Underwater 3D Reconstruction by Interleaving Multimodal SLAM and Incremental Gaussian Splatting

Daniel Yang[1,2], Jungseok Hong[1], John J. Leonard[1], and Yogesh Girdhar[2]

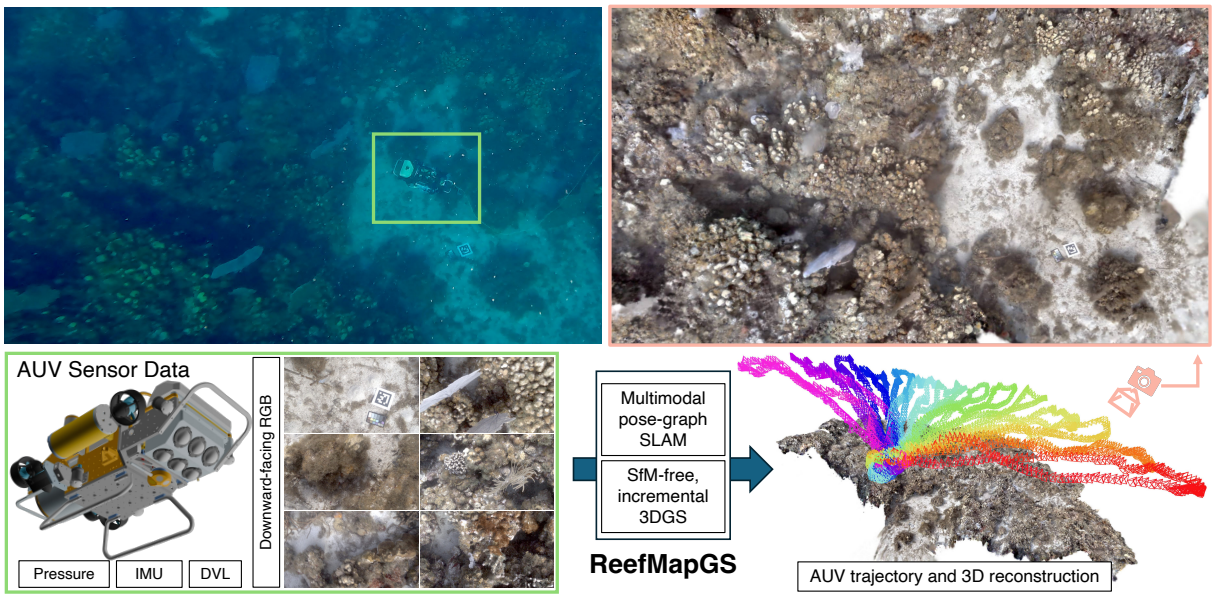

Fig. 1: **ReefMapGS** integrates multimodal pose-graph SLAM with 3D Gaussian Splatting to enable rapid, dense reconstruction of underwater environments like coral reefs without needing computationally expensive structure-from-motion pipelines. On the top left, we show an image captured by a snorkeler on the water surface passively observing a robot performing a visual benthic survey of a coral reef. Combining the AUV-collected monocular imagery with other vehicle sensor data, ReefMapGS can rapidly estimate the whole robot trajectory while also providing a high-quality, dense 3DGS reconstruction of the whole scene. The top right view is synthetically rendered from the obtained 3DGS reconstruction.

***Abstract*— 3D Gaussian Splatting enables high-quality, efficient 3D scene reconstruction, but relies on accurate camera poses typically obtained from computationally intensive structure-from-motion, making it unsuitable for field robotics. We propose ReefMapGS , an incremental 3DGS reconstruction framework that replaces SfM with pose-graph optimization-based SLAM, leveraging multi-modal sensors (e.g., acoustic, inertial, pressure, and visual) to estimate camera poses and their associated uncertainties. ReefMapGS builds an initial model from a high-certainty region and progressively expands by interleaving local tracking of new observations with 3DGS scene optimization. Refined poses are fed back into the pose graph for global trajectory optimization, tightly coupling reconstruction and localization. We show COLMAP-free 3D reconstruction of two geometrically complex underwater reef sites and improved global pose estimation of our AUV over complex foveated survey trajectories spanning up to 700 m.**

I. Introduction

Recent advances in autonomous underwater vehicles (AUVs) have enabled the mapping of challenging underwater environments, essential for environmental exploration and monitoring [1]–[5]. Many scientific and ecological tasks, such as coral reef monitoring and large-scale spatiotemporal environmental surveys, require high-quality, dense scene representations. The quality of dense reconstruction depends on the input images and the accuracy of camera pose estimation. However, obtaining these inputs for underwater scenes remains challenging due to degraded visual conditions (e.g., light attenuation, turbidity, color distortion, and low-texture scenes) and limited sensing capabilities.

Offline structure-from-motion (SfM) and multi-view stereo (MVS) methods can estimate poses in batch, but they have a high computational cost. Visual simultaneous localization and mapping (SLAM) methods can estimate poses incrementally and build maps in real time, but they are fundamentally unreliable underwater due to the challenges mentioned above. To address this issue, state-of-the-art underwater SLAM systems integrate multimodal sensors, combining inertial, velocity, and depth measurements with visual and/or acoustic cues in pose-graph optimization frameworks. While effective for long-distance trajectory estimation, these systems produce sparse maps, posing a key bottleneck for downstream tasks that require dense reconstruction, such as scientific analysis.

