Hybrid Visual SLAM for Multi-session Precise Localization: Application to a Coastal Cliff in Normandy

Diego Navarro [1,2], Ezio Malis [2], Raphael Antoine [1], and Philippe Martinet [2]
[1]ENDSUM team. Cerema Normandie-Centre, France
[2]ACENTAURI team. Centre Inria d'Université Côte d'Azur, France
e-mail: diego-navarro.tellez@inria.fr

***Abstract*—In the context of smart structure maintenance, the positioning of data measurements (e.d. radar, thermal camera, etc..) is crucial. Most of this data is meant to be collected in the proximity of the structure. In this paper, we address the challenge of accurately localizing a drone within GPS-deprived environments. This issue arises particularly near large structures, where the GPS signal can be significantly distorted or entirely absent. One of the most common solutions to the problem is to use a vision sensor and a Simultaneous Localization and Mapping (SLAM) system to reconstruct the environment and localize the drone. However, existing SLAM approaches may not be robust and precise enough, especially when the cameras lose perspective due to the proximity of the structure. In this paper, we propose a novel framework that computes a dense map of the environment to exploit it using direct odometry which excels in precise localization. The main contribution of this paper is the use of dense maps that enable localization in scenarios with narrow perspective. Experiments in realistic simulated environments demonstrate the capability of the system to localize the drone within 16 centimeter accuracy and to outperform existing State Of The Art approaches.**

Keywords-robotics; autonomous vehicles; vision and scene understanding; volumetric image representation.

I. Introduction

Coastal cliff monitoring has become increasingly critical in the face of climate change, as rising sea levels and more frequent extreme weather events accelerate coastal erosion [1]. Cliffs are particularly vulnerable to processes such as landslides and collapses, which threaten both natural ecosystems and human settlements. Monitoring using Unmanned Aerial Vehicles (UAVs) may allows for early detection of instability, helping to mitigate the risks to infrastructure and communities while informing coastal management strategies to adapt to a changing environment. Commonly used onboard monitoring instruments include cameras and LiDAR (Light Detection and Ranging) systems. Recently, geophysical systems have been developed to enable subsurface imaging directly from UAVs. Among them, Ground-penetrating radar (GPR) has proven to be a powerful non-invasive tool for rocky cliffs, enabling 2D or 3D visualization of subsurface conditions and the detection of internal structural weaknesses [2]. For optimal results, the antennas must be positioned very close to the surface, typically within a few centimeters, to maximize signal-to-noise ratio, without compromising the security of the flight. Performing close-range measurements poses a major challenge for conventional localization methods that rely on GNSS systems, especially for UAVs. This scenario introduces several issues regarding the accuracy of the measurements: on one hand, satellite signal occlusion caused by proximity to surfaces or multi-path interference can compromise GPS (Global Positioning System) positioning accuracy. On the other hand, onboard applications are limited by the vehicle's weight capacity and energy consumption, reducing operational efficiency. Among the different tools used for localization in GPS-denied environments, SLAM (Visual Simultaneous Localization and Mapping) methods, especially feature-based sparse approaches, have gained considerable popularity [3]. Notable examples include methods like ORB-SLAM [4], which are particularly advantageous when computational resources are limited. However, these methods face challenges when dealing with repetitive textures or low-textured images.

Prior work has explored the creation of dense maps from sparse key-frames, as demonstrated by Zhang's work on stereo mapping [5]. Zhang and Shu introduced a dense mapping module into the ORB-SLAM2 workflow, enabling the generation of dense maps from the key-frame database. This method explored the post-processing of the mapping session, but it did not address the problem of localization in close-range inspection scenarios. Kerb et al. [6] introduced a novel approach to modeling and rendering dense maps using Gaussian splatting. This work revisited an older rasterization technique, now more feasible due to advancements in hardware architectures and rendering methods. As a result, it reignited interest in dense mapping approaches, leading to the emergence of new SLAM techniques inspired by [6] within the open-source community.

