SLAM IN SPACE

by

Andrew Balotin

Advised by Dr. Karthik Dantu

December 2024

Submitted to the

Faculty of the Graduate School of

the University at Buffalo, State University of New York

in partial fulfilment of the requirements for the

degree of

Master of Science

Department of Computer Science and Engineering

Table of Contents

List of Tables

List of Figures

Abstract

This paper presents a comparative performance study of several Visual Simultaneous Localization and Mapping (VSLAM) systems in the context of space environments. We investigate the hypothesis that terrestrial based systems are not suited for space environments. These systems range from direct and indirect methods to deep learning based approaches. We evaluate LDSO, DSM, ORB-SLAM3, VINS-FUSION, and DROID-SLAM on space relevant datasets such as the Perseverance Landing, Ingenuity, Perseverance Rover, and the OSIRIS-Rex mission. These datasets were curated from public NASA mission data and included monocular images and IMU data if available. Each dataset exhibits unique challenges such as dim illumination, low texture surfaces, and sparse feature environments. Implementation of each SLAM system was completed through dockerized environments and ran on a dedicated NVIDIA 4090 GPU. Observational results suggest that terrestrial based systems often fail under the difficult conditions of space. Both the direct (LDSO, DSM) and indirect (ORB-SLAM3, VINS-FUSION) systems showed very limited success in outputting pose estimation and easily lost tracking. However, the deep learning system, DROID-SLAM, showed potential with consistent pose estimation across datasets. We conclude that many terrestrial SLAM techniques are inadequate for this domain. This can be potentially improved by more relevant deep

learning feature extraction or integration of multi sensor systems.

Chapter 1

Introduction

1.1 Definition of Problem

Simultaneous Localization and Mapping (SLAM) is a key component in autonomy which enables vehicles to explore and understand unfamiliar environments without relying on a preexisting map. SLAM allows a robot to construct or update a map while simultaneously tracking its own position through that map. In order to do this, the SLAM process requires two critical tasks to happen at once: Localization and Mapping. Localization determines the robot's position and orientation, or pose, relative to its environment. Mapping constructs a 2D or 3D environment based on sensor input. In order to complete both of these tasks, the systems go through a series of steps. SLAM systems first receive input from a variety of sensor data such as camera images, LiDAR, IMU measurements, etc [1]. The system then completes a pose estimation by finding and matching features between data intervals. After, loop closure is performed to identify where a robot has been previously and correct any error in feature position. Lastly, it performs bundle adjustment to optimize and refine camera pose by minimizing the error. Over the last three decades, there have been many developments to SLAM with a large variety of SLAM approaches, such as filter based techniques, graph optimizations, and visual algorithms [1]. Yet despite decades of progress in this field, applying SLAM methods directly to space based missions is still a challenging topic.

As highlighted by recent studies [2], translating SLAM into space environments poses several difficulties. Space missions operate in unstructured environments with sparse features and limited illumination. The lack of identifiable visual and structural features prevents SLAM systems from performing consistently in a repeatable manner. Many of the most popular datasets that modern systems are evaluated on, such as EUROC [3] and KITTI [4], mainly capture indoor and urban environments which lack the texture variability and lighting extremes observed in space.

1.2 Importance

Space missions increasingly require onboard autonomy to be able to safely navigate the surrounding terrain. Although ground support has traditionally been the backbone for many spacecraft operations, communication delays such as the 26-minute round trip to mars, make real time control infeasible as human intervention cannot happen quick enough for critical scenarios. Furthermore, widely used techniques like Stereo Photoclinometry (SCP) which delivers accurate topological information about a small celestial body (SCB), still rely on human oversight. Meanwhile, methods for autonomous roving applications are still not advanced enough to traverse extreme terrain surfaces, such as the lunar poles or high grade surfaces [5]. The drive for spacecraft autonomy is pushed by a desire to increase mission accuracy, robustness, and reliability. While there have been many successful improvements to robotics and autonomy recently, most techniques are not mature enough to broaden the access to space. These challenges emphasize the need for real time SLAM solutions to allow a spacecraft to navigate complex environments with minimal to no ground intervention.

