3D Foundation Model-Based Loop Closing for Decentralized Collaborative SLAM

Anonymous Authors

***Abstract*—Decentralized Collaborative Simultaneous Localization and Mapping (C-SLAM) techniques often struggle to identify map overlaps due to significant viewpoint variations among robots. Motivated by recent advancements in 3D foundation models, which can register images despite large viewpoint differences, we propose a robust loop closing approach that leverages these models to establish inter-robot measurements. In contrast to resource-intensive methods requiring full 3D reconstruction within a centralized map, our approach integrates foundation models into existing SLAM pipelines, yielding scalable and robust multi-robot mapping. Our contributions include: (1) integrating 3D foundation models to reliably estimate relative poses from monocular image pairs within decentralized C-SLAM; (2) introducing robust outlier mitigation techniques critical to the use of these relative poses; and (3) developing specialized pose graph optimization formulations that efficiently resolve scale ambiguities. We evaluate our method against state-of-the-art approaches, demonstrating improvements in localization and mapping accuracy, alongside significant gains in computational and memory efficiency. These results highlight the potential of our approach for deployment in large-scale multi-robot scenarios.**

***Index Terms*—Collaborative SLAM, 3D Vision**

I. Introduction

Decentralized Collaborative Simultaneous Localization and Mapping (C-SLAM) is a critical capability for multi-robot systems operating in unknown environments. In these scenarios, multiple robots must explore the environment independently, while exchanging information to build a shared global map. This task becomes particularly challenging when the robots' viewpoints differ significantly due to varying trajectories and diverse sensor placements, making it difficult to indentify overlapping map sections and generate reliable inter-robot loop closures [1]. Moreover, when network issues or bandwidth limitations prevent centralized processing, loop closure detection must occur between the agents with minimal data transmission, as exchanging entire maps is too bandwidth-intensive.

To address these challenges, we introduce a novel loop closing approach that leverages recent advancements in 3D foundation models [2], [3] to handle extreme viewpoint variations without map sharing to find overlaps. Specifically, our approach capitalizes on the ability of 3D foundation models to perform up-to-scale relative pose estimation from pairs of monocular images, even in cases of opposite viewpoints and unknown image domains thanks to extensive pretraining on large-scale datasets. We name our solution Foundation model-based LOop Closing Collaborative SLAM (FLOCC-SLAM).

FLOCC-SLAM is designed for seamless integration with existing single-robot odometry systems, enabling robust, efficient, and easily deployable C-SLAM on top of individual robot localization pipelines, assuming these provide locally consistent, metric-scale odometry (e.g., VIO). During pose graph optimization, our method utilizes the metric scale from odometry to infer the correct scale of inter-robot measurements obtained from 3D foundation models.

This paper introduces three main contributions:

- the integration of a 3D foundation model (MASt3R [3]) into a decentralized C-SLAM pipeline to estimate relative poses between robots based on monocular image pairs, providing a means to detect inter-robot loop closures in situations with limited viewpoint overlap;
- outlier detection and uncertainty modelling designed for 3D relative pose estimation with MASt3R, to reduce the occurrence of spurious loop closures;
- a set of specialized pose-graph optimization formulations to merge individual robot maps, resolving 3D scale ambiguities of the generated loop closures and refining the overall localization accuracy.

Overall, our contributions enable us to leverage recent advances in 3D foundation models to enhance the robustness of C-SLAM and its performance across a wider range of environments and multi-robot missions. We evaluate the performance of our approach against state-of-the-art decentralized C-SLAM algorithms on several multi-robot dataset sequences. Our results demonstrate that the powerful representations generated by 3D foundation models enable substantial improvements in localization accuracy while reducing computational and memory overhead through specialized optimization and keyframe sparsification. This makes our approach a promising solution for large-scale multi-robot deployments in unknown environments, where inter-robot collaboration is crucial for efficient exploration and mapping.

II. Background and Related Works

In C-SLAM, multiple robots work together to build a shared map and localize within it. By sharing sensor data and detecting overlap between their maps, the robots can improve their individual localization and create a globally consistent view of the environment across the robots. In this section, we present the background and related work on C-SLAM, as well as on image-based relative-pose estimation.

1) Collaborative SLAM: In C-SLAM, robots typically perform SLAM individually and then share information about their maps to fuse them into a globally consistent estimate of the traversed environment. Similar to single-robot SLAM, C-SLAM is typically divided in two parts: the front-end, which is responsible for feature extraction and data association, and the back-end, which manages state estimation [4].

Preprint submitted to IEEE Robotics and Automation Letters as Submission for RA-L only
Received July 21, 2025 20:06:20 Pacific Time

One of the most challenging task of the front-end is to efficiently detect and compute inter-robot loop closures within the overlapping regions. These loop closures correspond to connections between independent robots' estimates that can be discovered when the same places are visited by different robots. They serve as stitching points to merge local maps into a global representation of the environment. To efficiently merge large maps, loop closure detection is typically performed in two stages: place recognition, to identify possible map overlaps, followed by the registration to compute the 3D relative pose between the individual overlaps.

The back-end then estimates the most likely poses and map based on measurements collected from all robots. Our work focuses on pose-graph formulations of SLAM, where features are marginalized into inter-pose measurements, as this approach is generally more efficient for larger maps [4].

However, the perceptual aliasing phenomenon, where distinct places are mistaken for the same location, can cause front-end techniques to produce spurious measurements. Several methods have been proposed to address this pervasive issue in SLAM, such as Pairwise Consistency Maximization [5], or Graduated Non-Convexity (GNC) [6].