In this paper, we present a V-SLAM framework tailored for close structure inspection using UAVs. To overcome UAV constraints, we employ a two-step system. In the initial step, a stereo camera captures images during a mapping flight, using a feature-based method for ego-pose estimation [7]. The system's modular design allows for flexibility in choosing various ego-localization methods. The resulting dense map serves dual purposes: aiding mission planning and enhancing localization in the second step. In this subsequent stage, precise localization is achieved using a monocular camera, addressing weight and energy constraints posed by embedded measurement equipment such as radar and thermal cameras.

The rest of the paper is structured as follows: In Section II, we will describe the proposed method, beginning by the mapping workflow and then describing the scanning functioning. Then, in section III the results of the method will be presented, as well as the conditions of test. Finally, the

performance of the method and future work will be discussed in section IV.

II. Proposed Method

In this section, we present a novel Visual SLAM (V-SLAM) method that integrates feature-based and direct methods to generate a dense global map useful for odometry and navigation tasks. The generated dense map plays a pivotal role in localizing the agent when the SLAM system faces challenges, such as loss of perspective or low-texture environments. Its richness offers an additional layer of information to estimate poses in situations where traditional methods may fail.

During the localization step in our target use case, as the agent approaches structures, the descriptors generated by feature-based methods during the mapping session may become invisible. This typically occurs due to a change in scale of the incoming image flow, resulting in low density data. Our dense map overcomes this obstacle by storing maximal information, enabling the agent to formulate better pose hypotheses.

A. Mapping

1) Input: During the mapping flight the ego-localization system is expected to select a set of key-frames composed of a pair of stereo images $\mathbf{I}_k$ and an estimated pose ${}^0\hat{\mathbf{T}}_k$. The selection criteria may vary, but the frames shall respect some conditions for photogrammetry reconstruction such as the overlap and the coverage of the target area. Once the key-frames have been collected, the system carries the point-cloud registration process.

The registration process (see Figure 1) is carried out by aligning the incoming image $\mathbf{I}_k$ with a render of the global map $\mathbf{I}_v$ and its depth map $\mathbf{D}_v$ at the current estimated pose ${}^0\hat{\mathbf{T}}_k$. The alignment procedure leverages the map data observable at the current pose $\mathbf{M}_v$.

2) Rasterizer: Inspired by the work of [6], the framework uses the EWA (Elliptical Weighted Average) volume splatting method [8] to render (the process of converting the point-cloud into an image) $\mathbf{I}_v$ and $\mathbf{D}_v$ without need of a mesh. To do so, the EWA method propose a projection of the covariance matrices by means of a projection model linearized around the center of the ellipsoid $\mathbf{u}_j$.

$$\mathbf{J}_j = \begin{pmatrix} \frac{f_{px}}{(\mathbf{u}_j)_z} & 0 & -\frac{f_{px}(\mathbf{u}_j)_x}{(\mathbf{u}_j)_z^2} \\ 0 & \frac{f_{px}}{(\mathbf{u}_j)_z} & -\frac{f_{px}(\mathbf{u}_j)_y}{(\mathbf{u}_j)_z^2} \\ \frac{(\mathbf{u}_j)_x}{|\mathbf{u}_j|} & \frac{(\mathbf{u}_j)_y}{|\mathbf{u}_j|} & \frac{(\mathbf{u}_j)_z}{|\mathbf{u}_j|} \end{pmatrix} \tag{1}$$

Specifically, we use an implementation of the EWA method that suppose an irregular volume which implies non-circular kernels (see Figure 2). As proposed in the original EWA method, the weights are gathered from a lookup table using a radial distance function defined by the conic matrix $\mathbf{Q}_v$ (inverse of the covariance $\mathbf{V}_v$). The radial distance is computed using the ellipsoid equation and the difference in pixels between the center of the ellipsoid projected in the image plane $\mathbf{p}_v$ and the cells inside the kernel $\mathbf{p}_i$ (see Figure 3).

$$r_{vi}(\mathbf{d}_{vi}) = \mathbf{d}_{vi}^T \mathbf{Q} \mathbf{d}_{vi} \quad \text{where} \quad \mathbf{d}_{vi} = \mathbf{p}_v - \mathbf{p}_i \tag{2}$$

The radial function is scaled by the lookup table size so every pixel in the kernel corresponds to a given integer index in the lookup table (r_{vi} in Figure 3).

