Systematic Comparison of ORB-SLAM2 and LDSO based on Varying Simulated Environmental Factors

Adam Kalisz[a], Tong Ling[b], Florian Particke[c], Christian Hofmann[d] and Jörn Thielecke[e]
Department of Electrical, Electronic and Communication Engineering, Information Technology (LIKE),
Friedrich-Alexander-Universität Erlangen-Nürnberg, Am Wolfsmantel 33, Erlangen, Germany

Keywords: Visual, Simultaneous, Localization, And, Mapping, SLAM, ORB, LDSO, DSO, Comparison.

Abstract: Although the number of outstanding but highly complex Visual SLAM systems which are published as open source has increased in recent years, they often lack a systematic evaluation of their weaknesses and failure cases. This work systematically discusses the key differences of two state-of-the-art Visual SLAM algorithms, the indirect ORB-SLAM2 and the direct LDSO, by extensive experiments in varying environments. The evaluation is principally focused to the trajectory accuracy and robustness of the algorithms in specific situations. However, details about individual components used for the estimation of trajectories in both systems are presented. In order to investigate crucial aspects, a custom dataset was created in a 3D modeling software, Blender, to acquire the data for all experiments. The experimental results demonstrate the strengths and weaknesses of the systems. In particular, this research contributes insight into: 1. The influence of moving objects in a usually static scene. 2. How both systems react on periodicly changing scene lighting, both local and global. 3. The role of initialization on the resistance to dynamic changes in the scene.

1 INTRODUCTION

A robust localization and mapping in unknown environments is a highly desirable ability of autonomous robots. Visual Simultaneous Localization And Mapping (SLAM), as a realization using only cameras, gains increasing attention due to its simple configuration and low cost. Visual SLAM (VSLAM) can be classified into monocular, stereo, and RGB-D SLAM based on the type of camera used. At each time step, a monocular camera provides only a single frame, a stereo camera two frames from different lenses, and a RGB-D camera usually can capture a RGB image and a depth image. It is worth mentioning that a monocular SLAM, though with the simplest hardware configuration, should be performed more carefully due to the scale ambiguity (i.e., it can only retrieve the camera pose and the environment to an unknown scale).

A well designed sensor fusion approach is able to solve some of the fundamental problems of monocular VSLAM. For example, it can cooperate with Global Positioning System (GPS) in an autonomous car or robot to provide a more accurate localization or with other sensors in a GPS-denied area (Schleicher et al., 2009)(Shi et al., 2013).

However, the requirement on the accuracy of estimations in VSLAM's varies depending upon applications. Even though there has been a remarkable development over the last 30 years and many existing SLAM systems have already achieved fulfilling accuracy, studies such as (Cadena et al., 2016), (Huang and Dissanayake, 2016), and (Taketomi et al., 2017) have pointed out that there are still challenges and possible improvement directions.

Comparing the current development of VSLAM can provide a guide for specific future improvements which can be useful for any VSLAM system. In this work, we systematically analyze two state-of-the-art SLAM systems, Oriented FAST and rotated BRIEF (ORB)-SLAM2 (Mur-Artal and Tardós, 2017) and Direct Sparse Odometry with Loop Closure (LDSO) (Gao et al., 2018). ORB-SLAM2 is a newer version of ORB-SLAM (Mur-Artal et al., 2015) with improved feature extraction and a global optimization as well as new interfaces for stereo and RGB-D input. LDSO is extended based on DSO (Engel et al., 2018) by modifying feature extraction and adding a

[a] https://orcid.org/0000-0001-5428-5433
[b] https://orcid.org/0000-0002-9110-9890
[c] https://orcid.org/0000-0003-2761-2616
[d] https://orcid.org/0000-0003-1720-6948
[e] https://orcid.org/0000-0001-6671-6341