Recently, 3D Gaussian Splatting (3DGS) [6] has emerged as an efficient and differentiable representation that produces

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA {dxyang, jungseok, jleonard}@mit.edu

[2]Applied Ocean Physics & Engineering Department, Woods Hole Oceanographic Institution, Woods Hole, MA, USA {yogi}@whoi.edu

high-fidelity, dense reconstructions and enables pose refinement via gradient-based optimization. This advancement has motivated the development of 3DGS-based SLAM systems [7]–[9]. However, these assume reliable visual conditions or RGB-D sensing, assumptions that do not hold underwater.

We address this gap by tightly coupling multimodal pose-graph SLAM with incremental 3DGS model updates in a closed loop. Starting from regions of low pose uncertainty near a known landmark, we incrementally expand the dense scene representation. As the 3DGS model improves, refined camera poses via differentiable rendering are fed back into the pose graph to improve global trajectory estimation. This bidirectional refinement allows pose estimation and dense reconstruction to mutually improve over extended trajectories, even with noisy poses, distorted images, and minimal visual overlap. In contrast to previous 3DGS-based SLAM systems, our approach leverages a known landmark used in foveated, rosette-shaped surveys performed by AUVs in coral reef benthic surveys as well as the complementary strengths of multimodal SLAM and differentiable scene representations. The resulting pipeline is bundle-adjustment-free and capable of dense, incremental reconstruction at scales relevant to real-world underwater surveys, spanning hundreds of meters while also improving global pose accuracy. Here, we describe our approach, ReefMapGS, and perform qualitative and quantitative analysis on two challenging, real-world coral reef surveys, showing both improved 3D reconstruction quality and less error in robot trajectory estimation.

II. Related works

A. Bundle adjustment-free 3D reconstruction

Recent feed-forward 3D models reduce reliance on SfM by predicting point maps directly from pairs of images, from which camera parameters can be estimated. DUSt3R [10] introduced a framework for dense per-pixel 3D pointmap regression, enabling end-to-end reconstruction from arbitrary image pairs without camera calibrations. Many works build upon DUSt3R to improve dense feature matching [11] and handle multiple views [12], [13] or a stream of images [13], [14]. Other works like VGGT [15] utilize feed-forward transformers to simultaneously attend to and jointly reason over all input views. These models rely on learned geometric priors and dense correspondences that can initialize or regularize poses. However, when texture is scarce and viewpoints are suboptimal (e.g., underwater scenes), they tend to struggle.

B. Underwater SLAM

Underwater SLAM is substantially more challenging than terrestrial or aerial SLAM due to degraded visual conditions, limited sensing bandwidth, and the absence of GPS [16], [17]. Optical imagery suffers from attenuation, color distortion, turbidity, and low-texture scenes, causing vision-only and visual-inertial SLAM systems to fail or drift even under moderate underwater conditions [18]. To achieve robust long-horizon navigation, underwater SLAM systems therefore rely on multi-modal sensor fusion, commonly integrating IMU, DVL, and depth measurements within factor-graph or pose-graph optimization frameworks [4]. Visual or acoustic sensing is typically used to detect features and provide relative pose constraints. Underwater SLAM systems such as [19] focus on robust trajectory estimation using sparse representations. In contrast, recent works including [20]–[22] yield dense maps, but typically evaluate on relatively short trajectories rather than long-horizon mapping and rely on stereo camera inputs.

C. Dense SLAM

Dense underwater reconstruction has relied on SfM and MVS pipelines (e.g., COLMAP [23] and Metashape [24]), which are computationally intensive and unsuitable for field deployments. One method [25] replaces these pipelines with learning-based ego-motion estimation [26] for underwater mapping, but remains limited to simple trajectories and low-fidelity reconstructions. Radiance field methods such as NeRF [27] and 3DGS [6] can bridge sparse SLAM and dense reconstruction. 3DGS in particular provide a computationally efficient, differentiable representation for high-fidelity geometry and appearance. Several SLAM systems [7]–[9] incorporate 3DGS to jointly optimize poses and scene models, but assume well-calibrated, illumination-consistent inputs with high inter-frame overlap—conditions often violated in benthic imagery from low-cost AUVs due to optical degradation through water and imprecise calibration. To address these challenges, we integrate multi-modal pose-graph optimization with dense 3DGS reconstruction, enabling bundle-adjustment-free large-scale underwater reconstruction with improved global pose estimation.

III. Method

A. Problem formulation

We aim to construct a dense 3D representation of an underwater environment while estimating robot, and thus camera, poses through fusing multimodal sensor data. We assume that a robot performs a survey of an underwater scene centered around a fixed, known landmark. While here we consider a robot navigating along a rosette trajectory, a flower-petal pattern as shown in Fig. 4, ReefMapGS is applicable to other survey trajectories centered around a fixed, known landmark.