In this work, we specifically focus on key challenges associated with decentralized C-SLAM, specifically generating pairwise inter-robot loop closures and maintaining scale consistency, all without exchanging complete maps. A number of decentralized C-SLAM systems have been developed in recent years. DSLAM [7] was an early approach that leveraged compact learned descriptors to enable efficient distributed place recognition in the front-end. Subsequent systems, such as Kimera-Multi [8], extended this work by incorporating robust distributed back-end solvers. Another system, Swarm-SLAM [9] built upon these advances by introducing a sparsification strategy that prioritizes inter-robot loop closures, significantly improving front-end efficiency. Most recently, DVM [10], released concurrently with this paper, is the first system for decentralized monocular C-SLAM that executes both its front-end and back-end entirely onboard each agent.

A. Relative Pose Estimation

Producing globally consistent maps in C-SLAM heavily relies on the ability to accurately perform relative pose estimation, which involves performing 3D registration between two keyframes from different robots.

The geometric methods usually employed in monocular setups [11] can only estimate relative poses up-to-scale, meaning the absolute metric scale of the transformation remains unknown. While scale information for consecutive images can be recovered by combining image-based estimates with motion sensors such as IMUs [12], these sensors cannot be used to recover the scale of relative poses between non-consecutive images. Moreover, compared to consecutive images, non-consecutive ones typically exhibit larger viewpoint differences, where traditional keypoint-based matching methods—such as those relying on hand-crafted features—often struggle. Recent advancements in monocular depth estimation – such as DPT [13] – offer promising alternatives for scale estimation. However, despite their potential, these methods often require domain-specific fine-tuning, which can limit their generalizability when mapping unknown environments.

To overcome viewpoint limitations, learning-based methods like SuperPoint [14] and SuperGlue [15] use deep-learning techniques to enhance keypoint-matching robustness, incorporating reasoning across the entire image to improve performance. The field has also introduced new datasets and benchmarks, such as the Map-Free challenge [16] which focuses on scenarios with drastic viewpoints and illumination changes, requiring the emergence of new techniques to succeed, such as MicKey [17] and DUSt3R [2]. The first predicts metric correspondences directly in 3D camera space instead of the usual 2D pixel space, while DUSt3R reformulates the image matching problem as a pairwise 3D reconstruction task, predicting and aligning 3D pointmaps to estimate relative poses. Extending DUSt3R, MASt3R [3] regresses local features and explicitly trains for pairwise matching.

Building on these recent advances, MAST3R-SfM [18] and MAST3R-SLAM [19] integrate these 3D foundation models into complete mapping and localization frameworks. Similar to traditional SfM and SLAM systems, these methods reduce computational complexity by selecting keyframes instead of processing all images, constructing a graph where keyframes with overlapping viewpoints are linked—analogous to the pose graphs commonly used in SLAM. However, while highly effective in the single-robot case, these approaches are less effective for multi-robot SLAM because they rely on monolithic global optimization, jointly refining poses and scene geometry. This involves centralized processing, where all keyframe images must be collected and processed on a central server, which requires continuous high-bandwidth communication with all robots. Moreover, in multi-robot settings, the number of keyframes and constraints, from odometry and inter-robot loop closures, grows faster than with a single robot, making real-time onboard computation with large models impractical.

In contrast, our approach is specifically designed for the constraints of decentralized systems. By leveraging 3D foundation models for robust inter-robot loop closures from just a pair of images, we avoid large map exchanges between robots. Instead of a monolithic global optimization, our method distributes the expensive encoding step and enables optimization to be handled by a dynamically elected robot, as in [9], thus respecting the communication and computation bottlenecks inherent to decentralized operation.

III. Method

Our pose-graph visual C-SLAM approach assumes that each robot has access to a pre-existing localization source with correct metric scale—such as those obtained from visual-inertial [12], stereo [20], or LiDAR-based [21] systems. We define the input pose estimates of a robot α as $T_{\alpha,0}, \ldots, T_{\alpha,n} \in \mathrm{SE}(3)$, where $T_{\alpha,i}$ represents the pose estimate of robot α at each of the $i \in n$ keyframes along its trajectory. Alongside each keyframe, our approach also takes as input the corresponding image frame I_i^α, which will be used for inter-robot loop-closure detection.

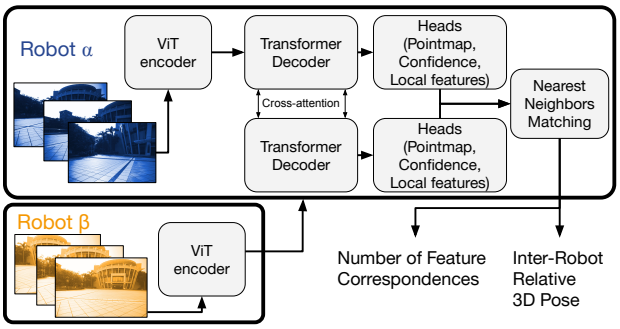


Fig. 1: Illustration of our inference pipeline, highlighting the inter-robot processing of loop closures.

A. Place Recognition

The first step of our pipeline is to perform place recognition between the maps of different robots. By identifying locations visited by two or more robots, we can create inter-robot loop closures that link the individual robot pose graphs into a consistent, shared localization estimate. For place recognition, while [18], [19] utilize MASt3R-encoded features with Aggregated Selective Match Kernels [22], we chose CosPlace [23] for its strong out-of-the-box performance. It enables us to demonstrate state-of-the-art C-SLAM using pretrained models without any calibration needed to adapt to a new domain, a key advantage for deployment in uncontrolled environments.

When two or more robots are within communication range, the robots exchange their compact CosPlace descriptors with one another. Once a robot α has received descriptors from a neighboring robot β, it compares them with its own descriptors using cosine similarity. For each pair of keyframes (I_i^α, I_j^β), we compute the cosine similarity score $s_{\beta,j}^{\alpha,i}$. The best match for each keyframe is determined through a nearest neighbor search and, if the best match similarity exceeds a threshold, the corresponding pair of keyframes (I_i^α, I_j^β) is considered an inter-robot loop-closure candidate. These candidates are then passed to the subsequent registration step, where the relative pose $T_{\beta,j}^{\alpha,i}$ between the two keyframes is computed.

