

Regular Article

# GPGM-SLAM: a Robust SLAM System for Unstructured Planetary Environments with Gaussian Process Gradient Maps

Riccardo Giubilato<sup>1</sup>, Cedric Le Gentil<sup>2</sup>, Mallikarjuna Vayugundla<sup>1</sup>,  
Martin J. Schuster<sup>1</sup>, Teresa Vidal-Calleja<sup>2</sup> and Rudolph Triebel<sup>1</sup>

<sup>1</sup>Institute of Robotics and Mechatronics, German Aerospace Center (DLR)

<sup>2</sup>Centre for Autonomous Systems, University of Technology Sydney

**Abstract:** Simultaneous Localization and Mapping (SLAM) techniques play a key role towards long-term autonomy of mobile robots due to the ability to correct localization errors and produce consistent maps of an environment over time. Contrarily to urban or man-made environments, where the presence of unique objects and structures offer unique cues for localization, the appearance of unstructured natural environments is often ambiguous and self-similar, hindering the performances of loop closure detection. In this paper, we present an approach to improve the robustness of place recognition in the context of a submap-based stereo SLAM based on Gaussian Process Gradient Maps (GPGMaps). GPGMaps embed a continuous representation of the gradients of the local terrain elevation by means of Gaussian Process regression and Structured Kernel Interpolation, given solely noisy elevation measurements. We leverage the imagelike structure of GPGMaps to detect loop closures using traditional visual features and Bag of Words. GPGMap matching is performed as an SE(2) alignment to establish loop closure constraints within a pose graph. We evaluate the proposed pipeline on a variety of datasets recorded on Mt. Etna, Sicily and in the Morocco desert, respectively Moon- and Mars-like environments, and we compare the localization performances with state-of-the-art approaches for visual SLAM and visual loop closure detection.

**Keywords:** planetary robotics, SLAM, localization, extreme environments, learning

## 1. Introduction

Within the last decades, the field of mobile robotics underwent a significant technological leap in terms of enhanced mobility, sensory perception and planning of actions. The resulting increased capabilities are crucial, within our modern society, as they provide tools for disaster mitigation [Klamt et al., 2018], search and rescue operations [De Cubber et al., 2013, Delmerico et al., 2019] and exploration of hazardous and unreachable environments, such as in the case of exploration

---

Received: 14 June 2021; revised: 15 July 2022; accepted: 26 July 2022; published: 10 August 2022.

**Correspondence:** Riccardo Giubilato, Institute of Robotics and Mechatronics, German Aerospace Center (DLR),

Email: [riccardo.giubilato@dlr.de](mailto:riccardo.giubilato@dlr.de)

This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2022 Giubilato, Gentil, Vayugundla, Schuster, Vidal-Calleja and Triebel

DOI: <https://doi.org/10.55417/fr.2022053>

of planetary bodies [Wedler et al., 2021]. In many occasions, the robotic assets need to operate autonomously, without relying on the intervention of human operators. This is the case of robotics for the exploration of Mars or the Earth's moon, where communication delays limit the possibility of manual intervention [Heverly et al., 2013]. On Earth, the scenario of subterranean exploration imposes similar constraints, due to the possibility of communication dropouts [Otsu et al., 2020].

The principal focus of current planetary rover missions lies on traversing from one target of scientific interest to the next [Heverly et al., 2013], therefore moving faster between goals is a key requirement for increasing the overall return of the mission [Farley et al., 2020, Allouis et al., 2011, Marc et al., 2020]. Future prospects for planetary exploration, however, will likely involve sample return [Mattingly and May, 2011] or cooperation of robotic agents to build infrastructure [Heinicke and Foing, 2021, Cesaretti et al., 2014, Thangavelautham et al., 2020]. In this case, the predominant paths of robots would often intersect themselves while revisiting sites. The need to be precisely localized with respect to any target during operations, as well as the need of accurate maps of the terrain for efficient planning of actions, requires centimeter-level localization. This is orders of magnitude lower of what is achievable relying on pure dead reckoning, as well as on global localization techniques that make use of DEMs [Chiodini et al., 2017, Nefian et al., 2014, Carle et al., 2010] or imagery [Wu et al., 2019] built from satellite observations.

A solution to this problem is provided by the technique referred to as Simultaneous Localization and Mapping (SLAM) [Durrant-Whyte and Bailey, 2006]. SLAM consists in utilizing sensory inputs to concurrently build a map, or a representation of the environment, and localize an observer with respect to it. Visual SLAM techniques exploit cameras for ego-motion estimation, mapping and place recognition and, to this day, are mature enough to find commercial application in the domains of autonomous driving [Cadena et al., 2016a] or augmented reality (AR) [Li et al., 2017]. Traditional Visual SLAM approaches [Mur-Artal and Tardos, 2017, Qin et al., 2018], based on detection and tracking of visual features, are proven to be accurate and robust against failures and drift in the pose estimation over time, thanks to well-developed loop closure detection techniques [Galvez-Lopez and Tardos, 2012, Cummins and Newman, 2008], which allow to establish global optimization problems to correct trajectory and map by enforcing pose constraints while revising places.

However, the typical scenarios where this family of algorithms is developed and tested comprise often man-made or, in general, feature-rich environments [Geiger et al., 2013, Sturm et al., 2012]. As an example, the task of visual place recognition from the point of view of a moving car is subject to strong prior constraints on the camera perspective: the car would likely revisit a place from the same direction or the opposite, as its motion is constrained by the lanes on the road. Contrarily, visual place recognition in unstructured natural environments is challenging due to aliasing and lack of unique visual or structural features that suggest unambiguously a previously visited location [Giubilato et al., 2020]. This is the case in the planetary exploration scenario, where unique landmarks are scarce and hardly recognizable due to the environment appearance as well as the perceptive behaviors of mobile robots performing autonomous navigation in such context. The necessity to use visual inputs for the purpose of traversability estimation and hazard avoidance results in arbitrary viewpoints for cameras, usually mounted on pan-tilt units, hindering the likelihood of collecting images of the same landscapes from repeatable positions and view directions. This is further aggravated by the fact that robots do not necessarily traverse the same paths as there are typically no designated roads or pathways on such rough terrains.

This paper builds on our previous work [Le Gentil et al., 2020] where we introduced the concept of Gaussian Process Gradient Maps (GPGMaps): local representations of the environment obtained by fitting an elevation model on submaps [Schuster et al., 2019] through Gaussian Process regression and applying linear operators to compute the spatial derivatives. Thus, instead of relying on the visual similarity of camera images, the similarity of the terrain elevation is leveraged to both select and validate matches between GPGMaps by means of common visual techniques. Through Gaussian Process regression, GPGMaps represent the environment in a continuous manner, robust to measurement noise, allowing to detect the similarity of the terrain regardless of the direction of travel and the distance of the viewpoints. In doing so, we rely on the assumption that the terrain is

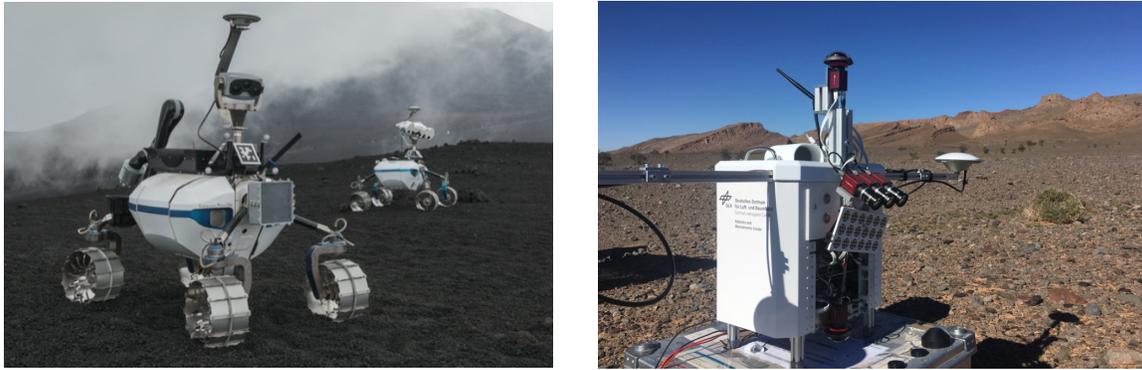
not locally flat and contains at least a minimum amount of structure. Although the presence of rocks or any unique and salient structure is not necessary for effectively matching GPGMaps, the terrain is required to be characterized by ever so tiny differences in elevation such that the resulting gradient images from the Gaussian process representation contain any details. As we rely on the gradient of elevation, even just pebbles or ripples in the soil due to wind or erosion, peculiarities that are typical of any unstructured natural environment and terrains that are traversable from wheeled rovers, are enough to permit features to be extracted. We exploit GPGMaps to build a graph-based SLAM architecture targeted at stereo vision-based perception systems for robots operating in challenging natural and unstructured environments.

Thanks to the involvement in the Helmholtz alliance ROBEX “Robotic Exploration of EXtreme Environments” [Wedler et al., 2017] in 2012-2017 and the project ARCHES “Autonomous Robotic Networks to Help Modern Societies” [Schuster et al., 2020], 2018-2022, the DLR (German Aerospace Center) Institute of Robotics and Mechatronics is conducting specific research on enabling autonomy for an team of heterogeneous robots which comprises an UAV, ARDEA [Lutz et al., 2020], and a planetary-like rover, the Lightweight Robotic Unit (LRU) [Schuster et al., 2017]. Within the final demonstration mission of project ROBEX, a variety of datasets has been recorded with accurate differential GPS ground truth, where the LRU robots perform different tasks, e.g., waypoint navigation, autonomous exploration, and long-range traversing [Vayugundla et al., 2018]. The mission took place on Mt. Etna, Sicily, a volcanic environment designated as a Moon-analogue site and especially challenging for the purpose of localization for the reasons previously introduced.

In the context of collecting datasets in remote planetary-like environments, in 2018, the DLR Institute of Robotics and Mechatronics recorded the Morocco-Acquired Dataset of Mars-Analogue eXploration (MADMAX) [Meyer et al., 2021] on a variety of Mars-analogue sites in Morocco. This dataset specifically addresses the purpose of benchmarking localization algorithms in challenging Mars-like scenarios and was recorded with a perception system, the Sensor Unit for Planetary Exploration Rovers (SUPER), analogous to the one of the LRU rovers and the ARDEA UAV. Figure 1 shows a variety of views of the Etna and Morocco sites respectively on the top and bottom rows, highlighting the similarity of both locations to lunar and martian scenes. Figure 2 gives instead an impression of the LRU rover and the SUPER system.



**Figure 1.** Impressions of the Moon-like landscape on the volcano Mt. Etna (top) and of the Mars-like landscape in the Moroccan desert (bottom).



**Figure 2.** Pictures of the Lightweight Rover Unit (LRU) rovers on Mt. Etna (left) and the Sensor Unit for Planetary Exploration Rovers (SUPER) handheld system in the Morocco desert (right).

By evaluating the proposed SLAM system on the Mt. Etna and Morocco datasets, we analyze the performance of our approach in unstructured environments, which closely represent the challenges in localization and mapping brought by planetary-like scenarios where no prior assumptions can be made. We furthermore compare the localization performances of GPGM-SLAM with state of the art visual SLAM systems such as ORB-SLAM2 [Mur-Artal and Tardos, 2017] and VINS-Mono [Qin et al., 2018], as well as its pure place recognition capabilities with the visual loop closure detector iBoW-LCD [Garcia-Fidalgo and Ortiz, 2018]. We therefore highlight the specific problems that hinder the chances of establishing loop closures and report our lessons learned for the researchers working on SLAM systems on similar field robotics applications.

More specifically, this paper presents the following contributions going beyond our previous works on GPGMaps:

- we improve on [Le Gentil et al., 2020] by implementing an efficient GP regression method based on Structured Kernel Interpolation (SKI) [Wilson and Nickisch, 2015] to infer the gradient of the terrain elevation using solely noisy elevation measurements. SKI allows us to sample the GP output on a denser domain and reduce the computational complexity from  $\mathcal{O}(n^3)$  to  $\mathcal{O}(n)$ . Furthermore, we propose a variation of the SKI method, which permits to generate gradients with only elevation as input.
- we present a full SLAM system based on GPGMaps with loop closing capabilities showing unprecedented localization performances on completely unstructured scenarios.
- we test our pipeline on a variety of challenging datasets recorded with the LRU rover and the SUPER sensor box during field tests on Mt. Etna, Sicily and in the Morocco desert, which are respectively designated as Moon- and Mars-analogue environments, for durations of travel up to 1 hour. We demonstrate higher localization performances compared to our current SLAM system, targeted at teams of heterogeneous robots [Schuster et al., 2019], and compared to the state of the art visual SLAM systems ORB-SLAM2 and VINS-MONO.
- we analyze in detail the properties of structure- and visual-based loop closure detection in the context of unconstrained camera trajectories in natural environments. We then demonstrate how our approach leads to higher chances of detecting previously observed terrain structures compared to pure visual similarity, also when revisiting places from opposite travel directions.

This paper is structured as follows. In Section 2, we give an overview of related works about SLAM in outdoor unstructured environments, in Section 3, we introduce the components of our SLAM system and the integration of GPGMap matching as an alternative to our baseline configuration. In Section 4 we introduce the concept of GPGMap and in Section 5 we explain how GPGMaps are used in the context of detecting loop closures. Section 6 presents a preliminary assessment of the precision of selecting candidate GPGMap matches using the BoW paradigm. Section 7 contains the evaluation

of GPGM-SLAM on a variety of sequences in planetary analogue environments and is followed by a discussion on the different behaviour of structure- and visual-based loop closure detection. Finally, we draw the conclusions and potential developments of GPGM-SLAM in Section 8.

## 2. Related Works on Outdoor SLAM

Visual localization in challenging outdoor scenarios has been successfully performed for many decades now. A notable example is the implementation of Visual Odometry (VO) onboard the Mars Exploration Rovers (MER) Spirit and Opportunity [Olson et al., 2003] to decrease localization drift accumulated from the wheel slippage on soft grounds. This allowed the rovers to travel on longer traverses for many years [Maimone et al., 2007] without the need of constant external supervision, as was required for NASA’s Mars Sojourner rover, which was teleoperated from ground control through the video feedback of the lander cameras [Wilcox and Nguyen, 1998] and used wheel odometry and a gyroscope to navigate towards close targets. Although VO techniques have been proven to be a robust manner to estimate a vehicle’s ego motion, the ever present drift, albeit low, is never compensated. To this end, Simultaneous Localization and Mapping (SLAM) techniques [Durrant-Whyte and Bailey, 2006] have been developed to solve both the localization and mapping problems concurrently, reducing localization drift thanks to the ability of establishing loop closures when revisiting known places. The earliest formulations of the probabilistic SLAM problem leveraged Extended Kalman Filters (EKF) [Bailey et al., 2006] and particle filters [Montemerlo et al., 2003]. The first is essentially limited by the computational complexity being  $O(n^2)$  with  $n$  the increasing number of state coordinates including robot poses and landmarks. The second approach, although solving the intrinsic problems of EKF-SLAM, namely computational complexity through independence of landmarks and assumption of Gaussianity of the motion and sensor models [Grisetti et al., 2005], is limited by the memory consumption, proportional to the number of particles, and, consequently, the poor scalability to higher dimensional spaces (e.g., for 6D SLAM instead of planar). A turning point was reached with graph-based formulations [Thrun and Montemerlo, 2006, Grisetti et al., 2010] and smoothing approaches [Kaess et al., 2008, Kaess et al., 2012] where the sparsity of the SLAM problem was leveraged to increase computational efficiency.