Let $\mathcal{X} = \{\mathbf{X}_k\}_{k=0}^{K}$ denote the discrete sequence of robot camera poses, where each pose $\mathbf{X}_k \in SE(3)$. $\mathcal{Z}$ denotes the sensor measurements which arrive asynchronously at sensor-specific timestamps: Monocular RGB images $\mathcal{I} = \{(t_i, \mathbf{I}_{t_i})\}_{i=1}^{N_{\text{rgb}}}$ from a downward-facing calibrated camera, linear velocities $\mathcal{V} = \{(t_j, \mathbf{v}_{t_j})\}_{j=1}^{N_{\text{dvl}}}$, $\mathbf{v}_{t_j} \in \mathbb{R}^3$ from a DVL, angular velocity $\mathbf{\Omega} = \{(t_j, \boldsymbol{\omega}_{t_j})\}_{j=1}^{N_{\text{imu}}}$, $\boldsymbol{\omega}_{t_j} \in \mathbb{R}^3$ and linear acceleration $\mathcal{A} = \{(t_j, \mathbf{a}_{t_j})\}_{j=1}^{N_{\text{imu}}}$, $\mathbf{a}_{t_j} \in \mathbb{R}^3$ from an IMU, and depth measurements $\mathcal{D} = \{(t_\ell, d_{t_\ell})\}_{\ell=1}^{N_{\text{depth}}}$, $d_{t_\ell} \in \mathbb{R}$ from a pressure sensor. Note that depth here refers to distance from the sea surface, not depth from the camera sensor.

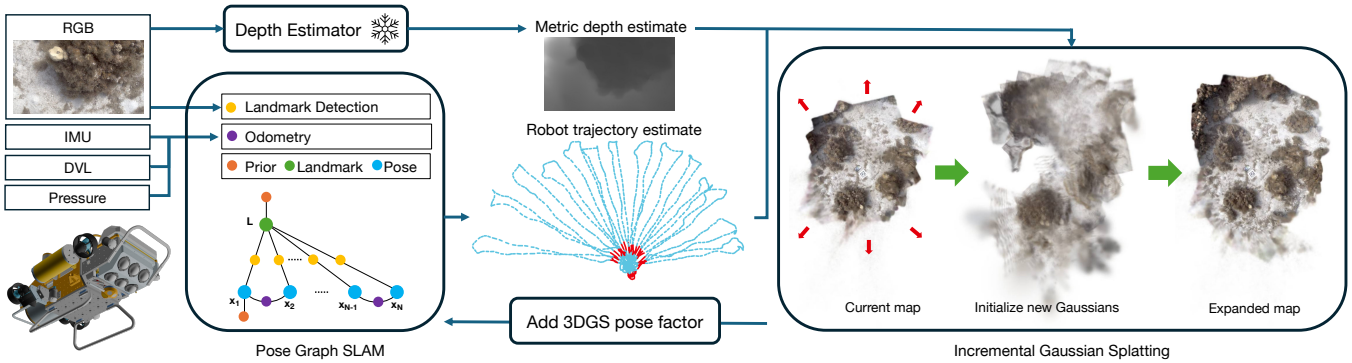

Fig. 2: **System overview** ReefMapGS takes as input RGB images and inertial, acoustic, and pressure sensor data from an AUV. Sensor data is fused into odometry while the known, central landmark is detected from RGB frames. Together, this information is integrated into a factor graph to estimate the whole AUV trajectory, shown in the center. We incrementally build a 3DGS model of the scene, starting from the region of highest certainty near the center. The solid blue line shows observations incorporated into the current map. From this current map, we expand and incorporate new observations at the frontier, the solid red line. The observations are aligned with the current map starting from the pose estimate from the factor graph. Together with depth information from a metric depth estimation model, we initialize new Gaussians and then optimize the model, yielding an expanded map. We repeat this process to incorporate the remaining observations, shown in the dashed blue line.

B. Pose-graph optimization for trajectory estimation

We formulate the pose estimation problem as a factor graph optimization problem, where we estimate the robot's camera pose $\mathcal{X}$ and landmark $\mathcal{L}$ via Maximum a Posteriori (MAP) inference given sensor measurements $\mathcal{Z}$.

$$\mathcal{X}^\star, \mathcal{L}^\star = \underset{\mathcal{X}, \mathcal{L} \in SE(3)}{\arg\max}\ p(\mathcal{X}, \mathcal{L} \mid \mathcal{Z}). \tag{1}$$

We first estimate odometry using an Extended Kalman Filter (EKF). We also pre-process the raw IMU measurements using a complementary filter [28] to obtain a stable, real-time orientation estimate. We fuse the filtered IMU data, depth, and DVL velocity measurements with an EKF [29] to obtain odometry data.

To build a factor graph, we incorporate prior knowledge of a static landmark located at the center of each survey site. We add both odometry and landmark measurement factors into the graph, each associated with robust noise models (e.g., Huber) to handle potential outliers. The final pose trajectory is estimated by solving the factor graph using Levenberg–Marquardt (LM) optimization.

C. Incremental 3D Gaussian Splatting

ReefMapGS utilizes a 3D Gaussian Splatting (3DGS) map representation, parameterizing the scene with isotropic 3D Gaussians and using zero order spherical harmonics. Each 3D Gaussian has 8 parameters: mean $\boldsymbol{\mu} \in \mathbb{R}^3$, scale $\boldsymbol{S} \in \mathbb{R}^1$, opacity $\boldsymbol{o} \in \mathbb{R}^1$, and color $\boldsymbol{c} \in \mathbb{R}^3$. This reduced parameterization lowers memory cost and simplifies optimization. Typically, a whole 3DGS model is optimized given a dataset of RGB images, corresponding poses, and a point cloud to initialize the 3DGS model, the latter two of which are typically obtained from SfM (e.g., COLMAP [23]). All views of the scene are jointly optimized and needed from the start.

Our underwater field-robotics context violates these assumptions: poses are estimated from noisy multimodal sensor fusion, point clouds are generated from a monocular depth estimator, and the fixed landmark in the pose-graph introduces spatially varying uncertainty, where poses near the landmark are tightly constrained while those further away show much higher uncertainty.