In this paper, popular terrestrial SLAM systems were evaluated on newly created space datasets derived from public NASA mission imagery. Specifically, several VSLAM systems with different feature extraction techniques were selected. The purpose was to confirm the hypothesis that terrestrial based SLAM systems do not work in a space environment due to the extreme constraints. By identifying which systems and feature extraction techniques fail, we provide insights into which SLAM systems can adapt for future autonomous space exploration.

1.3 Historical Background

SLAM has evolved much from its original form in the late 1980s. The initial breakthrough of SLAM established statistical foundations for geometric uncertainty in robotics mapping. Many early SLAM methods used the Extended Kalman Filter approach, which was the backbone of the probabilistic SLAM approaches. However, this faced lots of computational challenges since it needed to maintain and update very large state vectors which included the robots pose and map [6]. This gave way to more efficient filter based methods and eventually evolved to visual SLAM. Visual Slam, at the top level, is broken down into two distinct categories: direct and indirect. Indirect based visual slam extracts feature points from textures in a scene. It keeps track of the descriptor points though a scene and is able to match them in sequential frames. Indirect systems are computationally expensive but are precise and robust. Direct based systems, on the other hand, use pixel-level data to estimate the camera motion directly and build an optimization problem to minimize the overall photometric error. Direct methods will track a pixel's brightness, intensity, color, etc. Direct methods often face large optimization problems and various lighting conditions will negatively impact the system's accuracy [7]. Since both of these categories alone have some disadvantages, other SLAM methods like deep learning have been developed to mitigate some of these challenges. These deep learning methods mainly use a CNN architecture to extract difficult features and optimize the pose. Many of these deep learning systems, however, are difficult to run in real time and require large processing power due to demanding requirements.

Much of the related work for SLAM in space was completed for small celestial bodies (SCB) and employs a variety of methods. Some work focuses on Rao-Blackwellized Particle Filter solutions and uses a monocular camera for VSLAM in order to land on an SCB [8][9]. This work ensures that at least one full rotation of the SCB is made during descent in order to complete loop closure and minimize error. Meanwhile, other systems use the asteroid shape and motion and apply an expectation conditional-maximization (ECM), improving accuracy for a landing site selection [10]. Other work uses optimization based SLAM frameworks that use the spacecraft and asteroid rigid dynamics to create 3D point clouds for a real time alternative to Stereo Photoclinometry (SPC) [11][12]. Similarly, some VSLAM methods strictly use a monocular camera approach and triangulate the surface features to construct a 3D model [13]. For an SCB lacking identifiable landmarks, a topography model is used to extract landmarks and match them with monocular camera images to estimate pose estimation [14]. However, this method requires the topographical map to be pre generated and does not have the ability to generate a real time model of the SCB. While these works focus on mapping and landing on an SCB, there are others that focus on maneuvering around them. A factor-graph SLAM approach was used to incorporate sensor measurements, Earth relative positions, and monocular camera images to navigate around an asteroid [[15]. AstroSLAM extends this by incorporating orbital motion constraints to achieve better performance over standard inertial based methods [16]. Beyond VSLAM setups, some work such as Active Asteroid-SLAM, integrates LiDAR scans for point cloud matching, enabling the estimation of a spacecraft's state while simultaneously building the map [17]. Collectively, these works establish the necessity for on-board, real time SLAM approaches to support autonomous navigation in space environments.

Chapter 2

Solution

2.1 Dataset Creation and Selection

In order to test the terrestrial SLAM systems, space datasets were created using publicly available NASA mission data. There were five unique datasets that were curated. These datasets included the perseverance rover, ingenuity, Mars perseverance rover landing, OSIRIS-Rex, and the lunar reconnaissance orbiter. Each dataset was downloaded as a set of .png images along with IMU data if applicable. Each system was stored in a separate local folder on a local lab server.