Visual SLAM is traditionally performed through detection and tracking of visual features [Bay et al., 2006, Rublee et al., 2011, Lowe, 2004]. The task of recognizing previously visited places relies on detecting the occurrence of already observed features, mostly by means of aggregation techniques such as Bag of Words [Galvez-Lopez and Tardos, 2012] or Vectors of Locally Aggregated Descriptors (VLAD) [Arandjelovic and Zisserman, 2013]. For a comprehensive summary of visual place recognition techniques, we refer the reader to [Lowry et al., 2016]. In recent years, visual SLAM systems, such as ORB-SLAM2 [Mur-Artal and Tardos, 2017] or LSD-SLAM [Engel et al., 2014], reached a maturity such that they can successfully be deployed in mobile robots or devices to perform mapping and localization in feature-rich environments. To this day, however, while excelling in the fields of augmented reality and autonomous driving [Bresson et al., 2017, Cadena et al., 2016b], modern visual SLAM systems have yet to demonstrate successful long term operations in unstructured and homogeneous scenarios, as also demonstrated by the aims of DARPA’s program Robotic Autonomy in Complex Environments with Resiliency (RACER). The self-similarity and lack of diverse visual features, in fact, challenges the ability of recognizing already visited places and therefore the possibility to compensate localization drift. This is aggravated by the strict dependency of visual place recognition on the repeatability of viewpoints, which is not guaranteed for the case of unconstrained robot navigation in outdoor scenes. To this end, earlier approaches, targeting the specific application scenario of exploring planetary-like environments, leveraged visual saliency to segment repeatable regions in the image and track them through visual features [Bajpai et al., 2016]. Although compensating for the self-similarity of the environment and minimizing the computational load, the possibility of detecting loop closures is still dependent on the repetition of viewpoints. The presence of pan-tilt mechanisms on planetary rovers allows to plan the camera viewpoints in order to maximize the accuracy of VO [Otsu et al., 2017] and potentially facilitate the detection

of loop closures, the success of which is however dependent on the localization uncertainty and can become hard to tackle. A different path towards visual place recognition belongs to the SeqSLAM approach [Milford and Wyeth, 2012, Siam and Zhang, 2017], proven to work with some adaptations also on very challenging unstructured and self-similar scenarios [Grixa et al., 2018]. This approach, however, consists in matching continuous sequences of images, which poses again strong constraints on the sequence of viewpoints and is more targeted to path following or homing tasks.

The utilization of range sensors in the context of SLAM helps to overcome the limitations of traditional visual sensing. Earlier approaches leveraged plane scanning LiDARs mounted on pan-tilt units to produce scans with 360° coverage, using the scans for either obstacle detection [Schafer et al., 2008] or trajectory planning and localization [Rekleitis et al., 2009] in outdoor environments. Following the recent democratization of LiDAR sensors, many odometry and SLAM systems have been developed using 3D LiDARs as the main sensing source [Zhang and Singh, 2014, Shan and Englot, 2018, Le Gentil et al., 2021] and demonstrated to work reliably and accurately in urban landscapes where the surrounding structures (e.g., buildings, cars, and trees) well constrain rigid registration algorithms such as Iterative Closest Point (ICP). Even though modern LiDAR odometry methods achieve high accuracy in pose estimation on long travels, the ever growing localization drift must be compensated by establishing loop closures. Many approaches for loop closure detection based on LiDAR scans have been developed recently and follow different strategies. Similarly to visual methods, the detection of 3D features, such as SHOT [Salti et al., 2014] or FPFH [Rusu et al., 2009] and their variations, allow to perform place recognition by matching keypoints from point clouds in moderately unstructured scenarios. In [Guo et al., 2019], SHOT descriptors augmented by LiDAR intensity are used in a probabilistic voting scheme to find candidate point cloud matches from a database. Intensity of LiDAR measurements is used also in [Cop et al., 2018], to produce spatial histograms as global point cloud descriptors. Other approaches employ projection of ranges or intensity from point clouds to perform feature extraction on images. NARF (Normal Aligned Radial Features) are used in [Steder et al., 2011] in a Bag of Word scheme to find candidate matches from urban and indoor scenes and discard ambiguous point clouds through a self-similarity test. Other approaches rely on global descriptors of LiDAR scans, such as ScanContext [Kim and Kim, 2018], or on segmentation and accumulation of distinct point clusters, such as SegMap [Dubé et al., 2018, Dubé et al., 2020]. The dependency on specific assumptions about the target environment, such as the presence of evident structures that can be easily segmented either by removing a traversable ground or by matching primitive shapes [Pierzchała et al., 2018], makes these approaches unsuitable for generic natural and planetary-like landscapes. Furthermore, to this day no mechanical or, in general, 3D LiDAR sensor has been deployed on an autonomous rover mission for planetary exploration, with the exception of NASA's Ingenuity helicopter which carried an 1D LiDAR altimeter for the purpose of altitude estimation. Therefore, in our work, we will leverage structural information but in the context of stereo vision.

Among SLAM systems that specifically address the target application of rovers in planetary-like scenarios, [Hidalgo-Carrió et al., 2018] implements a SLAM system where an odometry model is inferred through the usage of Gaussian Process regression to complement a visual pipeline where the extracted keyframes are added to a graph in an adaptive manner, as to maximize the information gain and minimize the computational load. The overall system has been successfully tested on analogue planetary terrains, where however the authors highlight the challenging nature of the acquired images, difficult to use for loop closure detection due to the strong perceptual aliasing. The authors of [Geromichalos et al., 2020] propose a SLAM system based on a particle filter that includes global localization constraints. 2.5D local maps are built from stereo data and matched to the current map using a variant of ICP, whose scores are used to resample particles. Global localization is performed by performing template matching between local and global elevation images. The method is interesting and it shares with our system the idea of using local elevation and registration techniques from the domain of image matching to perform localization. However, the usage of plain elevation, compared to its gradient, is insufficient to provide enough details for matching, especially if the extent of local maps is limited and therefore less likely to contain salient structural features.

Furthermore, the transformation of point clouds to 2.5D maps loses the 3D information that is contained in the original point clouds, which can be used for the purpose of mapping in case of overhanging structures [Schuster et al., 2019] or to refine matching by means of ICP-like techniques.

### 3. The SLAM Architecture

GPGM-SLAM is built upon a distributed visual SLAM system based on submaps, designed to enable localization and mapping within teams of heterogeneous robots equipped with stereo vision systems [Schuster et al., 2019], such as the Lightweight Rover Unit (LRU) rovers [Schuster et al., 2017] and the UAV ARDEA [Lutz et al., 2020] developed at the DLR (German Aerospace Center) Institute of Robotics and Mechatronics. On both systems, stereo images are processed with the Semi-Global Matching (SGM) algorithm [Hirschmüller, 2007] implemented on an FPGA to obtain per-pixel depth information, which is used to evaluate the terrain traversability but also to compute visual odometry.

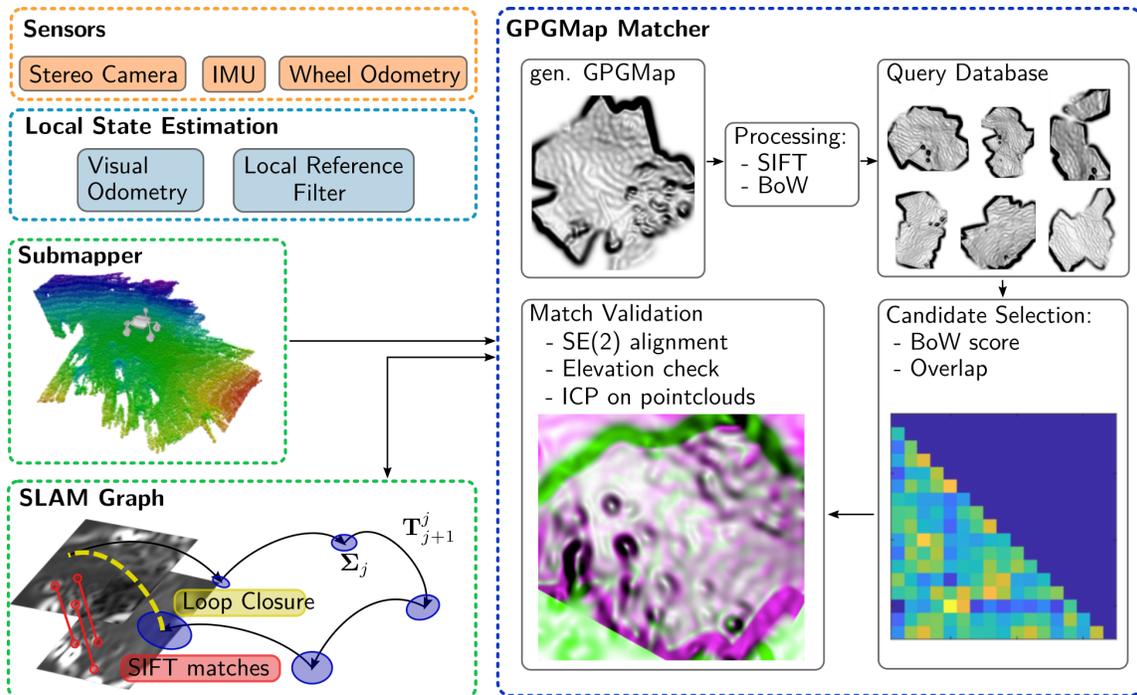
Local state estimation is performed as a fusion of visual odometry, IMU measurements and, if available, wheel odometry. The fusion is based on a loosely coupled approach and estimates 6D poses as well as sensor biases using an Extended Kalman Filter (EKF), implemented as a Local Reference Filter [Schmid et al., 2014]. It allows long-term operation by enabling a consistent estimation of all the unobservable state variables within the system, i.e., translations in  $x$ ,  $y$ ,  $z$  and yaw angle, with locally bounded uncertainties. Switching to a new local reference frame is actively triggered by a set of conditions, which include the length of the local trajectory, the accumulated yaw rotation of the robot body, and exceeding a threshold on the state covariances.

Each local reference frame defines also the origin of a submap, which is a local representation of the environment with a bounded size and error due to the aforementioned frame switching conditions. It represents the geometry of the environment in form of a 3D point cloud as well as an additional probabilistic 3D voxel grid. Submaps are aggregated from stereo depth measurements using the pose estimated from the local reference filter. In the original SLAM system from [Schuster et al., 2019], submaps distinguish obstacles and traversable parts, and obstacles are not only utilized for local navigation but also to define 3D keypoints to extract C-SHOT features using the normals of the full point cloud. With that, the annotated submap point clouds are utilized for the goal of establishing loop closures by matching 3D features and validating a transformation from Hough3D clusters followed by a 4D ICP step.

Submap matches produce rigid transformation constraints between the associated local reference filters, which define the nodes of a graph optimized by iSAM2 [Kaess et al., 2012] using the GTSAM library. Subsequent nodes are constrained by interpose constraints between the local reference frames defined by the local state estimation. Loop closures constrain nonadjacent nodes, and are followed by an update of the iterative iSAM2 optimizer. The optimized poses are finally used to correct the appearance of the full map, which is built by assembling the rigid submaps' point clouds and voxel grids according to the positions of their local reference frames.

In the course of this paper, we will refer to this configuration of the SLAM system as a baseline for comparison. Furthermore we will also refer to a variation of the system [Giubilato et al., 2021], which refers to an improved formulation of local submaps with the inclusion of visual keyframes captured from the camera images along the local trajectory. In this variation, keyframes are used to compute ORB features on locations of the image with valid depth values. ORB features are finally matched to compute interpose constraints between keyframes, which are constrained to the local reference frames from visual-inertial constraints.

The system denoted GPGM-SLAM, see Figure 3, substitutes the point cloud-based submaps with GPGMaps to increase the performances in the loop closure detection stage. GPGMaps are built by leveraging Gaussian Processes to infer a continuous elevation given the submap point cloud as a set of training points. This allows to overcome the traditional issues with the point cloud data structure, e.g., noise and occlusions, to the end of matching on challenging unstructured scenes. Compared to the baseline system, GPGMaps are not only matched given the overlap from the prior

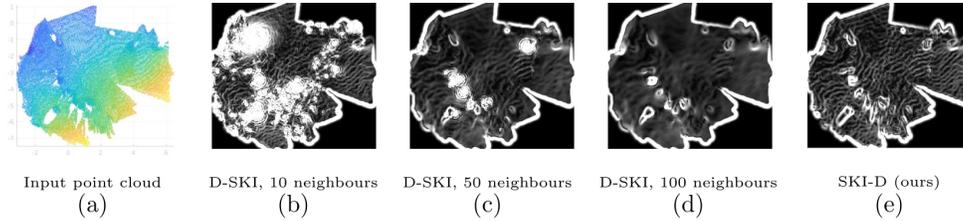


**Figure 3.** Overview of the GPGM-SLAM pipeline: (a) GPGMaps are generated using 3D submaps from aggregated stereo clouds. Visual features, computed on the gradient images, are used to create BoW vectors and detect loop closures from a database. (b) Validated GPGMap matches are used to establish loop closure constraints in a pose graph, where submap origins (blue ellipses) are joined by VIO (Visual-Inertial Odometry) pose constraints.

on poses, but also using a Bag of Words approach with SIFT features computed on the gradient of the elevation. Each GPGMap also contains the “parent” submap to refine the final transformation with a 4D ICP step. It is important to mention that the pipeline we propose is independent of the odometry source and modality, which benefits the generality of the approach. The only requirement is the presence of a depth sensor (e.g., a stereo camera), as depth images are used to construct submaps and, subsequently, GPGMaps. We will present details on construction of GPGMaps in the following section.

#### 4. Gaussian Process Gradient Maps

Gaussian Process (GP) regression [Rasmussen and Williams, 2006] is a probabilistic, nonparametric regression method of growing popularity in the fields of robotics perception and state estimation. Its data-driven nature is especially suited to represent outdoor unstructured environments as shown in [Vasudevan et al., 2009, Vasudevan, 2012, Sun, 2018]. In our previous work [Le Gentil et al., 2020], we introduced the concept of GPGMap to address the challenge of noisy and sparse geometric data in the context of submap-based place recognition in unstructured planetary environments. GPGMaps are a continuous and probabilistic representation of the gradient of the terrain elevation. In this representation, a direct prediction of the elevation’s derivatives is inferred from 3D point clouds of the system’s environment using GPs and linear operators [Särkkä, 2011]. While GP models allow for data-driven accurate interpolation, they suffer from a cubic  $\mathcal{O}(n^3)$  computational complexity for the first inference and linear  $\mathcal{O}(n)$  for each additional prediction, with  $n$  the number of sample data. In our previous work [Le Gentil et al., 2020], this constraint was simplistically addressed with a naive index-based downsampling of the submap point clouds. While improving computation time, this naive method reduces the data information used in the inference.



**Figure 4.** Inference of the gradient of the terrain elevation using D-SKI [Eriksson et al., 2018] and the proposed SKI-D method. (a) shows the input point cloud (colours correspond to different elevation values). (b), (c), and (d) are the output of D-SKI inferences using both as inputs the elevation information and *ad hoc* normals computed using 10, 50, and 100 closest points, respectively. (e) shows the output of the inference of the gradient directly from the point cloud data with the proposed SKI-D method.