We therefore propose leveraging the fixed, known landmark to build the 3DGS scene incrementally, starting from the region of lowest uncertainty near the landmark and progressively incorporating new observations at the frontier, as shown in Fig. 2. Concretely, we discretize the robot trajectory by distance from the landmark into rings of paired poses and image observations. Before incorporating each new ring, we locally refine its pose estimates by fitting observations against the scene optimized thus far (left of Fig. 3). Given a 3DGS model of the partial scene, a new image, and an initial camera pose estimate, we minimize reconstruction error and back-propagate gradients through the 3DGS model to the pose, as implemented in gsplat [30]. This requires sufficient visual overlap and fails if the initial pose is too far from the true pose. Scene quality and pose accuracy thus improve synergistically as the model is refined through photometric reconstruction and adaptive density control.

Local refinement, however, corrects only individual poses. To propagate refinements globally, we add the refined poses as 3DGS-derived factors in the factor graph. For each refined pose $i \in \mathcal{I}_{\text{3DGS}}$, we add a cost term:

$$J_{\text{ext}}(\mathcal{X}) = \sum_{i \in \mathcal{I}_{\text{3DGS}}} \left\| \operatorname{Log}\left((\tilde{\mathbf{X}}_i^{\text{3DGS}})^{-1}\, \mathbf{X}_i \right) \right\|^2_{\Sigma_i^{\text{3DGS}}} \tag{2}$$

where $\tilde{\mathbf{X}}_i^{\text{3DGS}}$ is the refined pose for a pose $\mathbf{X}_i$ and Σ_i^{3DGS} encodes its predefined uncertainty. The refined trajectory and landmark estimates are:

$$\mathcal{X}^{\star\star}, \mathcal{L}^{\star\star} = \arg\min_{\mathcal{X},\, \mathcal{L}} \left(J_{\text{base}}(\mathcal{X}, \mathcal{L}) + J_{\text{3DGS}}(\mathcal{X}) \right) \tag{3}$$

The addition of these 3DGS camera factors can cause a large shift in the global trajectory, as shown in Fig. 3 from light gray to dark purple, particularly in uncertain regions far from the landmark. Thus, before expanding the frontier of the 3DGS model and incorporating a new set of image observations, we perform this global trajectory optimization. Specifically, we only perform global trajectory optimization if the average uncertainty of camera poses at the frontier

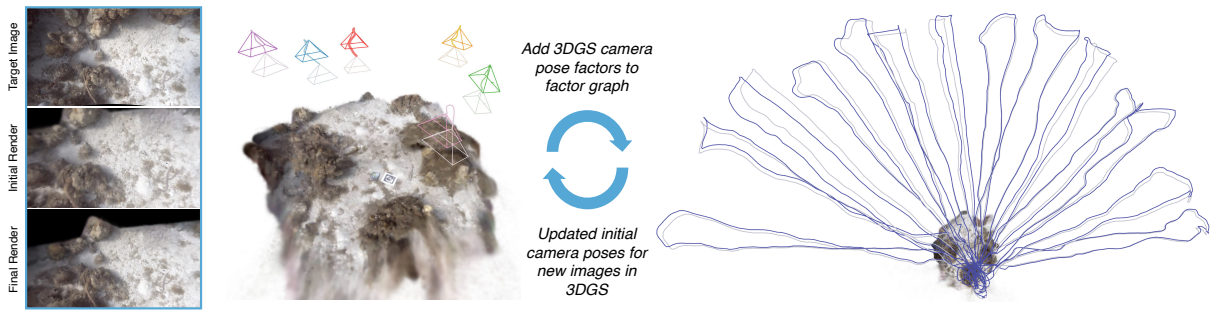

Fig. 3: **Incremental, radially-expanding 3DGS optimization framework.** Local camera pose refinement occurs as we radially and incrementally expand our 3DGS-based map representation, incorporating new cameras and their visual data into the scene. Given a current 3DGS map, target image, and initial camera pose estimate, we can refine the camera pose with gradient descent (as shown from the light to dark frustums). Sample target, initial pose render, and final pose render images are shown on the left for the blue frustum. As camera poses are refined, we add these refined camera pose estimates as factors into our pose-graph to globally update the whole robot trajectory, as shown between the light (before) and dark (after) overlayed trajectories. Local pose refinement aligns new observations before expanding the scene, while global trajectory optimization propagates these changes to poses far away.

region has not increased significantly, more than doubled, from the central, lowest uncertainty seed region. In other words, when the factor graph is most certain about robot states, but these state estimates do not align with the 3DGS scene, we must incorporate our additional 3DGS camera factors to correct these biases.