B. Registration

While previous techniques [8], [9] relied on 3D registration using hand-crafted image features that are highly sensitive to viewpoint changes, we leverage recent foundation models for 3D image matching [2], [3]. These models can infer relative poses between images from significantly different viewpoints, enabling the detection of more inter-robot loop closures.

Whereas traditional C-SLAM methods use stereo, RGB-D, or LiDAR data for relative pose estimation with accurate scale, we rely solely on monocular images and resolve the scale ambiguity at a later stage (see Section III-C). For registration, each image of a loop closure candidate (I_i^α, I_j^β) is encoded locally on its corresponding robot into a compressed latent vector, using the pre-trained and frozen ViT encoder of the MASt3R model [3], that can later be decoded to reconstruct the scene and estimate the relative pose, as illustrated in Fig. 1. The encoding of I_j^β is shared with robot α for decoding, feature matching, and inter-robot relative pose inference. This process yields a relative pose measurement $T_{\beta,j}^{\alpha,i}$ along with the number of feature correspondences between the two frames.

As MASt3R is prone to occasional failures [3], we introduce a novel confidence estimation metric. To compute it, robot α also performs MASt3R inference on a consecutive pair of its own images, $(I_{i-1}^\alpha, I_i^\alpha)$, for which a reliable odometry estimate exists. We then compute the ratio $r_{\beta,j}^{\alpha,i}$ of feature correspondences between the inter-robot loop closure pair (I_i^α, I_j^β) and the intra-robot odometry pair $(I_{i-1}^\alpha, I_i^\alpha)$:

$$r_{\beta,j}^{\alpha,i} = \frac{|(I_i^\alpha, I_j^\beta) \text{ feature correspondences}|}{|(I_{i-i}^\alpha, I_i^\alpha) \text{ feature correspondences}|} \tag{1}$$

Since an odometry pair typically has high overlap and many correspondences, this ratio $r_{\beta,j}^{\alpha,i}$ effectively normalizes the confidence of a loop closure match.

We use this ratio first to filter out failed registrations that fall below a threshold R_{thr}, and second to weight successful measurements in the pose graph. For successful loop closures, we map the ratio to a probability $p_{\beta,j}^{\alpha,i}$ using a logistic function:

$$p_{\beta,j}^{\alpha,i} = \left[1 + \exp(-k \cdot (r_{\beta,j}^{\alpha,i} - 1))\right]^{-1}. \tag{2}$$

The probability and its parameter k can be manually tuned, or learned if training data from the deployment or a similar environment is available a priori [24].

The confidence probability is then multiplied by the information matrix associated with each measurement. The information matrix acts as a weight during pose graph optimization, ensuring that high-confidence loop closures have a greater influence on the final solution than low-confidence ones. For each inter-robot loop closure, we define the cost function $\phi_{\beta,j}^{\alpha,i}$:

$$\phi_{\beta,j}^{\alpha,i} = \left\| T_{\alpha,i}^{-1} \cdot T_{\beta,j} - \bar{T}_{\beta,j}^{\alpha,i} \right\|^2_{\Omega_{\beta,j}^{\alpha,i}(p_{\beta,j}^{\alpha,i})} \tag{3}$$

where $\overline{T}_{\beta,j}^{\alpha,i}$ is the relative pose measurement, and $\Omega_{\beta,j}^{\alpha,i}$ is the measurement information matrix which is proportional to the confidence $p_{\beta,j}^{\alpha,i}$. This cost function can then be incorporated into a global nonlinear least-squares problem alongside odometry constraints for multi-robot pose graph optimization.

C. Multi-Robot Pose Graph And Loop Scale Optimization

In our framework, multi-robot pose graph optimization ensures that the maps and estimated trajectories of all robots are aligned within a common reference frame, providing a unified representation of the environment. The optimization problems are formulated as a factor graph, where the variables represent the unknown quantities (e.g., the robot poses), and the factors define functions over subsets of these variables.

a) Base Multi-Robot Factor Graph: The base optimization problem consists primarily of two types of factors: odometry factors, which link consecutive keyframe poses, and loop closure factors, which link non-consecutive keyframes. Odometry cost functions for robot α are defined as follows:

$$\phi_{\alpha,i}^{\alpha,i-1} = \left\| T_{\alpha,i-1}^{-1} \cdot T_{\alpha,i} - \bar{T}_{\alpha,i}^{\alpha,i-1} \right\|^2_{\Omega_{\text{odom}}} \tag{4}$$

where $\bar{T}^{\alpha,i-1}_{\alpha,i}$ is the relative pose measurement, and Ω_{odom} is the corresponding information matrix.

Thus, for the multi-robot optimization problem, we minimize the sum of all cost functions:

$$\boldsymbol{T}^* = \underset{\boldsymbol{T}}{\operatorname{argmin}} \sum_{\alpha\in\mathcal{R}} \sum_{i\in(1:n_\alpha)} \phi^{\alpha,i-1}_{\alpha,i} + \sum_{(\alpha,i),(\beta,j)\in L_{\mathcal{R},\mathcal{R}'}} \phi^{\alpha,i}_{\beta,j} \quad (5)$$

where $\mathcal{R}$ denotes the set of all robots, n_α represents the number of keyframes for robot α, $L_{\mathcal{R},\mathcal{R}'}$ is the set of loop closures linking different robots, and $\boldsymbol{T}$ is the set of all poses forming the robots trajectories. The set of robots $\mathcal{R}$ in Eq. (5) can involve more than two robots; in fact, we perform the optimization with all neighboring robots, detecting and incorporating inter-robot loop closures into the optimization problem for each pair. Also, we do not consider intra-robot loop closures $\phi^{\alpha,i}_{\alpha,k}$, as we aim to isolate the effects of inter-robot loop closures, but they could straightforwardly be integrated into the formulation as an additional cost function.