Nevertheless, as our scenario contemplates a decrease in the point density, instead of deforming the kernel with a low-pass filter (described by an identity matrix), we change the shape of the lookup weights for an exponential decay function in the table w_{tab} (see Figure 3). This allows the rasterizer to fill the unpopulated areas with the information of the neighbors without excessively blurring the points in the center of close kernels.

As the first key-frame is used as reference and therefore directly saved into the map, no rendering is needed. However, for the next frames the map manager will collect the visible map points $\mathbf{M}_v$, their colors $\mathbf{C}_v$ and its respective covariance matrices $\mathbf{V}_v$ into a *virtual* surface that will serve for rendering. The surface is then projected into the image space to generate a new image and depth-map $(\mathbf{I}_v, \mathbf{D}_v)$. This process involves selecting a weight from the look-up table w_{tab} based on the radial distance index derived from the ellipsoid equation r_{vj}.Each color value $\mathbf{c}_v$ is then aggregated, weighted by the selected weight w_{vj}.

$$\mathbf{c}_j = \frac{1}{\sum^v w_{vj}} \sum^v w_{vj} \mathbf{c}_v \quad \text{where} \quad w_{vj} = w_{tab}(r_{vj}(\mathbf{d}_{vj}))) \tag{3}$$

3) Dense visual odometry (DVO): Then, the DVO module [9] computes the pose ${}^v\tilde{\mathbf{T}}_k$ that represents the estimated error between the estimated pose and the real pose. The corrected pose of the Keyframe can then be composed from the estimated pose and the error estimation.

The corrected pose is then used to align the incoming point-cloud data with the global map data. This map is build as a graph of *surfaces* $\mathbf{S}_s$, composed of a set of 3D points $\mathbf{M}_s$, a set of colors $\mathbf{C}_s$, a set of covariance matrices $\mathbf{V}_s$ and a pose ${}^0\hat{\mathbf{T}}_s^*$. This serves to model a fictitious camera that organizes the data in a grid corresponding to an image plane $\mathbf{I}_s$. This simplifies the process of updating the map and the selection of points to render.

4) Point-cloud computation: Once the pose is corrected, a new data-cloud is computed and filtered. In the current implementation, the depth-maps are computed by two algorithms chosen depending on the scenario. The Stereo Block Matching algorithm of the OpenCV [10] library for unstructured environments and the CREStereo pre-trained model for structured environments. The new depth-map is compared with the existing data in the map, if this information is too different, that region of the map is considered an anomaly. Abnormal depth estimations can appear for many reasons, specially with StereoBM, errors in the depth estimation or unveiling of occluded objects. Spurious information in known regions of the map tends to be visually similar but spatially incoherent. In contrast, revealed object are both visual and