In this work, to improve on the processing time and to allow the use of denser information, we employed a variant with derivatives of the Structured Kernel Interpolation (SKI) scheme [Wilson and Nickisch, 2015]. SKI is a GP-kernel approximation method based on interpolation/inducing points used to alleviate the cubic complexity of regular GP regression. Work on derivatives with SKI-based GP interpolation [Eriksson et al., 2018], so-called D-SKI, was later introduced to leverage the SKI scheme and the signal’s derivatives (as input) to improve the inference output. D-SKI allows for the inference of the signal’s derivatives but requires observations of the derivatives in the first place. Unfortunately these derivatives, which in our case are equivalent to the normal vectors of the submap point clouds, are not readily available as an observation. Instead, if required as input, the normal vectors have to be computed from the point cloud using numerical methods that require manual parameter tuning. As an example, to generate Figure 4, we estimate the submap normal vectors with a closest-neighbor search and a principal component analysis to provide D-SKI GPs with both elevation and derivative information. By looking at the gradient inferences with different parameters for the closest-neighbor search, it is clear that the output greatly depends on the *ad hoc* derivative estimation method and the original spatial point distribution. One can see that Figure 4(b) provides good details of the small ripples in the sand but performs poorly around bigger “features” in the environment, while (d) shows the larger features clearly but misses the details of the ripples in the sand. Using D-SKI over model-based generated derivatives (as opposed to actual observations of the derivatives) fails the data-driven nature of GP regression and can significantly degrade its performance. In this work, we propose a variation of the SKI scheme to directly infer the elevation derivatives using elevation measurements solely. We will further refer to this variation as SKI-D. Figure 4(e) demonstrates the advantage of the proposed method by providing clear gradient information both for the small ripples in the sand and the larger features.

This section first presents the required GP regression and SKI background before detailing the concepts involved in the generation of GPGMaps as shown in Figure 6.

#### 4.1. Gaussian Process regression and SKI

Let us consider a signal  $h$ , function of  $t$ , with a GP [Rasmussen and Williams, 2006] as

$$h(t) \sim \mathcal{GP}(0, k_h(t, t')). \quad (1)$$

The function  $k_h(t, t')$  is called the kernel function and represents the covariance between two instances of the signal  $h$ :

$$\text{cov}(h(t), h(t')) = k_h(t, t'). \quad (2)$$

Given noisy  $N$  observations of that signal

$$y_i = h(t_i) + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma_y^2), \quad (i = 1, \dots, N), \quad (3)$$

the mean and variance of  $h$  at any input  $t$ ,  $h^*(t)$  and  $\sigma_h^*(t)$  respectively, can be inferred as

$$h^*(t) = \mathbf{k}_h(t, \mathbf{t}) \left[ \mathbf{K}_h(\mathbf{t}, \mathbf{t}) + \sigma_y^2 \mathbf{I} \right]^{-1} \mathbf{y}, \quad (4)$$

$$\sigma_h^*(t) = k_h(t, t) - \mathbf{k}_h(t, \mathbf{t}) \left[ \mathbf{K}_h(\mathbf{t}, \mathbf{t}) + \sigma_y^2 \mathbf{I} \right]^{-1} \mathbf{k}_h(t, \mathbf{t})^\top, \quad (5)$$

with  $\mathbf{y}$  the column vector of the noisy observations  $[y_1, \dots, y_N]^\top$ ,  $\mathbf{k}_h(t, \mathbf{t})$  the row vector of covariances of  $h$  between the inference input and the observations' inputs  $[k_h(t, t_1), \dots, k_h(t, t_N)]$ , and  $\mathbf{K}_h(\mathbf{t}, \mathbf{t})$  the covariance matrix of the observations  $[\mathbf{k}_h(t_1, \mathbf{t})^\top, \dots, \mathbf{k}_h(t_N, \mathbf{t})^\top]$ . Note that this formulation assumes the same level of noise for all the observations.

As discussed above, the main drawback of GP regression is its computational complexity of  $\mathcal{O}(n^3)$  due to the matrix inversion in (4). The SKI principle [Wilson and Nickisch, 2015] relies on the use of inducing points  $U$  arranged as a regular grid. By formulating the approximation  $\mathbf{K}_{\mathbf{t}, U} \approx \mathbf{W}_{\mathbf{t}} \mathbf{K}_{U, U}$  with  $\mathbf{W}_{\mathbf{t}}$  a matrix of interpolation weights, it is possible to approximate  $\mathbf{K}_{\mathbf{t}, \mathbf{t}} = \mathbf{K}_h(\mathbf{t}, \mathbf{t}) + \sigma_y^2 \mathbf{I}$  as

$$\mathbf{K}_{\mathbf{t}, \mathbf{t}} \approx \mathbf{W}_{\mathbf{t}} \mathbf{K}_{U, U} \mathbf{W}_{\mathbf{t}}^\top = \mathbf{K}_{\text{SKI}}. \quad (6)$$

By choosing a local interpolation method, the matrix  $\mathbf{W}_{\mathbf{t}}$  can be extremely sparse and the inversion  $\mathbf{K}_{\text{SKI}}^{-1} \mathbf{y}$  is solved using a linear conjugate gradient only requiring matrix vector products. Similarly the covariance vector  $\mathbf{k}_h(t, \mathbf{t})$  is computed with a fixed amount of operations as  $k_h(t, t') \approx \sum_i w_i(t) k_h(u_i, t')$ . It results in a computationally efficient method to infer the signal's mean for any input  $t$ : linear time  $\mathcal{O}(n)$  inference with  $\mathcal{O}(1)$  for additional predictions.

## 4.2. SKI-D for Gaussian Process Gradient Maps generation

Unless in the context of specific exploration tasks (e.g., exploration of caves or lava tubes [Léveillé and Datta, 2010]), the environment surrounding the rover is unlikely to contain overhanging or vertical structures. Accordingly, we assume that the environment can be modeled as a 2.5D map without losing significant structural details. Given that submaps are gravity-aligned, their elevation, or the  $z$  component of the point cloud, can be expressed as a function  $f(\mathbf{x})$  with  $\mathbf{x}$  the  $x$  and  $y$  spatial coordinates within the submap's local reference frame. Considering the elevation function modelled with a zero-mean GP as

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k_f(\mathbf{x}_i, \mathbf{x}_{i'})),$$

$$z_i = f(\mathbf{x}_i) + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma_z^2), \quad (7)$$

the elevation information can be inferred at any  $x$ - $y$  input using (4) and (6). Leveraging the application of linear operators for GP regression [Särkkä, 2011], the derivatives of the elevation can exactly be inferred solely as a function of the noisy elevation observations:

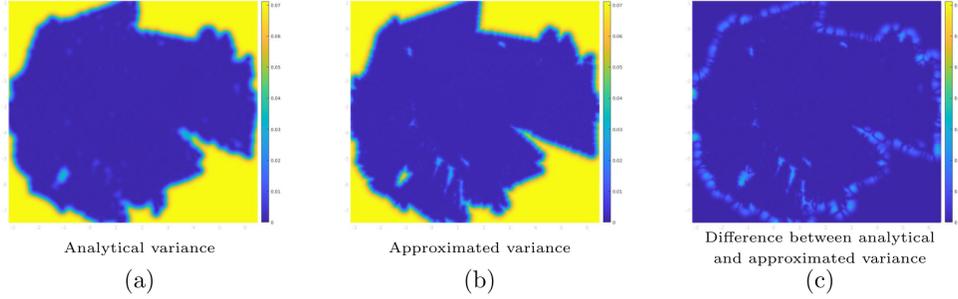
$$\frac{\partial f(\mathbf{x})}{\partial x} = \frac{\partial \mathbf{k}_f(\mathbf{x}, \mathbf{X})}{\partial x} \left[ \mathbf{K}_f(\mathbf{X}, \mathbf{X}) + \sigma_z^2 \mathbf{I} \right]^{-1} \mathbf{z},$$

$$\frac{\partial f(\mathbf{x})}{\partial y} = \frac{\partial \mathbf{k}_f(\mathbf{x}, \mathbf{X})}{\partial y} \left[ \mathbf{K}_f(\mathbf{X}, \mathbf{X}) + \sigma_z^2 \mathbf{I} \right]^{-1} \mathbf{z}. \quad (8)$$

As per [Eriksson et al., 2018], based on the SKI formulation, the differentiation operator can be applied to the kernel as  $\frac{\partial k_f(\mathbf{x}, \mathbf{X})}{\partial x} \approx \sum_i \frac{\partial w_i(\mathbf{x})}{\partial x} k_f(\mathbf{u}_i, \mathbf{X})$  and  $\frac{\partial k_f(\mathbf{x}, \mathbf{X})}{\partial y} \approx \sum_i \frac{\partial w_i(\mathbf{x})}{\partial y} k_f(\mathbf{u}_i, \mathbf{X})$ . Consequently, the elevation's derivatives are approximated as

$$\frac{\partial f(\mathbf{x})}{\partial x} \approx \left( \frac{\partial \mathbf{w}(\mathbf{x})}{\partial x} \mathbf{K}_{U, U} \right) \mathbf{K}_{\text{SKI}}^{-1} \mathbf{z}, \quad (9)$$

$$\frac{\partial f(\mathbf{x})}{\partial y} \approx \left( \frac{\partial \mathbf{w}(\mathbf{x})}{\partial y} \mathbf{K}_{U, U} \right) \mathbf{K}_{\text{SKI}}^{-1} \mathbf{z}. \quad (10)$$



**Figure 5.** Comparison of the analytical variance (a) computed with (5), and the proposed approximation (b). (c) is the difference between (b) and (a).

The gradient component of the GPGMaps is then obtained as the norm of the derivatives:

$$g(\mathbf{x}) = \sqrt{\left(\frac{\partial f(\mathbf{x})}{\partial x}\right)^2 + \left(\frac{\partial f(\mathbf{x})}{\partial y}\right)^2}. \quad (11)$$

The generation of the GPGMaps is done by sampling both the elevation and its gradient following a 2D Cartesian grid. In our experiments, the GPGMaps are inferred using the squared exponential kernel (also known as radial basis function) with a grid resolution of 1 cm.

### 4.3. Elevation’s variance approximation

Unfortunately, the SKI scheme as presented in [Wilson and Nickisch, 2015] and [Eriksson et al., 2018] does not allow for efficient inference of the signal’s variance. In order to provide a proxy for the variance of the elevation inferences, we leverage the kernel function between an inference input  $\mathbf{x}$  and the centroid of the noisy observations that are present in an arbitrary radius around the inference input. For efficiency, a k-d tree is used for the neighbor search in the  $x$ - $y$  plane. If no point is present in the radius search, the closest neighbor is used instead of the neighbors’ centroid. Formally, the approximation of the variance is defined as  $\tilde{var}(f(\mathbf{x})) = k_f(\mathbf{x}, \mathbf{x}) - \frac{(k_f(\mathbf{x}, \tilde{\mathbf{x}}))^2}{k_f(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) + \sigma_z^2}$ , with  $\tilde{\mathbf{x}}$  being the centroid of the neighbors (in the noisy observation’s input  $\mathbf{X}$ ) in a certain radius around  $\mathbf{x}$ , or simply the closest observation input if none is present in the radius search.

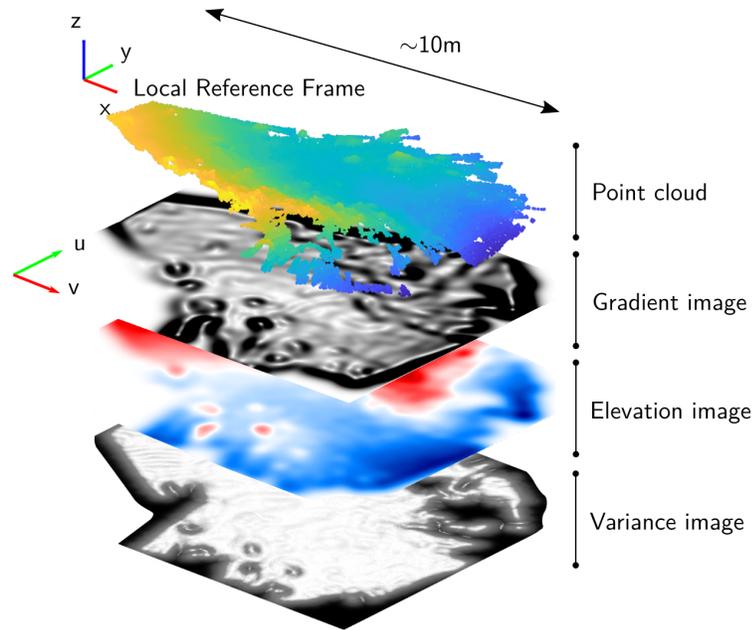
Figure 5 shows the analytical variance from (5) side-by-side with the proposed approximation. Overall our approximation tends to be conservative by slightly overestimating the variance. The difference is shown in Figure 5(c). Less than 6% of the inferred values differ from the analytical ones by more than 10% of the signal’s variance. These are present in “high” variance areas and therefore have very little impact on the rest of our SLAM framework as per the use of variance-based binary masks (as presented in the following section) to ignore high variance areas.

## 5. Loop Closure with GPGMaps

Establishing loop closures in our SLAM system is accomplished by attempting pairwise matching of GPGMaps. The procedure leverages the representation of elevation, variance, and gradient of elevation as images to use elements of standard image registration pipelines. More specifically, as new GPGMaps are created, SIFT features are extracted on the gradient image. To avoid extracting features on uncertain regions, where the elevation is inferred far from neighboring training points, the variance image is used to establish a binary mask. Let the  $i$ th GPGMap of a mapping session be denoted as

$$\mathcal{G}_i = \langle \mathbf{T}_{\text{LRF}}^w, P, I, V, G, K, F \rangle_i \quad (12)$$

where  $\mathbf{T}_{\text{LRF}}^w$  is the pose of Local Reference Frame for the  $i$ th GPGMap with respect to a world coordinate frame.  $P$  is the original point cloud,  $I$  and  $V$  are elevation and variance images,  $G$



**Figure 6.** Anatomy of a GPGMap. A point cloud, inherited from the parent submap, is used to train a Gaussian Process which infers elevation in a spatially continuous manner using SKI. The colormap of the elevation image highlights negative and positive elevation values, which are normalized on a zero mean. The variance of the elevation is instead approximated given the distance to the closest training point and the length scale of the squared exponential kernel, darker colors suggest higher variances which grow rapidly with the distance from the closest train point (the variance image is trimmed for the purpose of visualization, however the high variance regions extend to infinity in the spatial domain). The gradient image encodes the magnitude of the spatial gradients of the elevation image and is used to compute SIFT features for loop closure detection and matching of GPGMaps.

is the elevation gradient,  $K$  and  $F$  represent the sets of keypoints and feature descriptors. The task of establishing matches between GPGMaps is split in two main parts. First, a preliminary candidate selection is made relying either on appearance or prior pose information. Second, the candidate matches are validated by attempting to fit an  $SE(2)$  transformation between gradient images. The relative pose between the local reference frames of matched GPGMaps enforces loop closure constraints in the SLAM graph.

### 5.1. Loop Closure Detection

As new GPGMaps are created, the ensemble of SIFT descriptors computed on the gradient image is translated into a lightweight representation using the Bag of Words paradigm. More specifically, we use a modified version of the DBoW2 [Galvez-Lopez and Tardos, 2012] library introduced in [Giubilato et al., 2020]. By means of a vocabulary of visual words, built appropriately considering the potential features that can appear during execution, the Bag of Words model in DBoW2 summarizes feature occurrences into a map of indexes and corresponding weights, called BoW vector. The weighting scheme is based on the *tf-idf* (i.e., term frequency–inverse document frequency) metric, which weights the effectiveness of a word to suggest a visual context depending on the number of times the word was observed in the images used to build the vocabulary. That ensures that common features have little impact on the similarity metric between BoW vectors. Finally, an additional weight is applied to the feature occurrences to account for the incompleteness of the vocabulary. Vocabularies built on sparse and partial data might not fit well the operational scenario, and the feature descriptors parsed to the vocabulary might lay far from the closest word in the descriptor

space. This distance is accounted for, as explained in more detail in [Giubilato et al., 2020], to downweight the contribution of the word to the similarity of BoW vectors.