Each SLAM system relies on different data formats and file organization to run local datasets. To accommodate multiple SLAM systems, all the space datasets were restructured to match the EUROC standard, which dictates a specific folder layout and file path format. Most, if not all VSLAM systems are set up to run EUROC datasets. Since EUROC data encodes the timestamps into the .png file names and most of the downloaded mission data did not include raw timestamps, each image's index was used as the file name. This was done to seamlessly run the local datasets on each SLAM system. Next, every dataset was converted into a robotic operating system (ROS) .bag file. This format is widely used in SLAM research to combine data like images, IMU data, timestamps, and others into a single file. Many SLAM systems accept and use .bag files to run data.

2.2 SLAM Systems Tested

The VSLAM systems chosen for this project consisted of a mix between indirect, direct, and deep learning feature extraction methods. The table of the selected VSLAM systems and their extraction techniques is given below. By choosing at least one system from each feature category, it helped ensure a balanced analysis.

Table 2.1: Selected VSLAM Systems

VSLAM System	**Feature Extraction**
LDSO	Direct
DSM	Direct
DROID-SLAM	Deep Learning
ORB-SLAM3	Indirect
VINS-FUSION	Indirect

ORB-SLAM3 is a popular indirect SLAM system that extracts and tracks keypoints by using ORB (Oriented FAST and Rotated BRIEF) descriptors. It is able to support monocular, stereo, and visual inertial configurations. ORB-SLAM3 is considered one of the most robust and accurate systems in available literature [18]. VINS-FUSION is a SLAM system designed for a combination of IMU data and camera systems. While it is an indirect feature system, its main purpose is to fuse camera data and inertial measurements into a pose graph optimization to allow for more stable tracking [19].

Direct Sparse Odometry with Loop Closure (LDSO) is a direct monocular SLAM system that can utilize any image pixel that has sufficient intensity gradient. This makes it robust in potentially featureless areas. The system also reliably completes loop closure which results in overall performance comparable to state of the art feature systems [20]. Direct Spare Mapping (DSM) is also a direct monocular SLAM system that focuses on minimizing photometric pixel error by using photometric bundle adjustment (PBA). This system has a persistent map which handles observations [21]. Both of these systems focus on the image pixels rather than feature descriptors like indirect systems.

DROID-SLAM is a deep learning based SLAM system that iteratively refines camera poses and depth maps using learned feature extraction. This system uses a gated recurrent unit (GRU) within a dense bundle adjustment framework which makes it robust and potentially well suited for spare environments [22].

2.3 Experimental Procedure

Every SLAM system used different versions of linux and dependencies. In order to get around this, the systems were each put in their own individual Docker container. This allowed us to prevent any dependency conflicts and troubleshoot each system where needed. Each SLAM system was downloaded onto a local lab computer and ran on a dedicated Nvidia rtx 4090 gpu. All systems ran headless.

Each SLAM system was evaluated by two different metrics for each space dataset. The first metric was tracking percentage. This metric is used to determine how well the SLAM system is able to keep track of features frame by frame. The metric is quantified by looking at the system output results and evaluating the pose estimation. Each pose keyframe has an associated timestamp. If the system did not get through the entire dataset before losing tracking, the last time stamp in the output is taken and compared with the final timestamp in the dataset. This gives a tracking percentage and shows how many frames the system was able to track in the given dataset. The next metric is absolute trajectory error (ATE). This metric is used to compare the accuracy of the output pose estimation from the system. It is the difference between the estimated trajectory and the ground truth for each dataset.

2.4 Results and Observations

Due to an unforeseen hard drive corruption on the server that hosted our experimental outputs, a significant portion of the results was lost. Instead, the observational data will be summarized and highlighted for each system. Below is a table showing the observational results for LDSO. LDSO showed some

Table 2.2: LDSO Results

Dataset	**Tracking Observation**
Perseverance Landing	Minimal
Ingenuity	Minimal
Perseverance Rover	Minimal
OSIRIS-Rex	None

tracking for the mars datasets. However for the asteroid dataset, it showed no tracking at all and routinely failed to provide any pose estimation data. This is somewhat expected as the very dim lighting makes it difficult for direct based systems to perform well.

