Periodic Bundle Adjustment to Mitigate Drift Error in ORB-SLAM3 for Autonomous Vehicle

Hilda Widyastuti*[1,2], Bambang R. Trilaksono[1], Egi Muhammad Idris Hidayat[1], and Widyawardana Adiprawita[1]

[1] School of Electrical Engineering and Informatics, Institut Teknologi Bandung;
[2] Politeknik Negeri Batam
33218013@std.stei.itb.ac.id

Abstract. Visual SLAM systems commonly encounter challenges with drift errors arising from incremental position estimations between frames. As autonomous vehicles traverse longer distances, these systems are prone to an accumulation of drift errors. ORB-SLAM3 mitigates this issue through optimizations applied to motion-only bundle adjustment, local bundle adjustment, and full bundle adjustment. Once loop closure is detected, the full bundle adjustment optimizes all positional and landmark data; however, in situations where loop closure does not occur or occurs over extended distances, resulting in a significant accumulation of drift errors. An additional optimization trigger, independent of loop closure, must address this. This study proposes the periodic bundle adjustment method, which optimizes the system based on the number of keyframes. Implementing this method within ORB-SLAM3 reduces the RMSE APE value compared to ORB-SLAM3 without periodic bundle adjustment on the KITTI Odometry dataset.

Keywords: ORB-SLAM3, Autonomous Vehicle, Optimization.

1 Introduction

Autonomous vehicle has the potential to offer a safer mode of transportation compared to human-driven car [1]. In autonomous vehicle technology, precise knowledge of the car's location and awareness of its surroundings are crucial [2]. Autonomous vehicle use the Global Positioning System (GPS) or Simultaneous Localization and Mapping (SLAM) to determine their location. SLAM is a technique that estimates the vehicle's position using onboard sensors while simultaneously constructing a map of the environment based on the data gathered by those sensors [3].

L. Lumombo et al. (eds.), *Proceedings of the 7th International Conference on Applied Engineering (ICAE 2024)*,
Advances in Engineering Research 251,
https://doi.org/10.2991/978-94-6463-620-8_21

SLAM is preferred over GPS for several reasons: (1) SLAM provides superior accuracy in enclosed environments, such as tunnels, tall buildings, or underground areas, where GPS signals may be obstructed or inaccurate. SLAM can operate effectively in such conditions without relying on external signals [4] . (2) SLAM allows vehicle to map their surroundings while simultaneously determining their position on the map [5] . (3) In urban settings, GPS signals may suffer from multipath effects or reduced accuracy due to reflections from tall buildings [6]. (4) SLAM can function in environments where GPS signals are unavailable, such as indoor spaces or beneath bridges, by utilizing local sensor data [2]. (5) SLAM enables vehicle to detect changes in their environment, allowing them to update maps and navigation routes in real-time [7] . (6) The integration of SLAM with GPS provides an additional layer of security and redundancy, ensuring uninterrupted navigation in the absence of GPS signals [1] .

This study utilizes ORB-SLAM3 to map the location of autonomous vehicle [8]. However, both visual SLAM and visual odometry are prone to drift errors from frame-by-frame position estimations [2], [8], [9], [10], [11]. As autonomous vehicle traverses longer distances, these drift errors accumulate. ORB-SLAM3 addresses this issue by optimizing through motion-only bundle adjustment (BA), local bundle adjustment, and full bundle adjustment [8]. Full bundle adjustment optimizes the vehicle's position and landmark data, typically after loop closure detection or map merging [8]. However, in some scenarios, significant drift errors can accumulate due to the absence of loop closure, delayed loop closure, or the failure to merge maps, thus necessitating an alternative optimization trigger [9].

An additional trigger is required to initiate the optimization process rather than relying solely on loop closure or map merging. For instance, RoomSLAM initiates optimization based on the presence of a room. Once the robot completes its traversal, optimization is performed on its position, landmark, and wall data within that room [10] . Similarly, the approach in [11] uses a sliding window optimization that covers the previous and current keyframes. Although room-based optimization is conceptually interesting for autonomous vehicle, its implementation is challenging in outdoor environments where indoor-like rooms are absent. Drawing inspiration from RoomSLAM, this study introduces a periodic bundle adjustment optimization process based on the number of keyframes.