Kalisz, A., Ling, T., Particke, F., Hofmann, C. and Thielecke, J.
Systematic Comparison of ORB-SLAM2 and LDSO based on Varying Simulated Environmental Factors.
DOI: 10.5220/0008879401730180
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 4: VISAPP*, pages 173-180
ISBN: 978-989-758-402-2; ISSN: 2184-4321

loop closure functionality. There are three reasons for choosing these two systems. Firstly, their core ideas are different. ORB-SLAM2 is a feature-based (indirect) method, while LDSO is a direct method. Through the comparison, insight into these mechanisms is expected to be obtained, which benefits the future development in the two directions or a hybrid one. Secondly, they are both sparse and real-time capable methods, ensuring a fair comparison. Thirdly, both algorithms have achieved high accuracy on several benchmark datasets. Despite many kinds of improvement proposed for these two systems so far, they still represent the current state of monocular Visual SLAM and deserve a more systematic analysis. To the best of our knowledge, there is still no systematic analysis of ORB-SLAM2 and LDSO for a monocular camera.

Throughout this article we are investigating their robustness and accuracy in challenging environments which we can fully control. Due to the fact that LDSO only accepts monocular frames, our work is limited to the case with a monocular camera.

2 RELATED WORK

In the past work, the experimental comparison was usually based on the benchmark datasets such as TUM-monoVO (Engel et al., 2016), TUM RGB-D (Sturm et al., 2012), EuRoC MAV (Burri et al., 2016), and KITTI (Geiger et al., 2013) as well as ICL-NUIM (Handa et al., 2014).

Earlier research compared version one of ORB-SLAM with other existing algorithms (Huletski et al., 2015)(Li et al., 2016). In addition to the experimental comparison, a theoretical comparison of numerous monocular VSLAMs was made in (Younes et al., 2017).

This was extended by a quantitative comparison of the former versions of both methods ORB-SLAM with DSO in (Engel et al., 2018). Engel et al. ran DSO and ORB-SLAM on the TUM-monoVO, the EuRoC MAV and the synthetic ICL-NUIM dataset. In the experiments with TUM-monoVO and ICL-NUIM, DSO outperformed ORB-SLAM regarding the trajectory accuracy and robustness. In the other experiments, ORB-SLAM achieved a better accuracy but showed worse robustness. Besides, they also studied the tracking accuracy on the TUM-monoVO dataset with artificial geometric and photometric noise. As expected, ORB-SLAM is more robust to the geometric noise and DSO is more robust to the photometric noise.

(Yang et al., 2018) investigated special aspects of ORB-SLAM and DSO including photometric calibration, motion bias and rolling shutter effect. DSO has proven to be more sensitive to the photometric calibration and the rolling shutter effect. Besides, both methods showed a large performance bias when running forward and backward on a dataset.

The first comparison of LDSO with ORB-SLAM2 was made by (Gao et al., 2018). In their work, LDSO and ORB-SLAM2 have achieved comparable trajectory accuracy on the KITTI dataset. While running on the EuRoC MAV, ORB-SLAM2 showed better accuracy and LDSO better robustness.

Finally, we would like to note, that this work is the extension of our own open source evaluation pipeline. A brief investigation of DSO using synthetic data was performed by the authors in (Particke et al., 2018) and the datasets in this work have been generated using the B-SLAM-SIM framework from (Kalisz et al., 2019).

3 SYSTEM OVERVIEW OF ORB-SLAM2 AND LDSO

This section theoretically compares ORB-SLAM2 and LDSO from the overall system overview to their individual strategies in each module.

Generally, they consist of three parts: Firstly, the tracking to predict the current camera pose and decide if a new keyframe is necessary. Secondly, the mapping to optimize several but not all keyframes and map points, and eliminate some of them when necessary. Thirdly, the loop closing to correct drift of the camera pose by loop detection and pose graph optimization. Accurate trajectories and maps are only possible after a careful initialization, which is introduced specifically when discussed. Two special techniques in ORB-SLAM2 are the relocalization, namely recognizing a place where the camera was, and the global bundle adjustment to optimize the whole trajectory and map. The relocalization occurs after the tracking is lost, while the global BA is only performed if the initialization or loop closure is finished. The key differences in their architecture and design ideas are summarized briefly in Table 1.