Let be  $\mathbf{v}_i$  the BoW vector for GPGMap  $\mathcal{G}_i$ :

$$\begin{aligned} \mathbf{v}_i &= \{\langle \text{id}_0, \bar{w}_0 \rangle, \dots, \langle \text{id}_j, \bar{w}_j \rangle\}, \\ \bar{w}_j &= w_j^* \cdot \text{idf} \cdot \text{tf}, \end{aligned} \quad (13)$$

where  $\text{id}_j$  is the index of the  $j$ th word and  $\bar{w}_j$  is the corresponding weight, which is computed as the product of all weighting factors including  $w_j^*$  which encodes the distance to the closest visual word and  $\text{idf} \cdot \text{tf}$ , which is the tf-idf term. All new GPGMaps are compared with a database of the GPGMaps previously built from the beginning of the session to identify potential matches. The similarity of GPGMaps is computed in a fast and approximate manner by comparing the BoW vectors through a similarity score  $s(\mathbf{v}_i, \mathbf{v}_{db})$ , computed as in [Galvez-Lopez and Tardos, 2012]:

$$s(\mathbf{v}_i, \mathbf{v}_{db}) = 1 - \frac{1}{2} \left| \frac{\mathbf{v}_i}{|\mathbf{v}_i|} - \frac{\mathbf{v}_{db}}{|\mathbf{v}_{db}|} \right| \in [0, 1]. \quad (14)$$

For every GPGMap, we select the two most similar ones and attempt match validation.

In addition, all GPGMaps are associated with a Local Reference Frame, the pose of which is added to the SLAM graph. Thus, it is possible at all times to use the prior pose information to select matching candidates based on spatial overlap. This is not only the strategy employed by our baseline SLAM system [Schuster et al., 2019] but also from the majority of structure-based SLAM systems [Ebadi et al., 2020, Mendes et al., 2016] usually relying on LiDAR point clouds. Specifically, we compute a bounding box for each GPGMap considering the extend of the original point cloud. The bounding box is then inflated considering the variance on the pose coordinates. For every new GPGMap, we compute the Intersection over Union (IoU) of its bounding box with the GPGMaps in the database. Furthermore, we monitor the status of the pose graph, maintained by the SLAM node, and in case significant changes are observed the IoU is recomputed for all pairs of GPGMaps. Nonmatched and sufficiently overlapping GPGMaps are then selected as candidates for match validation.

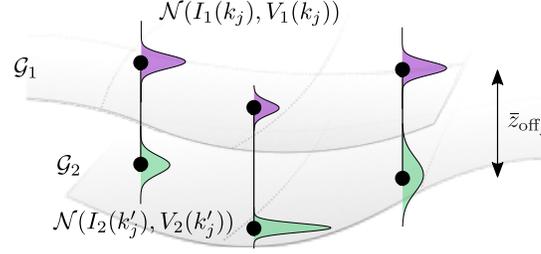
A match queue is built by adding first the candidate matches based on BoW similarity and secondly the candidates from overlap. This way, in case a match of the first type is validated, the pose graph is immediately updated correcting poses and covariances of all GPGMaps. This allows to more accurately compute the overlap metric and, as covariances decrease, to reduce the number of candidates to be validated, reducing the computational overhead.

## 5.2. Loop Closure Validation

In this step of the pipeline, candidate GPGMap matches are validated to select true matches and compute a final rigid transformation between the original point clouds that is used to constrain their local reference frames in the SLAM graph. The validation is articulated in two steps. A first stage concerns the gradient images and, more specifically, their SE(2) alignment through SIFT features matching. A second stage consists in deriving the final transformation between the original point clouds given a prior alignment from the first step.

### 5.2.1. SE(2) Alignment of Elevation Images

The SIFT features  $f_1$  and  $f_2$  of a candidate GPGMap pair  $(\mathcal{G}_1, \mathcal{G}_2)$  are matched to determine an SE(2) transformation between the gradient images  $G_1$  and  $G_2$ . Feature matching is performed either in a brute force manner or using the ratio test [Lowe, 2004]. The matching strategy is selected by the user at runtime. In our experiments we find that the ratio test allows to efficiently reject outliers. However, in the more challenging sequences, the stricter check results in losing good matches compared to the simpler brute force approach, performed in both directions. The choice between the two can be therefore made empirically depending on the appearance of the environment. The resulting set of matching features is used to fit an SE(2) transformation between the images in a RANSAC scheme as explained in [Le Gentil et al., 2020]. A first check is performed on the number



**Figure 7.** Estimation of the  $z$  offset from the aligned elevation  $I$  and variance  $V$  images of GPGMaps  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . The SE(2) transformation for the alignment is obtained from SIFT correspondences  $k_j$  and  $k'_j$  computed on the gradient images.

of inliers, which must be higher than a predefined threshold. This value is preliminarily set to 5 and is generally valid for all test datasets.

Compared to [Le Gentil et al., 2020], to reduce the computational overhead, we here accept the result of RANSAC based solely on the elevation of matched features instead of the whole elevation images. Let be  $I_1(k_j)$  and  $I_2(k'_j)$  be elevations corresponding to the locations of the matched SIFT keypoints in GPGMaps  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . Let also  $V_1(k_j)$  and  $V_2(k'_j)$  be the associated variances. Having computed the SE(2) transformation that aligns  $I_1$  to  $I_2$ , the goal here is to determine the remaining parameter to align the point clouds in the 3D space, which is the  $z$  offset. This offset is due to the fact that elevations are determined relatively to the local reference frame of  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , leading to potential differences between the local elevations of matching SIFT keypoints. Indeed, a possibility to directly determine the transformation between the reference frames of GPGMaps  $\mathcal{G}_1$  and  $\mathcal{G}_2$  would be to build sets of 3D points associated to matched SIFT keypoints and solve the problem as a 3D-3D alignment. However, as a variance is known over the elevation,  $z$  offsets can be more formally represented as Gaussian distributions (see Figure 7):

$$z_{\text{off},j} = \mathcal{N}(I_1(k_j) - I_2(k'_j), V_1(k_j) + V_2(k'_j)). \quad (15)$$

For all  $n$  correspondences, the offset that best aligns in  $z$  the two GPGMaps can be estimated as

$$\bar{z}_{\text{off}} = \frac{\sum_{j=1}^n \frac{I_1(k_j) - I_2(k'_j)}{V_1(k_j) + V_2(k'_j)}}{\sum_{j=1}^n \frac{1}{V_1(k_j) + V_2(k'_j)}}, \quad (16)$$

which is the average of all offsets weighted on their variance, or the results of weighted least squares regression for a constant function.

After applying  $\bar{z}_{\text{off}}$  to the elevation, we perform a final check to ensure that the pair of GPGMaps is a true match. This is done by examining the Bhattacharyya distances between the aligned elevations  $z_1$  and  $z_2$ :

$$\begin{aligned} z_1 &= \mathcal{N}(I_1(k_j) - \bar{z}_{\text{off}}, V_1(k_j)), \\ z_2 &= \mathcal{N}(I_2(k'_j) - \bar{z}_{\text{off}}, V_2(k'_j)), \end{aligned} \quad (17)$$

$$D_B(z_1, z_2) = \frac{1}{4} \ln \left( \frac{1}{4} \left( \frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} + 2 \right) \right) + \frac{1}{4} \left( \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \right).$$

If the GPGMap match is true, the  $z$  offset should determine low Bhattacharyya distances for all keypoint pairs. Therefore we accept the match if more than 70% of the pairs lead to  $D_B(z_1, z_2) < t_{D_B}$  where  $t_{D_B}$  is set to 2 from empirical considerations.

### 5.3. Point Cloud Alignment

Having validated a match between a pair of GPGMaps, a transformation between the original point clouds has to be derived in order to establish a loop closure constraint in the pose graph.

A transformation  $\mathbf{T}_{\mathcal{G}_2}^{\mathcal{G}_1} \in SE(2)$  is known from the last step as well as an offset  $\bar{z}_{\text{off}}$  that aligns the elevations  $I_1$  to  $I_2$ . A transformation in the 3D space from the Local Reference Frames of the original point clouds is obtained as

$$\mathbf{T}_{\text{LRF}_1}^{\text{LRF}_2} = \mathbf{T}_{\mathcal{G}_2}^{\text{LRF}_2} \mathbf{T}_{\mathcal{G}_1}^{\mathcal{G}_2} \mathbf{T}_{\text{LRF}_1}^{\mathcal{G}_1} \quad (18)$$

$$= \begin{bmatrix} \mathbf{I} & rx_o & & \\ & ry_o & & \\ & & 0 & \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}(\phi) & rt_x & & \\ & rt_y & & \\ & & \bar{z}_{\text{off}} & \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I} & rx'_o & & \\ & ry'_o & & \\ & & 0 & \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (19)$$

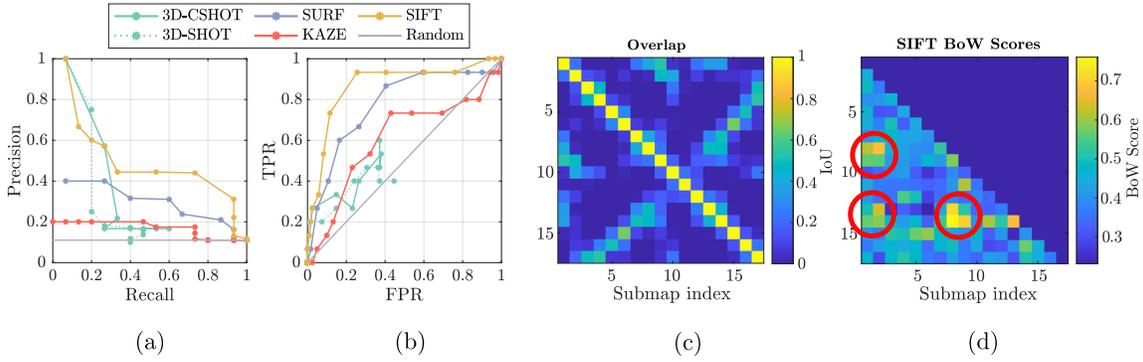
where  $\mathbf{R}(\phi)$  is the rotation component of  $\mathbf{T}_{\mathcal{G}_1}^{\mathcal{G}_2}$  with roll and pitch angles set to zero,  $r$  is the resolution of elevation and variance images (m/pix), and  $x_o$  and  $y_o$  are the origins of the elevation and variance images in the Local Reference Frame of the original submaps. To account for potential misalignments after validation, we perform a 4D ICP on the point clouds using  $\mathbf{T}_{\text{LRF}_1}^{\text{LRF}_2}$  as the initial transformation. Finally, the RMSE error of the point-to-point distances at the last iteration of ICP is used to approximate a covariance for the refined transformation, as suggested in [Brand et al., 2015], which is then used to establish a loop closure constraint in the pose graph.

## 6. Evaluation of Loop Closure Detection

We perform here a first evaluation of the performances that can be expected in recollecting similar GPGMaps from a database. As visual place recognition using the Bag of Words paradigm is very mature and has been implemented in many visual SLAM systems, we aim to evaluate the feasibility of utilizing the same techniques and apply them to the structure-oriented case. Furthermore, this preliminar assessment also aims at motivating the choice of the visual descriptor which we utilize in GPGM-SLAM given the superior classification performances.

Here we implement a Bag of Word framework in the MATLAB environment, in order to easily analyze various descriptor choices. Although the performances might differ from the DBow2 library that we use in GPGM-SLAM, we expect them to be comparable and, most of all, we expect the different descriptors to exhibit a similar behaviour between the two implementations. We evaluate the SIFT [Lowe, 2004], SURF [Bay et al., 2006] and KAZE [Alcantarilla et al., 2012] feature descriptors, extracted on the gradient images from the Etna sequences, see the dataset presentation in Section 7. We purposely leave out of this evaluation the ORB descriptor [Rublee et al., 2011] as it was not considered as a candidate while developing the pipeline. This is due to the fact that ORB relies on detection of corners, which are densely present in images of man-made and natural environments as well, but fundamentally absent from the smooth gradient images of GPGMaps. A vocabulary of features is constructed using the gradient images from a subset of GPGMaps built while evaluating GPGM-SLAM on the sequences. To every GPGMap, a single BoW vector is associated given the set of descriptors extracted on the relative gradient image. BoW vectors are evaluated pairwise computing an L1 distance as similarity metric as suggested in [Galvez-Lopez and Tardos, 2012]. The smaller the distance, the more SIFT descriptors are shared between the original gradient images and, therefore, the more likely are the GPGMaps to represent the same environment. To evaluate the precision of detecting true GPGMap matches, we collect GPGMaps built within the etna\_03 sequence, which is a sequence that well represents the volcanic environment on Mt. Etna, containing both stones and sandy slopes. We determine an approximation of ground truth GPGMap matches utilizing the poses computed during SLAM, and computing the IoU (Intersection-over-Union) of bounding boxes of GPGMaps to compute a value of overlap. We empirically set a value of 0.3 as a threshold of IoU, as done also in our previous work [Le Gentil et al., 2020]. For this purpose, the poses compute after SLAM are accurate enough to discriminate between overlapping and non overlapping GPGMaps.

Figure 8 reports precision-recall and ROC (Receiver Operating Characteristic Curve) curves built by varying a threshold between the minimum and maximum BoW scores to classify GPGMap pairs



**Figure 8.** Comparison of the classification performances of matching GPGMap pairs using different visual feature descriptors in a BoW scheme with traditional 3D feature matching in a RANSAC scheme. The results refer to the *etna\_03* sequence. A precision-recall plot (a) as well as a ROC (Receiver Operating Characteristic Curve) (b) illustrate the different performances of the evaluated classifiers compared to the performances of a random selection. (c) and (d) are heatmaps of the spatial overlap between GPGMaps compared with the pairwise scores of BoW vectors built using SIFT descriptors. Red circles highlight examples of true loop closures corresponding to high BoW scores and overlap.

as matching. The precision-recall metrics are defined as such:

$$P = \frac{T_p}{T_p + F_p}, \quad R = \frac{T_p}{T_p + F_n}, \quad (20)$$

where  $T_p$ ,  $F_p$ , and  $F_n$  are true positives, false positives and false negatives. Positive and negative relates to the decision of the classifier while being true and false denotes the real classification, in our case depending on the definition of ground truth matches. In this sense, the higher the precision, the higher is the fraction of correct classifications and the higher the recall, the more numerous they are. The ROC curve shows instead the relationship between TPR (True Positive Rate) and FPR (False Positive Rate), which denote the probability of correctly classifying positives and negatives. They are respectively defined as

$$\text{TPR} = \frac{T_p}{T_p + F_n}, \quad \text{FPR} = \frac{T_n}{T_n + F_p}. \quad (21)$$

Among all the evaluated visual descriptors, SIFT is the one which obtains the highest classification performances. Note that these performances are related to nonvalidated candidate GPGMap matches. We expect, in fact, that the precision after validation will be 100% in order to avoid validating any false GPGMap match. Overall, the SIFT descriptor allows here to select a wider set of true GPGMap matches and therefore facilitating the task of detecting numerous loop closures along the trajectory. Figure 8 also compares the overlap of GPGMaps, or IoU values, and the BoW scores between all GPGMap pairs, highlighting with red circles three examples of correct loop closure detections.