We optimize our 3DGS map, $\mathcal{G}$, with the a modified uncertainty-aware version of the original 3DGS reconstruction loss [6] and an additional edge-aware total variation loss [31] on the rendered depth, $\mathcal{L}_{Z_{TV}}$, promoting smoothness while allowing for variation along areas of high gradient in the observed camera image (e.g., edges). We incorporate a DINOv2 [32] based uncertainty modeling component similar to past works [8], [33], [34] by projecting 3D-aware features [35] through a shallow MLP, $\mathcal{P}$, to an uncertainty map, β, with modulates the 3DGS L1 reconstruction loss:

$$\mathcal{L}_{\mathcal{G}} = (1-\lambda_1)\left|\left|\frac{I-\hat{I}}{\beta_{\text{detach}}}\right|\right|_1 + \lambda_1 \mathcal{L}_{\text{SSIM}} + \lambda_2 \mathcal{L}_{Z_{TV}} \quad (4)$$

The uncertainty MLP, $\mathcal{P}$, is optimized similar to [8], with the modified SSIM loss and two regularization terms from [34]. One regularization term, $\mathcal{L}_{\text{unc-var}}$ minimizes the variance of uncertainty for similar features while the other, $\mathcal{L}_{\text{unc-log}} = \log \beta$ prevents β from a degenerate solution of high uncertainty everywhere. $\mathcal{P}$ is jointly optimized with the 3DGS map, $\mathcal{G}$, with the β detached in the reconstruction loss and renders from $\mathcal{G}'$ detached in the uncertainty loss.

$$\mathcal{L}_{\mathcal{P}} = \frac{\mathcal{L}'_{\text{SSIM}}}{\beta^2} + \lambda_3 \mathcal{L}_{\text{unc-var}} + \lambda_4 \mathcal{L}_{\text{unc-log}} \quad (5)$$

D. Using monodepth estimators to generate pseudo depth

While 3DGS normally relies on initialization from a point cloud, typically a byproduct of SfM, or rely on expensive sampling through methods like MCMC [36], ReefMapGS avoids using structure-from-motion entirely. We utilize a DepthAnythingV2 [37] model fine-tuned to output metric-depth within our underwater benthic imagery. We generate a pseudo ground truth dataset to fine-tune this model, by fitting dense depth from an off-the-shelf relative depth pre-trained DepthAnythingV2 model with the sparse point cloud obtained from Metashape, similar to previous works [38]. In the absence of stereo vision or a depth sensor on our AUV, we utilize the outputs of the metric depth model to initialize new Gaussians within our 3DGS scene.

Method	Tektite		Yawzi	
	Length (m)	RMSE (m)	Length (m)	RMSE (m)
Metashape (ref)	347.305	0.000	695.504	0.000
GTSAM	347.198	0.328	701.151	0.283
ORB-SLAM3	17.714	0.202	6.931	0.222
DROID-SLAM	18.343	4.305	142.180	4.532
VGGT-SLAM	11.938	5.029	83.476	6.085
MASt3R-SLAM	52.540	5.496	48.608	1.398
MonoGS	-	f	-	f
WildGS-SLAM	139.286	4.408	681.151	4.488
ReefMapGS	361.458	0.135	721.253	0.229

TABLE I: **Trajectory estimation metrics.** ATE RMSE is calculated in meters. Some methods are unable to track all frames, so we report RMSE over those frames that are tracked as well as total path length. Lower RMSE is better but the path length should also be closer to the reference. Best results using non-oracle data are highlighted as first , second , and third . f denotes complete tracking failure. For RMSE, lower is better ↓ .

IV. Experiments

A. Dataset collection

We evaluate ReefMapGS on coral reef benthic survey data collected at two sites in the US Virgin Islands, Tektite and Yawzi, both biologically active with a patchy mix of sand, seagrass, and hard and soft corals [39]. An AprilTag [40] is placed at the center of the reef. Data is collected by CUREE [1], a low-cost underwater vehicle equipped with a 4K downward-facing monocular camera, a Waterlinked DVL A50, an ICM-20602 IMU, and a BlueRobotics Bar30 pressure sensor. CUREE follows fixed rosette trajectories at a target altitude of 2 m above the seafloor: a half-rosette at Tektite (347.3 m, 27.0 min) and a full rosette at Yawzi (695.5 m, 43.2 min).

RAW imagery is white-balanced and converted to sRGB; the consistent survey altitude and downward-facing geometry yield visually uniform data with minimal distance-dependent effects. Images are rectified using the vehicle's default calibration, and we use only a center crop (preserving aspect ratio) to reduce peripheral distortion. Notably, unlike typical room-scale datasets, our top-down benthic imagery exhibits significantly less inter-frame overlap, further reduced by this cropping. Sample images are shown in Fig. 1.

B. Evaluation metrics

We evaluate our framework through two different criteria, visual fidelity of the observed images in the scene reconstruction and absolute trajectory error (ATE) of the robot trajectory. Visual fidelity is evaluated with peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and perceptual distance (LPIPS). The ground truth reference of the robot trajectory is approximated using Metashape to estimate the pose of each image collected, and we use evo [41] to align the reference before computing the ATE RMSE in meters. We also report the trajectory length, as not all methods successfully track the full image sequence, providing context for interpreting the ATE RMSE metric.

C. Comparative baselines

We evaluate our framework against other visual SLAM methods: ORB-SLAM3 [42], DROID-SLAM [43], MASt3R-SLAM [44], VGGT-SLAM [45], MonoGS [9], WildGS-SLAM [8]. With these methods, we focus on the ATE RMSE, as not all methods provide a dense reconstruction from which reconstruction quality can be quantitatively evaluated. We also evaluate against a baseline multimodal landmark SLAM implementation with GTSAM [46], suggesting an upper bound on ATE RMSE using solely multimodal sensor data without integrating dense visual information.