In contrast to ORB-SLAM2 computing the geometric error, LDSO takes the geometric error into account only when closing loops, and calculates the photometric error in its tracking and mapping parts. Instead of directly computing the intensity difference, LDSO attempts to compute the irradiance (the energy falling onto a small patch of the imaging sensor) difference. In order to convert between a brightness intensity and an irradiance value, LDSO expects a pho-

Table 1: Overview about the main differences in both SLAM systems.

Component	ORB-SLAM2	LDSO
Point Management	Image pyramid Uniform Grid FAST detector ORB descriptor Triangulation Survival of the fittest	Uniform Grid Large gradients Shi-Tomasi score Inverse depth Status label
KeyFrame Management	Covisibility graph Survival of the fittest Observation test	Sliding window Optical flow Camera translation Exposure time
Initialization	Homography Fundamental matrix	First frame is fixed Photometric error Converged optimization
Tracking	Indirect Constant velocity Relocalization (BoW)	Direct 83 motion hypotheses Photometric error
Mapping	Local bundle adjustment	Sliding window bundle adjustment

tometric calibration of images, and builds a photometric model as explained in (Engel et al., 2016). The parameters of this model will also join in optimization processes. The purpose of this photometric model is to enhance the robustness of the system, because the irradiance constancy, as compared to the brightness constancy, more easily holds in the real world considering the automatic exposure changes and non-linear response functions in modern off-the-shelf cameras. If the images are not photometrically calibrated previously, predefined parameters will be used. Generally, ORB-SLAM2 can be considered as a fully feature-based method, whereas LDSO is basically a direct method.

As an indirect method, ORB-SLAM2 can handle large baseline motions thanks to the feature matching. Due to the implementations, ORB-SLAM2 can maintain a global consistency of the trajectory by relocalization and loop closure, and speeds up the bundle adjustment and pose graph optimization by means of the innovative covisibility graph and essential graph. Furthermore, ORB-SLAM2 builds two geometric models, the homography and fundamental matrix, to ensure an accurate and robust initialization. Nevertheless, these two models still require a sufficient parallax to reduce the uncertainty and some degenerate cases are still not avoidable, e.g. degenerate homography happens when one of the camera center lies in the world plane. Also, this feature-based methodology demands an environment with prominent features. Although LDSO also extracts features from images, it can better cope with a texture-less environment due to an adaptive threshold for high-gradient pixel detection. Because of the trait of the direct approach, LDSO can achieve a sub-pixel accuracy. However, the deficiency in the relocalization function limits the global localization capability of LDSO in spite of integrating the loop closure function. Despite a coarse-to-fine tracking, LDSO is still less efficient at handling large camera motions than ORB-SLAM2.

4 EXPERIMENTAL SETUP

This chapter describes the preparation for a systematic evaluation in order to make this process transparent to the reader.

For a systematic evaluation, datasets used should cover as many cases as possible. Two difficulties arise here, namely, how to keep control variables constant and how to obtain the ground truth trajectory. Many existing real-world datasets such as KITTI were recorded in a dynamic environment and have translational and rotational motion, sometimes even motion blur, which makes a systematic evaluation intractable. Other synthetic datasets such as ICL-NUIM were recorded in static scenes, but they only cover limited cases. Based on these limitations, we decided to create new datasets to benchmark ORB-SLAM2 and LDSO using the open source software Blender[1].

[1] https://www.blender.org/

Visual SLAM usually creates a world coordinate system with the first tracked frame as its origin, where camera poses are expressed relative to it. However, the ground truth trajectory is usually extracted from a different coordinate system. In order to quantitatively compare the estimated path with the ground truth, we first align both trajectories with support for the collinear case as discussed in (Barfoot, 2017).

