A Framework for Collaborative Multi-Robot Mapping using Spectral Graph Wavelets

Lukas Bernreiter¹, Shehryar Khattak², Lionel Ott¹, Roland Siegwart¹, Marco Hutter² and Cesar Cadena¹

Abstract
The exploration of large-scale unknown environments can benefit from the deployment of multiple robots for collaborative mapping. Each robot explores a section of the environment and communicates onboard pose estimates and maps to a central server to build an optimized global multi-robot map. Naturally, inconsistencies can arise between onboard and server estimates due to onboard odometry drift, failures, or degeneracies. The mapping server can correct and overcome such failure cases using computationally expensive operations such as inter-robot loop closure detection and multi-modal mapping. However, the individual robots do not benefit from the collaborative map if the mapping server provides no feedback. Although server updates from the multi-robot map can greatly alleviate the robotic mission strategically, most existing work lacks them, due to their associated computational and bandwidth-related costs. Motivated by this challenge, this paper proposes a novel collaborative mapping framework that enables global mapping consistency among robots and the mapping server. In particular, we propose graph spectral analysis, at different spatial scales, to detect structural differences between robot and server graphs, and to generate necessary constraints for the individual robot pose graphs. Our approach specifically finds the nodes that correspond to the drift's origin rather than the nodes where the error becomes too large. We thoroughly analyze and validate our proposed framework using several real-world multi-robot field deployments where we show improvements of the onboard system up to 90% and can recover the onboard estimation from localization failures and even from the degeneracies within its estimation.

Keywords
Multi-Robot Mapping, Spectral Graph Theory

Introduction
Over recent years, an abundance of localization and mapping frameworks have been proposed and successfully deployed in various robotic scenarios. As part of this research, many traditional SLAM challenges have been fully or partially addressed. Despite this development, new challenges readily arise with the need for more robotic autonomy and the deployment of heterogeneous robotic teams in large-scale environments. In particular, an increase in the number of deployed robots and autonomy requires a higher degree of robustness and efficiency. At the same time, scalability and persistence across all systems become a pertaining issue. While it is difficult to maintain a consistent estimate of the environment across all employed systems, it is an essential prerequisite for operating robotic teams in applications like disaster response or search and rescue.

With the recent advent of high-bandwidth mobile networks such as 5G networks, collaborative robotic approaches have received increased attention in the robotics community due to their improved practical feasibility. A promising research direction is to employ a centralized mapping approach. Centralized servers running in the local network or a remote cloud environment have more computational capacity than individual robots. Therefore, they can perform expensive operations such as global optimizations, loop closing, and exploitation of all available sensor data to improve accuracy and overcome onboard failures.

Most collaborative mapping approaches focus on building accurate maps on the server and ignore the use of global multi-robot information to provide localization corrections to individual robots. Especially in centralized settings without feedback, pose estimation discrepancies may arise between robots during large missions leading to severe drift between robot and server maps resulting in increased optimization time at the server for collaborative mapping. Therefore, it is desirable to provide additional constraints to improve onboard estimation and collaborative mapping performance for large-scale multi-robot missions.

Furthermore, multi-robot missions often deploy a heterogeneous set of robots, e.g., aerial and ground robots, which additionally might rely on heterogeneous sensory systems. Carrying a diverse set of multi-modal sensors onboard and effectively utilizing different algorithms for, e.g., localization and mapping, can be highly beneficial for the deployment as it becomes more flexible and robust. However, no common layer sharing data to improve pose

¹Autonomous Systems Lab, ETH Zurich, Switzerland
²Robotics Systems Lab, ETH Zurich, Switzerland

Corresponding author:
Lukas Bernreiter, Autonomous Systems Lab, ETH Zurich, Zurich, Switzerland 8092.
Email: berlukas@ethz.ch
estimation and mapping estimates among the employed robots is readily available with diverse sensory systems. Hence, a sensor modality-invariant approach that can incorporate and communicate relevant consistency information among robots while maintaining low network bandwidth requirements is essential for large-scale multi-robot field deployments.

This paper proposes a novel multi-robot pose graph consistency approach independent of the underlying robot pose estimation processes. Our proposed approach relies only on a sparse abstraction of the estimated poses in $SE(3)$. Moreover, the framework operates in the graph spectral domain of the pose graphs to identify structural anomalies in the individual robot pose graphs using a multi-scale analysis. By examining the structural components of the pose graphs at different scales, our system identifies discrepancies in the local and coarser neighborhoods and adds corresponding constraints to improve the pose estimation accuracy of individual robots and make the individual robot and collaborative server maps consistent with each other. The key contributions of this paper are:

- Graph spectral analysis of pose graphs to identify discrepancies between onboard and server pose graphs.
- Automatic adaptive inference of multi-scale constraints to correct onboard estimation failures.
- Comparison against current state-of-the-art approaches on datasets and a thorough quantitative analysis on large-scale multi-robot field deployments presented to validate the proposed approach.