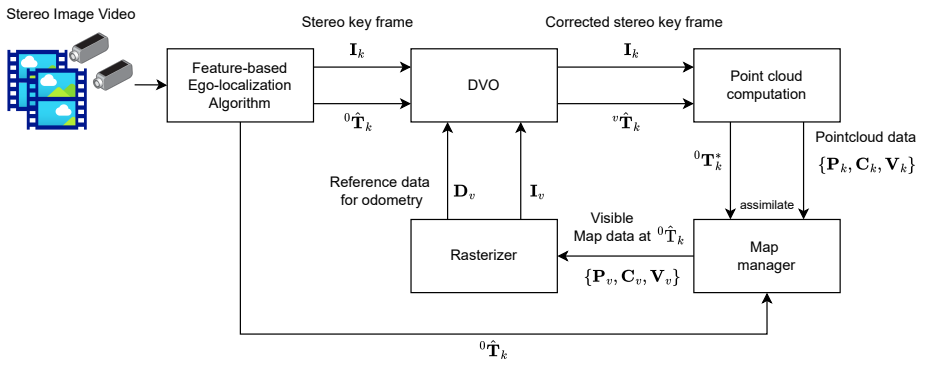

Figure 1. Workflow for the mapping phase. Most computations are currently computed offline due to constraints related to the current implementation. One frame takes 300 seconds to be treated, execution time profiles indicate that 90% of the time is expended in object instances creations and single-threaded operations that could be parallelized.

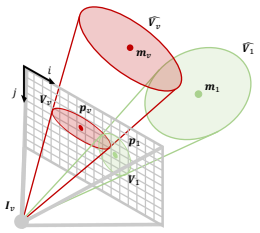

Figure 2. The 3D covariance matrices in camera frame $\hat{\mathbf{V}}_1, \ldots, \hat{\mathbf{V}}_v$ are mapped as ellipsoid regions in the image space $\mathbf{V}_1, \ldots, \mathbf{V}_n$ through perspective projection linearized around the center of the ellipsoids $\mathbf{m}_1, \ldots, \mathbf{m}_v$ whose coordinates in the image plane are $\mathbf{p}_1, \ldots, \mathbf{p}_v$. Then the colors are merged by means of a weighted average.

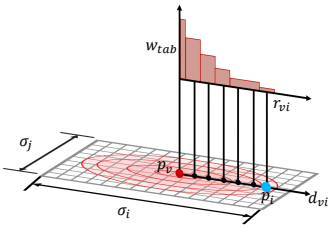

Figure 3. The weights are gathered from a lookup table using a radial function defined by the covariance matrix. This distance is computed for each pixel covered by the ellipsoid of the kernel $\mathbf{p}_i$ around the projected center $\mathbf{p}_v$. The size of the kernel is determined by the covariance matrix.

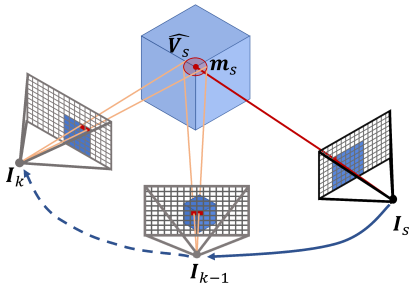

Figure 4. point-cloud alignment is achieved through the minimization of distance between the renderings at the estimated positions $\mathbf{I}_v$ and the key-frame images $\mathbf{I}_k$. The set of points that fall in the same pixels of $\mathbf{I}_s$ are used to update the covariance matrix of the map ellipsoids $\mathbf{m}_s$. The projection done is similar to the rendering process (linearization around the center of the ellipse).

spatially incoherent. This can be exploited to reject defective stereo images of noisy depth computations at the same time that legitimate new information is kept.

5) Map manager: Accepted data-clouds are then processed by the map manager which decides either to generate a new surface or update an existing one. This decision is done based in multiple criteria. The first criterion is the distance from the last reference surface, a straightforward measure ensuring proximity for alignment in image space. The second one is the overlap of the reference surface and the incoming point-cloud. If visible area of the key-frames changes significantly, a new surface is generated to maximize information gathering. The third criterion is a custom measure based on the coverage of interesting areas. Even with substantial overlap, is still possible to neglect some interesting areas observed just once or twice during the mapping session. The current implementation uses saliency maps computed by the Static Saliency module of OpenCV, a human-centered measure of local image relevance, but objective relevance metrics will be tested to enhance the probability of convergence in localization algorithms. At the moment, the system computes a salient map of the incoming point-cloud and compare it with the reference surface saliency map projected in the same image space. If interesting areas are not covered by the reference surface, the system will generate a new one.