2. Related Work

2.1 Visual SLAM System

The process of SLAM allows autonomous vehicle to create a map of their surroundings while using this map to establish their position simultaneously [12], [13]. Visual SLAM [14], [15] refers to a subset of SLAM techniques that utilize only images for mapping the environment and determining the vehicle's position. In SLAM, a autonomous vehicle moves through its environment, making relative observations of unknown reference points using onboard sensors.

The estimation of a moving camera's poses can be achieved through various methods, which can be categorized into direct, semi-direct, and indirect approaches. Direct approaches assume that the pixel intensities of the same point projected across multiple frames remain constant, making them sensitive to factors such as motion blur and changes in lighting. By minimizing the photo metric error between pixels across multiple frames, these techniques can estimate the camera poses [16]. Indirect approaches, also known as feature-based approaches [13], [17], [5], evaluate the geometric errors of sparse extracted feature points and optimize camera poses by minimizing the reprojection errors of the same points across multiple frames. Keyframe based SLAM is a type of feature-based method that utilizes a graph-based approach to address local metric mapping, loop closure detection, and global map optimization issues through parallel processes. Keyframes represent historical camera poses where significant features are matched and measured, and they are incorporated into the final map, as other frames are used for local tracking but are eventually discarded. This method enables real-time bundle adjustment on keyframes, facilitating efficient map adjustment [18].

The system developed in this study is ORB-SLAM3 [8], which performs sparse scene reconstruction and camera pose estimation by extracting ORB features from input images. The framework includes tracking thread, local mapping thread, loop closure and map merging thread, and full bundle adjustment thread, is widely regarded as state-of-the-art in pose estimation accuracy. ORB-SLAM3 incorporates key concepts such as keyframes, covisibility graphs, ATLAS, Oriented FAST and Rotated BRIEF(ORB), and Bundle Adjustment (BA).

2.2 Bundle Adjustment (BA)

The system is optimized using BA to minimize the cost function and reduce the reprojection error of all observations. BA is a critical element of visual SLAM, focusing on enhancing the camera poses and the positions of 3D points in the scene. This process aims to improve the accuracy of camera localization and map points by minimizing the variance between estimated feature points based on 3D points and camera poses and the detected feature points in the image. Implementing BA effectively reverses system drift and error accumulation, leading to improved overall accuracy and stability of the system. However, the high computational complexity of BA, especially when handling large datasets, requires substantial computational resources, posing challenges for real-time implementation [19].

ORB-SLAM3 uses motion-only BA to optimize camera poses in the tracking thread, local BA to optimize keyframe and point windows in the local mapping thread, and full BA to optimize all keyframes and points after loop closure [13]. This optimization is carried out with the help of the g2o framework's Levenberg-Marquardt algorithm [22].

According to [13], motion-only BA focuses on 1) optimizing the camera's orientation$R \in SO$and position, 2) minimizing the reprojection error between 3D points in world coordinates $X^i \in R^3$and the corresponding keypoint x_s^i, where $x_s^i \in R^3$,$i \in X$, X represents the set of all matches.

Local BA optimizes the set of covisible keyframes $K_L$and all the points observed in those keyframesP_L. While other keyframes, K_F, that observe the points P_L, contribute to the cost function, they remain excluded from the optimization process. The mathematical formula of local BA is in equations (1) and (2). On the other hand, full BA is a specific case of local BA, in which all keyframes and map points are optimized, except for the initial keyframe, which is kept fixed to eliminate gauge freedom [13].

$$\{X^i, R_l, t_l | i \in P_L, l \in K_L\} = \underset{X^i, R_l, t_l}{argmin} \sum_{k \in K_L \cup K_F} \sum_{j \in X_k} \rho\big(E(k,j)\big) \tag{1}$$

$$E(k,j) = \left\| x_s^j - \pi_s(R_k X^j + t_k) \right\|_{\Sigma}^2 \tag{2}$$

where :