Related Work

In this section, we review the state-of-the-art collaborative multi-robot localization and mapping approaches as well as the current applications of graph signal processing and degeneracy and failure detection.

Collaborative Multi-Robot Mapping

Collaborative multi-robot approaches can be distinguished into centralized (Deutsch et al. 2016; Schmuck and Chli 2019; Karrer et al. 2018) and distributed solutions (Cunningham et al. 2013; Dong et al. 2015). Deutsch et al. (2016) proposed a vision-based centralized multi-robot SLAM approach where a mapping server performs loop closures and replaces robot pose graphs with corrected graphs. A similar approach was proposed by Schmuck and Chli (2019) in which robots send local maps to a mapping server which then returns optimized keyframes and landmarks to each robot to include in their onboard optimizations, thus increasing the bandwidth requirements for real-world robot deployments. The work of Van Opdenbosch and Steinbach (2019) proposes an encoding and decoding of visual features during the transmission of the maps to reduce the required bandwidth. CoSLAM (Zou and Tan 2013) proposes to make use of GPU computing to circumvent the need for large computational processes and improve the speed of onboard optimizing tasks, hence requiring a GPU onboard individual robots.

Different from vision-only approaches, LAMP (Ebadi et al. 2020; Chang et al. 2022) proposes a large-scale collaborative multi-modal SLAM framework. However, their proposed approach does not provide any pose corrections from the centralized server to the individual robots.

In contrast to centralized approaches, distributed approaches require each robot to run a full onboard SLAM solution (Dong et al. 2015) and share marginalized information with other robots (Cunningham et al. 2013), thus making full information available to each robot. Additionally, they have the advantage of scaling well to large swarms of robotic systems (Zieglar et al. 2021) but typically increase the onboard compute requirements significantly.

A crucial aspect of multi-robot SLAM is the ability to incorporate inter-robot loop closures. Kim et al. (2010) aims to achieve consistent maps across multiple robots independently of the employed sensing modalities by detecting loop closures between robots and connecting their pose graphs. In the same direction, Mangelson et al. (2018, 2019) aim to robustly select inter-robot loop closure candidates by maintaining pair-wise consistent measurements. More recently Lajoie et al. (2020) proposed a distributed system with distributed loop closure detection.

The more robots are deployed for a specific task, the more information needs to be processed, potentially leading to delays or longer processing times, especially for components such as the factor graph optimization. Recently, COVINS (Schmuck et al. 2021) demonstrated a collaborative deployment of 12 individual agents while maintaining a reasonable collaborative trajectory error. Although their system propagates optimized poses from the centralized server back to individual agents, the poses are only used for drift quantization by comparing the optimized to the onboard estimate. Thus, the onboard pose estimations are not corrected.

Concluding, many existing approaches are limited to a single modality only (Lajoie et al. 2020; Karrer et al. 2018; Deutsch et al. 2016) often incorporated in tightly coupled multi-robot frameworks, exchanging large data structures such as descriptors (Tian et al. 2022), partial or complete (Schmuck and Chli 2019) factor graphs. As a consequence, the systems become less flexible and maintain little versatility for the application of different robotic tasks. Conversely, this paper proposes to detect discrepancies between the robot graphs using spectral analysis and a sparse abstraction of the server graph to generate an individual set of constraints for each robot. Hence, the proposed approach achieves high accuracy and mapping consistency while maintaining low network and compute requirements.

Failure and Degeneracy Detection

Pose estimation from onboard sensors is subject to drift (accumulation of small errors) and to degeneracies (errors due to specific sensor modality’s deficiency). Recognizing such errors enables corrective actions to avoid possible catastrophic losses (e.g., platform crashes and wrong decision making). However, evaluating the quality of poses or maps is not trivial when no ground truth is available for comparison. In Schwertfeger and Birk (2013), a metric to assess the quality of the maps was proposed by matching topological graphs from the robot with a ground truth map. Some research also approaches the problem using redundant estimation systems (Sundvall and Jensfelt 2006) to find inconsistencies. Moreover, the recent work of Nobili et al. (2018) learns a...
A good introduction and overview of graph signal processing (2014) proposes an anomaly detection for spatial proximity (2019) infers a failure type based on the distribution of the residual error. The work of Zhang et al. (2016), proposes to analyze the structure of the constraints using the eigenvalues to derive a degeneracy factor.

We approach the problem differently by taking into account the underlying graph structure, precisely its spectral properties. Thus, making our approach independent of the employed sensor system and enabling us to evaluate the discrepancies at multiple scales to be more precise when resolving the spurious estimations.