DSM actually gave no results at all. It did not initialize for some and lost

Table 2.3: DSM Results

Dataset	Tracking Observation
Perseverance Landing	None
Ingenuity	None
Perseverance Rover	None
OSIRIS-Rex	None

tracking after 1-2 frames for the rest. This is on par with what we expected for direct based systems.

Table 2.4: DROID-SLAM Results

Dataset	Tracking Observation
Perseverance Landing	Good
Ingenuity	Good
Perseverance Rover	Good
OSIRIS-Rex	Good

DROID-SLAM actually completed on all datasets. It was able to successfully maneuver and produce results for each dataset. This was somewhat expected as deep learning systems are better at extracting complex features.

Table 2.5: ORB-SLAM3 Results

Dataset	Tracking Observation
Perseverance Landing	Minimal
Ingenuity	Minimal
Perseverance Rover	Minimal
OSIRIS-Rex	Minimal

ORB-SLAM3 had minimal results for all the datasets. This was an expected result as the featureless space makes it difficult to produce ORB descriptors. The Figure below shows SIFT tracking of ORB descriptors between two sequential images in the Ingenuity dataset. In successful feature tracking, the matched feature lines would be horizontal across images. However, in Figure 2.1 they

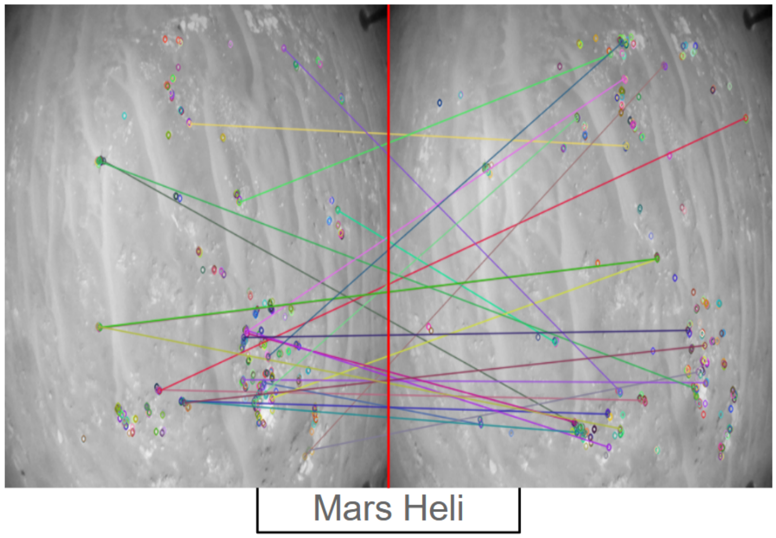

Figure 2.1: Ingenuity ORB Tracking

are sporadic, showing that the features are being matched incorrectly between frames. This emphasizes the difficulties in successful feature tracking in space, even with the most robust terrestrial SLAM systems.

Table 2.6: VINS-FUSION Results

Dataset	**Tracking Observation**
Perseverance Landing	Minimal
Ingenuity	Minimal
Perseverance Rover	Minimal
OSIRIS-Rex	Minimal

Lastly, VINS-FUSION also showed minimal results. It also showed minimal tracking between all the datasets.

Chapter 3

Conclusions

3.1 Summary of Findings

Our analysis of multiple SLAM systems ranging from indirect to direct and deep learning approaches, revealed noticeable differences between the Mars and asteroid datasets. It showed that direct based systems struggled in low light environments and often failed to initialize and maintain tracking beyond a few frames. DROID-SLAM, a deep learning based system, showed promising feature tracking after successfully outputting pose estimation for all datasets. Indirect systems like ORB-SLAM3 and VINS-FUSION demonstrated minimal tracking success, highlighting the difficulties of a visually sparse environment.