X^i : 3D points in world coordinates

R : camera's orientation

t : camera's position

K_L: set of covisible keyframes

P_L: all the points observed in the covisible keyframes

ρ: Huber cost function

x_s^i: set of 3D keypoint, where$i \in X$; X is the set of all matches

E (k,j) : error between observed data and the model's prediction

π_s: stereo projection function

Σ: covariance matrix associated to the scale of the keypoint

2.3 Previous Studies on Bundle Adjustment

Several studies have explored bundle adjustment. As shown in table 1, the research in [2] based on ORB-SLAM2, performs optimization in areas containing traffic signs. The scope of the optimization includes all keyframes, with adjustments applied to both ORB data points and ORB points within the traffic sign bounding box. In [8], three types of optimizations are implemented: motion-only bundle adjustment for optimizing camera poses, local bundle adjustment triggered upon the creation of a new keyframe, which optimizes the current keyframe and its neighboring keyframes, and full bundle adjustment conducted after loop closure detection. In contrast, the study in [9] applies optimization to every frame. The research in [10], based on ORB-SLAM2, performs optimization in every room. The estimated items are the robot position, object position, and wall position. The study in [11] is not based on ORB-SLAM2 and ORB-SLAM3. It uses sliding window, with coverage consisting of previous keyframe and current keyframe. It consists of two types of optimization, namely point-based and object-based optimization, object-based optimization is performed on the previous keyframe. The results of object-based optimization are input for point-based optimization. Point-based optimization is performed on the current keyframe.

Table 1. The comparison of related research.

Ref.	Approach		
	Basic algorithm	Optimization items	Place or time of optimization

Ref.	Approach		
	Basic algorithm	Optimization items	Place or time of optimization
[2]	ORB-SLAM2	optimization is applied to ORB data points and ORB points located within the traffic sign bounding box	optimization is performed in areas containing traffic signs
[8]	ORB-SLAM2	camera pose optimization	motion-only bundle adjustment
		current keyframe and its neighboring keyframes	when a new keyframe is formed, local bundle adjustment is performed.
		all keyframes	full bundle adjustment performed after loop closure is detected
[9]	not based on ORB-SLAM2 and ORB-SLAM3	poses and 3D reconstruction	done in every frame
[10]	ORB-SLAM2	robot position, object position, and wall position	optimization is done in each room generated
[11]	not based on ORB-SLAM2 and ORB-SLAM3	using a sliding window, with coverage consisting of the previous keyframe and the current keyframe	• object-based optimization is performed on the previous keyframe. The result becomes input for point-based optimization • point-based optimization is performed on the current keyframe
the present work	ORB-SLAM3	all keyframes and points on the map	full bundle adjustment every 100 keyframes

3. Periodic Bundle Adjustment

The focus of the present work is on ORB-SLAM3. As shown in Figure 1 about ORB-SLAM3 diagram with periodic bundle adjustment, a frame undergoes processing within the tracking thread. If the frame meets the criteria for becoming a keyframe, it is passed to the local mapping thread for further processing. In this thread, keyframes are optimized through local bundle adjustment. The keyframes are added to a keyframe container list. When the number of keyframes in the container list reaches a specified threshold (100 keyframes), a comprehensive optimization is performed using the full bundle adjustment method, optimizing all vehicle position data and previously generated map point (landmark) data. After the full BA is completed, the keyframe container list is cleared for the next cycle of keyframe processing.

As shown in Figure 1, we introduce the new method, check keyframe total, which serves the following functions :

1. It adds keyframes that pass from the local keyframe culling process to the container list.
2. It checks whether the number of keyframes in the container list has reached 100. If 100 keyframes are accumulated, the full BA thread is executed. If not, the full BA thread is only executed when a loop is detected, or a successful map merge occurs.
3. The keyframe container list is cleared after the full BA thread is executed.

The overall ORB-SLAM3 algorithm, which incorporates periodic bundle adjustment (with bold sign), is detailed in Algorithm 1.