After trajectory alignment, appropriate error metrics should be used to evaluate the difference between the estimated trajectory and ground truth. Two frequently applied metrics are the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE). Advantages and disadvantages of ATE and RPE are already discussed in (Kümmerle et al., 2009), (Sturm et al., 2012), and (Zhang and Scaramuzza, 2018). The following evaluations use ATE as the error metric.

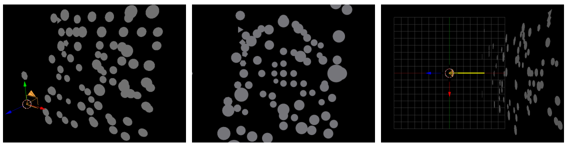
5 RESULTS AND EVALUATION

This chapter describes experiments and evaluation with respect to different factors. The results of running ORB-SLAM2 and LDSO on an ideal synthetic dataset is the reference for the investigation of different aspects including dynamic environments with moving objects as well as the change of image brightness. It is noteworthy that both SLAM systems run on each dataset 100 times to mitigate the impact of randomness, which arises from non-deterministic algorithms such as RANSAC (Fischler and Bolles, 1981) used to reject outliers in the pose prediction stage. Results are primarily represented in the form of cumulative plots, where the x-axes depict the percentage of tracked frames, root mean square translational error, and root mean square rotational error respectively, while y-axes denote the number of runs, in which a value, less than or equal to a certain x-value, was obtained. Ultimately, all observations and indications from the research are summarized.

5.1 Static Scene

We evaluate both algorithms on an environment called *Polygon World* which is built from blob, corner and edge features. It contains 81 polygons with no texture but the same solid color. The environment is designed to be static, which means there are no moving objects and no illumination change. It is deliberately kept as simple as possible, which however makes it only suitable to generate image sequences with camera movements in a small area.

From Polygon World, a reference sequence of 250 frames was generated as illustrated in Figure 1. The camera moves straightforward with a constant velocity towards polygons and captures 20 frames per second. In total, the camera moves a distance of 5 m.



(a) Polygon World. (b) Camera view. (c) Top view.

Figure 1: An illustration of the reference sequence from the synthetic world named *Polygon World*. It contains 81 different polygons ranging from 3 to 83 vertices. The camera is highlighted in the various perspective views and the top view shows a yellow line which represents the ground truth trajectory in the evaluation.

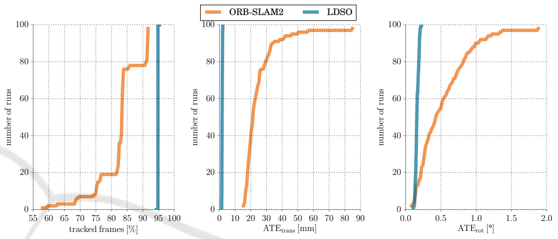


Figure 2: Results on the reference dataset from Polygon World. In the total 100 runs, ORB-SLAM2 failed once, in which it did not initialize. Due to no detected loops, the sliding-window-related trajectory of LDSO is evaluated.

Figure 2 demonstrates the results of 100 runs on the reference dataset from Polygon World. It can be observed that with this sequence, LDSO outperforms ORB-SLAM2 regarding the root mean square translational and rotational error. In addition, LDSO shows much less randomness as opposed to ORB-SLAM2.

The reason for these is the initialization. As discussed before, an accurate initialization is vital for the subsequent tracking. To initialize, ORB-SLAM2 applies the RANSAC algorithm to select detected feature correspondences for the computation of a homography or a fundamental matrix, while LDSO performs a joint optimization, which is more deterministic.

In this sequence, although the camera views different polygons, it does not mean the detected corners have distinct feature descriptors. Due to the inherent descriptor computation algorithm and the resolution of the images, numerous features are seen similar to each other. As a result, ORB-SLAM2 cannot discard incorrect correspondences between frames (see Figure 3). If the erroneous correspondences are picked for the initialization, there can be two consequences: On the one hand, ORB-SLAM2 is very likely to obtain an invalid homography or fundamental matrix and needs to iterate the picking process, causing a slow initialization and further fewer tracked frames.