We also compare the performances of a traditional approach for structure registration, using the original point clouds that are used to generate the GPGMap. 3D-SHOT and 3D-CSHOT refer to matching keypoints from high curvature regions [Giubilato et al., 2020] using the SHOT and CSHOT descriptors [Salti et al., 2014] followed by RANSAC to determine a relative 6D rigid transformation given pairs of matching 3D keypoints. The precision-recall curves for these configurations are determined by varying the distance threshold between descriptors to get keypoint matches between the two point clouds. As the precision-recall curves show, the structure-based approach is able to correctly select a small set of matching pairs and the performances degrade with an immediate loss of precision for low recall values. Note that this alternative is order of magnitudes more computationally expensive than evaluating the similarity between BoW vectors and therefore completely infeasible for larger GPGMap sets. Furthermore, the approach depends on the possibility of extracting 3D

keypoints which, as will be later discussed, is the main limitation of structure-based approaches in our target scenarios.

The results presented here refer only to a single sequence out of the ones used for evaluating the full pipeline, which are introduced later in Section 7. Nonetheless, the sequence is representative of an environment where a balanced amount of featureless sandy slopes and aggregates of rocks permit the observer to collect sufficiently unambiguous SIFT features to discriminate between matching and nonmatching GPGMaps. Instead, other sequences within our datasets are either too strongly constrained by traversability (i.e., within these sequences, the rovers were always keeping a safe distance to rocks or smaller structures due to being collected while demonstrating autonomy) or just too imbalanced, with a small amount of potential GPGMap matches.

## 7. Performance Evaluation of SLAM System in Planetary-like Environments

We evaluate the localization performances of GPGM-SLAM on two datasets recorded in planetary-like environments on Mt. Etna, Sicily and in the Morocco desert. The first dataset [Vayugundla et al., 2018] was recorded in 2017 during the final demonstration mission of the Helmholtz project ROBEX, where the volcanic environment of Mt. Etna appears as a lunar landscape. The lack of visual and structural features, along with the repetitiveness of the visual appearance, makes place recognition a hard task, challenging the ability of robots to recognize previously visited places and correct localization drift. Similarly, the second dataset, named MADMAX and collected on the Moroccan desert [Meyer et al., 2021] aims at evaluating localization algorithms in challenging outdoor environments. The appearance of this dataset is similar to the one of Martian landscapes, and covers a wide variety of scenarios. A summary of the sequences used to test GPGM-SLAM is given in Tables 1 and 2.

GPGM-SLAM is evaluated in a comparative performance analysis with our baseline SLAM system, in two configurations, as well as the plain output of the visual-inertial odometry. A global trajectory is assembled by transforming in world coordinates the local trajectories that belong to each submap, or GPGMap. Let be  $\mathbf{p}^{G_i} = \{p_j, j = 1, \dots, n\}^{G_i}$  trajectory of the camera belonging to GPGMap  $i$ . The full trajectory at the end of a SLAM session can be reconstructed through the optimized poses from the pose graph  $\mathbf{T}_{G_i^w}$ . The reconstructed trajectory is aligned to a D-GPS ground truth leveraging temporal correspondences. The first corresponding positions between trajectory and ground truth are fixed, and Horn’s quaternion-based method is used to find the rotation matrix that ensures the best fit. We then evaluate the Root Mean Square (RMS) error between corresponding positions to define a final performance metric:

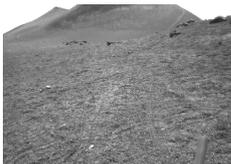
$$\text{RMSE} = \sqrt{\frac{1}{n_c} \sum_{j=0}^{n_c} \|p_j - p_j^{\text{D-GPS}}\|^2}. \quad (22)$$

Furthermore, we compute the final error, or the absolute trajectory error evaluated at the final correspondence with the D-GPS track, as well as the relative errors which are normalized with respect to the trajectory length. For all sequences and for each compared configuration of the SLAM system, we perform 5 runs to account for nondeterministic effects due to RANSAC and scheduling of processes in the operative system. The algorithms are tested on a laptop equipped with an Intel i7-4810MQ CPU and 16 Gb of RAM, whose computational performances are on par with the PC mounted on the LRU rover.

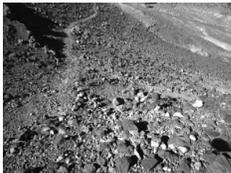
### 7.1. Mt. Etna Datasets

Sequences from the Mt. Etna datasets are recorded from the perspective of the LRU rover while performing a variety of tasks involving different levels of autonomy and environments. More specifically, the `etna_02` sequence is recorded while imparting direct commands to the rover through a joystick with the aim of traversing a long path with two overlapping loops to evaluate long-term localization [Grixa et al., 2018]. As the rover is fully operated, the pan-tilt mechanism of the camera

**Table 1.** Presentation of the Etna sequences

Sequence	Time [min]	Length [m]	Example Image	Description
etna_01	8	30		<ul style="list-style-type: none"> <li>- Short rectilinear travel forward and back to the start. Rover is teleoperated</li> <li>- Sandy, featureless flat ground</li> </ul>
etna_02	65	820		<ul style="list-style-type: none"> <li>- Long distance travel. Rover is teleoperated, pan-tilt mechanism not used, camera pointing to the same direction</li> <li>- Sandy and hilly terrain, does not contain rocks or obstacles</li> </ul>
etna_03	25	125		<ul style="list-style-type: none"> <li>- Rover performs waypoint-based autonomous navigation, using often the pan-tilt mechanism to assess traversability.</li> <li>- Hilly terrain containing plenty of visual and structural features</li> </ul>
etna_04	39	112		<ul style="list-style-type: none"> <li>- Autonomous exploration sequence, no user intervention. The rover moves frequently using the pan-tilt mechanism, therefore jerky camera motions pose challenges to the state estimation</li> <li>- Traversable terrain, few rocks scattered and observed at a safe distance</li> </ul>
etna_05	40	170		<ul style="list-style-type: none"> <li>- As etna_04</li> </ul>

**Table 2.** Presentation of the Morocco sequences

Sequence	Time [min]	Length [m]	Example Image	Description
E0	15	223		<ul style="list-style-type: none"> <li>- Traverse on a hiking path through a rocky slope. SUPER frequently stops and looks around.</li> <li>- Traversing same exact path on the way back from the opposite direction, start and end of the trajectory coincide.</li> <li>- Rough terrain with big stones.</li> </ul>
E1	16	309		<ul style="list-style-type: none"> <li>- As E0</li> </ul>
E2	25	374		<ul style="list-style-type: none"> <li>- Same environment as E0 and E1. Big traverse off the hiking path with little to no overlap. Return on the initial spot while traversing on previously observed places.</li> </ul>

head is not used, and cameras are generally pointed towards the ground. This, together with the locally straight trajectory, result in very narrow submaps which contain little structure (see impression in Table 1) and are very challenging to match. A similar type of environment is observed in etna\_01 although in this sequence, together with etna\_03, the rover is navigating autonomously across waypoints. In this case, the obstacle avoidance mechanism requires to frequently observe the environment in the proximity of the rover, which causes submaps to both have a wider extent and include more obstacles and peculiar structures. The etna\_03 sequence is the easiest sequence of the set for the purpose of establishing loop closures through submap and GPGMap matching. The rover follows here a trajectory shaped as a “figure 8” with two long overlapping sequences of about 15 meters length, the first on a rocky area and on the same driving direction, the second on a flat and sandy area and traversed from the opposite direction. In the etna\_04 and etna\_05 sequences, the rovers move in a fully autonomous manner as a tradeoff between exploring a scene, covering as much ground as possible, and revisiting places [Lehner et al., 2017]. These two sequences are the most challenging, in this evaluation, for the purpose of loop closure detection. Fully autonomous behaviors require the rovers to acquire as much information as possible about the environment, this results in frequent and jerky motions of the camera head. This limits the spatial extent of submaps, as new ones are more frequently triggered by rapid increase in the covariance of local pose estimation. In order to maximize traversability, the environment also contain very little amount of rocks and patterns on the sand to recognize. Furthermore, unlike the other datasets, the trajectories from etna\_04 and etna\_05 do not overlap after long distances (e.g., start to end) but contains only few and short term loops, which limits the effectiveness of establishing loop closures to the overall localization accuracy. However, the challenging nature of the appearance of the environment allows to benchmark the ability of detecting hard to find loop closures.

Results for the Etna sequences are summarized in Table 3, where GPGM-SLAM is compared with our baseline SLAM system as well as the visual-inertial odometry alone (i.e., no submap matching), denoted “filter.” Bold faces denote the best results for all the sequences, and it is immediately apparent how GPGM-SLAM outperforms all configurations, delivering better fitting trajectories to the D-GPS track and therefore lower maximum and final errors.

As all SLAM configuration share the same inputs, i.e., submaps, the sole reason for the higher performances of GPGM-SLAM is the ability to establish more, and more sparsely distributed, constraints in the pose graph from GPGMap matches. Results are particularly impressive for the

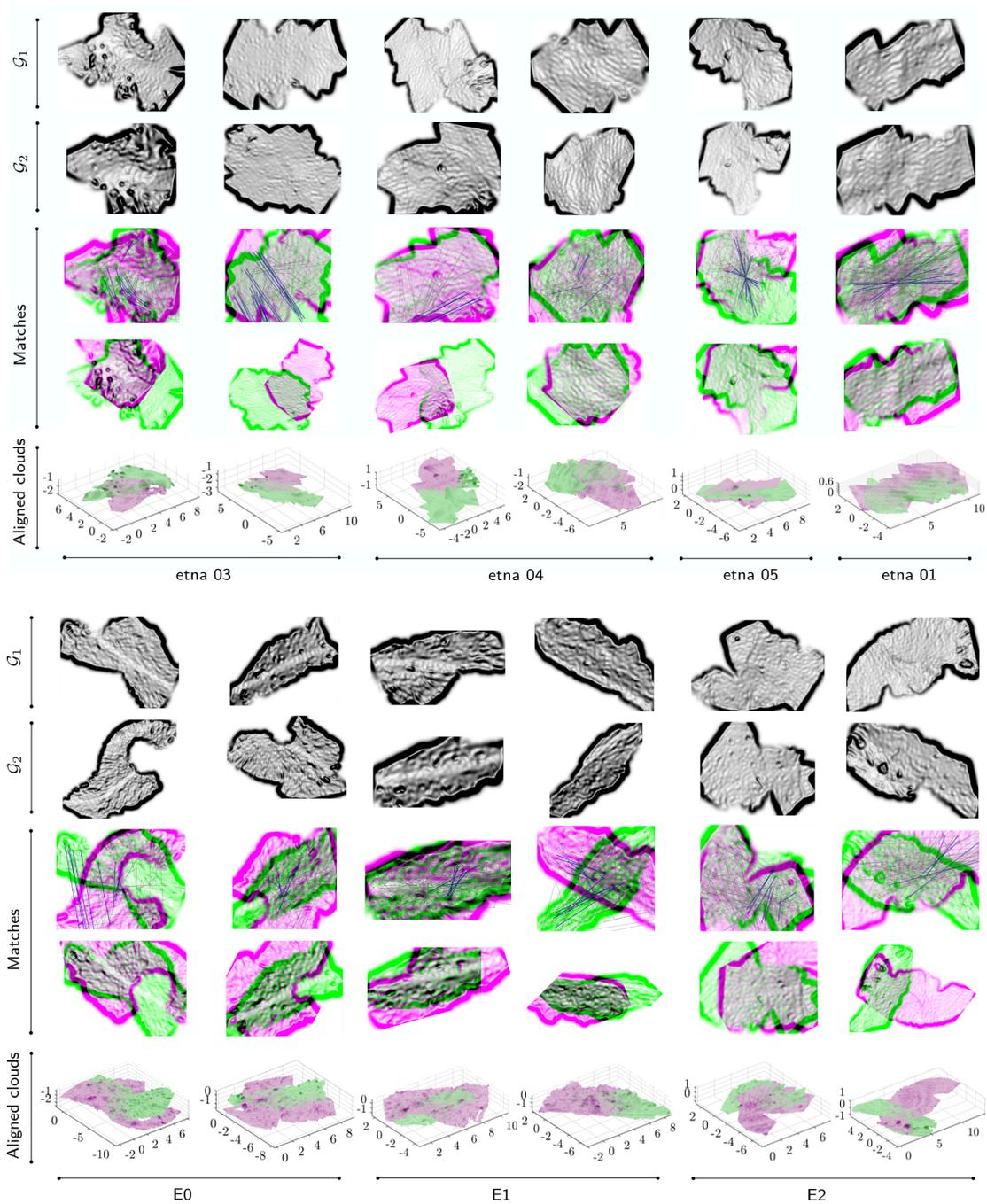
**Table 3.** Results for the Etna dataset averaged on 5 runs. Relative errors are normalized on the approximate length of the trajectory. Matches are average number of validated submap matches

Dataset	ETNA											
Sequence	etna_01				etna_02				etna_03			
Config	filter	3D	3D-KF	GPGM	filter	3D	3D-KF	GPGM	filter	3D	3D-KF	GPGM
RMSE (avg)	0.44	0.50	0.48	<b>0.41</b>	9.39	7.88	9.06	<b>2.84</b>	0.92	0.44	0.37	<b>0.34</b>
EndErr (avg)	0.98	1.23	1.18	<b>0.67</b>	32.26	26.44	30.37	<b>10.95</b>	1.05	1.34	1.00	<b>0.88</b>
RMSE (%)	0.75	0.87	0.82	<b>0.71</b>	1.14	0.96	1.10	<b>0.35</b>	0.75	0.36	0.30	<b>0.27</b>
MaxErr (%)	1.68	2.11	2.03	<b>1.15</b>	3.93	3.22	3.70	<b>1.33</b>	0.85	1.08	0.81	<b>0.70</b>
Matches	-	-	-	<b>0.83</b>	-	0.25	-	<b>19.4</b>	-	1.2	3	<b>8</b>

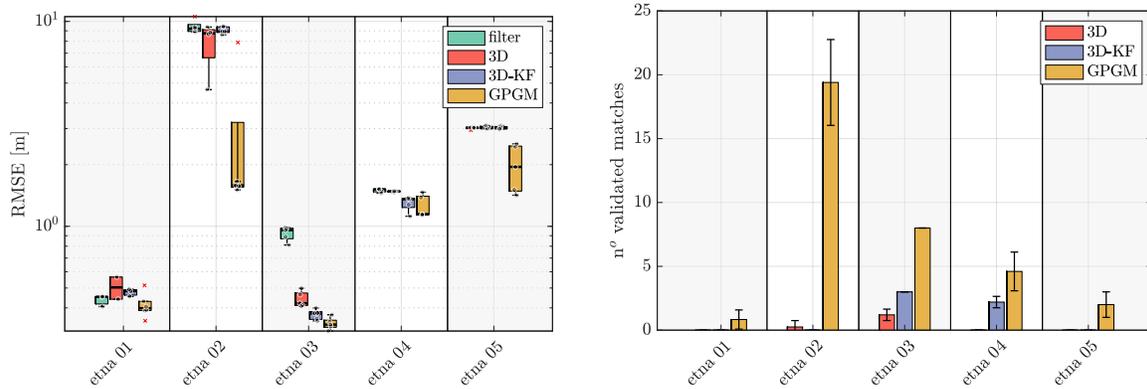
  

Dataset	ETNA							
Sequence	etna_04				etna_05			
Config	filter	3D	3D-KF	GPGM	filter	3D	3D-KF	GPGM
RMSE (avg)	1.50	1.48	1.30	<b>1.25</b>	3.01	3.04	3.03	<b>1.97</b>
EndErr (avg)	2.94	2.80	2.55	<b>2.27</b>	7.73	7.75	7.85	<b>4.98</b>
RMSE (%)	1.33	1.31	1.15	<b>1.11</b>	1.78	1.79	1.79	<b>1.16</b>
MaxErr (%)	2.60	2.48	2.25	<b>2.01</b>	4.56	4.57	4.63	<b>2.93</b>
Matches	-	-	2.2	<b>4.6</b>	-	-	-	<b>2</b>

etna\_02 and etna\_01 cases, where submaps do not contain rocks or evident structures in general. In fact, on etna\_01 no submaps are matched from CSHOT correspondences and on etna\_02 only one match happens and only in 1 run out of 5. GPGM-SLAM instead matches consistently around 20 GPGMaps on etna\_02, when the rover drives on the same path for the second time, as well as 1 to 2 GPGMaps on etna\_02 out of a total of 8 submaps in the sequence. Figure 9 displays a



**Figure 9.** Impressions of GPGMap matches for the Etna (top) and Morocco (bottom) datasets.  $\mathcal{G}_1$  and  $\mathcal{G}_2$  denote the gradient images of the GPGMap pair, "Matches" denotes validated SIFT feature matches as well as aligned gradients. The last row shows the original point clouds from submaps aligned in the 3D space.