The challenge with the relative poses measured with MASt3R is that their scaling is often inconsistent or very imprecise [18], [19]. Thus, to properly integrate these measurements into the optimization problem, we must either determine the scale in advance using other sensors or, as we propose here, treat the loop closure scale as an optimization variable. Naïvely, we can leverage the correctly scaled odometry measurements between the two odometry poses to estimate the loop closure scale. Assuming that the same scaling factor applies to both odometry and loop closure:

$$\bar{t}^{\alpha,i}_{\beta,j} = \frac{\left\|\bar{t}^{\alpha,i-1}_{\alpha,i}\right\|}{\left\|t^{\alpha,i-1}_{\alpha,i}\right\|} \cdot t^{\alpha,i}_{\beta,j} \quad (6)$$

where $t^{\alpha,i-1}_{\alpha,i}$ and $t^{\alpha,i}_{\beta,j}$ are the relative translations output by MASt3R, and $\bar{t}^{\alpha,i-1}_{\alpha,i}$ is the known translation from odometry. While we cannot guarantee that the loop closure and odometry share the same scaling factors, this often provides a reasonable initial guess, as discussed in Section IV-C.

b) Independent Scales Multi-Robot Factor Graph: In this paper, we go further and propose to explicitly optimize the loop closures' scale. For this, we draw inspiration from [25], which introduced scaling factors for pedestrian trajectories estimated via IMU dead-reckoning and refined using sporadic UWB distance measurements. While this work focused on rescaling odometry estimates, we instead apply scaling to the non-consecutive relative pose measurements.

As a first step, we decompose the measured relative pose $\bar{T}^{\alpha,i}_{\beta,j} \in \mathrm{SE}(3)$, from MASt3R, associated with a loop closure as $\bar{R}^{\alpha,i}_{\beta,j} \in \mathrm{SO}(3)$, and $\bar{t}^{\alpha,i}_{\beta,j} \in \mathbb{R}^3$, $s^{\alpha,i}_{\beta,j}$.

where $\bar{R}^{\alpha,i}_{\beta,j}$ is the measured relative rotation matrix, $\bar{t}^{\alpha,i}_{\beta,j}$ is the measured translation vector between the two poses, and $s^{\alpha,i}_{\beta,j}$ adjusts the magnitude of the translation vector to the correct scale. Together, the scaled relative pose $\hat{T}^{\alpha,i}_{\beta,j}$ is:

$$\hat{T}^{\alpha,i}_{\beta,j} = \begin{bmatrix} \bar{R}^{\alpha,i}_{\beta,j} & s^{\alpha,i}_{\beta,j} \cdot \bar{t}^{\alpha,i}_{\beta,j} \\ 0 & 1 \end{bmatrix} \quad (7)$$

We then define a new loop closure cost function that incorporates the scale value:

$$\hat{\phi}^{\alpha,i}_{\beta,j} = \left\| T^{-1}_{\alpha,i} \cdot T_{\beta,j} - \hat{T}^{\alpha,i}_{\beta,j} \right\|^2_{\Omega^{\alpha,i}_{\beta,j}(p^{\alpha,i}_{\beta,j})} \quad (8)$$

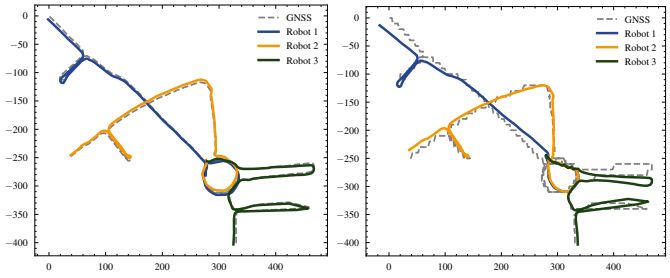
To optimize this novel factor, we need to provide the corresponding derivatives. Specifically, since we use GTSAM [26] as our factor graph optimization framework, we provide the analytical measurement Jacobian matrices for efficient computation, which are evaluated in the tangent space at the current estimate during optimization: $\boldsymbol{H}_{T_{\alpha,i}} = -\mathrm{Adj}(\mathrm{inv}(T^{-1}_{\alpha,i} \cdot T_{\beta,j}))$, $\boldsymbol{H}_{T_{\beta,j}} = \boldsymbol{I}$, and $\boldsymbol{H}_s = [\boldsymbol{0} - \bar{t}^{\alpha,i}_{\beta,j}]^\top$. Using $\hat{\phi}^{\alpha,i}_{\beta,j}$ instead of $\phi^{\alpha,i}_{\beta,j}$ in Eq. (5) we derive the optimization problem where the scale of each loop closure is treated independently.

c) Smoothed Scales Multi-Robot Factor Graph: We also introduce a third formulation, which includes an additional factor that links scale factors from related loop closures to ensure they remain similar. We cluster the loop closures that link relative poses fewer than ten keyframes apart. These clusters typically occur when the robot revisits a specific area for an extended period, resulting in multiple loop closures being detected within a single large overlap between the maps. Previous work have explored identifying such clusters to detect outliers among loop closure measurements [27]. In our case, the intuition behind linking them is that, since MASt3R is data-driven, relative poses inferred from a similar image domain might exhibit similar scaling factors.

The cost function to link the scale values, along with its corresponding Jacobians, are defined as $\phi_{s_{i,j}} = \|s_j - s_i\|^2_{\Omega_s}$, $\boldsymbol{H}_{s_i} = -1$, and $\boldsymbol{H}_{s_j} = 1$. where the scale residual $\phi_{s_{i,j}}$ is defined as the difference between scale estimates s_i and s_j from two different loop closures, ensuring consistent scale across the loop closure cluster. The corresponding Jacobians, $\boldsymbol{H}_{s_i}$ and $\boldsymbol{H}_{s_j}$, are the partial derivatives of $\phi_{s_{i,j}}$ with respect to s_i and s_j, respectively. For completeness, in our experimental analysis we also consider the scenario where all loop closures share the same scale value. Our formulation is designed for decentralized execution. Following [9], the optimization is performed on a dynamically elected robot when a group of robots meet, with the resulting estimates broadcast back to the neighbors. Fully distributed solvers for our scale-aware formulations are a promising direction for future work.