Finally, there is a pixel size check. Lower depth values indicate that the camera is closer to the surface. When this occurs, a new node is created and initialized with data from a neighboring node. The covariance matrices for the new node are adjusted based on the characteristics of the incoming image, ensuring that the map keeps all available information to enhance performance during the next phase.

New key surfaces $\mathbf{S}_s$ are initialized with the information

from the surrounding surfaces and the incoming point-cloud. The matches of points are obtained through alignment in the image space. By projecting the 3D ellipsoids into the same image plane, the system can compare the points of the surfaces with the incoming point-cloud(see Figure 4). In the empty cells, the grid assimilates the values of the neighbors or the incoming point-cloud. In populated cells, the new value is averaged with the current one if the distance between them is within a certain threshold. Otherwise, the system overrides the existing point with the new one. This update process mimics the splatting method for rendering so the misalignment of image pixels and reference pixels do not affect the quality of the update. The covariance matrix $\mathbf{V}_s$ is merged with a new one with the shape of the current pixel size. The center of the ellipsoid is updated as the result of an iterative mean computation from all the point observations after alignment. Currently, the fusion of the color information $\mathbf{C}_s$ is done with a mean computation in the LAB color space with the intention to better capture the perceptual changes in the color. After processing all key-frames, the system has generated a surface graph that will be used in the localization step.

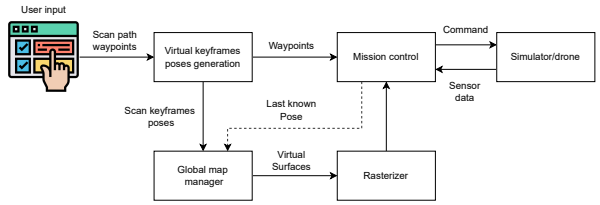

Figure 5. The global map manager will take any pose and generate a virtual surface that will be used to generate a depth and color images necessary for the odometry.

B. Localization

1) Input: In this step, a scanning path is defined either by an algorithm or by the user. In the current state of the framework, the user shall choose the shape of the scan path. After defining the way-points of the scanning path, the mission control will generate a set of virtual key-frames to assist the scanning operation.

2) Virtual frames generation: First, the map produced by the offline registration is loaded. Then, the system generates a set of frames with a co-visibility criteria (see Figure 5). By projecting the scanning path's lines onto the virtual image plane, the visible distance covered by the virtual camera's FOV (field of view) is computed. With this information, the next pose of the virtual camera is chosen by overlap criterion. In the current implementation, the path generator computes the portion of the parametric line between two way-points that is covered by the virtual camera FOV. The selection of the point along this line is based on achieving the specified percentage of overlap. This chosen point becomes the image center for the next virtual image frame. The path generator maps this pixel to the 3D space and adds an offset to get the next pose of the virtual camera. This offset places the camera at a distance behind the scan trajectory, relative to the expected position, allowing the image to cover a larger area than the real camera will capture. This approach ensures that incoming images along the scan path can be reliably tracked from at least one virtual key-frame.

3) Frame management: During the scan path, the global map manager module picks the closest virtual key-frame for localization. The selection is based on an estimation of the current pose got by the mission control module. If the DVO module fails to converge, a new virtual key-frame is generated by the rasterizer at the last known pose to restart the tracker. Since the DVO module estimates the pose from the incoming images to the reference virtual key-frames, the system can recover the global pose without accumulating any errors. The localization is carried out with the same direct odometry method used during the global map generation. However, it is worth noting that the DVO module can localize the agent using different camera configurations than those used during mapping. This can be done by configuring the rasterizer to emulate the camera settings from the incoming images.

III. Results

This section presents the results of the experiments to characterize the performance of the method. First, the test conditions and the baseline for comparison are described, followed by a discussion of the method's performance.

