Evaluation of a Simultaneous Localization And Mapping algorithm in a dynamic environment using a red green blue - depth camera

Ardhisha Pancham[*1, 2], Daniel Withey[1], Glen Bright[2]

[1]MIAS, MDS, Council for Scientific and Industrial Research, Pretoria, South Africa
[2]Department of Mechanical Engineering, University of KwaZulu-Natal, South Africa
apancham@csir.co.za, dwithey@csir.co.za, brightg@ukzn.ac.za

[*]Corresponding author: Ardhisha Pancham, apancham@csir.co.za, Ph: +27-128412468

Abstract. Simultaneous Localization And Mapping (SLAM) assumes a static environment. In a dynamic environment, the localization accuracy and map quality of SLAM may be degraded by moving objects. By removing these moving objects SLAM performance may improve. Oriented FAST (Features from Accelerated Segment Test) and Rotated BRIEF (Binary Robust Independent Elementary Features) (ORB)-SLAM [1] is a state-of-the-art SLAM algorithm that has shown good performance on several Red Green Blue - Depth (RGB-D) datasets with a moving camera in static and dynamic environments. ORB-SLAM is robust to moderate dynamic changes [1]. However, ORB-SLAM has not been evaluated with a moving RGB-D camera and an object moving at a range of specific linear speeds. This paper evaluates the performance of ORB-SLAM with a moving RGB-D camera in a dynamic environment that includes an object moving at a range of specific linear speeds. Results from experiments indicate that a moving object at lower speeds, in the range tested, degrades the performance of ORB-SLAM and by removing the moving object the performance of ORB-SLAM improves.

Keywords: ORB-SLAM, SLAMIDE, SLAM, dynamic environment, RGB-D camera

1 Introduction

Simultaneous Localization And Mapping (SLAM) enables a mobile robot to construct a map of an unknown, static environment and localize itself simultaneously [2]. Real world environments, however, are dynamic and contain moving objects that may lead to localization errors and reduce map quality. The performance of SLAM In Dynamic Environments (SLAMIDE) may be improved by the removal of moving objects or detecting and tracking these moving objects.

Expensive sensors, such as laser scanners [3] have been used to solve SLAMIDE. Recently, research has involved the use of low-cost sensors such as cameras [4], [5], and RGB-D cameras [1], which generate both color and depth data. However, questions remain regarding sensor type, methods for differentiating stationary and moving objects [6], and how best to track moving objects and predict their positions over time.

Oriented FAST (Features from Accelerated Segment Test) and Rotated BRIEF (Binary Robust Independent Elementary Features) (ORB)-SLAM [1] is a state-of-the-art SLAM algorithm that has shown good performance on several Red Green Blue - Depth (RGB-D) datasets with a moving camera in static and dynamic environments. ORB-SLAM is robust to moderate dynamic changes [1]. However, ORB-SLAM has not been evaluated with a moving RGB-D camera and an object moving at a range of specific linear speeds.

This paper evaluates the performance of ORB-SLAM with a moving RGB-D camera in a dynamic environment that includes an object moving at a range of specific linear speeds. A Vicon motion capture system is used to determine ground truth positions for the moving camera. Experiments show that a moving object at lower speeds, in the range tested, degrades the performance of ORB-SLAM and by removing the moving object the performance of ORB-SLAM improves.

The remainder of this work is organized as follows. Section 2 explains ORB-SLAM, section 3 describes the experimental methods, section 4 contains the experiments and results, and section 5 concludes the paper.

2 ORB-SLAM

A detailed explanation of ORB-SLAM is given in [1]. It builds on Parallel Tracking and Mapping (PTAM) [7] and other algorithms. ORB features are used as they are computationally efficient and rotation invariant [8]. ORB-SLAM comprises of three parallel threads: tracking, local mapping and loop closing, as shown in Fig. 1. The tracking thread performs camera localization and new keyframe decision. The local mapping thread carries out new keyframe processing, local bundle adjustment and redundant keyframe removal. The loop closing thread performs loop detection and closure [1].

Extensive evaluations of ORB-SLAM have demonstrated its excellent accuracy and robustness [9]. It is robust to moderate dynamic changes [1], not affected by brightness variations and offers computational efficiency. However, it is unsuitable for environments without features, similar features may cause incorrect loop closures and drift arises without loop closures [10].