**Graph Signal Processing**

Spectral graph theory is an active research area and has gained popularity in the past years in the context of robotics. Spectral graph theory approaches have been proposed for robotic mapping (Brusiskil et al. 2007), planning (Indelman 2018), and more recently, in combination with graph neural networks for various robotic tasks (Chandra et al. 2020; Moon and Lee 2020). In general, graphs are irregular structures and are capable of modeling large, complex, and distributed problems (Mateos et al. 2019), e.g. Egilmez and Ortega (2014) proposes an anomaly detection for spatial proximity of graph nodes using spectral graph filtering. Furthermore, graph signal processing aims at applying signal processing techniques on graph structures, thus allowing the use of existing concepts such as the Laplacian operator (Sandryhaila and Moura 2014) and multi-scale analysis (Hammond et al. 2011, 2019). Similarly, Donnat et al. (2018) aims to learn a multi-scale structural embedding using graph wavelets by treating the wavelet coefficients as a probability distribution. A good introduction and overview of graph signal processing are presented in Ortega et al. (2018).

Our approach also performs a structural analysis of graph signals to detect discrepancies between the onboard and server graphs. Using localized graph wavelets in the graph domain, our approach directly compares the trajectories at different scales to estimate the severity of the inconsistency of the individual pose graphs.

**Preliminaries**

This section introduces the fundamental and necessary concepts for analyzing and comparing graph structures in the graph spectral domain. We first introduce the underlying methods to use graphs for modeling complex problems. Next, the analysis of harmonic signals in the Euclidean and graph domain are covered.

**Fundamental Graph Theory Review**

In this work, we exploit the graph structure that serves as the primary foundation for most modern SLAM backends (Cadena et al. 2016). In particular, we extract the pose information of the factor graphs, i.e., disregarding any other sort of constraints to visual landmarks, GPS sensors, etc. Thus, we consider in this work, weighted undirected graphs $G = (\mathcal{E}, \mathcal{V}, w)$ consisting of a set of nodes $\mathcal{V}$ with cardinality $N$, edges $\mathcal{E}$ and weights $w : \mathcal{E} \to \mathbb{R}^+$ denoting how strong two nodes are connected with each other.

A graph $G$ is uniquely described by $\mathcal{E}$, $\mathcal{V}$ and $w$ in the form of a weighted symmetric adjacency matrix $A \in \mathbb{R}^{N \times N}$ with $A_{n,m} > 0$ if two nodes $n$ and $m$ are connected. The weight can be chosen freely, such as, for instance, the spatial proximity, or the number of co-observed landmarks between nodes, but ought to measure the relationship between the nodes. Another fundamental construct in graph theory, is the degree matrix $D$, defined as a diagonal matrix with entries $D_{n,n} = \sum_{n'} A_{n,n'}$ where $n'$ are all incident nodes of $n$.

Finally, signals in traditional signal processing are often expressed as functions over time, such as $x(t) : \mathbb{R} \to \mathbb{R}$, mapping a scalar value to each discrete point in time $t$. In a similar vein, signals on graphs are defined as $x(n) : \mathcal{V} \to \mathbb{R}$, associating a scalar value to each node $n$ in the graph. While a traditional signal $x(t)$ changes over time, a signal defined on a graph $x(n)$ alters between the nodes in the graph, leading to certain variations within the signal. Analyzing these signal variations and, consequently, their trends can lead to a more fundamental understanding of the signal’s nature and is generally termed spectral or frequency analysis.

---

Figure 1. Overview of our approach. We consider multiple robots simultaneously exploring an environment and sending incremental maps to a centralized mapping server. The server accumulates all robot maps and jointly optimizes them. A relaxation of the collaborative multi-robot map is sent back to the robots, where a multi-scale graph spectral analysis is performed to identify discrepancies onboard and server maps and to generate necessary constraints for making them consistent.
**Euclidean and Graph Spectral Analysis Review**

Integral transformations such as the Fourier transform project a signal onto a Hilbert space enabling the analysis of the characteristic properties of that signal. The data in the projected space is a compact representation denoting the information that is more or less prevalent in the input data. However, different aspects, such as the duration, bandwidth, and discretization, need to be carefully considered for a correct analysis of band-limited signals and to avoid unfavorable effects such as leakage and Gibbs ringing.

The Fourier transform is a fundamental tool for various applications in many different fields ranging from image processing to data science (Rao et al. 2010). Specifically, the Fourier transform expresses a signal \( x(t) \) as a sum of basis functions for which a typical choice is the complex wave: \( \exp(2\pi j ft) \).