**Figure 10.** RMSE errors and submap or GPGMap matches for the Etna sequences.

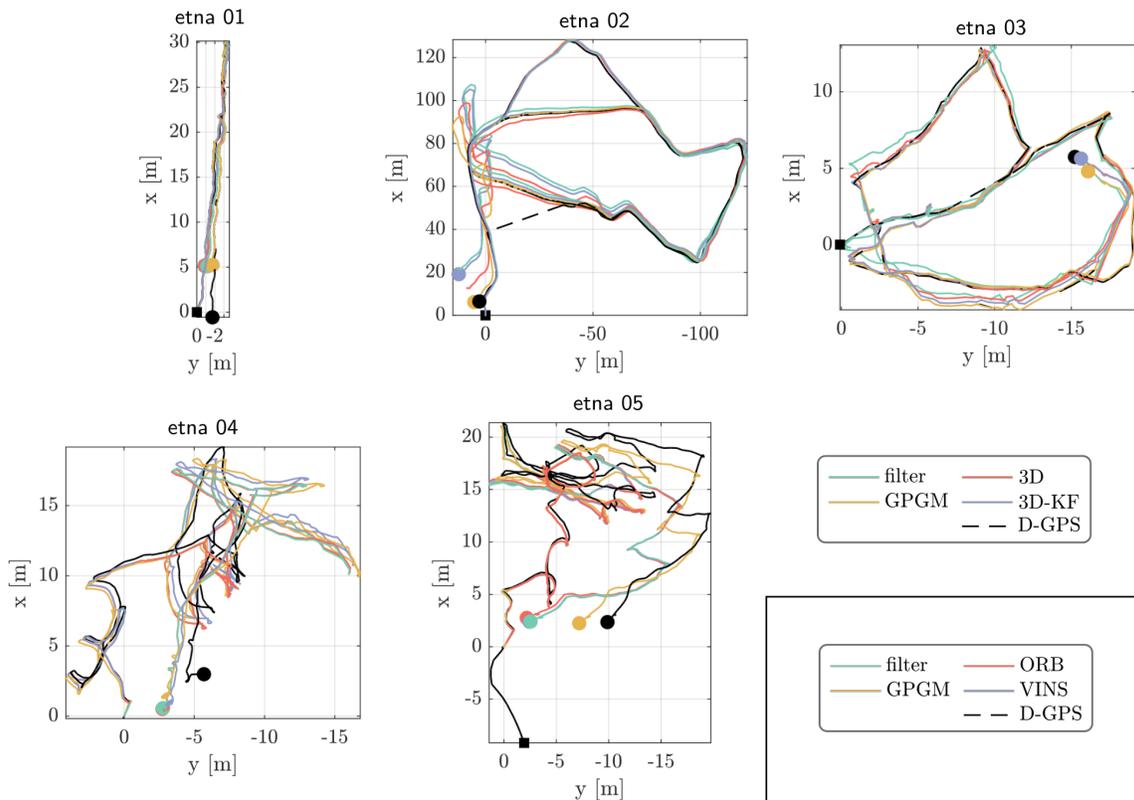
variety of GPGMap match examples including the alignment of gradient images and point clouds. Specifically for *etna\_01*, the lack of structure is immediately apparent also from the point cloud view, where rocks emerging from the sand are shaded in darker colors. Submap pairs from *etna\_02* have a similar appearance to the ones from *etna\_01* although being narrower. The *etna\_03* sequence is the easiest of the set, giving many opportunities to match either submaps or GPGMaps. The trend of RMSE errors from Figure 10, in fact, show decreasing errors for the 3D, 3D-KF, and GPGM configurations. The environment from *etna\_03* is rocky and the rover frequently repeats the same path also observing the scene from similar perspectives. This facilitates place recognition from multiple modalities, i.e., visual and structural perception, favoring the 3D-KF case over the 3D only configuration. Both of the approaches, however, are only able to match submaps from the parts of the sequence richer of structure, where also the rover revisits the place from the same driving direction. GPGM-SLAM instead, matches submaps also in the parts of the sequence where the rover revisits from the opposite driving direction, due to the inherent benefit of Gaussian Process regression to infer elevation where it is missing (e.g., obstructed from direct line of sight). Finally, for *etna\_04* and *etna\_05*, although obtaining lower errors than the baseline, performances of GPGM-SLAM in terms of absolute localization are not impressive due to the pose drift accumulated by jerky camera motions which are not compensated by the lack of long range loop closures. On *etna\_05*, the most challenging sequence on the dataset, the baseline SLAM system is unable to perform any submap matches, while GPGM-SLAM is able to match a few GPGMaps and partially compensate for localization drift. In Figure 11, all the best fitting trajectories are plotted against the D-GPS ground truth, reflecting the different behaviour of the rover as previously stated.

To contextualize the performance figures with respect to the state of the art of Visual SLAM systems, we attempted to evaluate ORB-SLAM2 [Mur-Artal and Tardos, 2017] and ORB-SLAM3 [Campos et al., 2020]. However, as cameras are mostly looking downward towards the sandy ground, and the visual appearance is highly ambiguous and aliased, both algorithms frequently lose tracking of ORB features. For ORB-SLAM2 this translate into failure, unless the camera revisits similar viewpoints. For ORB-SLAM3 instead, this translates into frequent creation of new maps, which the place recognition scheme is unable to match. As fragments of the reconstructed trajectories are too short to be evaluated (less than a couple of meters in general), we refer/postpone their evaluation to the Morocco dataset, where the camera motion from the hand-held unit facilitates visual tracking.

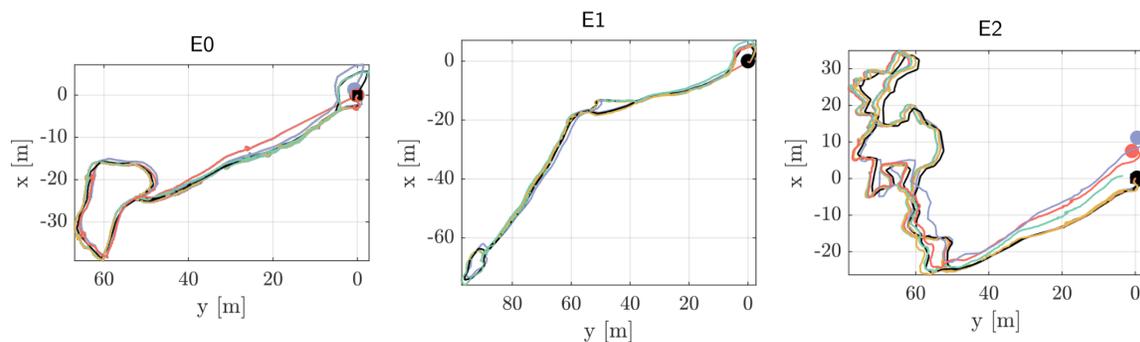
## 7.2. Morocco Datasets

The sequences denominated E0, E1, and E2 take place on and off an hiking trail located on a rocky slope in the “Kess Kess” formations, Erfoud area, Morocco. The sequences are named, for consistency, as within the released MADMAX dataset [Meyer et al., 2021]. Compared to the

## Etna sequences



## Morocco sequences

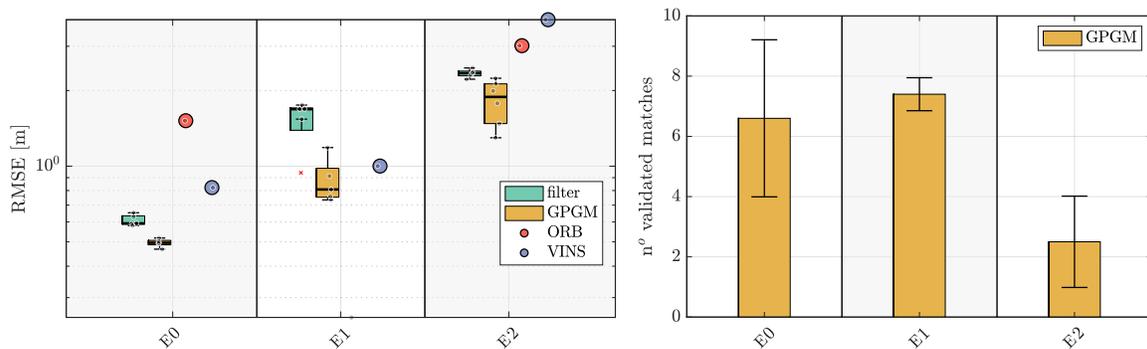


**Figure 11.** Trajectories for all sequences and algorithms on the Etna and Morocco datasets. Squares and circles highlight respectively the start and end of the trajectories.

sequences from the Etna dataset, the environment on these sequences is scattered with rocks, see the impressions from Table 2, whose uniformity again challenges the ability to recognize specific places. Furthermore, compared to the frequent panning motions of the LRU rover camera, in all the sequences from the Morocco dataset, the camera usually points forward along the direction of motion resulting in narrower GPGMaps and therefore comprising less structural details. As an additional challenge, while returning to the starting location, the path is traversed from the opposite direction. This compromises the ability of performing place recognition using visual cues and we specifically evaluate this in Section 7.5.

**Table 4.** Results for the Morocco dataset averaged on 5 runs, single runs for ORB-SLAM2 and VINS. Relative errors are normalized on the approximate length of the trajectory.

Dataset	MOROCCO											
Sequence	E0				E1				E2			
Config	filter	GPGM	ORB	VINS	filter	GPGM	ORB	VINS	filter	GPGM	ORB	VINS
RMSE (avg)	0.61	<b>0.50</b>	1.52	0.82	1.52	<b>0.88</b>	-	1.00	2.34	<b>1.82</b>	3.02	3.83
EndErr (avg)	1.14	0.33	<b>0.19</b>	1.29	3.42	0.48	-	<b>0.03</b>	6.45	<b>2.93</b>	7.44	11.36
RMSE (%)	0.25	<b>0.21</b>	0.68	0.37	0.46	<b>0.27</b>	-	0.32	0.54	<b>0.42</b>	0.75	1.01
MaxErr (%)	0.46	0.14	<b>0.09</b>	0.57	1.03	0.15	-	<b>0.01</b>	1.49	<b>0.68</b>	1.85	2.99
Matches	-	6.60	-	-	-	7.40	-	-	-	2.50	-	-

**Figure 12.** RMSE errors and submap or GPGMap matches for the Morocco sequences

We compare the performances of GPGM-SLAM against our visual-inertial filter as a baseline, as well as with ORB-SLAM2 and VINS-Mono in order to contextualize our results with well known algorithms from the SLAM community. For reasons of consistency, we employ the trajectories estimated from ORB-SLAM2 and VINS released as part of the MADMAX dataset. Note that, compared to the Etna dataset, the Morocco sequences are recorded with a hand-held unit and therefore do not include additional sources of odometry, such as wheel odometry, leading to a fairer comparison. A single trajectory from both algorithms is provided for every sequence, resulting from evaluating both with an optimal parameter set. Table 4 reports the errors of all tested algorithms and configurations, which are graphically represented in Figure 12. For the Morocco dataset, we avoided to report the performances of our baseline configuration 3D and 3D\_KF, already evaluated on the Etna sequences. The opposite viewpoints combined with the uniformity of the structure, lead to a significant degree of occlusions and complementarity between potentially matchable submaps. Therefore due to the lack of shared surface details and unique 3D keypoint detections, submap matching on the level of point clouds show poor performances and often producing establishing wrong loop closures. We consider therefore these approaches unable to produce satisfactory results on the Morocco sequences and we avoid to report performance metrics for them.

In all the Morocco sequences GPGM-SLAM produces the closest trajectory to the D-GPS ground truth, achieving RMSE errors as low as the 0.2% of the trajectory length. This is due to the many GPGMap matches established along the overlapping parts of the trajectories, even if traversed from the opposite direction. Examples of matched GPGMaps are collected in Figure 9. Interestingly, although both ORB-SLAM2 and VINS obtain higher RMSE errors, in E0 and E1 they score the lowest final errors due to the fact that they detect loop closures at the very end of the sequences, where the camera observes objects and equipment from similar perspectives. The inability of detecting visual loop closures within the trajectory, however, does not permit to recover the inevitable pose drift which degrades the fit to the D-GPS track. Results of ORB-SLAM2 are

**Table 5.** L1 distances between feature descriptors using the GP-based gradient or the standard SIFT over the elevation’s gradient (leveraging numerical differentiation over the elevation’s gradient).

Method	Avg inlier descriptor dist.	Avg outlier descriptor dist.	Nb loop detected
(i) SIFT variant using GP derivatives	1656.7	1704.0	2
(ii) Standard SIFT over elevation’s gradient	1692.1	2337.3	5

not reported for E1 as it was unable to reconstruct more than approximately 25% of the trajectory due to non recovered tracking loss.

### 7.3. SIFT features design choices

Using SIFT features on GPGMaps corresponds to the detection of cornerlike keypoints on the elevation’s gradient associated with descriptors computed based on the gradients of the elevation’s gradient. This set-up aims to justify empirically the choice of using SIFT over elevation’s gradient compared to other variations.

The standard procedure to match two images using SIFT features consists in computing differences of Gaussians at different scales on both images to extract salient keypoints. Then, descriptors based on orientation histograms of the intensity gradients are computed over scaled patches around each keypoints. Technically, it requires the algorithm to calculate the gradient of the images numerically. Finally, the descriptors are used to match features between the images using various distance functions and matching strategies. Regarding this pipeline and the availability of both the elevation and elevation’s gradients as imagelike data (using GP regression without or with differentiation linear operators), our first design choice concerns the data used for keypoint detection; should the keypoints be extract over the terrain elevation or its gradient?

Let us consider five true loop closures in the dataset etna\_03 with given SE(2) transformations between each GPGMap pairs. Keypoints are extracted in each submap using independently both the elevation and the elevation’s gradient. For each submap pair, the keypoints of one submap are projected in the second submap using the associated SE(2) transformation. Then, keypoint pairs across maps are classified as true matches or not according to a distance threshold of three centimeters. Using the elevation for keypoints extraction results in an average of 5.8 true matches per loop closure, while the gradient-based keypoints results in an average of 11.2 true matches. These results coincide with the intuition that the elevation’s gradient presents more cornerlike patterns than the elevation data itself. The rest of this experiments will only consider keypoints detected over the gradient image of the GPGMaps.

