LINE-BASED MONOCULAR SLAM FOR SPACECRAFT RPO

Iason Georgios Velentzas*, Mehregan Dor, and Panagiotis Tsiotras; Georgia Institute of Technology, Atlanta, GA, 30313, USA, *[ivelentzas3@gatech.edu]

Abstract. *Simultaneous Localization and Mapping (SLAM) algorithms have demonstrated considerable efficacy in ground robotics applications. However, their application to space-based scenarios is constrained by many challenges, such as harsh lighting, large distances between the observer and the target, as well as on-board computational limitations. With the goal of enhancing resilience against conditions encountered in orbit, we incorporate line features as a stronger geometric primitive. The algorithm demonstrates increased performance on synthetic datasets generated specifically to capture these characteristics, as well as real-life imagery from a NASA proximity operation to the Hubble Space Telescope.*

Introduction. As the number of objects in orbit increases, it is essential for the future spacecrafts to have the ability to autonomously repair or remove the already deployed objects. Active Debris Removal (ADR) comprises one of the strategic goals of the European Space Agency (ESA), while In-Space Assembly and Manufacturing (ISAM) is the goal of various flight projects (e.g. OSAM-1, OSAM-2, Robotic Refueling Missions). These applications require increased autonomy and robustness of the Rendezvous and Proximity Operations (RPO) subsystems.

We consider the problem of a Chaser (C) satellite and a non-cooperative Target (T) satellite that are located in different orbits and do not exchange any information between each other. The goal is for the C satellite to accurately estimate a map of the T satellite and localize itself with respect to T.

The main sensor in our scenario is a monocular camera, which is a lightweight and inexpensive sensor, favored for space applications. A major drawback of this sensor is that it is heavily affected by lighting conditions. The sun being the sole lighting source in our case creates reflections, occlusions, and shadowing, which play a significant role in the quality of the image measurement. This leads even state-of-the-art Visual SLAM (VSLAM) pipelines to fail and lose track of the target[1]. One of the main reasons for this failure is the nature of feature points. Firstly, points are typically concentrated on a few planar surfaces, which results in a sudden loss of a lot of the features during the rotation of the target. Secondly, point features are not reliable under the described drastically changing lighting conditions, where even light self-shadowing can lead to the detection of non-existent features. To mitigate these effects, we employ line features in the VSLAM solution, which typically lie at the intersections of planes and can provide more reliable and useful information than points.

We are building upon an existing solution of Asteroid Relative Navigation[2], which is modified for the Spacecraft RPO scenario. Our algorithm receives images captured from a monocular camera on the C satellite. We wish to estimate the rigid-body transformation T_{TC} between the arbitrary target-fixed frame **T** and the known chaser-fixed frame **C**, while also maintaining an estimate of the 3D point and line landmarks of the target satellite.

Proposed Approach. We employ a keyframe-based SLAM approach and hence, the pipeline consists of three main modules: Frontend, Keyframe Selection, and Backend.

Frontend. Point features are represented as ORB keypoints, which are efficient and robust against rotation changes and noise. Only the matched point pairs that respect the epipolar constraint within a threshold of 1 pixel are considered to be inserted into the graph. Line Segments are detected in the images using the ELSED[3] algorithm, which is outperforming other traditional line detection approaches both in computational time and average precision. The algorithm also uses a pixel jump to connect line segments, which makes the detection more robust against self-shadowing and occlusions.

Feature matching is based on the image gradients of the local area around each feature. In the spacecraft RPO scenario the images might be poorly textured, or suffer from sharp shadows and specular reflections. This might change the appearance of the target drastically over the course of even a few images and even though the features might still be detected, the change in their appearance might not allow for correct matching. We employ the Tracking submodule, which tracks the features across all of the input images. For a feature to be inserted into the graph it has to be tracked through all of the frames since the last keyframe, allowing for a small sliding window in the matching process. As a result, features are correctly matched in a frame-by-frame fashion and erroneous matches are excluded from the graph.

Keyframe Selection. Downsampling the input images to be inserted in the optimization scheme is typically accomplished through hand-crafted heuristics regarding the spatial or temporal distance of the frames, their optical flow or the number of tracked features. Given the specific characteristics of the relative motion that the spacecraft RPO scenario possesses, we deem critical to incorporate multiple heuristics in our keyframe selection framework to ensure sufficient appearance change under various extreme cases, such as pure rotation or loss of the majority of the features. We utilize the following heuristics: 1) optical flow, 2) estimated parallax, 3) number of tracked features and 4) percentage of tracked features and based on the current values of the heuristics, we adapt their thresholds.

Backend. We cast the incremental smoothing optimization problem as a factor graph using the iSAM2

4th Space imaging Workshop. Atlanta, GA.
7-9 October 2024.

engine[4] within the GTSAM[5] library framework. We encode the line segment measurements as nonlinear factors in the factor graph. The line segment endpoints are unstable, since they are highly affected by noise, occlusions and lighting conditions. Hence, we choose to use the corresponding infinite lines in our optimization scheme. The 3D lines are minimally parameterized by their orthonormal representation and the reprojection error is intuitively defined as the sum of the distances between the measured 2D line segment endpoints and the reprojection of the 3D line.