Despite the insights gained from this work, a complete solution still remains difficult. While it appears that deep learning based systems can improve feature extraction, there are still questions of real time performance, hardware compatibility, and computational overhead. Furthermore, the application toward even more extreme environments remains open.

3.2 Future Work

Further research is required in order to develop SLAM systems for space environments. Deep learning systems might benefit from more domain specific training such as datasets that include the surface of mars or a SCB. Furthermore, more work needs to be done for integrating these deep learning systems for a low power consumption spacecraft. Understanding how to combine a tensor processing unit with a critical robotics system is essential for deep learning. Lastly, exploring SLAM systems that integrate sensors like LiDAR or thermal imaging could potentially overcome the limitations of optical SLAM.

Bibliography

[1] B. Alsadik and S. Karam. The simultaneous localization and mapping (slam)-an overview. *Surveying and Geospatial Engineering Journal*, 2:01–12, 2021.

[2] R. Giubilato, W. Stürzl, A. Wedler, and R. Triebe. Challenges of slam in extremely unstructured environments: The dlr planetary stereo, solid-state lidar, inertial dataset. *IEEE Robotics and Automation Letters*, 7(4):8721–8728, 2022.

[3] M. Burri, J. Nikolic, P. Gohl, et al. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

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

[5] J.A. Starek, B. Açıkmeşe, I.A. Nesnas, and M. Pavone. Spacecraft autonomy challenges for next-generation space missions. In *Advances in Control System Technology for Aerospace Applications*, volume 460. 2016.

[6] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006.

[7] A. Tourani, H. Bavle, J.L. Sanchez-Lopez, and H. Voos. Visual slam: What are the current trends and what to expect? *Sensors*, 22, 2022.

[8] C. Cocaud and T. Kubota. Surf-based slam scheme using octree occupancy grid for autonomous landing on asteroids. 2010.

[9] C. Cocaud and T. Kubota. Autonomous navigation near asteroid based on visual slam. 2012.

[10] N. Takeishi, T. Yairi, Y. Tsuda, F. Terui, N. Ogawa, and Y. Mimasu. Simultaneous estimation of shape and motion of an asteroid for automatic navigation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2861–2866, 2015.

[11] F. Baldini, A. Harvard, S. Chung, I.A. Nesnas, and S. Bhaskaran. Autonomous small body mapping and spacecraft navigation. 2018.

[12] A. Rathinam and A. Dempster. Monocular vision based simultaneous localization and mapping for close proximity navigation near an asteroid. 2017.

[13] P. Tsiotras. Shape carving in space: A visual slam approach to 3d shape reconstruction of a small celestial body. 2020.

[14] O. Knuuttila, A. Kestilä, and E. Kallio. Synthetic photometric landmarks used for absolute navigation near an asteroid. *The Aeronautical Journal*, 124:1–20, 2020.

[15] M. Dor, K.A. Skinner, P. Tsiotras, and T. Driver. Visual slam for asteroid relative navigation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2066–2075, 2021.

[16] M. Dor, T. Driver, K.M. Getzandanner, and P. Tsiotras. Astroslam: Autonomous monocular navigation in the vicinity of a celestial small body - theory and experiments. 2022.

[17] D. Nakath, J. Clemens, and C. Rachuy. Active asteroid-slam. *Journal of Intelligent Robotic Systems*, 99:303–333, 2019.

[18] C. Campos, R. Elvira, J.J. Rodríguez, J.M.M. Montiel, and J.D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37:1874–1890, 2020.

[19] T. Qin, J. Pan, S. Cao, and S. Shen. A general optimization-based framework for local odometry estimation with multiple sensors. 2019.

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

[21] J.A. Zubizarreta, I. Aguinaga, and J.M. Montiel. Direct sparse mapping. *IEEE Transactions on Robotics*, 36:1363–1370, 2019.

[22] Z. Teed and J. Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Neural Information Processing Systems*, 2021.