IV. Experiments

We conducted our experiments on dataset sequences using a robot onboard computer NVIDIA Jetson AGX Xavier (32GB). We benchmark FLOCC-SLAM against the open-source Swarm-SLAM [9] in two configurations: stereo and LiDAR. In the stereo configuration, place recognition was performed using CosPlace [23], and pose registration was carried out with PnP and RANSAC [20]. In the LiDAR configuration, place recognition utilized ScanContext [28], and pose registration was done using TEASER++ [29]. We selected Swarm-SLAM as a strong baseline due to its use of 3D sensing, allowing us to demonstrate that our monocular-only loop closing approach can achieve comparable or even superior results without relying on 3D data.



(a) FLOCC-SLAM (b) Swarm-SLAM LiDAR

Fig. 2: S3E Dormitory trajectory estimates with FLOCC-SLAM versus Swarm-SLAM in LiDAR configuration.

For a thorough and fair evaluation, we also configured Swarm-SLAM with several back-end optimizers: the fast Levenberg-Marquardt solver (LM) [26], the robust GNC optimizer [6], which can detect and reject outliers among the loop closures, and Riemannian Block Coordinate Descent (RBCD) [8], a distributed solver that incorporates GNC for outlier rejection. Notably, Swarm-SLAM with RBCD in stereo mode closely resembles Kimera-Multi [8], except for place recognition—Kimera-Multi uses DBoW2 [30], while Swarm-SLAM employs CosPlace [23].

For FLOCC-SLAM, we tested three different factor graph formulations: base multi-robot pose graph, with independent scale values (**IS**), and with smoothed scale values (**SS**). Unless otherwise specified, the base formulation was used. Since the base formulation does not optimize scale, it relies only on the scale initialization technique presented in Eq. (6). For place recognition, we set the similarity threshold $s^{\alpha,i}_{\beta,j}$ to a permissive value of 0.1, as our pipeline robustly handles false positives in the subsequent registration stage.

We first evaluated the techniques on the S3E dataset [31], which features three robots navigating large, dynamic indoor and outdoor environments. We selected the most challenging complete sequences with minimal overlap and differing viewpoints. Additionally, we tested on the GrAco dataset [32], which involves six robots in a large outdoor environment. An ablation study was performed using the GrAco dataset. For evaluation, we compared all our results against the provided ground truth: high-precision GNSS data (for outdoor sections, with reported accuracy of ±1cm [31], [32]) and motion capture systems (for indoor sections, with millimeter-level precision [31]), using the *evo* package [33]. In our experiments, all techniques share the same odometry estimates to isolate the impact of map merging.

A. Full System Performance

In Fig. 2, we illustrate FLOCC-SLAM's performance on the challenging S3E Dormitory sequence, demonstrating a closer alignment with the ground truth compared to the Swarm-SLAM baseline. The visualizations are supported by the detailed results presented in Table I, where we report the *Average Translation Error* (ATE), the number of loop closures N, and the computation time required by the factor graph solver for each technique on four sequences.

The best performance was achieved by FLOCC-SLAM using the smoothed scales formulation (**SS**) optimized with the **LM** solver. This approach generated orders of magnitude more loop closures, significantly enhancing the accuracy of the resulting solution. To identify successful matches, we applied a correspondences ratio threshold R_{thr} of 0.3. While some outliers might have passed through, their effects were mitigated by the ratio-based confidence mechanism (see Eq. (2)), without needing a more computationally expensive robust solver like GNC. Even though a few accurate loop closures may suffice for map merging, our approach's ability to generate more loop closures is a significant advantage, especially in scenarios where potential loop closures are scarce. In addition, directly addressing the scale ambiguity—without relying on approximate scale estimates or costly outlier rejection mechanisms—significantly enhances performance.

The improved accuracy of FLOCC-SLAM with smoothed scales (SS) indicates that loop closure scales within clusters may indeed be correlated. This makes the additional computation time versus IS—observed in cases like the Dormitory sequence—a worthwhile trade-off when resources allow. Notably, on the S3E Dormitory sequence, the stereo Swarm-SLAM configuration was unable to merge all three trajectories due to the challenges of detecting loop closures from opposite viewpoints using traditional stereo matching techniques. Similarly, the LiDAR configuration of Swarm-SLAM performed poorly on this sequence due to significant perceptual aliasing in the structurally repetitive dormitory hallways.

To further analyze our proposed approach, the following subsections present an ablation study we conducted to evaluate the effectiveness and impact of the various novel components.

B. Relative Pose Estimation Accuracy and Robustness

In our first ablation experiment, we used the GrAco dataset [32], which involves six robots, and extracted 88,096 image pairs that are less than 10 meters apart according to GNSS data. We then applied our registration pipeline to each pair to obtain the relative poses.

The first issue we investigated was determining the validity of the matches. Due to perceptual aliasing, place recognition sometimes fails, incorrectly identifying different locations as the same. This challenge is exacerbated by DUSt3R [2], and subsequently MASt3R [3], which are now capable of matching images from almost any viewpoint with confidence, making it harder to verify which image matches correspond to valid relative poses and should be included as inter-robot loop closures in the pose graph. In Fig. 3, we evaluate the precision and recall of five different metrics. A match is considered an outlier if its translation error compared to the ground truth exceeds 2 meters, with the measurements scaled using ground truth data. We found that relying solely on CosPlace similarity is insufficient to distinguish registration inliers from outliers. Alternatively, we considered metrics based on MASt3R's outputs. We considered the average confidence produced by MASt3R and computed the confidence ratio between the loop match and the odometry match. We also considered the number of feature correspondences and the correspondences ratio.

TABLE I: Average translation error (ATE), number of loop closures (N) and optimization time on S3E sequences.