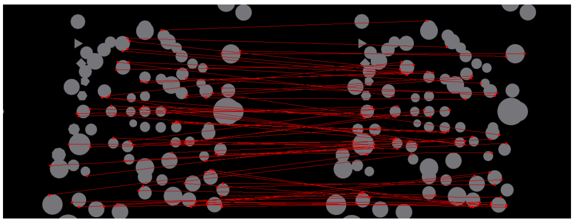


Figure 3: Erroneous correspondences detected by ORB-SLAM2. The figure is the concatenation of frame 1 (left) and 2 (right). A red dot is a detected feature and a red line linking two features represents an incorrect matching, which is empirically detected by checking if a feature moves more than 10 pixels between two frames. The number of erroneous matches takes up 14 percent of the total matches.

On the other hand, ORB-SLAM2 happens to initialize, but the recovered relative pose is probably inaccurate, leading to a larger error in the estimated trajectory.

Additionally, ORB-SLAM2 suffers from pixel discretization artifacts as mentioned in (Yang et al., 2018). That is, features' locations are represented by integers. This kind of representation is inexact and could severely affect the estimate's accuracy.

Furthermore, the camera moves very slowly in this sequence, causing only small parallax between adjacent frames. However, ORB-SLAM2 expects a noticeable change in perspective when initializing. Hence, ORB-SLAM2 waits for more frames until a large parallax and thus tracks fewer frames.

5.2 Dynamic Environment

An ideal environment for ORB-SLAM2 and LDSO should be static, which means no moving objects and no brightness change. However, the real world is always dynamic. For example, a landmark could move during the tracking, and the light intensity in the environment could change gradually resulting in a variation of the final image brightness.

5.2.1 Moving Object

There are many factors related to dynamic objects which can be delved into. We primarily focus on three of them, the number of moving objects and their moving directions as well as the time they start to move. The number of moving objects in our experiments is 1 or 40. The objects move forward with the same velocity as the camera, or they move in the direction which is perpendicular to the camera's moving direction. We investigate either all objects starting to move from the first frame or after 125 frames separately to evalute the influence of the initialization process in each SLAM system.

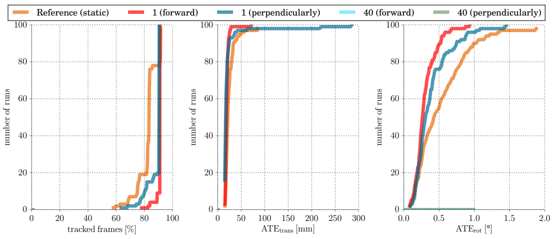


Figure 4: Results of ORB-SLAM2 where dynamic objects all start to move from the first frame.

Figure 4 depicts that ORB-SLAM2 cannot initialize when there are 40 objects moving from the beginning as the respective graph remains tiny. One moving object has only slight influence on the trajectory accuracy.

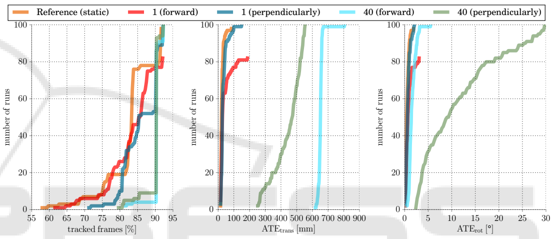


Figure 5: Results of ORB-SLAM2 where the dynamic objects all start to move after 125 frames.

If objects start to move after the system is initialized, ORB-SLAM2 can still track as depicted in Figure 5. However, the moving objects affect the trajectory accuracy. Compared with a single moving object, more moving objects lead to a less accurate trajectory. Furthermore, if the objects move together perpendicularly to the camera, the rotational error will increase significantly. Objects that are moving in the same direction as the camera influence the translational error of the camera.