Thus, the traditional Fourier transform is the inner product of the graph Fourier basis functions and, therefore, also to derive a Fourier transform in Eq. (2), the eigenvector \( \lambda x \) plays an important role w.r.t. the one-dimensional Laplace operator \( \Delta \) in \( \mathbb{R} \), i.e.

\[
\Delta \exp(2\pi j ft) = \frac{\partial^2}{\partial t^2} \exp(2\pi j ft) = 2j\pi f \frac{\partial}{\partial t} \exp(2\pi j ft) = -(2\pi f)^2 \exp(2\pi j ft).
\]

Based on the Helmholtz equation, i.e. \(-\Delta x = \lambda x\), the initial term \((2\pi f)^2\) corresponds to the eigenvalue while the later term \(\exp(2\pi j ft)\) denotes the eigenfunction.

The Laplace operator is a linear operation that expresses the divergence of a given function's gradient. Likewise, a similar intuition can be applied for deriving a Laplace operator for irregular graph domains. In concrete, the Laplace operator \(\Delta\) is replaced by the Laplacian matrix \(\mathcal{L}\) (Ricaud et al. 2019) measuring the variations within graphs. Generally, for an undirected graph with the two matrices \(A\) and \(D\), the Laplacian matrix can be obtained by

\[
\mathcal{L} = D - A,
\]

which is, by definition, a symmetric and positive semidefinite matrix. Hence, it can be decomposed into its eigenvalues \(\Lambda\) and eigenvectors \(U\), i.e. \(\mathcal{L} = U\Lambda U^\top\). Based on the findings in Eq. (2), the eigenvector \(U\) can be used to define the Fourier basis functions and, therefore, also to derive a Fourier transformation of graph signals. As a matter of fact, other constructions of the graph Laplacian \(\mathcal{L}\) such as the normalized graph Laplacian are also possible and are widely used in literature as different graph Fourier bases. We refer the interested reader to the work of Shuman et al. (2016) for a discussion on differences between the bases.

In particular, the graph Fourier transform of a graph function \(x\) is given by the expansion of \(x\) with the eigenvectors \(u_i\) of the graph Laplacian \(\mathcal{L}\) (cf. Eq. 3), i.e.

\[
X(\lambda_i) = \langle x(n), u_i(n) \rangle = \sum_n x(n)u_i^*(n),
\]

where \(\lambda_i\) is the \(i\)-th non-negative eigenvalue of \(\mathcal{L}\) and \(u_i^*\) is the complex conjugate of \(u_i\). The graph Fourier transform of a graph signal \(x\) is then given by \(X = U^\top x\). Moreover, the eigenvalues \(\Lambda\) are real values, and thus can be ordered and correspond to the graph frequencies, allowing a similar intuition as for traditional frequency analysis. Consequently, most of a graph signals energy is preserved in the lower bands of \(\Lambda\), and higher bands correspond to high oscillating frequencies.

The construction of the graph Fourier transform using Eq. 3 and Eq. 4 implies that all the spectral properties are given by the connection of the nodes. Thus, the modeling of the relationship between the individual nodes constitutes a crucial component of the system.

**Collaborative Multi-Robot Mapping**

This section details the proposed method, for which an overview is presented in Figure 1. Overall, the aim is to identify graph nodes with high drift that will lead to large errors and correct them with only a few constraints. The proposed approach comprises the following core components: (i) Onboard localization and mapping, (ii) Mapping server at the base station, and (iii) Pose graph comparison and correction.

**Centralized mapping and Localization**

Each robot performs onboard mapping and localization to provide an odometry estimate of its current position as well as to create a local map of the environment. Particularly, we don’t anticipate that the onboard system must perform any loop closure detection or re-localization methods to reduce their drift. These methods typically require large computational capacities and might, in real-world deployments, inhibit crucial processes like control, odometry, and navigation. Hence, in our setup, each robot solely focuses on estimating the relative motion and building an onboard map without optimizing it. The onboard map and pose estimates are then incrementally sent to a centralized mapping server for additional processing.

![Figure 2. Different employed graphs: a) Onboard visual-inertial graph, which will be incorporated into the global server map. b) Graph abstractions, including Kron reduction. c) Synchronized graphs. d) Multiscale spectral graph analysis. With the results from (d) the system queries poses in \(SE(3)\) to generate constraints.](image-url)
More specifically, the mapping server takes the role of a high-performance computing resource and acts as a centralized communications hub for all the robots. Hence, the mapping server constitutes the core component of our collaborative mapping approach. In particular, all incoming robot maps are accumulated, merged into a single collaborative multi-robot map, loop closed, and globally optimized. Consequently, all the computational-intense multi-robot operations are delegated to the central mapping server while the robots only perform the initial map building.

Since the onboard maps are sent incrementally, the mapping server needs to ensure continuous operation on the global multi-robot map, i.e., it does not run a set of operations once but rather constantly iterates over the global multi-robot map and, thereby, considering the most recent data.