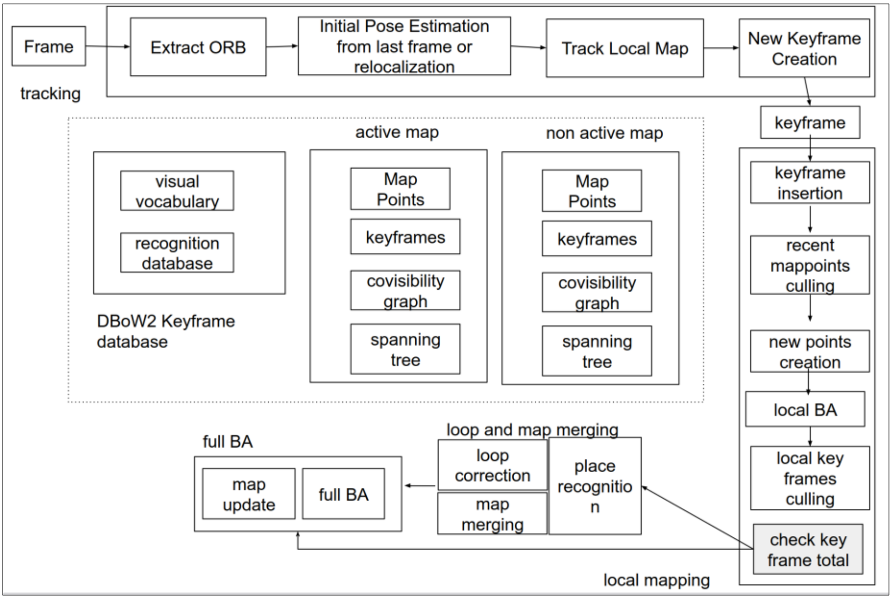


Fig. 1. ORB-SLAM3 Diagram with Periodic Bundle Adjustment.

Algorithm 1. ORB-SLAM3 algorithm with periodic bundle adjustment.

ORB-SLAM3 algorithm with periodic bundle adjustment input : stereo image dataset output : trajectory file showing autonomous vehicle positions steps : Read all images in one data sequence folder

```
for i=0 to num_image do
    //processing on an image in tracking thread
    ORB_points_extraction()
    position_determination()
    track_local_map ()
    optimization_with_motion_only_BA()
    //processing on local mapping thread
    KF = keyframe_generation()
    keyframe_insertion (KF)
    map_points_deletion_and_creation()
    neighborKF=keyframe_neighbor_determination()
    optimization_with_local_Bundle_Adjustment(KF, KF neighbor)
    delete_redundant_keyframes()

    //add periodic bundle adjustment
    //input keyframe to KeyFrame_container list
    KeyFrame_container.add (KF)
    number_of_key_frames = KeyFrame_container.size()
    if (number_of_key_frames = 100) then
        optimization_with_Full_Bundle_Adjustment (keyframe, 3D points in map)
        //empty the list of KeyFrame_container
        KeyFrame_container.empty()
    end if
    else
        detection_result = detect_loop_closure ()
        if (detection_result) then
            handle_loop_closure ()
            optimization_with_Full_Bundle_Adjustment (keyframe, 3D points in map)
        end if
    end if
end for
```

4. Experiments

4.1. Dataset and Metrics

Experiments were conducted using the widely recognized KITTI Odometry dataset [21], a real-world benchmark for autonomous driving. The dataset includes various driving scenarios captured through stereo cameras, such as rural roads. Sequences 00, 01,02, 03, 04, 05, 06, 07, 08, 09, and 10 were utilized for the odometry experiments. A detailed description of the dataset is provided in Table 2. The ground truth camera poses for each sequence were compared against the results obtained from the periodic bundle adjustment.

Table 2. Data KITTI Odometry.