3 Experimental methodology

3.1 Experimental setup

In the experiments, a moving object, a checkerboard (0.21 x 1.27 m) mounted on a Pioneer 3DX robot [11], was moved at a range of speeds (0.01, 0.1, 0.25, 0.5 m/s), in a straight line, in front of a feature-rich, scene. An RGB-D camera, an Asus Xtion Pro Live [12], was moved by hand in a straight line opposite to the direction of motion of the moving object, such that it captured the scene and the moving object, which moved to the center of the scene. Fig. 2 shows the test environment and the moving object.

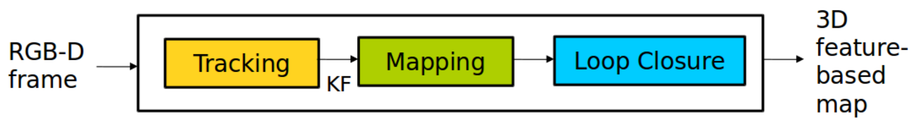

Fig. 1. ORB-SLAM threads, where KF is the keyframe.

Fig. 2. Test environment and moving object.

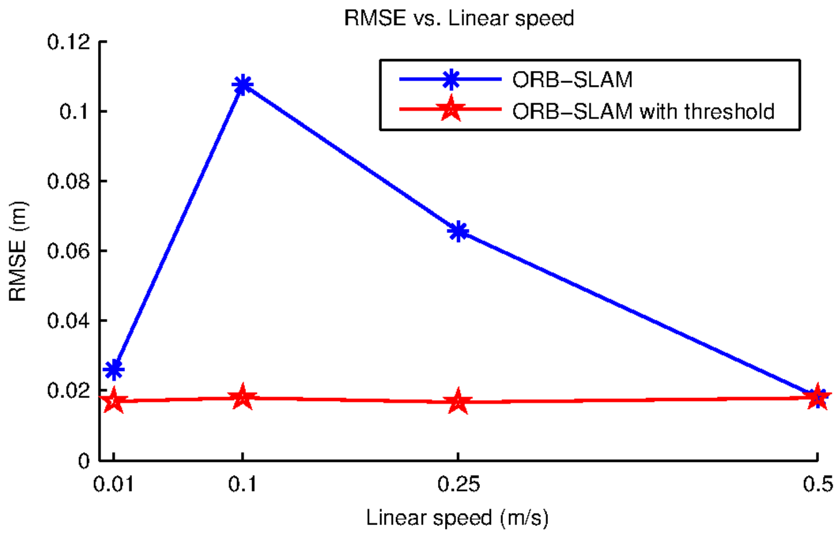

Fig. 3. RMSE versus linear speed.

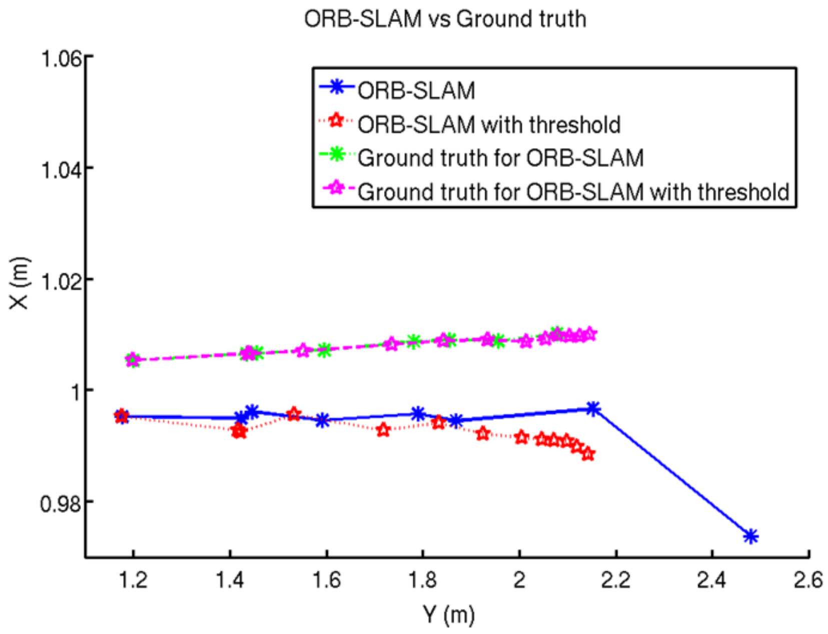

Fig. 4. Camera trajectories for one execution of ORB-SLAM, ORB-SLAM with threshold and ground truth for 0.1 m/s. The markers (stars and asterisks) for ORB-SLAM and ORB-SLAM with threshold indicate selected keyframes. Corresponding points are also marked on the ground truth trajectory.