A. Simulations with realistic data

The data used for the experiments was collected in a simulated environment in Airsim [11] Since our color profile for the color fusion is too simple, we decided to carry the simulation in a lighting-controlled environment. The observed site is the Sainte-Marguerite-sur-Mer cliff (Normandy), monitored in the framework of the Defhy3geo project [12]. The scene is composed by a segment of a cliff model reconstructed using the Agisoft [13] metashape software with centimeter-lever precision. A drone acquired geo-referenced images, allowing the generation of a 3D model of the cliff. The effective visible area is equivalent to a 60x20 meters wall with non-structured texture. This map demonstrates the performance of the system in low texture environments.

To benchmark the performance of our method, we use ORB-SLAM3 [7] in localization mode as a baseline. In this mode, the system focuses solely on agent localization, sacrificing mapping capabilities for computational efficiency. Both methods are submitted to the following test. First, a set of key-frames is selected during a *mapping flight*, a path far from the surface to inspect and done in an arbitrary way to simulate human pilot behavior. Then, the drone performs a "scanning flight", a path close to the surface with a structured path (see Figure 9) to simulate a measuring mission. We first tried to perform the scan mission with the ORB-SLAM3 in monocular mode, but the system gets lost when the agent is too close to the mapping trajectory. Then, we tried to perform the mission in stereo mode. Figure 6 displays the absolute translation error (ATE) in meters during the scan mission in this mode. Despite

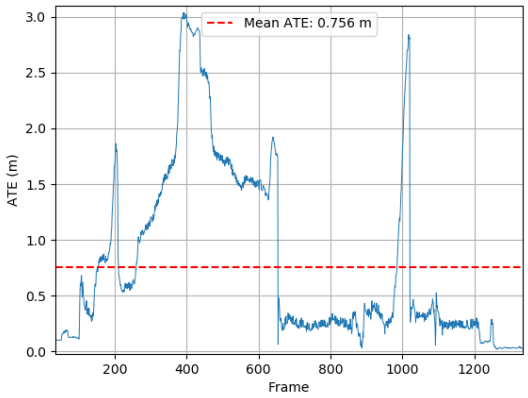

Figure 6. ORB-SLAM3 ATE (absolute translation error) in meters during the scan path with stereo camera. Being stereo the best configuration, the closest ORB-SLAM3 could get to the structure is 5 m from the structure, i.e. 8 m from the mapping path.

being the best configuration, the closest ORB-SLAM3 could get to the structure was 5m from the structure (8 meters from the mapping path). Prior to establishing correspondence with the map, localization error may escalate to 3m with possibility of recovery. Upon successful alignment with the map, the error returns to 15cm. Although the system is capable of maintaining this level of accuracy post-alignment, its performance falls short for close-range structure inspection.

The hardware used for the experiments is a custom-built computer with a Xeon processor and an Nvidia Quadro P2200 GPU. For the moment, the mapping is too slow for online use. The time of execution can be improved once the system is implemented with optimized code architecture for parallel and image processing. Our framework is built upon the following software components:

- Airsim as the simulation environment [11].
- ROS for communication between different software components.
- A Python-based offline registration node for map generation.
- A direct odometry algorithm from the OpenROX library (developed by the ACENTAURI team).
- A mission control module to load the map, generate the virtual frames and control the drone.

B. Performance

The first outcome of our method is the dense map. Figure 7 depicts the dense map generated by our approach alongside the reference ground-truth model. This map effectively captures the structural details of the environment as well as color information. Comparing the generated dense map with the ground truth, 50% of the points have less than 10cm of error, 36% are under 20cm and 10% under 30cm. In summary, the mean error of the points in the map is 11cm. The second result of our method is the capability to generate virtual frames.

In Figure 8, one can see a virtual frame generated by our method. As shown, our method is capable of generating

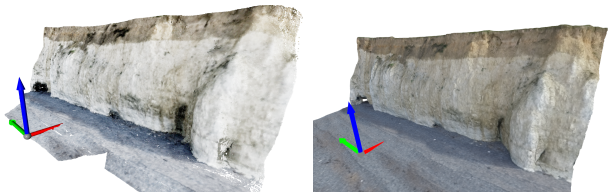

Figure 7. The map generated during the offline registration can be used to provide information for subsequent navigation tasks. The left image shows the dense map generated by our method, the right image shows the ground truth model.