Furthermore, at the end of each iteration, the mapping server constructs a global multi-robot graph from the optimized pose information and broadcasts it on the network. Notably, the sent global graph does not contain any sort of constraints such as IMU factors, visual terms, or LiDAR terms, but rather only comprises the positional and rotational information of the nodes.

Each time a robot receives a global graph, it replaces it with the previously received graph until the next update arrives. Thus, the comparison of the on-board estimation and the globally optimized solution of the server happens at the individual robots using the pose information in $SE(3)$.

**Global Multi-Robot Graph.** The global graph sent to the robots encapsulates the global knowledge of the environment in a compact representation and contains crucial information such as the last known positions of all robots. The global graph is built by defining representative nodes for each incoming submap. The representative nodes can be freely chosen but ought to reflect the robot trajectories to some degree. We found that keyframing heuristics such as minimum distance and rotation between consecutive nodes works well in our experiments.

Moreover, since the global map is continuously optimized, the graph is not immediately built after every operation, but only a reference to each node is maintained. When an update is triggered after the global optimization, the graph monitor (cf. Figure 1) retrieves the latest estimate of the multi-robot pose graph, builds the graph, and sends it to all robots.

A radius search is performed around each vertex in the graph, and the weight of adjacent nodes is calculated using a squared exponential function where the weight decreases with increasing distance $\Delta$, i.e.

$$w_D(n, m) = \exp \left( -\frac{\Delta(n, m)}{2\sigma^2} \right) \tag{5}$$

The distance function $\Delta(n, m)$ measures how much the nodes $n$ and $m$ are related to each other. Possible realizations of $\Delta(n, m)$ include measures of spatial proximity or viewpoint coherence, such as distance metrics using the global position $p \in \mathbb{R}^3$ or global orientation $R \in SO(3)$, among others. Although less generic, high-level information, such as the number of co-observed visual landmarks and semantic objects, can also be incorporated into $\Delta(n, m)$.

In our previous work (Bernreiter et al. 2022), we employed an abstraction from the factor graph, the so-called proxy graph, that only consisted of the positional information in $\mathbb{R}^3$ of the nodes in the factor graph. Since a drift in the onboard estimation or a degenerate state will always be recognized as a difference in the positions of the estimates. However, this does not accurately capture rotational drifts since the positional difference will manifest only on the consecutive nodes.

Therefore, in this paper, we employ a combination of spatial proximity and viewpoint coherence using a distance metric in $SE(3)$ (Barfoot 2017). Specifically, between two poses $T_n, T_m \in SE(3)$ we compute

$$\Delta(n, m) = \sqrt{\langle \xi^{\wedge}(n, m), \xi^{\wedge}(n, m) \rangle} = \sqrt{-\text{tr} \left( \xi^{\wedge}(n, m) M \xi^{\wedge\top}(n, m) \right)} \tag{6}$$

where $M$ is a set of weights for rotation and translation and $\wedge: \mathbb{R}^6 \rightarrow \mathbb{R}^{4 \times 4}$. The distance metric $\xi_{n,m} \in \mathfrak{se}(3)$ is defined as $\xi_{n,m} = \ln \left( T_n^{-1} T_m \right)^\vee$ with $\vee: \mathbb{R}^{4 \times 4} \rightarrow \mathbb{R}^6$ and $\ln: SE(3) \rightarrow \mathfrak{se}(3)$.

It is important to note that other realizations are possible distance metrics, e.g., in $\mathbb{R}^2$ using the Euclidean distance or in $SO(3)$ using the distance between rotations (Trefethen and Bau III 1997), i.e.

$$\Delta_r(n, m) = \text{tr} R_m R_n^{-\top}, \tag{7}$$

where $R \in SO(3)$ and represents the orientation.

**Graph Reduction.** Our framework allows for multiple different reduction schemes, for which all approaches operate on the server graph $G_{\text{server}}$ at the centralized mapping server.

When the global graph $G_{\text{server}}$ is broadcasted over the network, each robot synchronizes the onboard estimation with the nodes in $G_{\text{server}}$ based on their timestamp (cf. Figure 2c). Since $G_{\text{server}}$ was built based on the odometry estimates, there is a direct mapping between the nodes.

Moreover, it is important to note that the larger the server graph $G_{\text{server}}$ becomes, the higher the computational and bandwidth requirements are. This makes it undesirable to send the full graph to all robots in time-critical applications. As a consequence, $G_{\text{server}}$ is reduced using a Kron reduction (Dorfler and Bullo 2013), when the server graph reaches a predefined threshold of nodes. The Kron reduction removes a subset of nodes of the server graph while preserving the spectral properties and the adjacency matrix. The removal of the nodes is implemented as a Schur complement of $L$

$$L_{\text{reduced}} = L_{n,n} - L_{n,\tilde{n}} (L_{\tilde{n},\tilde{n}})^{-1} L_{\tilde{n},n}, \tag{8}$$

where $\tilde{n}$ are the nodes marked for removal.