Given the availability of the GP-inferred derivatives of the elevation, (9) and (10), one can consider using these analytical derivatives instead of the numerical differentiation of the elevation’s gradient performed when using standard SIFT features over GPGMaps. For each keypoint, we computed the SIFT descriptors using independently both the GP-inferred elevation derivatives (i) and the numerical gradient of the elevation’s gradient as per the classic use of SIFT features (ii). Note that these approaches are fundamentally different as (i) uses  $\frac{\partial f(\mathbf{x})}{\partial x}$  and  $\frac{\partial f(\mathbf{x})}{\partial y}$ , while (ii) uses  $\frac{\partial g(\mathbf{x})}{\partial x}$  and  $\frac{\partial g(\mathbf{x})}{\partial y}$  with  $g(\mathbf{x})$  defined in (11).<sup>1</sup> We analysed the matching distances for inliers and outliers using both techniques. The results are shown in Table 5. The significantly bigger difference between inlier and outlier distances for the standard SIFT descriptors confirms that the gradient of the elevation’s gradient (ii) offers more distinctive features and lead to better loop-detection performances. Additionally, the extra computation needed for the numerical differentiation is not penalising the real-time abilities of our system as shown in the following section (the descriptors computation represents less than 3% of the total GPGMap generation and processing time).

<sup>1</sup>The derivatives of  $g(\mathbf{x})$  cannot be directly inferred using linear operators over GP kernels as per the nonlinear operations in (11).

**Table 6.** Times for principal GPGMap processing steps, averaged on all datasets.

Step	Time (avg) [s]
create gpgpmap	6.22
pose callback	0.01
compute descriptors	0.19
compute bow vector	0.01
compute scores	0.01
(avg. / Submap)	<b>6.43</b>
match validation: RANSAC	0.01
match validation: ICP	0.62

#### 7.4. Timings

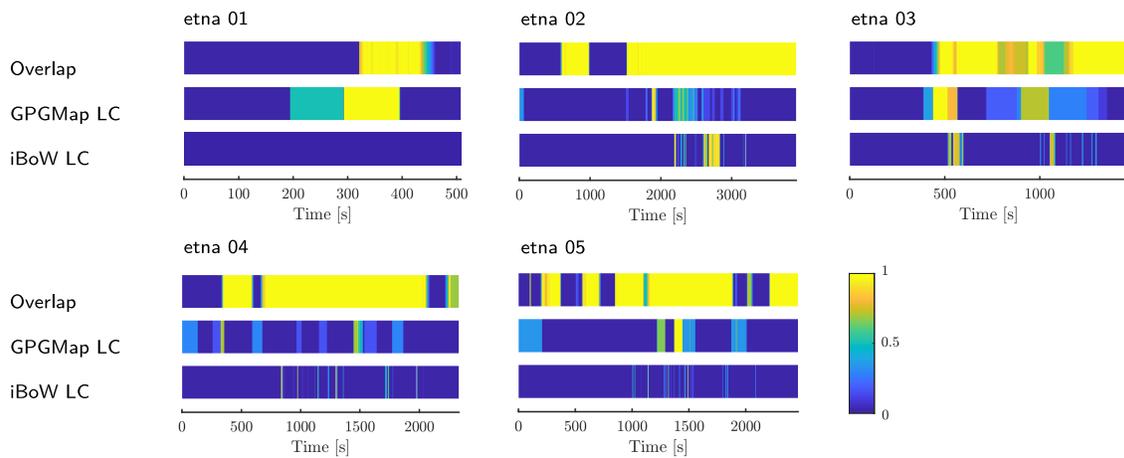
Table 6 reports the execution time for all the principal parts of GPGM-SLAM involved during the processing of GPGMaps. All times are referred to single thread implementations, except during GP inference of elevation where 2 threads are used, and are measured while evaluating the pipeline using a laptop equipped with an Intel i7-4810MQ with 16 GB of RAM. All algorithms are implemented in C++ with the only exception of SKI-D, which is executed by a Matlab process. Nevertheless, the times are comparable, if not equal, to those that can be expected from the computers onboard the LRU rovers. The most computationally expensive step is the creation of GPGMaps. The highest contributions are in order the Gaussian Process inference from SKI (with 1 centimeter resolution) and the computation of the approximate variance. The second highest time is spent on the extraction of SIFT features while all the remaining tasks, including generation of BoW vectors and retrieval from the database, are performed in less than 10 milliseconds. It must be noted that the extent of submaps greatly affects the inference time for the elevation: frequent panning motions of the camera head lead to accumulating points in the original submap in all directions, therefore growing the set of training points for the Gaussian process. For larger submaps, the time of generating GPGMaps can reach up to 5 or 6 seconds in the tested sequences. The computation time must, however, be compared with the time span of an individual submap which generally varies from 20 to 60 seconds while the camera is moving. Therefore the generation of GPGMap from submaps only takes a small fraction of the temporal extent of submaps, inducing a small and relatively constant delay between concluding the creation of one submap and performing loop closure detection with the respective GPGMap. In terms of match validation, which only occurs while evaluating candidate matches, fitting an SE(2) transformation between gradient images requires a negligible amount of time. The highest contribution to the validation time is the 4D ICP step on the original point clouds, which takes in average about 0.6 seconds.

#### 7.5. Comparison with visual relocalization

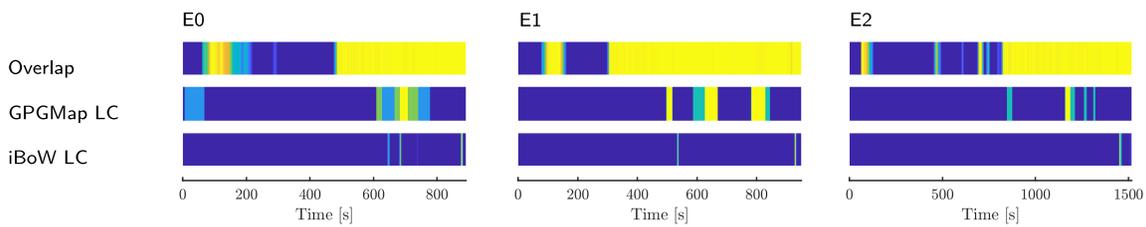
We compare the loop closure detection performances of GPGM-SLAM with the ones of a state of the art visual loop closure detector, iBoW-LCD [Garcia-Fidalgo and Ortiz, 2018]. This method is a recent incremental loop closure detector which eliminates the need of a vocabulary of visual word and exploits the concept of visual islands to group consequent images and reduce the computational time to detect a loop closure. The authors report higher retrieval performances than DBoW2, used by ORB-SLAM2, and therefore we use it to evaluate the different behavior of visual-only methods to detect loop closures on our challenging datasets.

We implement iBoW-LCD extracting a maximum of 1500 ORB features per image, using the left frame of the stereo setup. Loop closure validation is performed by fitting a fundamental matrix using RANSAC as implemented in the OpenCV library. We consider a visual loop closure successful if a minimum of 30 inliers is selected after RANSAC. All parameters are selected empirically such that the chance of establishing false loop closures is minimized.

## Etna sequences



## Morocco sequences

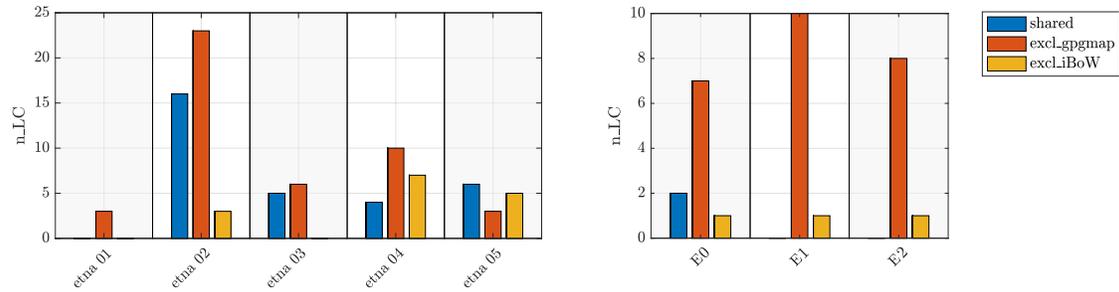


**Figure 13.** Frequency of loop closure occurrences for GPGM-SLAM and the visual loop closure detector iBoW-LCD. Heatmaps show the frequency over 5 runs for each algorithm, to account for randomness. The first field, denoted overlap, is proportional to the proximity of the camera for every time step to a past location, if closer than 5 meters. For GPGM-SLAM, a single loop closure occurrence covers the time interval of the relative GPGMap, while for iBoW-LCD a single loop closure occurrence is related to a keyframe match and slightly inflated for visualization purposes. All values are normalized between 0 (blue) and 1 (yellow).

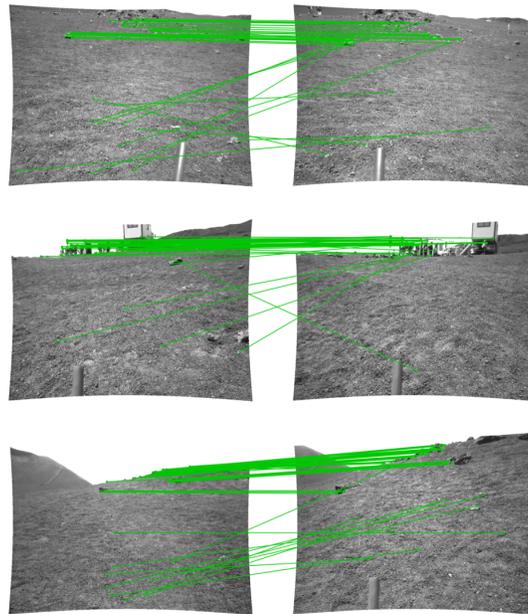
In Figure 13, we report the frequency of loop closures from both GPGM-SLAM and iBoW-LCD. For GPGM-SLAM, loop closures consists in validated GPGMap matches and their occurrence is registered in the time frame of the query GPGMap, or the one that is matched with one from the past. For iBoW-LCD an occurrence of visual loop closure is registered as a sparse event in a specific time. The time refers to the extent of the D-GPS track. We visually compare the frequencies of loop closure detections with an overlap value. The overlap is computed proportionally to the proximity of the current camera pose, taken from the D-GPS ground truth, with the closest one from the past trajectory. Figure 14 reports the number of shared and exclusive loop closures from GPGM-SLAM and iBoW-LCD. As loop closures are defined differently from the two algorithms, we consider an occurrence for iBoW-LCD the presence of any frame matches within the time span of an individual GPGMap. Occurrences of loop closures are either “shared” if they happen within the time frame of the same GPGMap, otherwise they are “exclusive” for one of the two methods. A higher number of exclusive matches denotes a higher loop closure detection power.

The aim of this evaluation is to highlight the different properties of both sensing modalities, i.e., structural properties for GPGM-SLAM and visual appearance for iBoW-LCD. Specifically, we want to evaluate the benefits of matching structure instead of images in unstructured scenarios where the observer path is unconstrained, but also highlight the limitations of performing loop closure detection with a single modality.

Although the environment on the Etna sequences is mostly sandy with a only a few clusters of rocks scattered around, the frequent panning of the camera leads to frequently observing the



**Figure 14.** Number of shared or exclusive occurrences of loop closures between GPGM-SLAM and iBoW-LCD. One occurrence of visual loop closure from iBoW-LCD is determined by the presence of any number of frame matches within the time interval of a GPGMap. Occurrences are either shared, if both algorithms procure a loop closure for the same GPGMap interval, otherwise exclusive for one of them.



**Figure 15.** Examples of loop closures detected by iBoW-LCD within the time span of GPGMaps that were not matched by GPGM-SLAM. Visual features driving the detection of a loop closure lie far from the stereo sensing range, which limits the spatial extent of GPGMaps. Note how wrong feature matches happened in the proximity of the cameras, on sandy terrains. These examples suggest the benefit of embedding multiple modalities to detect loop closures in the context of SLAM.

surrounding landscape at a distance. On etna\_01, the rover mostly observes a featureless ground without using the pan-tilt mechanism, and returns to the starting position from the opposite direction. In this case, only GPGM-SLAM is able to match GPGMaps and close loops, while iBoW-LCD does not detect any. Differently, on etna\_03 and etna\_02, iBoW-LCD detects many loop closures as the rover revisits frequently the same places from very close positions and viewpoints. However, GPGM-SLAM is able to match more numerous portions of the trajectory and because of that, matches are either shared between the two algorithms or detected exclusively by GPGM-SLAM, see Figure 14. The different modality of iBoW-LCD favors it in the etna\_04 and etna\_05 sequences, where many visual loop closure happen in places where GPGM-SLAM does not detect any similarity between GPGMaps, see the high number of exclusive loop closures for iBoW-LCD from Figure 14. iBoW-LCD, in fact, detect loop closures given image features which lie far beyond the range of stereo measurements, a few examples are given in Figure 15.

The sequences from the Morocco datasets are challenging for visual loop closure detection, as most of the overlapping parts of the trajectory are traversed from the opposite direction. While the elevation, inferred from the Gaussian Process, is continuous and its gradient offers repeatable features regardless of the camera viewpoint, the appearance of the images is completely different. This explains the higher performances of GPGM-SLAM in this case, or the high number of exclusive matches reported in Figure 14.

The presence of exclusive loop closure detections from iBoW-LCD in both the Etna and Morocco datasets suggests that multiple modalities could be exploited to improve the chances of relocalization. GPGMap matches in fact rely on the proximity of the environment to the camera, as the input point clouds are built from stereo depth, while visual loop closures can happen while observing landmarks far from the sensing range of a stereo setup.

## 8. Conclusions

In this paper, we presented GPGM-SLAM, a submap-based SLAM system for mobile robots based on Gaussian Process Gradient Maps targeted at stereo vision systems. The continuous elevation inferred by a Gaussian Process is used to compute gradient images which contain sufficient and repeatable visual information to make structure-oriented place recognition possible in challenging self-similar and ambiguous environments. Real time operations are permitted by Structured Kernel Interpolation (SKI) which reduce the computational load for inferring the local elevation by orders of magnitude compared to the standard GP formulation with  $O(n^3)$  time complexity. We tested GPGM-SLAM on sequences from two datasets captured on Mt. Etna, Sicily and in the Morocco desert, both planetary-like environments, demonstrating higher performances than our baseline SLAM system [Schuster et al., 2019] as well as state of the art visual SLAM systems. Through a specific analysis of the loop closure detection capabilities of our approach compared to visual approaches, we demonstrated that the performance improvements are principally due to the ability of matching GPGMaps produce while traversing on opposite directions. We in fact find out that the slow and unconstrained motion of mobile robots while operating in the field, often constrained by task specific requirements such as obstacle avoidance and traversability estimation, prevents the possibility to make any assumption on viewpoint repeatability which is, instead, paramount in common datasets oriented to autonomous driving. The inherent properties of Gaussian Processes, to compensate measurement noise and missing data in the training set (due to holes in the original point clouds), make robust matching possible and opens the possibility of performing robust multi-agent SLAM with different instrument setups in unstructured environments following unconstrained paths. Furthermore, we found that the robustness of our approach arises also from a limited dependency on the parameters, further reduced by employing Gaussian Processes, which are nonparametric. Indeed, the set of parameters and hyperparameters was initially set after empirical considerations and never tuned for the specific sequences. Finally, we highlighted the fact that, although generally performing worse in our specific datasets, visual place recognition is not limited by the availability of depth and therefore the approaches can be fused exploiting multimodality for more comprehensive relocalization capabilities [Giubilato et al., 2021]. As an example, visual features corresponding to points that lay beyond the stereo range could be utilized to enforce either heading constraints, or unscaled 2-view geometries by means of estimating the Essential matrix [Younes et al., 2017]. As part of additional future works, we plan to exploit semantic segmentation of known and unknown terrain classes to enhance the understanding and categorization of data within GPGMaps, allowing to improve matching performances in the face of extremely ambiguous structural cues.

## Acknowledgment

This work was supported by the UA-DAAD 2018 Australia-Germany Joint Research Cooperation Scheme, project COSMA (contract No. 57446007) and by the Helmholtz Association, project ARCHES (contract No. ZT-0033).