Seq.	Environment Description	Number of images
00	Dynamic environment, lots of parked cars, there are motorbikes running	4.541
01	Dynamic environment, there are other cars driving	1.101
02	Dynamic environment, there are other cars driving	4.661
03	Dynamic environment, there are other cars driving; there are parked cars	801
04	Dynamic environment, there are other cars running a small amount	271
05	Static environment, lots of parked cars, no cars moving	2.761
06	Static environment, lots of parked cars, no cars moving	1.101
07	Static environment, lots of parked cars, no cars moving	1.101
08	Dynamic environment, lots of parked cars, motorbikes running, bicycles running	4.071
09	Dynamic environment, there are other cars driving; there are parked cars	1.591
10	Static environment, lots of parked cars, no cars moving	1.201

The metrics commonly used to separately assess the global and local consistency of a SLAM trajectory are absolute pose error (APE) [22]. This paper utilizes the root mean square error (RMSE) of APE to gauge the performance of SLAM.

4.2. Simulation Procedures and Results

This study extends ORB-SLAM3 by incorporating periodic bundle adjustment. The Levenberg-Marquardt algorithm, implemented within the g2o graph optimization framework, performs all nonlinear least-squares optimizations. As shown in Figure 3, periodic BA is compared with the ground truth trajectory (GT) using the KITTI Odometry dataset. The trajectory obtained from ORB-SLAM3 with periodic BA is more closely aligned with the ground truth than the trajectory produced by ORB-SLAM3 alone. These results indicate that the integration of periodic BA in ORB-SLAM3 enhances localization performance. The implementation of periodic BA contributes to more accurate pose estimation for autonomous vehicle.

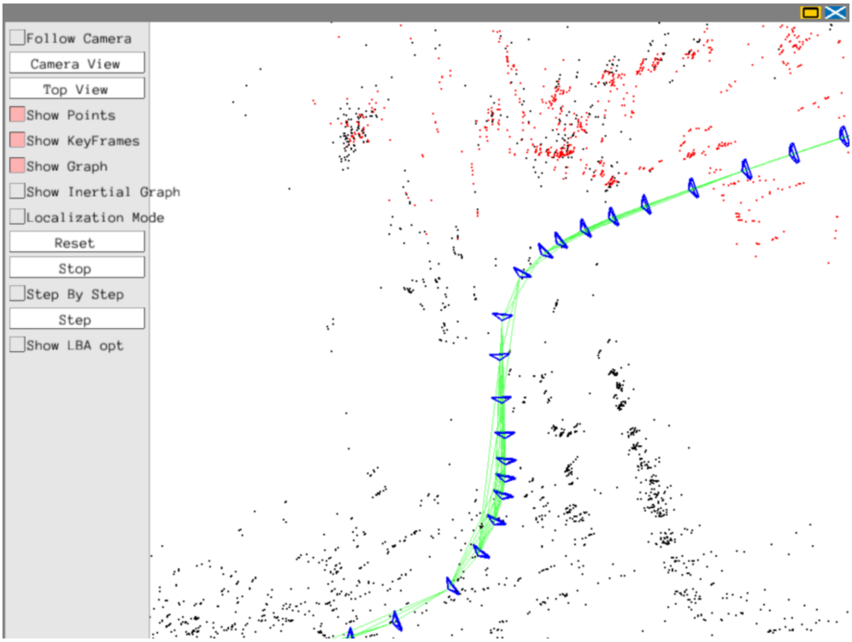
4.3. Ablation Study

Figure 2 presents two illustrations. Figure 2 (a) shows the ORB points generated in each frame, while figure 2 (b) depicts the trajectory, highlights the historical positions of the autonomous vehicle. Two experiments were conducted as follows:

1. Experiment 1: Used ORB-SLAM3 with the KITTI Odometry dataset.
2. Experiment 2: Used ORB-SLAM3 with the addition of periodic bundle adjustment, where optimization was performed for every 100 keyframes. It also used KITTI Odometry dataset.

In both experiments, the vehicle trajectory files generated by the autonomous vehicle were compared against the ground truth data from the KITTI dataset to calculate the RMSE APE.

(a)



(b)

Fig. 2. Graphical User Interface of ORB-SLAM3.