3.2 RGB-D data

The Robot Operating System (ROS) [13] was used to capture RGB-D camera images of the experiments in ROS bag files [14]. The bag files together formed the dataset for the experiments and served as the inputs to ORB-SLAM.

A Vicon motion capture system was used to record the ground truth of the camera. The ground truth camera pose was computed at the average time between the color and depth image pairs from the ASUS.

3.3 Evaluation measure

Errors in the camera pose were produced by motion of the moving object through the camera field of view. RMS Error (RMSE) was used to evaluate the translation error of the camera pose. RMSE is defined in (1), where n is the number of estimates, $\hat{x}_i$ is the i^{th} estimate, and x_i is the corresponding ground truth.

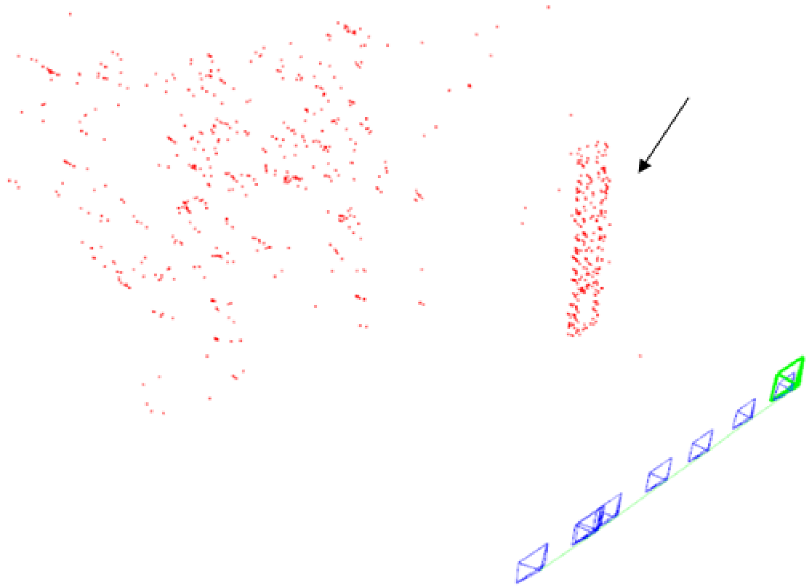

Fig. 5. Map from ORB-SLAM showing spurious measurements indicated by arrow.

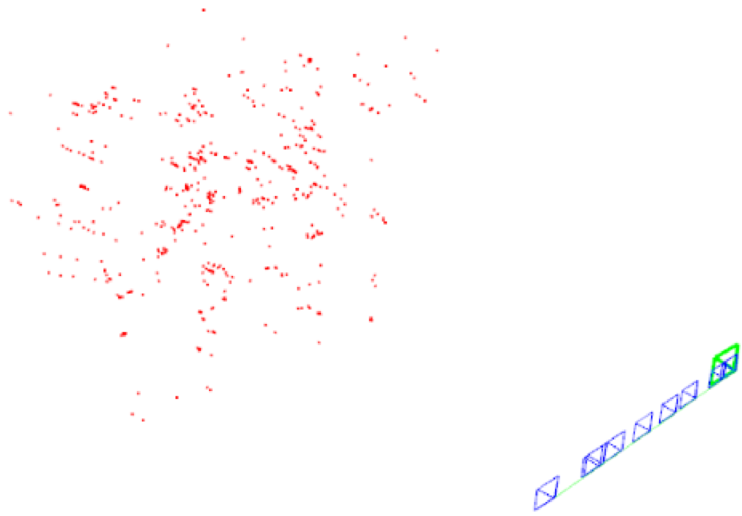

Fig. 6. Map from ORB-SLAM with threshold showing no spurious measurements.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2} \quad \textbf{(1)}$$

4 Results

The experiments were conducted with the ROS on a Linux computer, with an Intel Core i7 - 3720 CPU, with 2.6 GHz clock speed and 8 GB of RAM.