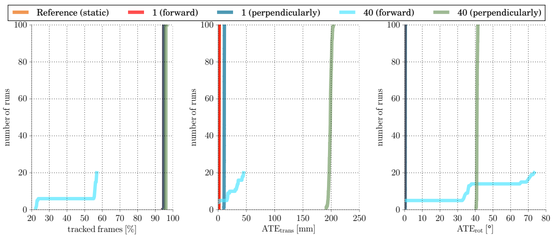


Figure 6: Results of LDSO where the dynamic objects all start to move from the first frame.

Similar conclusions can also be drawn from the results of LDSO depicted in Figures 6 and 7. One difference from ORB-SLAM2 is that LDSO was also able

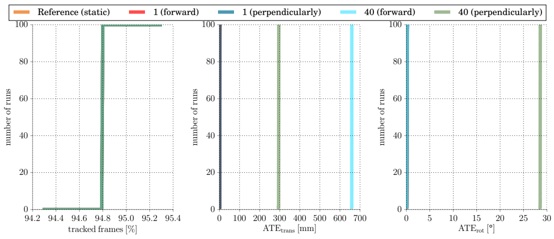
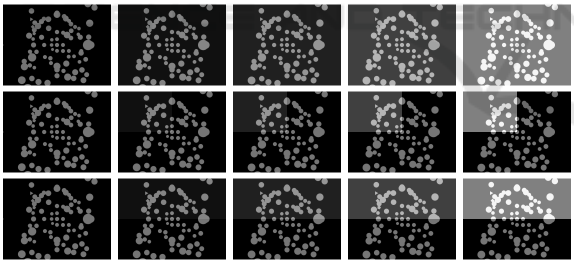


Figure 7: Results of LDSO where the dynamic objects all start to move after 125 frames.

to initialize when 40 objects started to move at the beginning, but the estimation contains larger translational and rotational errors.

5.2.2 Illumination Change

Regarding the illumination change, two situations should be considered, namely local and global variation. These variations in the real world lead to pixel intensity changes in images. Therefore, new sequences were generated directly by modifying the pixel intensities in the reference dataset from Polygon World. Like in the last section, the illumination change can also happen from the beginning or after a successful initialization. Based on the previous results, we assume ORB-SLAM2 and LDSO must have initialized in the first 125 frames of the reference dataset.



(a) $t = 1$. (b) $t = 2$. (c) $t = 3$. (d) $t = 4$. (e) $t = 5$.

Figure 8: An illustration of global (top row), 1/4 local (middle row) and 1/2 local (bottom row) illumination variation.

To model a global variation, the pixel intensity values in an image were added with 0, 16, 32, 64, 128 periodically across the whole image, whereas for a local variation only a quater or half of the image was altered as demonstrated in each row of Figure 8. The results of ORB-SLAM2 are demonstrated in Figures 9 and 10. For comparison, LDSO is summarized in Figures 11 and 12.

As can be seen in Figure 9, ORB-SLAM2's initialization and trajectory accuracy are not highly affected by the local illumination change starting from the beginning, but a global variation could lead to an estimated trajectory with a large error. If the intensity starts to vary after 125 frames, ORB-SLAM2 is more likely to estimate an inaccurate trajectory.

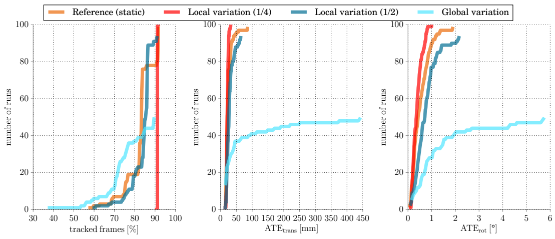


Figure 9: Results of ORB-SLAM2 when the illumination starts to change from the beginning.