		Campus			Teaching			Square			Dormitory		
		ATE (m)	N	Time (s)	ATE (m)	N	Time (s)	ATE (m)	N	Time (s)	ATE (m)	N	Time (s)
FLOCC-SLAM	GNC	19.25 ± 6.81	1122	27.45	2.40 ± 0.76	2274	41.58	9.53 ± 3.87	226	9.25	46.00 ± 30.36	175	34.79
	IS-LM	5.50 ± 2.16	1122	2.46	1.87 ± 0.65	2273	3.46	5.64 ± 2.46	226	1.41	3.84 ± 1.35	175	1.50
Monocular loop closures	SS-LM	**5.41 ± 2.13**	1122	1.82	**1.84 ± 0.67**	2273	4.74	**5.07 ± 2.56**	226	1.63	**2.41 ± 0.99**	175	11.05
Swarm-SLAM	RBCD	9.41 ± 7.59	29	286.63	3.59 ± 1.56	110	143.31	113.58 ± 80.99	6	120.33	7.41 ± 5.32	226	147.80
	GNC	10.50 ± 9.47	29	21.38	3.69 ± 1.52	110	13.97	164.83 ± 99.39	6	36.45	10.18 ± 5.47	226	26.14
LiDAR loop closures	LM	10.21 ± 8.97	29	1.23	3.85 ± 1.63	110	0.99	165.94 ± 98.98	6	1.73	8.34 ± 4.82	226	4.02
Swarm-SLAM	RBCD	6.79 ± 7.08	16	744.65	2.23 ± 0.94	27	840.53	42.97 ± 25.27	7	2.46		✗	
	GNC	6.56 ± 7.07	16	48.64	2.25 ± 1.15	27	23.85	9.73 ± 6.67	7	15.34		✗	
Stereo loop closures	LM	6.56 ± 7.07	16	2.90	2.20 ± 1.05	27	3.91	33.37 ± 22.98	7	1.27		✗	

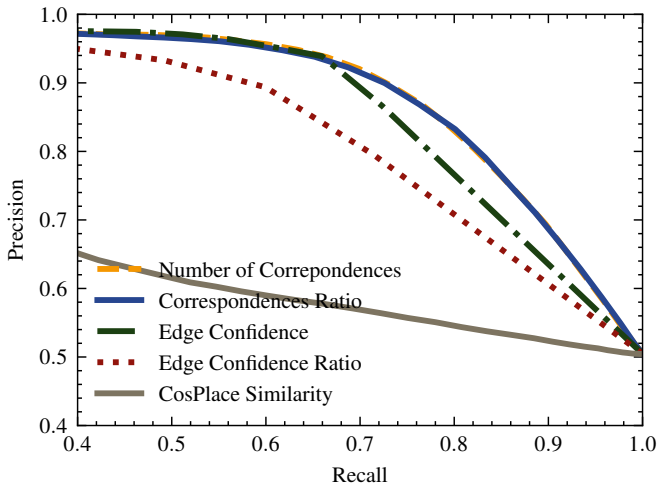


Fig. 3: Precision-Recall curves for various inlier/outlier detection techniques. MASt3R-based correspondences combined with the loop-to-odometry ratio yield the best performance.

The results show that both the number of correspondences and the correspondences ratio provided strong performance, with the ratio being preferable due to its unitless nature, making it easier to tune compared to the number of correspondences, which is affected by the image size and field of view.

C. Multi-Robot Scale and Pose Graph Optimization

In the following experiments, we evaluate the full pose graph solution involving the full 6-robot GrAco sequence. We mapped the feature correspondences ratio to a confidence metric (see Eq. (2)) to weight the inter-robot loop closures during pose graph optimization.

In Tables II and III, we analyze the effects of different factor graph formulations, and the impact of the odometry backbone. We tested two odometry backbones: VINS-Mono [12], making the system fully monocular, and the more accurate LiDAR-based LIO [21], to demonstrate our approach's effectiveness when the odometry has near-perfect scale. While monocular odometry yielded reasonable results, inaccuracies in scale at various parts of the trajectory—particularly at the start, where the online scale estimation had not yet converged—led to reduced performance. Since the GrAco dataset lacks VIO calibration sequences, all robot trajectories were affected by these initial scale inaccuracies. In real-world scenarios, proper calibration remains critical for effective VIO deployments. Results from our approach combined with the LIO-based solution demonstrate impressive accuracy, with errors below 4 meters over 3.5 kilometers of trajectories.

For each backbone, we compared the formulations previously introduced. For completeness, we also report results for both the non-robust LM solver and the robust GNC solver for each of our factor graph formulations. Additionally, we report the computation time on both our robot onboard computer, and a desktop server equipped with an AMD Ryzen 7 3700X CPU and an NVIDIA RTX 3070 GPU. This comparison is interesting given that some other existing C-SLAM approaches offload state estimation to servers [34]. Note that the pose graph optimization runs on CPU, while both place recognition and registration are perfomed on GPU. We provide each result with different scale initialization techniques, whether using ground truth, directly the raw output from MASt3R, or our odometry-based scaling (see Eq. (6)). The results in Tables II and III confirm that our formulations with independent or smoothed scales achieve the best—or near-best—performance, without the need for the expensive robust optimization of GNC. This shows that the use of a robust solver offers limited benefits. In fact, it can decrease performance when paired with the smoothed scale formulation, potentially because the outlier rejection mechanism of GNC conflicts with our confidence-based weighting. Compared to the several minutes required by GNC, our optimization runs in under a second on the server and just a few seconds on the onboard computer.

D. Resource Efficiency

In addition to the computational efficiency provided by our scale-aware optimization compared to robust methods, our approach offers further efficiency advantages. Specifically, we split the registration into two stages where keyframes are encoded locally on each of the two robots involved in the loop closure, and decoding is done on only one of the robots, as outlined in Fig. 1. This way, a keyframe part of multiple loop closures only needs to be encoded once—particularly useful when managing a large number of loop closure candidates. In Fig. 4, we show the computation gain on the GrAco dataset against a naïve baseline where encoding of both images is

TABLE II: Comparison of different pose graph optimization formulations with a VINS-Mono odometry backbone.