The choice of which nodes to keep in the reduced graph should rely upon the information content and how prominent a specific node is. Most interestingly, the magnitude of the eigenvalues of $L$ denotes how much information each graph frequency contains. Therefore, the nodes to keep during the Kron reduction are selected according to the largest eigenvalues in $\Lambda$.

As a next step, we perform a graph spectral analysis using $G_{\text{server}}$ to identify drifts in the onboard estimation.

**Spectral Analysis of Graph Signals**

The analysis of spectral components of band-limited signals is a well-established and widely used technique in engineering and research. This paper uses the theory of spectral analysis
and signal processing defined on graphs. In contrast to the standard spectral analysis, graphs do not assume any underlying manifold. Therefore, they are well-suited for many robotic applications where graph structures such as pose graphs play a significant role.

**Graph Comparison.** After the robot has received a global update, a chronological synchronization is performed, yielding a one-to-one mapping of the global graph $G_{\text{server}}$ and onboard estimates. Next, the pose information at the nodes of each graph is used to create the functions $f$ and $h$ for the server and robot estimates, respectively. In particular, the pose information for each node in the graph is used to compute the relative distance to the origin yielding a unique expression for each node.

Generally, wavelets are well-known to be very efficient and flexible for a variety of different tasks in signal processing problems Hammond et al. (2019). In traditional wavelet analysis, a signal $x(t)$ is projected onto a scaled ($a$) and shifted ($b$) wavelet $\psi$, i.e.

$$W(a,b) = \langle x(t), \psi_{a,b}(t) \rangle_t = \int_{-\infty}^{\infty} \frac{1}{a} \psi^\star \left( \frac{t - b}{a} \right) x(t) dt. \tag{9}$$

Using Parseval’s theorem, i.e., $\langle f, g \rangle = \langle F, G \rangle$, eq (9) can also be expressed with the Fourier-transformed signal: $W(a,b) = \langle X(\omega), \Psi_{a,b}(\omega) \rangle$. Analogously, the graph wavelet transform can be derived using the graph Fourier transform and a wavelet filter kernel on $L$. For more details, we refer the interested reader to the work of Hammond et al. (2011, 2019).

By construction, graph wavelets have the property of being localized on the graph (Tremblay and Borgnat 2014; Donnat et al. 2018), and, therefore, directly relate to its structural properties. The realization of a graph wavelet $\psi_{s,n}$ for a scale $s$ and node $n$ is given by

$$\psi_{s,n} = UG_{s}(\Lambda)U^\top \delta_n \tag{10}$$

$$= UG_{s}(\Lambda)\Delta_n, \tag{11}$$

where $\delta_n$ and $\Delta_n$ are a Dirac centered at vertex $n$ in the graph and graph spectral domain, and $G_s$ the wavelet filter bank at scale $s$. In other words, the filter bank $G_s$ acts only on the eigenvalues of the graph, i.e., $G_{s}(\Lambda) = \text{diag}(g(s\lambda_1), ..., g(s\lambda_N))$ and is multiplied with the graph Fourier-transformed Dirac, followed by an inverse transform $U$. Since $\psi_{s,n}$ lies in the graph domain, we can compute the wavelet coefficients for a graph signal $f$ using

$$W_{s,n} = \psi_{s,n}^\top f. \tag{12}$$

It is important to note that the wavelet $psi$ is calculated only for the server graph, while we compute graphical wavelet coefficients for both server and onboard estimation. Since $psi$ does not change as long as the global server graph does not change, we only need to compute Eq. (10) once per version of the global graph.

We employ the Meyer wavelet due to its good localization in the graph and frequency domain. Generally, larger values for $s$ compress the filter function $g$, while smaller values of $s$ stretch $g$. Thus, very large scales do not capture drifts precisely anymore. We found that using less than ten scales provides a good amount of information with different granularity on the inconsistencies.

The graph signal $f$ is localized for each node in the graph and can be chosen arbitrarily but should reflect the characteristics of each node, similar to the edge weights in Eq. (5). In this work, we utilize the distance metric defined in Eq. (6) from the local map origin $o$ to every node $n$ in the graph, i.e., $\Delta(n, o)$.

In summary, we compute wavelet coefficients $W$ up to scale $s_{\text{max}}$ for the server and onboard graphs, which constitute a feature vector that represents multiscale structural information for each node in the graph. Moreover, by comparing the coefficients for the server and onboard estimates, our algorithm can efficiently identify structural differences between them.