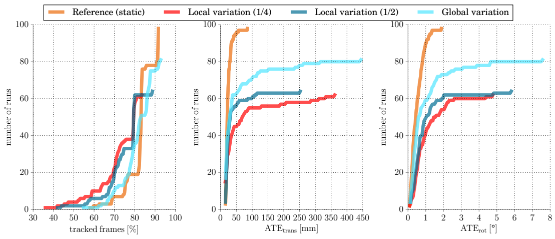


Figure 10: Results of ORB-SLAM2 when the illumination starts to change after 125 frames.

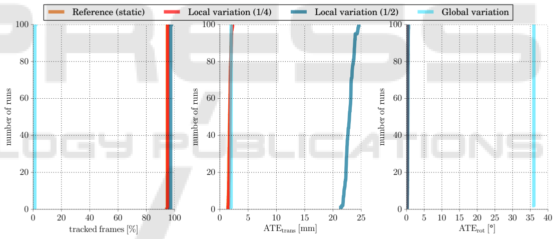


Figure 11: Results of LDSO when the illumination starts to change from the beginning.

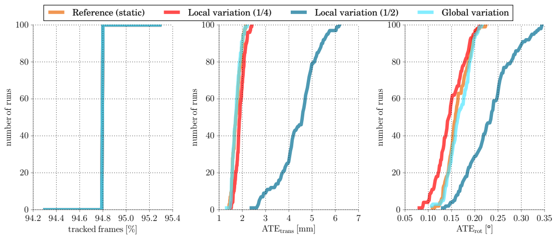
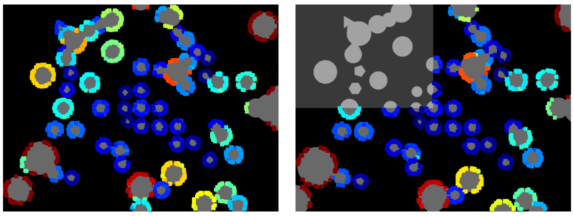


Figure 12: Results of LDSO when the illumination starts to change after 125 frames.

Compared to ORB-SLAM2, LDSO is more sensitive to illumination change. It failed very soon after the initialization in all runs on the sequence with global variation starting from the beginning. Additionally, a local variation in the half of images increases the position error considerably.

One unanticipated finding was that if the local



(a) Frame 107. (b) Frame 133.

Figure 13: Examples of depth maps estimated by LDSO. These are color-coded (near: dark red, far: dark blue) depth maps on two keyframes, which are used for the pose prediction of next frames. It can be seen that those features whose intensity values have changed greatly were discarded by LDSO.

variation happens only in one quarter of an image, LDSO is barely affected, which indicates LDSO still has resistance to the brightness change to some extent. A reason for this resistance is that LDSO can identify the pixels with strong variation as outliers and neglect them while tracking (see Figure 13).

6 CONCLUSIONS

Generally, a distinct feature is crucial in both systems. Although ORB, which is utilized in ORB-SLAM2, may outperform other existing features (Tareen and Saleem, 2018), it was not as robust as expected in the experiments. There were always inevitable erroneous correspondences of features. Besides, the features' locations are not exact due to discretization artifact. These are important reasons for ORB-SLAM2's randomness and less accurate estimate. An improvement in the future could be developing a more robust feature with sub-pixel accuracy. Another suggestion is that a smoothing method should be performed as the first step when a new frame arrives, but with the caution that the details in the image should be retained.

In the experiments about dynamic environments, we have found out:

1. It is quite challenging for ORB-SLAM2 and LDSO to initialize in a dynamic environment. However, if the dynamic change happens after the initialization, its influence will become much smaller.

2. If the change is not apparent enough (few moving objects or a small area with illumination change), ORB-SLAM2 and LDSO are able to identify outliers and neglect them during the estimation.