	GT Scale			MASt3R Scale			Odometry Scale		
	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)
LM	4.76 ± 4.50	1.60	0.33	39.93 ± 44.43	3.63	0.91	39.59 ± 45.68	3.27	0.79
GNC	4.89 ± 4.51	31.09	10.28	11.98 ± 5.21	32.93	10.49	5.56 ± 4.07	35.86	11.78
IS-LM	6.99 ± 6.47	3.27	0.79	7.11 ± 6.81	3.24	0.78	7.89 ± 7.01	3.49	0.87
IS-GNC	6.34 ± 6.51	104.21	29.65	6.40 ± 6.82	96.54	27.38	7.86 ± 7.01	112.94	31.98
SS-LM	7.43 ± 6.93	3.89	0.94	8.23 ± 7.34	3.50	0.82	7.42 ± 7.06	3.95	0.94
SS-GNC	7.45 ± 7.06	166.24	45.99	91.50 ± 59.82	296.27	85.31	58.94 ± 62.61	305.20	85.38

TABLE III: Comparison of different pose graph optimization formulations with a LIO odometry backbone.

	GT Scale			MASt3R Scale			Odometry Scale		
	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)
LM	2.75 ± 1.36	1.44	0.29	10.36 ± 3.65	1.44	0.30	6.90 ± 3.95	1.51	0.30
GNC	2.73 ± 1.36	29.61	9.58	9.25 ± 2.98	29.12	9.56	2.95 ± 1.32	30.71	9.89
IS-LM	3.45 ± 2.03	3.16	0.74	3.14 ± 1.80	2.83	0.66	3.41 ± 1.96	3.01	0.70
IS-GNC	3.45 ± 2.00	93.70	26.45	3.12 ± 1.77	87.59	24.63	3.40 ± 1.94	94.17	26.31
SS-LM	2.92 ± 1.93	4.06	0.95	3.29 ± 1.94	3.42	0.80	3.11 ± 1.81	3.85	0.84
SS-GNC	2.93 ± 1.90	156.52	41.42	3.23 ± 1.92	139.69	37.32	13.89 ± 11.23	342.36	89.20

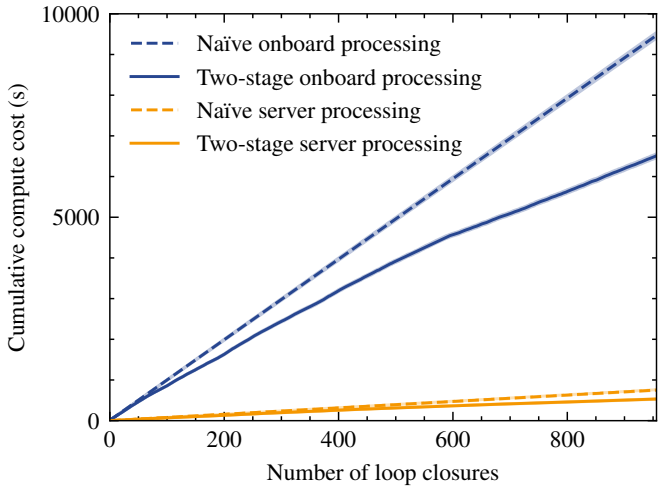


Fig. 4: Computational savings from two-stage MASt3R processing in C-SLAM. Splitting encoder and decoder processing reduces computation by avoiding redundant re-encoding of images with established loop closures. This is especially beneficial for onboard computers, where MASt3R processing is significantly more demanding than on a desktop server.

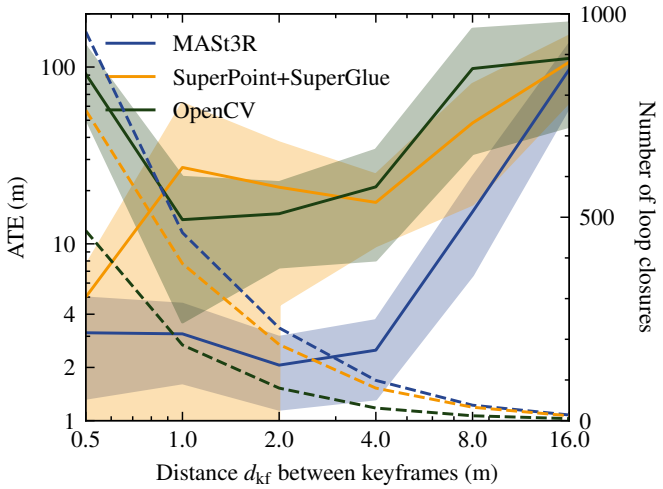


Fig. 5: Average Trajectory Error (ATE) versus keyframe distance, each registration technique is shown in a different color. Solid lines indicate the ATE with shaded regions representing standard deviation; dashed lines show the number of detected loop closures. Increasing keyframe distance reduces memory usage but typically degrades accuracy. Our approach maintains lower error, even with sparser keyframes.

done for all candidates. We can see that MASt3R inference is quite costly on a robot onboard computer (in blue) compared to a server with a larger GPU (in orange). This matches our observations for the pose graph optimization in Tables II and III, where onboard computing is approximately 3 to 4 times slower than the server. Thus, further gains could be achieved through more advanced load-sharing strategies between robots and servers.