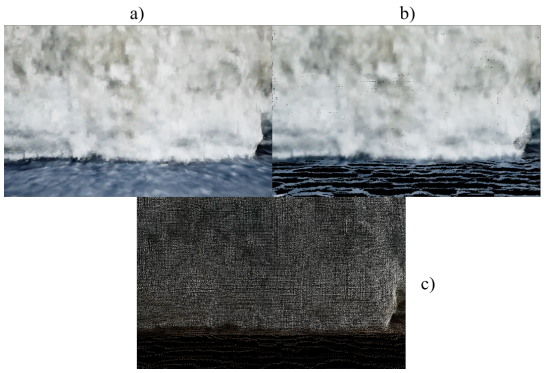

Figure 8. Virtual frames. a) Image seen by the camera, b) image rendered from the dense model and c) centers of the kernels used for rendering.

useful images even with relative low density point-clouds. Our method's third contribution is global pose estimation, demonstrating the system's resilience in adverse conditions. Figure 9 illustrate the dense map with the set of virtual reference poses. The Figure 10 shows that the agent was able to maintain the global pose estimation with mean ATE (absolute translation error) of 0.16m at 2m from the structure.

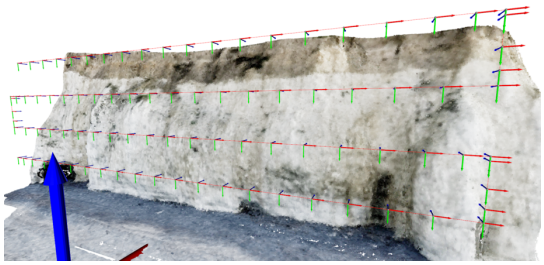

Figure 9. Path followed by the agent during the scan mission, a set of virtual poses is generated and a virtual image is rendered.

Pose estimation accuracy depends on the stereo pair-derived point cloud quality and dense odometry module accuracy. While odometry is fairly precise, depth estimation can be improved. The significant error between frames 1300 to 1500 in Figure 10 results from poor map quality in that area. This region has not been correctly modeled during mapping, which shows as under-performance of the registration module. Despite this, our method exhibits a accuracy similar to that of ORB-SLAM3 but operates at a distance 4 meters closer to the structure (2m

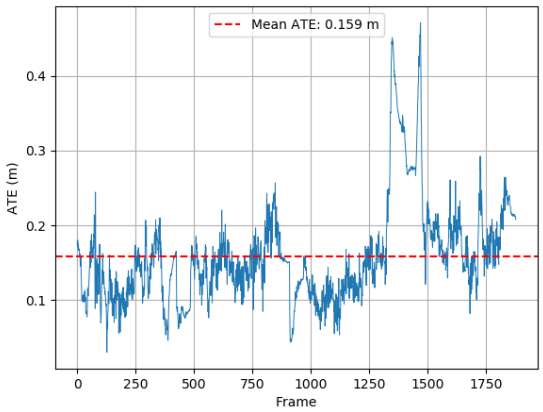

Figure 10. Our method exhibits consistent ATE behavior over 1900 frames, showing stability at various distances across multiple experiments. However, a notable shift occurs when the model is observed in regions with extremely low-texture conditions.

from the structure and 11m from the mapping path). It is worth mentioning that the position estimation results showed are not treated by any kind of filter, the error metrics were computed only with the output of the odometry module which computes an instantaneous pose estimation from the last estimated pose but does not keep track of the dynamic evolution of the images.