Table 3. The comparison RMSE APE of ORB-SLAM3 and ORB-SLAM3 with periodic BA, measured in meters.

Sequence	ORB-SLAM3 (1)	ORB-SLAM3 with periodic BA (2)	RMSE APE (2) versus (1)	RMSE APE (2) below 0.5 m
00	0.862	0.864	unchange	no
01	4.087	6.829	increase	no
02	3.410	3.221	decrease	no
03	0.329	0.340	increase	yes
04	0.162	0.195	increase	yes
05	0.370	0.393	increase	yes
06	0.383	0.494	increase	yes
07	0.582	0.393	decrease	yes
08	3.027	2.399	decrease	no
09	0.954	0.960	unchange	no

Sequence	ORB-SLAM3 (1)	ORB-SLAM3 with periodic BA (2)	RMSE APE (2) versus (1)	RMSE APE (2) below 0.5 m
10	1.394	1.374	decrease	no

Fig. 3. The comparison RMSE APE of ORB-SLAM3 and ORB-SLAM3 with periodic BA, measured in meters

4.4. Discussion

Table 3 and figure 3 compares the RMSE and APE, both measured in meters, across different sequences from the ORB-SLAM3 experiments with and without the use of periodic bundle adjustment. The analysis can be broken down as follows:

1. RMSE APE in ORB-SLAM3 without Periodic BA

   The RMSE values reflect the accuracy of ORB-SLAM3 in estimating the camera's pose within each sequence, where lower values indicate better performance. Whereas the APE quantifies the absolute difference between the estimated and ground truth poses. Lower APE values correspond to better accuracy. In sequence 03, 04, 05, 06, ORB-SLAM3 performs relatively well without periodic BA, achieving RMSE values below 0.5 meters, demonstrating its ability to maintain pose accuracy over time. However, in sequences such as 00, 01, 02, 07, 08, 09 and, 10, the RMSE APE are high. Notably, in sequence 01, the RMSE reaches 4.087 meters, indicating a drift in pose estimation accuracy, likely due to the accumulation of errors that periodic BA could reduce.

2. RMSE APE in ORB-SLAM3 with Periodic BA

   Bundle adjustment is critical in refining camera trajectories and 3D map points. In ORB-SLAM3, periodic BA corrects the accumulated errors from incremental pose estimation, resulting in more globally consistent maps and trajectories. By periodically optimizing the maps, BA helps prevent severe drift and stabilizes the system.

   In this study, periodic BA on ORB-SLAM3 produced three groups of results, namely (1) able to increase the accuracy of ORB-SLAM3, which occurred in four data sequences, namely 02, 07, 08, and 10, (2) decreased the accuracy of ORB-SLAM3, which occurred in five data sequences, namely 01, 03, 04, 05, and 06, (3) did not affect the accuracy of ORB-SLAM3, which occurred in two data sequences, namely data sequences 00 and 09. In the second group, although periodic BA decreased the accuracy of ORB-SLAM3, the resulting RMSE APE value was still below the threshold of position accuracy for autonomous vehicles, which was 0.5 meters, in four of the five data sequences. The results of this study indicate that periodic BA effectively reduces cumulative deviation and corrects pose errors by periodically re-optimizing the map.

There are weaknesses when applied to data sequence 01 because the accuracy decreased significantly from 4,087 meters to 6,829 meters. In addition, in data sequences 00, 01, 02, 08, 09, and 10, the RMSE value of APE ORB-SLAM3 is still above 0.5 meters, although it has been reduced by periodic bundle adjustments, namely in data sequences 02, 08, and 10. This is due to errors during the initialization process, depicted in Figures 4. The data sequences 00, 01, 02, 08, 09, and 10 have characteristics consisting of many trees and highway environments with few landmarks. This will be addressed in further research.