To measure reconstruction quality, we evaluate our framework with various permutations of input data, specifically the camera poses and initialization points, and the 3DGS optimization procedure, either in batch where all images and poses are available initially or incrementally, following the radially outward optimization from region of lowest to highest pose uncertainty as described in Sec. III-C. Some input permutations, those utilizing poses or the pointcloud from Metashape, are unrealistic to obtain in an "in the wild" field robotics context and are thus considered to use oracle information. We also ablate our framework with and without the global optimization step, where 3DGS camera factors are added into the pose graph to re-optimize the trajectory. We annotate each of the methods in Tab. II as such. For example, Metashape+sfm+batch means poses from Metashape, initialization points from SfM, and 3DGS optimization in batch as typically done. On the other hand, GTSAM+md+inc means camera poses from our factor graph, initialization points from a depth estimator, and 3DGS optimization incrementally. As an ablation, +reopt indicates if global reoptimization of the factor graph occurs during incremental 3DGS optimization. We also compare reconstruction quality with the WildGS-SLAM [8] output. All experiments are run with a Ryzen 9 7900X CPU and RTX 6000 Ada GPU.

V. Results

A. Pose estimation accuracy

Average trajectory error RMSE and estimated path length are shown for all methods quantitatively in Tab. I and qualitatively in Fig. 4. ReefMapGS achieves the lowest RMSE on Tektite (0.135 m) and the second lowest on Yawzi (0.229 m), while closely matching the reference path length. Although

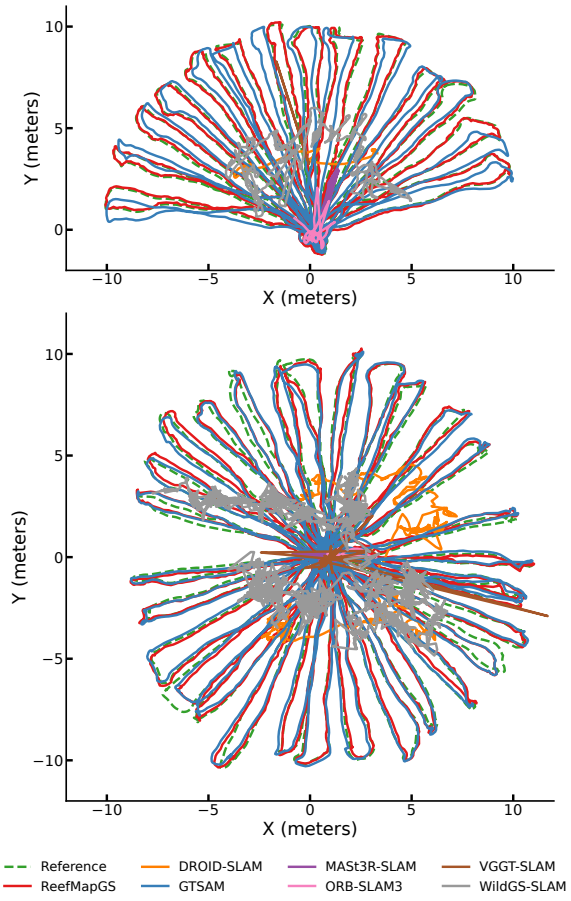

Fig. 4: **Qualitative trajectory evaluation** Top-down view of robot trajectories estimated by ReefMapGS and other baselines methods are plotted against the reference trajectory estimated by Metashape, dashed green, at both reef sites, Tektite (above) and Yawzi (below).

ORB-SLAM3 improves upon the RMSE by less than a centimeter on Yawzi, it tracks only a few meters of the full trajectory. Importantly, ReefMapGS reduces the ATE RMSE relative to GTSAM by 58.8% on Tektite and 19.0% on Yawzi, demonstrating that incorporating a dense 3DGS-based map representation into pose-graph-based multimodal SLAM yields more accurate full-trajectory estimation.

Learning-based and 3DGS-based SLAM methods all struggle with our data, with RMSEs an order of magnitude larger (4-5 m range), significant relative to the 10 m rosette radius. These methods track only short segments of smoother motion before losing tracking, particularly at the sharp turns at the ends of each rosette petal. We hypothesize that learning-based methods struggle with our data due to its top-down viewpoint, narrow field of view, and atypical semantics, which make it out-of-distribution from typical training data. ORB-SLAM3 tracks a short segment near the center with competitive ATE RMSE before losing tracking, highlighting the continued value of classical pipelines. We also note that GTSAM and ReefMapGS provide near metric-scale trajectories by leveraging inertial, velocity, and depth measurements, whereas purely visual methods are off by a scale factor.

	Method	Tektite PSNR ↑	Tektite SSIM ↑	Tektite LPIPS ↓	Tektite RMSE (m) ↓	Yawzi PSNR ↑	Yawzi SSIM ↑	Yawzi LPIPS ↓	Yawzi RMSE (m) ↓
Oracle	Metashape+sfm+batch	20.447	0.499	0.886	**0.000**	19.977	0.492	0.918	**0.000**
	Metashape+sfm+inc	**22.395**	**0.577**	**0.673**	0.070	**22.522**	**0.594**	**0.680**	0.064
	Metashape+md+batch	20.273	0.501	0.880	**0.000**	20.060	0.496	0.917	**0.000**
	Metashape+md+inc	21.975	0.564	0.702	0.066	22.158	0.583	0.701	0.059
In the wild	GTSAM+md+batch	17.846	0.461	0.945	0.328	17.683	0.467	0.958	0.283
	GTSAM+md+inc	21.051	0.542	0.767	0.336	20.746	0.541	0.799	0.331
	WildGS-SLAM	10.253	0.242	0.980	4.407	9.779	0.272	0.993	4.488
	ReefMapGS (GTSAM+md+inc+reopt)	22.219	0.567	0.721	0.135	21.226	0.551	0.777	0.229

TABLE II: **Reconstruction and tracking metrics across different methods.** We ablate across different configurations of input data and 3DGS optimization and with WildGS-SLAM. Initial poses can come from `GTSAM` or `Metashape`, Gaussians are initialized either from the SfM point cloud, `sfm`, or depth estimator outputs, `md`, and the 3DGS can be optimized either in batch, `batch`, or incrementally following our method, `inc`. Input data configurations that use oracle data from SfM are grayed. Best results using non-oracle data are highlighted as first and second, while best results among all are **bolded**.

