos, B., Fácil, J.M., Civera, J., Neira, J.: DynaSLAM: tracking, mapping, and inpainting in dynamic scenes. IEEE Robot. Autom. Lett. **3**(4), 4076–4083 (2018)