Another resource efficiency gain comes from reducing the number of keyframes to process and store in memory. However, reducing keyframes typically leads to decreased accuracy as the mapping becomes more coarse and fewer loop closures can be detected. In Fig. 5, we assess how increasing the distance between keyframes—and thus proportionally reducing their number—affects the ATE of the solution. We also compared this with loop closures computed using SuperPoint+SuperGlue and OpenCV. While other methods show a rapid performance decline as the keyframe distance increases, our approach maintains low ATE values even with distances of up to 4 meters. This demonstrates its ability to support sparser and more memory-efficient C-SLAM solutions without sacrificing accuracy. The efficiency gains of using sparser keyframes extend beyond memory savings, as sparse maps can also be compared—via place recognition and registration—using less communication bandwidth and in less time. High-accuracy sparse maps are beneficial for

deployments on small robots or consumer electronics.

V. Conclusion

In this paper, we introduced FLOCC-SLAM, a novel decentralized collaborative SLAM approach that leverages 3D foundation models to address the challenges of multi-robot loop closure detection. By incorporating monocular pose estimation and scale optimization, FLOCC-SLAM effectively improves inter-robot loop closure computation in scenarios where current methods struggle due to large differences in robot viewpoints. Our experimental results, and ablation study, demonstrate that FLOCC-SLAM outperforms state-of-the-art approaches, particularly in terms of accuracy. Furthermore, we showed that when accurate odometry with the correct scale is available, it is possible to easily and efficiently integrate up-to-scale loop closures in multi-robot pose graph optimization. Our approach, however, relies on the availability of metric-scale odometry from the single-robot systems to anchor the scale of the global map. It also assumes the odometry is stable enough to provide a reliable baseline for our confidence metric.

Looking ahead, there is considerable potential for tighter integration between 3D foundation models and C-SLAM systems to improve measurement accuracy and the overall representation of explored environments. Progress in this area would also be greatly accelerated by the creation of more diverse, large-scale C-SLAM datasets that feature varied sensor types and challenging, realistic multi-robot trajectories.

References

[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[2] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "DUSt3R: Geometric 3D Vision Made Easy," in *Proceedings of the IEEE/CVF CVPR Conference*, 2024, pp. 20 697–20 709.

[3] V. Leroy, Y. Cabon, and J. Revaud, "Grounding Image Matching in 3D with MASt3R," in *Proceedings of ECCV 2024: 18th European Conference on Computer Vision*, Nov. 2024, pp. 71–91.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[5] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise Consistent Measurement Set Maximization for Robust Multi-Robot Map Merging," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2916–2923.

[6] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, Apr. 2020.

[7] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-Efficient Decentralized Visual SLAM," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2466–2473.

[8] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, Aug. 2022.

[9] P.-Y. Lajoie and G. Beltrame, "Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 475–482, Jan. 2024.

[10] J. Bird, J. Blumenkamp, and A. Prorok, "DVM-SLAM: Decentralized Visual Monocular Simultaneous Localization and Mapping for Multi-Agent Systems," Mar. 2025.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. USA: Cambridge University Press, 2003.

[12] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[13] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision Transformers for Dense Prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 12 159–12 168.

[14] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *2018 IEEE/CVF CVPR Conference Workshops*, Jun. 2018, pp. 337–33 712.

[15] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 4937–4946.

[16] E. Arnold, J. Wynn, S. Vicente, G. Garcia-Hernando, Á. Monszpart, V. Prisacariu, D. Turmukhambetov, and E. Brachmann, "Map-Free Visual Relocalization: Metric Pose Relative to a Single Image," in *Computer Vision – ECCV 2022*, 2022, pp. 690–708.

[17] A. Barroso-Laguna, S. Munukutla, V. A. Prisacariu, and E. Brachmann, "Matching 2D Images in 3D: Metric Relative Pose from Metric Correspondences," in *Proceedings of the IEEE/CVF CVPR Conference*. IEEE Computer Society, Jun. 2024, pp. 4852–4863.

[18] B. Duisterhof, L. Zust, P. Weinzaepfel, V. Leroy, Y. Cabon, and J. Revaud, "MASt3R-SfM: A Fully-Integrated Solution for Unconstrained Structure-from-Motion," Sep. 2024.

[19] R. Murai, E. Dexheimer, and A. J. Davison, "MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors," Dec. 2024.

[20] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[21] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5135–5142.

[22] G. Tolias, T. Jenicek, and O. Chum, "Learning and Aggregating Deep Local Descriptors for Instance-Level Recognition," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, Proceedings, Part I*, Aug. 2020, pp. 460–477.

[23] G. Berton, C. Masone, and B. Caputo, "Rethinking Visual Geo-Localization for Large-Scale Applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4878–4888.

[24] Y. Tian, K. Khosoussi, and J. P. How, "A resource-aware approach to collaborative loop-closure detection with provable performance guarantees," *The International Journal of Robotics Research*, Sep. 2020.

[25] P.-Y. Lajoie, B. H. Baghi, S. Herath, F. Hogan, X. Liu, and G. Dudek, "PEOPLEx: PEdestrian Opportunistic Positioning LEveraging IMU, UWB, BLE and WiFi," in *ICC 2024 - IEEE International Conference on Communications*, Jun. 2024, pp. 3518–3523.

[26] F. Dellaert et al., "Georgia Tech Smoothing And Mapping (GTSAM)," http://gtsam.org/.

[27] F. Wu and G. Beltrame, "Cluster-based Penalty Scaling for Robust Pose Graph Optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6193–6200, Oct. 2020.

[28] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4802–4809.

[29] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[30] D. Galvez-López and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[31] D. Feng, "S3E: A Multi-Robot Multimodal Dataset for Collaborative SLAM," Jul. 2024.

[32] Y. Zhu, Y. Kong, Y. Jie, S. Xu, and H. Cheng, "GRACO: A Multimodal Dataset for Ground and Aerial Cooperative Localization and Mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 966–973, Feb. 2023.

[33] M. Grupp, "Evo: Python package for the evaluation of odometry and SLAM." 2017.

[34] P. Schmuck, T. Ziegler, M. Karrer, J. Perraudin, and M. Chli, "COVINS: Visual-Inertial SLAM for Centralized Collaboration," in *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2021, pp. 171–176.