## ORCID

Riccardo Giubilato  <https://orcid.org/0000-0002-3161-3171>  
 Cedric Le Gentil  <https://orcid.org/0000-0002-9790-5935>  
 Mallikarjuna Vayugundla  <https://orcid.org/0000-0002-9277-0461>  
 Martin J. Schuster  <https://orcid.org/0000-0002-6983-3719>  
 Teresa Vidal-Calleja  <https://orcid.org/0000-0002-5763-9644>  
 Rudolph Triebel  <https://orcid.org/0000-0002-7975-036X>

## References

- Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). Kaze features. In *European conference on computer vision*, pages 214–227. Springer.
- Allouis, E., Jorden, T., Patel, N., and Ratcliffe, A. (2011). Sample fetching rover: lightweight rover concepts for mars sample return. *ASTRA*.
- Arandjelovic, R. and Zisserman, A. (2013). All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1578–1585.
- Bailey, T., Nieto, J., Guivant, J., Stevens, M., and Nebot, E. (2006). Consistency of the ekf-slam algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568. IEEE.
- Bajpai, A., Burroughes, G., Shaikat, A., and Gao, Y. (2016). Planetary monocular simultaneous localization and mapping. *Journal of Field Robotics*, 33(2):229–242.
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417.
- Brand, C., Schuster, M. J., Hirschmüller, H., and Suppa, M. (2015). Submap matching for stereo-vision based indoor/outdoor slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5670–5677.
- Bresson, G., Alsayed, Z., Yu, L., and Glaser, S. (2017). Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016a). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016b). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332.
- Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., and Tardós, J. D. (2020). Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam. *arXiv preprint arXiv:2007.11898*.
- Carle, P. J., Furgale, P. T., and Barfoot, T. D. (2010). Long-range rover localization by matching lidar scans to orbital elevation maps. *Journal of Field Robotics*, 27(3):344–370.
- Cesaretti, G., Dini, E., De Kestelier, X., Colla, V., and Pambaguian, L. (2014). Building components for an outpost on the lunar soil by means of a novel 3d printing technology. *Acta Astronautica*, 93:430–450.
- Chiodini, S., Pertile, M., Debei, S., Bramante, L., Ferrentino, E., Villa, A. G., Musso, I., and Barrera, M. (2017). Mars rovers localization by matching local horizon to surface digital elevation models. In *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 374–379. IEEE.
- Cop, K. P., Borges, P. V., and Dubé, R. (2018). Delight: An efficient descriptor for global localization using lidar intensities. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3653–3660. IEEE.
- Cummins, M. and Newman, P. (2008). Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665.
- De Cubber, G., Doroftei, D., Serrano, D., Chintamani, K., Sabino, R., and Ourevitch, S. (2013). The eu-icarus project: developing assistive robotic tools for search and rescue operations. In *2013 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pages 1–4. IEEE.
- Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., et al. (2019). The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7):1171–1191.
- Dubé, R., Cramariuc, A., Dugas, D., Nieto, J., Siegart, R., and Cadena, C. (2018). SegMap: 3D segment mapping using data-driven descriptors. In *Robotics: Science and Systems (RSS)*.

- Dubé, R., Cramariuc, A., Dugas, D., Sommer, H., Dymczyk, M., Nieto, J., Siegwart, R., and Cadena, C. (2020). SegMap: Segment-based mapping and localization using data-driven descriptors. *The International Journal of Robotics Research*, 39(2-3):339–355.
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110.
- Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E., Thakur, A., Funabiki, N., Morrell, B., Wood, S., et al. (2020). Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 80–86. IEEE.
- Engel, J., Schöps, T., and Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer.
- Eriksson, D., Dong, K., Lee, E., Bindel, D., and Wilson, A. G. (2018). Scaling gaussian process regression with derivatives. In *Advances in Neural Information Processing Systems*, pages 6867–6877.
- Farley, K. A., Williford, K. H., Stack, K. M., Bhartia, R., Chen, A., de la Torre, M., Hand, K., Goreva, Y., Herd, C. D., Hueso, R., et al. (2020). Mars 2020 mission overview. *Space Science Reviews*, 216(8):1–41.
- Galvez-Lopez, D. and Tardos, J. D. (2012). Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197.
- Garcia-Fidalgo, E. and Ortiz, A. (2018). ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words. *IEEE Robotics and Automation Letters*, 3(4):3051–3057.
- 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.
- Geromichalos, D., Azkarate, M., Tsardoulias, E., Gerdes, L., Petrou, L., and Perez Del Pulgar, C. (2020). Slam for autonomous planetary rovers with global localization. *Journal of Field Robotics*, 37(5):830–847.
- Giubilato, R., Vayugundla, M., Schuster, M. J., Stürzl, W., Wedler, A., Triebel, R., and Debei, S. (2020). Relocalization with submaps: Multi-session mapping for planetary rovers equipped with stereo cameras. *IEEE Robotics and Automation Letters*, 5(2):580–587.
- Giubilato, R., Vayugundla, M., Stürzl, W., Schuster, M. J., Wedler, A., and Triebel, R. (2021). Multi-modal loop closing in unstructured planetary environments with visually enriched submaps. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
- Grisetti, G., Kümmerle, R., Stachniss, C., and Burgard, W. (2010). A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43.
- Grisetti, G., Stachniss, C., and Burgard, W. (2005). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2432–2437. IEEE.
- Grixa, I., Schulz, P., Stürzl, W., and Triebel, R. (2018). Appearance-based along-route localization for planetary missions. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6327–6334. IEEE.
- Guo, J., Borges, P. V., Park, C., and Gawel, A. (2019). Local descriptor for robust place recognition using lidar intensity. *IEEE Robotics and Automation Letters*, 4(2):1470–1477.
- Heinicke, C. and Foing, B. (2021). Human habitats: prospects for infrastructure supporting astronomy from the moon. *Philosophical Transactions of the Royal Society A*, 379(2188):20190568.
- Heverly, M., Matthews, J., Lin, J., Fuller, D., Maimone, M., Biesiadecki, J., and Leichty, J. (2013). Traverse performance characterization for the mars science laboratory rover. *Journal of Field Robotics*, 30(6):835–846.
- Hidalgo-Carrió, J., Poulakis, P., and Kirchner, F. (2018). Adaptive localization and mapping with application to planetary rovers. *Journal of Field Robotics*, 35(6):961–987.
- Hirschmüller, H. (2007). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341.
- Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F. (2012). isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235.
- Kaess, M., Ranganathan, A., and Dellaert, F. (2008). isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378.
- Kim, G. and Kim, A. (2018). Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809.

- Klamt, T., Rodriguez, D., Schwarz, M., Lenz, C., Pavlichenko, D., Droschel, D., and Behnke, S. (2018). Supervised autonomous locomotion and manipulation for disaster response with a centaurlike robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE.
- Le Gentil, C., Vayugundla, M., Giubilato, R., Stürzl, W., Vidal-Calleja, T., and Triebel, R. (2020). Gaussian process gradient maps for loop-closure detection in unstructured planetary environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1895–1902.
- Le Gentil, C., Vidal-Calleja, T., and Huang, S. (2021). In2laama: Inertial lidar localization autocalibration and mapping. *IEEE Transactions on Robotics*, 37(1):275–290.
- Lehner, H., Schuster, M. J., Bodenmüller, T., and Kriegel, S. (2017). Exploration with active loop closing: A trade-off between exploration efficiency and map quality. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6191–6198. IEEE.
- Léveillé, R. J. and Datta, S. (2010). Lava tubes and basaltic caves as astrobiological targets on earth and mars: a review. *Planetary and Space Science*, 58(4):592–598.
- Li, P., Qin, T., Hu, B., Zhu, F., and Shen, S. (2017). Monocular visual-inertial state estimation for mobile augmented reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 11–21. IEEE.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- Lowry, S., Sunderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., and Milford, M. J. (2016). Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19.
- Lutz, P., Müller, M. G., Maier, M., Stoneman, S., Tomic, T., von Bargen, I., Schuster, M. J., Steidle, F., Wedler, A., Stürzl, W., and Triebel, R. (2020). Ardea—an mav with skills for future planetary missions. *Journal of Field Robotics (JFR)*.
- Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186.
- Marc, R., Roberts, H., Weclowski, P., and Aspinall, D. (2020). An improved rover navigation system for long range traverses: Surpassing 1km per sol in planetary exploration. In *71st International Astronautical Congress (IAC)*.
- Mattingly, R. and May, L. (2011). Mars sample return as a campaign. In *2011 Aerospace Conference*, pages 1–13. IEEE.
- Mendes, E., Koch, P., and Lacroix, S. (2016). Icp-based pose-graph slam. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 195–200. IEEE.
- Meyer, L., Smišek, M., Fontan Villacampa, A., Oliva Maza, L., Medina, D., Schuster, M. J., Steidle, F., Vayugundla, M., Müller, M. G., Rebele, B., Wedler, A., and Triebel, R. (2021). The madmax data set for visual-inertial rover navigation on mars. *Journal of Field Robotics*, n/a(n/a).
- Milford, M. J. and Wyeth, G. F. (2012). Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE international conference on robotics and automation*, pages 1643–1649. IEEE.
- Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al. (2003). Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI*, volume 3, pages 1151–1156.
- Mur-Artal, R. and Tardos, 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.
- Nefian, A. V., Bouyssounouse, X., Edwards, L., Kim, T., Hand, E., Rhizor, J., Deans, M., Bebis, G., and Fong, T. (2014). Planetary rover localization within orbital maps. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1628–1632. IEEE.
- Olson, C. F., Matthies, L. H., Schoppers, M., and Maimone, M. W. (2003). Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229.
- Otsu, K., Agha-Mohammadi, A.-A., and Paton, M. (2017). Where to look? predictive perception with applications to planetary exploration. *IEEE Robotics and Automation Letters*, 3(2):635–642.
- Otsu, K., Tepsuporn, S., Thakker, R., Vaquero, T. S., Edlund, J. A., Walsh, W., Miles, G., Heywood, T., Wolf, M. T., and Agha-Mohammadi, A. A. (2020). Supervised autonomy for communication-degraded subterranean exploration by a robot team. In *2020 IEEE Aerospace Conference*, pages 1–9.
- Pierzchała, M., Giguère, P., and Astrup, R. (2018). Mapping forests using an unmanned ground vehicle with 3d lidar and graph-slam. *Computers and Electronics in Agriculture*, 145:217–225.

- Qin, T., Li, P., and Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020.
- Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.
- Rekleitis, I., Bedwani, J.-L., and Dupuis, E. (2009). Autonomous planetary exploration using lidar data. In *2009 IEEE International Conference on Robotics and Automation*, pages 3025–3030. IEEE.
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, pages 2564–2571.
- Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE.
- Salti, S., Tombari, F., and Di Stefano, L. (2014). Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264.
- Särkkä, S. (2011). Linear operators and stochastic partial differential equations in Gaussian process regression. In *International Conference on Artificial Neural Networks (ICANN)*, pages 151–158.
- Schafer, H., Hach, A., Proetzsch, M., and Berns, K. (2008). 3d obstacle detection and avoidance in vegetated off-road terrain. In *2008 IEEE International Conference on Robotics and Automation*, pages 923–928. IEEE.
- Schmid, K., Ruess, F., and Burschka, D. (2014). Local reference filter for life-long vision aided inertial navigation. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE.
- Schuster, M. J., Brunner, S. G., Bussmann, K., Büttner, S., et al. (2017). Towards autonomous planetary exploration the lightweight rover unit (LRU), its success in the spacebotcamp challenge, and beyond. *Journal of Intelligent & Robotic Systems*.
- Schuster, M. J., Müller, M. G., Brunner, S. G., Lehner, H., Lehner, P., Sakagami, R., Dömel, A., Meyer, L., Voderbauer, B., Giubilato, R., Vayugundla, M., Reill, J., Steidle, F., von Barga, I., Bussmann, K., Belder, R., Lutz, P., Stürzl, W., Smíšek, M., Maier, M., Stoneman, S., Prince, A. F., Rebele, B., Durner, M., Staudinger, E., Zhang, S., Pöhlmann, R., Bischoff, E., Braun, C., Schröder, S., Dietz, E., Frohmann, S., Börner, A., Hübers, H., Foing, B., Triebel, R., Albu-Schäffer, A. O., and Wedler, A. (2020). The arches space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration. *IEEE Robotics and Automation Letters*, 5(4):5315–5322.
- Schuster, M. J., Schmid, K., Brand, C., and Beetz, M. (2019). Distributed stereo vision-based 6d localization and mapping for multi-robot teams. *Journal of Field Robotics*, 36(2):305–332.
- Shan, T. and Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE.
- Siam, S. M. and Zhang, H. (2017). Fast-seqslam: A fast appearance based place recognition algorithm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5702–5708. IEEE.
- Steder, B., Ruhnke, M., Grzonka, S., and Burgard, W. (2011). Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1249–1255.
- 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.
- Sun, L. (2018). *Large-scale continuous 2.5D robotic mapping*. PhD thesis, Open Publications of UTS Scholars.
- Thangavelautham, J., Chandra, A., and Jensen, E. (2020). Autonomous robot teams for lunar mining base construction and operation. In *2020 IEEE Aerospace Conference*, pages 1–16. IEEE.
- Thrun, S. and Montemerlo, M. (2006). The graph slam algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429.
- Vasudevan, S. (2012). Data fusion with gaussian processes. *Robotics and Autonomous Systems*, 60(12):1528–1544.
- Vasudevan, S., Ramos, F., Nettleton, E., Durrant-Whyte, H., and Blair, A. (2009). Gaussian process modeling of large scale terrain. In *2009 IEEE International Conference on Robotics and Automation*, pages 1047–1053.
- Vayugundla, M., Steidle, F., Smisek, M., Schuster, M. J., Bussmann, K., and Wedler, A. (2018). Datasets of long range navigation experiments in a moon analogue environment on mount etna. In *ISR 2018; 50th International Symposium on Robotics*, pages 1–7.

- Wedler, A., Schuster, M. J., Müller, M. G., Vodermayr, B., Meyer, L., Giubilato, R., Vayugundla, M., Smisek, M., Dömel, A., Steidle, F., Lehner, P., Schröder, S., Staudinger, E., Foing, B., and Reill, J. (2021). German aerospace center’s advanced robotic technology for future lunar scientific missions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2188):20190574.
- Wedler, A., Vayugundla, M., Lehner, H., Lehner, P., et al. (2017). First results of the ROBEX analogue mission campaign: Robotic deployment of seismic networks for future lunar missions. In *68th International Astronautical Congress (IAC)*.
- Wilcox, B. and Nguyen, T. (1998). Sojourner on mars and lessons learned for future planetary rovers. Technical report, SAE Technical Paper.
- Wilson, A. and Nickisch, H. (2015). Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP). *Proceedings - International Conference on Machine Learning*.
- Wu, B., WK, P. R., Ludvig, P., Chung, A. S., and Seabrook, T. (2019). Absolute localization through orbital maps and surface perspective imagery: A synthetic lunar dataset and neural network approach. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3262–3267. IEEE.
- Younes, G., Asmar, D., Shamma, E., and Zelek, J. (2017). Keyframe-based monocular slam: design, survey, and future directions. *Robotics and Autonomous Systems*, 98:67–88.
- Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2.

**How to cite this article:** Giubilato, R., Gentil, C. L., Vayugundla, M., Schuster, M. J., Vidal-Calleja, T., & Triebel, R. (2022). GPGM-SLAM: a robust SLAM system for unstructured planetary environments with Gaussian process gradient maps. *Field Robotics*, 2, 1721–1753.

**Publisher’s Note:** Field Robotics does not accept any legal responsibility for errors, omissions or claims and does not provide any warranty, express or implied, with respect to information published in this article.