**Correcting Onboard Estimation**

For a server node $n$ with corresponding onboard node $n'$, the scale-wise distance is computed as:

$$d^s_{n,n'} = \|W_{s,n} - W_{s,n'}\|_2, \tag{13}$$

where $W_{s,n}$ and $W_{s,n'}$ were computed using Eq. (12) and the graph signals for server and onboard graph, respectively.

Intuitively, since graph wavelets are localized at a specific node $n$ in the graph (Tremblay and Borgnat 2014; Donnat et al. 2018), and large scales compress $g$, the process spreads further into the graph, leading to a description of the larger neighborhood. In contrast, since small scales stretch $g$, they yield a description of the closer neighborhood of $n$. It is also possible to combine multiple scales, e.g., by accumulating the distances in Eq (13) over multiple scales.

Consequently, three separate cases are distinguished, i.e., a large difference in the lower, mid, and higher scales of the coefficients (cf. Figure 3a-c). In our previous work (Bernreiter et al. 2022), we introduced thresholds for $d^s_{n,n'}$ to decide when to construct constraints for nodes exceeding it. Essentially, the threshold for the small, mid, and large scales needed to be set carefully for the graph and might require some tuning. In this work, we introduce the notion of only using the top distances for constructing the constraints. If a small-scale difference is amongst the top distances, a relative constraint is added between the direct neighbors of $n'$. Likewise, a corresponding constraint within a multi-hop distance of $n'$ is added for a top mid-scale difference. In the case of a top large-scale discrepancy, a constraint between the $k$-nearest submaps is added. It is important to note that the server and robot graph can be expressed in arbitrary and unknown frames.

---

Figure 3. Illustration of three different relative constraint types. Based on the scale of the structural difference, additional constraints are added to correct (a) adjacent, (b) close neighborhood, (c) between submaps, or (d) on degenerate poses.
since all different types of constraints are relative between nodes. Additionally, it should be noted that the constraints originate directly from the $SE(3)$ graph that was used for the spectral comparison.

**Updating the Onboard Graph** Each robot runs an onboard incremental graph optimization proposed by Kaess et al. (2012) to incorporate the odometry constraints as well as the additional constraints provided by our algorithm.

Since the mapping server will continuously optimize the global multi-robot map and incorporate new inter- and intra-robot loop closures, the constraints can change over time, i.e., existing constraints can be republished with some differences. Thus, the onboard graph manager module keeps a history of all the incorporated constraints and checks for each incoming constraint whether an already active constraint exists between the corresponding nodes. For efficiency, existing constraints are only updated when there is a reasonable difference in translation and rotation to the previous one. Otherwise, they remain unchanged.

New constraints are not directly added to the onboard graph, but we rather buffer all incoming constraints and apply them at once in a batch. This avoids the graph being accessed for every constraint that is identified.

Additionally, since the global multi-robot graph is modified over time, new constraints are also generated if previously labeled nodes have changed but have not been identified during the current comparison. This is important since loop closures at the mapping server can significantly alter the global graph. Accordingly, the newly generated constraints could contradict the already existing constraints in the onboard graph and, therefore, need to be updated.

**Experiments**

We thoroughly evaluate the proposed framework with its different configurations and demonstrate its real-world application using various datasets comprising aerial and legged robots. First, we validate our approach and compare its performance to the current state-of-the-art methods using the EuRoC (Burri et al. 2016) dataset sequences to simulate multi-robot deployments. Next, we demonstrate the real-world performance of our framework during a multi-robot autonomous exploration and mapping mission conducted in an underground tunnel system using ANYmal (Hutter et al. 2017) legged robots. Finally, a multi-robot experiment conducted in indoor and outdoor environments demonstrates the localization recovery for an individual agent in case of onboard localization failure.

For all experiments, we use a radius search of 7 m around the nodes in the global multi-robot map to construct the global graph using Eq. (5). The root-mean-square of the absolute trajectory error denoted as RMSE, is used as an evaluation metric for all experiments. Moreover, unless otherwise stated, our framework is continuously running along with the onboard estimation and performs the proposed spectral analysis every 20 s. At the end of each comparison, the onboard graph will be provided with the top 15 constraints.

Finally, the mapping server provides an updated multi-robot graph to the robots after performing one cycle of operations.

**EuRoC Dataset: Validation and Comparison**

We compare the performance of our proposed approach against the current state-of-the-art collaborative mapping frameworks (Karrer et al. 2018; Schmuck and Chli 2019; Campos et al. 2021) using the Machine Hall (MH) sequences from the EuRoC dataset to evaluate and validate our single- and multi-robot performance. For each sequence, ROVIO (Bloesch et al. 2017) is used to provide monocular visual-inertial odometry for individual aerial agents.