Method	Runtime (hh:mm) Tektite	Yawzi
COLMAP	10:40	17:14
Metashape	8:04	6:57
ORB-SLAM3	0:01	0:02
DROID-SLAM	0:03	0:05
VGGT-SLAM	0:08	0:06
MASt3R-SLAM	0:14	0:14
WildGS-SLAM	8:36	8:41
ReefMapGS	2:30	3:22

TABLE III: **Processing time.** ReefMapGS cuts down processing time significantly compared to open source (COLMAP) and proprietary (Metashape) SfM software, requiring roughly 3 hours for both scenes. While learning-based methods are fast none yield coherent results on our scenes.

B. 3D reconstruction quality

Tab. II shows the reconstruction quality of ReefMapGS and baselines across both the Tektite and Yawzi scenes. Across both scenes, ReefMapGS achieves higher reconstruction quality than when using the oracle data with batch optimization, Metashape+sfm+batch. When incorporating our incremental optimization with local pose refinement as new frontiers are incorporated (+inc instead of +batch), we see that across all input data scenarios, even when purely using SfM data, reconstruction quality increases significantly. This highlights the effectiveness of our incremental 3DGS optimization, ensuring scene coherence by optimizing it from the most certain region and aligning new observations with the reconstruction thus far.

We also see that initializing Gaussians with depth estimator data yields a slight, but not catastrophic, decrease in quality, as opposed to initializing Gaussians with the SfM point cloud (+md instead of +sfm). This highlights the role that depth images can use in 3DGS-based pipelines for visual SLAM and shows that monocular depth estimators can be a powerful replacement when stereo vision is unavailable, either from hardware limitations or when visual data quality prevents application of stereo matching and depth from disparity algorithms (e.g. low texture sandy regions).

Finally, we see that global re-optimization of the factor graph using 3DGS camera pose factors further increases reconstruction quality. This highlights how incorporating 3DGS camera pose factors, resulting from local camera refinement while incrementally optimizing the model, can increase model quality. Intuitively, if two petals of the rosette have overlapping visual content, but the trajectories have drifted apart due to an error in pose estimation closer to the central landmark, then local pose refinement may not perform this visual loop closure. With global pose refinement incorporating small corrections that accumulate to these large drifts, redundant visual structures from different trajectory petals can be reconciled, increasing reconstruction quality.

C. Running time

Tab. III shows runtimes for various methods; for COLMAP and Metashape, only camera pose estimation time is shown, excluding 3D reconstruction. SfM pipelines are computationally expensive, with COLMAP requiring roughly 11 and 17 hours for Tektite and Yawzi, respectively, and Metashape 8 and 7 hours. ReefMapGS completes both scenes in approximately 3 hours, 2–3x faster than the other 3DGS-based system, with local pose refinement being the primary bottleneck. Learning-based and classical methods process scenes in minutes, though none produced coherent outputs on our datasets, highlighting a promising future direction.

VI. Conclusions

We present ReefMapGS , a framework that tightly integrates multimodal pose-graph SLAM with incremental 3DGS for large-scale underwater reconstruction. From initial poses obtained by fusing IMU, DVL, depth, and landmark measurements, we incrementally build a dense 3DGS representation, locally refine camera poses via differentiable rendering, and feed refinements into the pose graph for global optimization, enabling pose estimation and reconstruction to mutually improve. Our approach reconstructs large-scale underwater scenes faster and without relying on computationally intensive SfM, making it practical for field deployments. We demonstrate reconstruction quality competitive with or exceeding that of using oracle SfM inputs and another 3DGS-based SLAM method, as well as low ATE compared to both a multimodal SLAM baseline lacking dense scene feedback and visual SLAM methods that fail on our challenging datasets. Future work includes extending to landmark-free surveys, exploring global re-optimization strategies, and enabling live onboard deployment. Overall, ReefMapGS provides a practical path toward reliable large-scale underwater mapping by closing the loop between multimodal SLAM and 3DGS.

REFERENCES

[1] Y. Girdhar, N. McGuire, L. Cai, S. Jamieson, S. McCammon, B. Claus, J. E. S. Soucie, J. E. Todd, and T. A. Mooney, "Curee: A curious underwater robot for ecosystem exploration," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 411–11 417.

[2] B. Joshi, M. Xanthidis, M. Roznere, N. J. Burgdorfer, P. Mordohai, A. Q. Li, and I. Rekleitis, "Underwater exploration and mapping," in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*. IEEE, 2022, pp. 1–7.

[3] M. Johnson-Roberson, M. Bryson, A. Friedman, O. Pizarro, G. Troni, P. Ozog, and J. C. Henderson, "High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology," *Journal of Field Robotics*, vol. 34, no. 4, pp. 625–643, 2017.