Regarding the initialization, the techniques used are not perfect. There are two questions which need to be answered. The first question is which frame should be used for the initialization. In ORB-SLAM2, two frames are chosen according to the number of features and correspondences. In LDSO, after the first frame is fixed as the reference, the next frame will be directly used for the image alignment. The second question is how to initialize the system, namely, how to estimate the relative pose and the landmarks' positions. ORB-SLAM2 makes use of the geometry information to compute the relative pose and then triangulation. In contrast, LDSO optimizes them together by providing initial guesses, expecting a convergence of the values in the photometric cost function. Despite the fact that LDSO has initialized in more runs in the experiments, the frames for the initialization should still be selected based on certain criteria to mitigate the dependency of the first frame, and the model should be general for all kinds of situations. An improvement may be using more views as introduced in (Hartley and Zisserman, 2003).

In future work we aim to investigate the differences between both SLAM systems by comparing their motion models and loop closure capabilities.

ACKNOWLEDGEMENTS

This work was supported by Siemens Mobility GmbH, Erlangen, Germany.

REFERENCES

Barfoot, T. D. (2017). *State Estimation for Robotics*. Cambridge University Press.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., and Siegwart, R. (2016). The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332.

Engel, J., Koltun, V., and Cremers, D. (2018). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625.

Engel, J., Usenko, V., and Cremers, D. (2016). A photometrically calibrated benchmark for monocular visual odometry. *arXiv preprint arXiv:1607.02555*.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Gao, X., Wang, R., Demmel, N., and Cremers, D. (2018). Ldso: Direct sparse odometry with loop closure. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2198–2204. IEEE.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

Handa, A., Whelan, T., McDonald, J., and Davison, A. J. (2014). A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE international conference on Robotics and automation (ICRA)*, pages 1524–1531. IEEE.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Huang, S. and Dissanayake, G. (2016). A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems*, 13(5):1729881416669482.

Huletski, A., Kartashov, D., and Krinkin, K. (2015). Evaluation of the modern visual slam methods. In *2015 Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)*, pages 19–25. IEEE.

Kalisz, A., Particke, F., Penk, D., Hiller, M., and Thielecke, J. (2019). B-slam-sim: A novel approach to evaluate the fusion of visual slam and gps by example of direct sparse odometry and blender. In *VISIGRAPP*.

Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., and Kleiner, A. (2009). On measuring the accuracy of slam algorithms. *Autonomous Robots*, 27(4):387.

Li, A. Q., Coskun, A., Doherty, S. M., Ghasemlou, S., Jagtap, A. S., Modasshir, M., Rahman, S., Singh, A., Xanthidis, M., O'Kane, J. M., et al. (2016). Experimental comparison of open source vision-based state estimation algorithms. In *International Symposium on Experimental Robotics*, pages 775–786. Springer.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163.

Mur-Artal, R. and Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Particke, F., Kalisz, A., Hofmann, C., Hiller, M., Bey, H., and Thielecke, J. (2018). Systematic analysis of direct sparse odometry. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6. IEEE.

Schleicher, D., Bergasa, L. M., Ocana, M., Barea, R., and Lopez, M. E. (2009). Real-time hierarchical outdoor slam based on stereovision and gps fusion. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):440–452.

Shi, Y., Ji, S., Shi, Z., Duan, Y., and Shibasaki, R. (2013). Gps-supported visual slam with a rigorous sensor model for a panoramic camera in outdoor environments. *Sensors*, 13(1):119–136.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE.

Taketomi, T., Uchiyama, H., and Ikeda, S. (2017). Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16.

Tareen, S. A. K. and Saleem, Z. (2018). A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–10. IEEE.

Yang, N., Wang, R., Gao, X., and Cremers, D. (2018). Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters*, 3(4):2878–2885.

Younes, G., Asmar, D., Shammas, E., and Zelek, J. (2017). Keyframe-based monocular slam: design, survey, and future directions. *Robotics and Autonomous Systems*, 98:67–88.

Zhang, Z. and Scaramuzza, D. (2018). A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7244–7251. IEEE.