IV. Conclusion and Future Work

In this paper, we introduced an innovative framework designed to tackle the challenge of localization close to structures where traditional methods may fail. Our approach focuses on creating and using a dense environmental map to enhance accuracy, particularly in scenarios involving narrow-perspective cameras and low-texture images. We achieve this by accurately estimating camera ego poses and applying the projection model for efficient matching of dense point clouds and generation of synthetic images.

This versatile map proves invaluable not only for localization but also for mission control and cost-effective creation of digital twins for multi-session autonomous navigation. In the current version of our method, we seek virtual key-frames based on pose estimations, which may deviate from actual surface positions. To further strengthen our system, we plan to integrate sensor fusion techniques, such as Ultra-Wideband (UWB) and Inertial Measurement Units (IMU), streamlining surface identification and simplifying the odometry initialization process. Likewise, ongoing work aims to improve the characterization of the surface modeling data-structures and data aggregation.

Despite these advances, several challenges remain. Although a dense map augments the robustness of the system across sessions, this map representation could benefit from richer visual data-representations. First, color information modeling can be improved by considering the dynamic nature of outdoors illumination across sessions. Secondly, the registration method could be formulated in a more effective way, instead of an optimistic computation of the covariance matrices with aligned point-clouds, the map could capture this information with a more realistic representation and less susceptible to errors of depth and pose estimation.

The results show that the quality of the localization depends heavily on the quality and coverture of the map. This implies that the criteria to create new map nodes are critical in the creation of a good quality map. Ongoing work focuses on the detection of relevant information in the surfaces, detail-centered registration and memory optimization. Likewise, the creation of reference of key-frames during localization present a research venue worth exploring. So far, the main challenge is to formulate data-wise criteria to efficiently manage the map registration and the reference rendering.

While the results presented here are based on simulations with realistic data, ongoing work has shown promising results in real-world scenarios, which align with the findings presented in this study.

References

[1] J. R. Shadrick *et al.*, “Sea-level rise will likely accelerate rock coast cliff retreat rates”, *Nature Communications*, vol. 13, no. 1, p. 7005, 2022.

[2] A. D. Switzer, C. Gouramanis, C. S. Bristow, and A. R. Simms, “Ground-penetrating radar (gpr) in coastal hazard studies”, in *Geological Records of Tsunamis and Other Extreme Waves*, Elsevier, 2020, pp. 143–168.

[3] Y. Dai, J. Wu, and D. Wang, “A review of common techniques for visual simultaneous localization and mapping”, *Journal of Robotics*, vol. 2023, no. 1, p. 8 872 822, 2023.

[4] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”, *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2016. DOI: 10.1109/tro.2017.2705103.

[5] B. Zhang and D. Zhu, “A Stereo SLAM System With Dense Mapping”, *IEEE Access*, vol. 9, pp. 151 888–151 896, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3126837.

[6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering”, *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, vol. 42, no. 4, Jul. 2023.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM”, *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, ISSN: 1941-0468. DOI: 10.1109/TRO.2021.3075644.

[8] M. Zwicker, H. Pfister, J. van Baar, and M. Gross, “Ewa volume splatting”, in *Proceedings Visualization, 2001. VIS ’01.*, 2001, pp. 29–538. DOI: 10.1109/VISUAL.2001.964490.

[9] A. I. Comport, E. Malis, and P. Rives, “Real-time Quadrifocal Visual Odometry”, *The International Journal of Robotics Research*, vol. 29, no. 2, pp. 245–266, 2010. DOI: 10.1177/0278364909356601.

[10] G. Bradski, “The OpenCV Library”, *Dr. Dobb’s Journal of Software Tools*, 2000.

[11] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles”, *CoRR*, vol. abs/1705.05065, 2017. arXiv: 1705.05065.

[12] T. Junique *et al.*, “Investigation of the geological and hydrogeological structure of chalk cliffs with visible, thermal infrared and electrical resistivity imaging”, *Journal of Hydrology*, vol. 630, p. 130 642, 2024.

[13] Agisoft LLC, *Agisoft metashape*, 2019.