[4] F. F. R. Merveille, B. Jia, Z. Xu, and B. Fred, "Advancements in sensor fusion for underwater slam: A review on enhanced navigation and environmental perception," *Sensors (Basel, Switzerland)*, vol. 24, no. 23, p. 7490, 2024.

[5] S. Williams and I. Mahon, "Simultaneous localisation and mapping on the great barrier reef," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2, 2004, pp. 1771–1776 Vol.2.

[6] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis *et al.*, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[7] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track and map 3d gaussians for dense rgb-d slam," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 21 357–21 366.

[8] J. Zheng, Z. Zhu, V. Bieri, M. Pollefeys, S. Peng, and I. Armeni, "Wildgs-slam: Monocular gaussian splatting slam in dynamic environments," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 11 461–11 471.

[9] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian Splatting SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[10] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 697–20 709.

[11] Y. Cabon, L. Stoffl, L. Antsfeld, G. Csurka, B. Chidlovskii, J. Revaud, and V. Leroy, "Must3r: Multi-view network for stereo 3d reconstruction," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1050–1060.

[12] B. P. Duisterhof, L. Zust, P. Weinzaepfel, V. Leroy, Y. Cabon, and J. Revaud, "Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion," in *2025 International Conference on 3D Vision (3DV)*. IEEE, 2025, pp. 1–10.

[13] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa, "Continuous 3d perception model with persistent state," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 10 510–10 522.

[14] H. Wang and L. Agapito, "3d reconstruction with spatial memory," *arXiv preprint arXiv:2408.16061*, 2024.

[15] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 5294–5306.

[16] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *IFAC conference of manoeuvering and control of marine craft*, vol. 88. Lisbon, 2006, pp. 1–12.

[17] J. J. Leonard and A. Bahr, "Autonomous underwater vehicle navigation," *Springer handbook of ocean engineering*, pp. 341–358, 2016.

[18] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Q. Li, N. Vitzilaios *et al.*, "Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7227–7233.

[19] S. Rahman, A. Q. Li, and I. Rekleitis, "Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1861–1868.

[20] W. Wang, B. Joshi, N. Burgdorfer, K. Batsosc, A. Q. Lid, P. Mordohaia, and I. Rekleitisb, "Real-time dense 3d mapping of underwater environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5184–5191.

[21] S. Xu, K. Zhang, and S. Wang, "Aqua-slam: Tightly-coupled underwater acoustic-visual-inertial slam with sensor calibration," *IEEE Transactions on Robotics*, 2025.

[22] J. Song, O. Bagoren, R. Andigani, A. Sethuraman, and K. A. Skinner, "Turtlmap: Real-time localization and dense mapping of low-texture underwater environments with a low-cost unmanned underwater vehicle," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1191–1198.

[23] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] "Agisoft Metashape: Professional Edition." [Online]. Available: https://www.agisoft.com/features/professional-edition/

[25] J. Sauder, G. Banc-Prandi, A. Meibom, and D. Tuia, "Scalable semantic 3d mapping of coral reefs with deep learning," *Methods in Ecology and Evolution*, vol. 15, no. 5, pp. 916–934, 2024.

[26] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[27] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[28] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for imus and margs," *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, 2015.

[29] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13*. Springer, 2016, pp. 335–348.

[30] V. Ye, R. Li, J. Kerr, M. Turkulainen, B. Yi, Z. Pan, O. Seiskari, J. Ye, J. Hu, M. Tancik *et al.*, "gsplat: An open-source library for gaussian splatting," *Journal of Machine Learning Research*, vol. 26, no. 34, pp. 1–17, 2025.

[31] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[32] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[33] J. Kulhanek, S. Peng, Z. Kukelova, M. Pollefeys, and T. Sattler, "WildGaussians: 3D gaussian splatting in the wild," *Advances in Neural Information Processing Systems*, vol. 38, 2024.

[34] W. Ren, Z. Zhu, B. Sun, J. Chen, M. Pollefeys, and S. Peng, "Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 8931–8940.

[35] Y. Yue, A. Das, F. Engelmann, S. Tang, and J. E. Lenssen, "Improving 2D Feature Representations by 3D-Aware Fine-Tuning," in *European Conference on Computer Vision (ECCV)*, 2024.

[36] S. Kheradmand, D. Rebain, G. Sharma, W. Sun, Y.-C. Tseng, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi, "3D Gaussian Splatting as Markov Chain Monte Carlo," *Advances in Neural Information Processing Systems*, vol. 38, 2024.

[37] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[38] M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, and J. Kannala, "Dn-splatter: Depth and normal priors for gaussian splatting and meshing," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025, pp. 2421–2431.

[39] N. Aoki, B. Weiss, Y. Jézéquel, A. Apprill, and T. A. Mooney, "Replayed reef sounds induce settlement of favia fragum coral larvae in aquaria and field environments," *JASA Express Letters*, vol. 4, no. 10, 2024.

[40] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.
[41] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.
[42] C. Campos, R. Elvira, J. J. G. Rodríguez, J. 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, 2021.
[43] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in neural information processing systems*, 2021.
[44] R. Murai, E. Dexheimer, and A. J. Davison, "MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
[45] D. Maggio, H. Lim, and L. Carlone, "Vggt-slam: Dense rgb slam optimized on the sl (4) manifold," *Advances in Neural Information Processing Systems*, vol. 39, 2025.
[46] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)