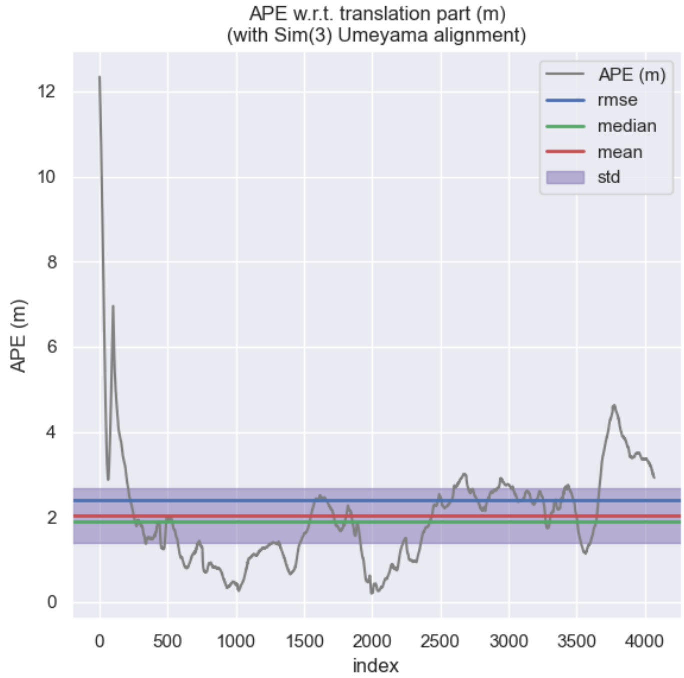


Fig. 4.APE w.r.t. translation part (m) (with Sim(3) Umeyama alignment) of ORB-SLAM3 with periodic BA on sequence 08 of KITTI Odometry dataset.

3. Comparison with related research

BA is widely regarded as essential for improving long-term localization accuracy in visual SLAM. Studies consistently demonstrate that systems integrating BA more frequently exhibit reduced error accumulation, particularly in large or dynamic environments. ORB-SLAM3 follows this trend, with the addition of periodic BA contributing to the observed performance gains in the Table 3. These findings align with other SLAM frameworks, such as RTAB-Map[23] and VINS-Mono[24], where BA is instrumental in minimizing pose drift and ensuring consistency in visual SLAM systems.

5. Conclusion

The integration of periodic BA in ORB-SLAM3 significantly enhances accuracy, as reflected in the reduction of RMSE APE across some sequences. Sequences with higher initial RMSE APE values, such as sequence 08, benefit the most from this optimization. These results are consistent with established findings in SLAM literature, where BA is recognized as an effective method for reducing drift and improving map consistency. In future work, we will try to 1) explore the computational complexity because the using of periodic bundle adjustment and 2) handle errors in the initialization process to reduce the RMSE APE of ORB-SLAM3.

Acknowledgments. This study was funded by Indonesia Endowment Fund for Education Agency.

References

1. Van Brummelen, J., O'Brien, B., Gruyer, D., Najjaran, H.: Autonomous vehicle perception : The technology of today and tomorrow. Transportation Research Part C **89**(February), 384–406 (2018), doi: 10.1016/j.trc.2018.02.012.

2. Zhang, Y., Zhang, H., Wang, G., Yang, J., Hwang, J.: Bundle Adjustment for Monocular Visual Odometry Based on Detections of Traffic Signs. EEE Transactions on Vehicular Technology **69**(1), 151–162 (2020), doi: 10.1109/TVT.2019.2954876.

3. Sualeh, M., Kim, G.: Simultaneous Localization and Mapping in the Epoch of Semantics: A Survey. International Journal of Control, Automation and Systems **17**(3), 729–742 (2019), doi: 10.1007/s12555-018-0130-x.

4. Abdelhafid, E. F., Abdelkader, Y. M., Ahmed, M., Rachid, D., Abdelilah, E. I.: Visual and light detection and ranging-based simultaneous localization and mapping for self-driving cars. International Journal of Electrical and Computer Engineering **12**(6), 6284–6292 (2022), doi: 10.11591/ijece.v12i6.pp6284-6292.

5. Mur-Artal, R., Montiel, J. M. M., Tardos, J. D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics **31**(5), 1147–1163 (2015), doi: 10.1109/TRO.2015.2463671.