The open-source implementation of ORB-SLAM [15] was evaluated with a moving RGB-D camera in a dynamic environment that included an object moving at a range of specific linear speeds.

The performance of ORB-SLAM was compared to a modified version where a distance threshold was used to remove the moving object from its known position in the environment. The modified version is referred to as ORB-SLAM with threshold.

Default parameters of ORB-SLAM were used. ORB-SLAM and ORB-SLAM with threshold were executed three times on each bag file in the experimental dataset, and the RMSE was calculated for the camera poses produced.

Fig. 3 shows the RMSE of ORB-SLAM and ORB-SLAM with threshold for the linear speeds tested. The error of ORB-SLAM increases as the speed decreases of the moving object from 0.5 to 0.1 m/s. The error is highest at 0.1 m/s (0.11 m) and lowest at 0.5 m/s (0.02 m).

ORB-SLAM with threshold has lower error than ORB-SLAM because the moving object is removed from the SLAM process. By removing the moving object, ORB-SLAM's performance improves by around 50% or more for linear speeds 0.01, 0.1 and 0.25 m/s.

Fig. 4 shows the camera trajectories for one execution of ORB-SLAM, ORB-SLAM with threshold and the corresponding ground truths for 0.1 m/s.

Spurious measurements from the moving object were included in the maps at all speeds for ORB-SLAM but were not visible in the map for ORB-SLAM with threshold as shown in Fig. 5 and Fig. 6.

5 Conclusion

This paper evaluated the performance of ORB-SLAM with a moving RGB-D camera in a dynamic environment that included an object moving at a range of specific linear speeds. Results showed that a moving object at lower speeds degraded localization performance, and spurious measurements from the moving object were included in the map. By removing the moving object, the performance of ORB-SLAM improved by around 50% or more for linear speeds 0.01, 0.1 and 0.25 m/s.

In the tests performed, the environment was known and, in this case, the moving object could be removed by using a distance threshold. In an unknown, dynamic environment, methods for moving object removal and/or methods for detecting and tracking moving objects should be applied to improve SLAM performance.

References

1. Mur-Artal, R., Montiel, J., Tardos, J.: ORB-SLAM: A versatile and accurate monocular SLAM system. In: IEEE Transactions on Robotics, vol. 31, pp.1147--1163 (2015)
2. Durrant-Whyte, H., Bailey, T.: Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. In: Robotics and Automation Magazine, vol. 13, pp.99--110 (2006)
3. Wang, C.-C., Thorpe, C., Thrun, S., Hebert, M., Durrant-Whyte, H.: Simultaneous localization, mapping and moving object tracking. The International Journal of Robotics Research 26(9), 889--936 (2007)
4. Migliore, D., Rigamonti, R., Marzorati, D., Matteucci, M., Sorrenti, D.: Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. In: Proceedings of International workshop on Safe navigation in open and dynamic environments application to autonomous vehicles (2009)
5. Perera, S., Pasqual, A.: Towards realtime handheld monoslam in dynamic environments. In: International Symposium on Visual Computing, pp.313--324 (2011)
6. Wolf, D., Sukhatme, G.: Mobile robot simultaneous localization and mapping in dynamic environments., 53--65 (2005)
7. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on, pp.225--234 (2007)
8. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp.2564--2571 (2011)
9. Mur-Artal, R., Tardos, J.: Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM. In: Robotics: Science and Systems, p.2015
10. Bove, C., Wald, A., Michalson, W., Donahue, M., LaPenta, J.: Collaborative Robotics Heads-Up Display Major Qualifying Project. (2016)
11. Omron Adept MobileRobots: PioneerP3DX. Available at: http://www.mobilerobots.com/researchrobots/PioneerP3DX.aspx
12. ASUSTeK Computer Inc: Xtion pro live. Available at: https://www.asus.com/3D-Sensor/Xtion/
13. ROS: ROS.org. Available at: http://wiki.ros.org/
14. ROS: ROS.org, Bags/Format. Available at: http://wiki.ros.org/Bags/Format
15. Mur-Artal, R.: raulmur/ORB_SLAM2. Available at: https://github.com/raulmur/ORB_SLAM2