Results. We provide qualitative results on the Frontend module's performance based on the real-life dataset from the NASA STS-125 Servicing Mission 4 (SM4) and quantitative results on the full pipeline based on a synthetic dataset of the Hubble Space Telescope.

Frontend. In Fig. 1 we visualize the tracked features between two consecutive keyframes of the SM4 dataset. The left images correspond to the last keyframe, while the right images correspond to the current keyframe. As we can see, the point features (top) are concentrated in high-textured surfaces of the satellite, while line features (bottom) are able to capture information about all the parts of the target.

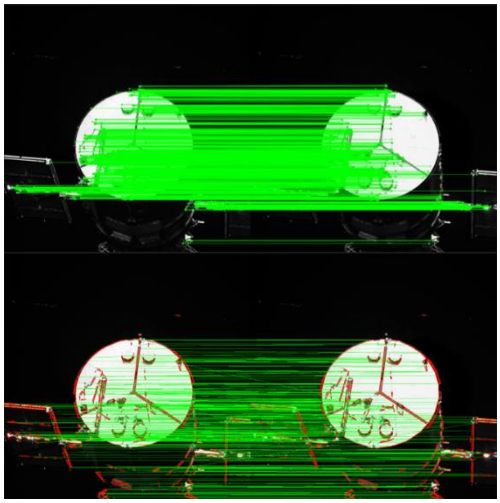

Figure 1. Feature tracking results between consecutive keyframes. Top: Keypoints. Bottom: Line segments.

Backend. We provide results regarding the localization performance of the pipeline (PL), when compared to the only-points version of it (P). Fig. 2 presents the position and Fig. 3 presents the attitude error as expressed in the Local Vertical Local Horizontal (LVLH) frame for a Blender dataset. The relative distance between the satellites is approximately 100 m throught the trajectory. We observe that the norm of both the position and attitude error decrease, which is a consequence of the higher quality and stronger geometric constraint enforced from the line features. Finally, Tab. 1 shows the Root Mean Squared Error (RMSE) of the position and the attitude.

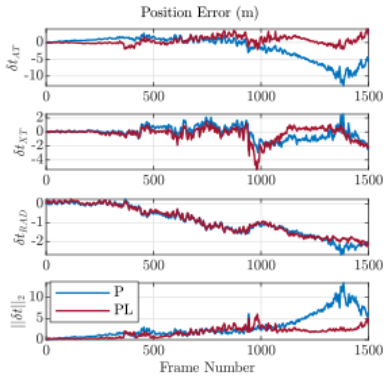

Figure 2. Position error (m) in LVLH. Rows 1-3 indicate the across track, cross track, and radial directions, while row 4 shows the error norm.

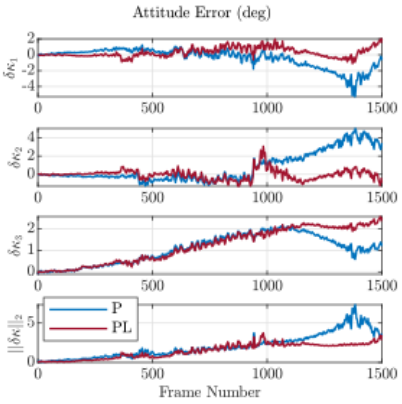

Figure 3. Attitude error (deg) in LVLH. Rows 1-3 indicate the across track, cross track, and radial directions, while row 4 shows the error norm.

Table 1. Error comparison between P and PL versions. The first number of each cell refers to position error (m) and the second number to attitude error (deg)

	P	PL	% improvement
AT	3.85 / 1.32	1.37 / 0.73	64.44 / 44.56
XT	1.09 / 1.82	1.13 / 0.64	-2.90 / 64.67
RAD	1.23 / 1.25	1.17 / 1.46	5.06 / -14.79
norm	4.19 / 2.57	2.12 / 1.75	49.29 / 31.54

References.

[1] M. Dor et al, "ORB-SLAM Applied to Spacecraft Non-Cooperative Rendezvous", *Space Flight Mechanics Meeting,* AIAA, Jan. 2018. DOI: 10.2514/6.2018-1963.

[2] M. Dor et al, "AstroSLAM: Autonomous Monocular Navigation in the Vicinity of a Celestial Small Body", 2022, arXiv:2212.00350

[3] I. Suarez et al, "ELSED: Enhanced Line SEgment Drawing", *Pattern Recognition*, 128, 2022, DOI: 10.1016/j.patcog.2022.108619/.

[4] M. Kaess et al, ""iSAM2: Incremental smoothing and mapping using the Bayes tree", *The International Journal of Robotics Research*, 31(2),2012,pp. 216–235, DOI: 10.1177/0278364911430419.

[5] F. Dellaert et al, "Factor Graphs for Robot Perception", *Foundations and Trends in Robotics,* 6(1-2), 2017, pp. 1-139