6. Bresson, G., Alsayed, Z., Yu, L., Glaser, S.: Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. IEEE Transactions on Intelligent Vehicles **2**(3), 194–220 (2017), doi: 10.1109/TIV.2017.2749181.

7. Wang, K., Zhao, G., Lu, J.: A Deep Analysis of Visual SLAM Methods for Highly Automated and Autonomous Vehicles in Complex Urban Environment. IEEE Transactions on Intelligent Transportation Systems **25**(9), 1–18 (2024), doi: 10.1109/TITS.2024.3379993.

8. Campos, C., Elvira, R., Rodriguez, J. J. G., Montiel, J. M. M., Tardos, J. D.: ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap

SLAM. EEE Transactions on Robotics **37**(6), 1874–1890 (2021), doi: 10.1109/TRO.2021.3075644.

9. Zhang, Y., Tosi, F., Mattoccia, S., Poggi, M.: GO-SLAM: Global Optimization for Consistent 3D Instant Reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3704–3714 (2023), doi: 10.1109/ICCV51070.2023.00345.
10. Rusli, I., Trilaksono, B. R., Adiprawita, W.: RoomSLAM: Simultaneous localization and mapping with objects and indoor layout structure. IEEE Access **8**, 196992–197004 (2020), doi: 10.1109/ACCESS.2020.3034537.
11. Liu, J., Gao, Y., Jiang, X., Fang, Z.: Online object-level SLAM with dual bundle adjustment. Applied Intelligence **53**(21), 25092–25105 (2023), doi: 10.1007/s10489-023-04854-4.
12. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: Part I. IEEE Robotics and Automation Magazine **13**(2), 99–108 (2006), doi: 10.1109/MRA.2006.1638022.
13. Mur-Artal, R., Tardos, J. D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017), doi: 10.1109/TRO.2017.2705103.
14. Fuentes-Pacheco, J., Ruiz-Ascencio, J., Rendón-Mancha, J. M.: Visual simultaneous localization and mapping: a survey. Artificial Intelligence Review **43**(1), 55–81 (2015), doi: 10.1007/s10462-012-9365-8.
15. Chen, C., Zhu, H., Li, M., You, S.: A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives. Robotics **7**(3) (2018), doi: 10.3390/robotics7030045.
16. Alismail, H., Browning, B., Lucey, S.: Photometric bundle adjustment for vision-based SLAM. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10114 LNCS**, 324–341 (2017), doi: 10.1007/978-3-319-54190-7_20.
17. Davison, Andrew J., Reid, Ian D., Molton, Nicholas D., Stasse, Olivier.: MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 1052–1067 (2007), doi: 10.1109/TPAMI.2007.1049.
18. Strasdat, H., Montiel, J. M. M., Davison, A. J.: Real-time monocular SLAM: Why filter?. In : Proceedings - IEEE International Conference on Robotics and Automation, pp. 2657–2664 (2010), doi: 10.1109/ROBOT.2010.5509636.
19. Cai, D., Li, R., Hu, Z., Lu, J., Li, S., Zhao, Y.: A comprehensive overview of core modules in visual SLAM framework. Neurocomputing **590**, 127760 (2024), doi: 10.1016/j.neucom.2024.127760.

20. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G2o: A general framework for graph optimization. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 3607–3613 (2011), doi: 10.1109/ICRA.2011.5979949.

21. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. International Journal of Robotics Research **32**(11), 1231–1237 (2013), doi: 10.1177/0278364913491297.

22. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE International Conference on Intelligent Robots and Systems, pp. 573–580 (2012), doi: 10.1109/IROS.2012.6385773.

23. Labbé, M., Michaud, F.: RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. Journal of Field Robotics **36**(2), 416–446 (2019), doi: 10.1002/rob.21831.

24. Qin, T., Li, P., Shen, S.: VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. IEEE Transactions on Robotics **34**(4), 1004–1020 (2018), doi: 10.1109/TRO.2018.2853729.