First, we evaluate the localization performance by comparing the onboard robot estimates, collaborative server estimates, and the proposed approach to the ground truth. In addition, we also compare current state-of-the-art approaches with results presented in Table 1. Since the onboard estimation does not have an initial collaborative result, we omit reporting any error. Each robot employs a $SE(3)$ distance as in Eq. (6), unless otherwise stated. Moreover, each robot does not employ graph reduction measures but utilizes the full graph to infer the constraints.

It can be noted that despite the more significant individual onboard error, the proposed framework still attains the lowest collaborative error. Furthermore, by correcting the onboard estimation using our multi-scale spectral approach, the lowest single-robot errors are also achieved, demonstrating the proposed approach’s effectiveness in correcting large onboard estimation errors. Next, using the experimental setup described in CVI-SLAM (Karrer et al. 2018), we demonstrate that the proposed approach can facilitate accurate pose estimation for individual robots by providing collaborative corrections, as shown in Table 2.

**Table 1.** RMSE comparison for the EuRoC dataset. The top part shows the results of single and collaborative approaches, while the bottom row shows the individual corrected results.

**Table 2.** Onboard pose RMSE after adding constraints from the centralized server for different dataset combinations.

---

\*Single robot results from Qin et al. (2018). Collaborative result from Schmuck and Chli (2019).

†Monocular visual-inertial results from Campos et al. (2021).

‡As reported in Schmuck and Chli (2019).

§Results from our previous work Bernreiter et al. (2022)
Finally, we investigate and analyze the effects of applying different reduction levels to the server graph $G_{server}$. This leads to a graph with fewer nodes being broadcasted to the individual robots, enabling a smaller network footprint. Thus, the server-to-robot communication requirements can be significantly reduced with a sparse relaxation of the dense server graph while still being able to reduce the onboard estimation error. This is particularly useful for very efficient bandwidth requirements or when the robots are only within connection range for a very short period of time. Table 3 states the RMSE of the onboard after applying three different levels of reduction to the server graph.

After the reduction, the server graph contains fewer nodes that are sent to the robots. Thus, it requires less bandwidth for the transmission to the individual robots. However, the synchronization of the onboard estimate with $G_{server}$ leads then also to fewer nodes which results in fewer constraints that can be added to the onboard factor graph, and the error steadily increases with each level. Nevertheless, even when removing more than 50% of the nodes in the server graph, the error does not significantly increase. Consequently, it is possible to use our approach to gain a reasonable improvement with only a few additional constraints in the onboard graph.

### Analysis of Robotic Drift Recovery

We conducted an experiment in a particularly challenging environment for LiDAR localization due to the absence of surrounding geometric structure to demonstrate the utility of collaborative mapping toward localization recovery for an individual robot in case of an onboard estimation failure. Two ANYmal robots were simultaneously deployed in an indoor office environment connected to an outdoor rooftop terrace. Each robot is equipped with a Velodyne VLP-16 LiDAR and a Sevensense Alphasense visual-inertial sensor. Both sensors are synchronized within the onboard computer.

Onboard robot odometry estimation and mapping are performed by CompSLAM (Khattak et al. 2020) and, along with the required visual and pointcloud data, are sent to the mapping server whenever the robots are within the communication range. The first robot performs a loop indoors while the second robot transitions outdoors through a narrow doorway, navigates a rectangular path, and returns indoors.

**Localization Recovery** Due to the absence of surrounding structures on the outdoor terrace, the onboard localization drifts significantly, skewing the onboard robot map. Nevertheless, the collaborative mapping approach is able to generate a consistent map of the environment, as shown in Figure 4, due to its inter- and intra-robot loop closure capabilities.

We created a ground-truth map using a Leica RTC360 scanner to evaluate the proposed collaborative mapping framework’s performance and quantify the effect of the integration of collaborative corrections on individual robot pose accuracy. Ground-truth robot poses were then computed by registering individual robot pointclouds against the ground-truth map following the approach of Ramezani et al. (2020). Furthermore, the integration of collaborative constraints enables localization recovery for the individual robot leading to a significant reduction in its pose error, as shown in Table 4, when compared to the ground-truth robot trajectory.

| Sequences | MH01 (80 m) | MH02 (73 m) | MH03 (130 m) |
|-----------|-------------|-------------|-------------|
| **Onboard** | 0.21 m | 0.29 m | 0.41 m |
| **No Reduction** | 0.026 m / 192 | 0.027 m / 178 | 0.033 m / 255 |
| **20% Reduction** | 0.240 m / 153 | 0.032 m / 142 | 0.040 m / 204 |
| **40% Reduction** | 0.041 m / 115 | 0.036 m / 142 | 0.048 m / 153 |
| **60% Reduction** | 0.048 m / 76 | 0.042 m / 71 | 0.046 m / 102 |

Table 3. Onboard RMSE after correction w.r.t. different Kron reduction levels. The right value denotes the number of nodes in the server graph after the reduction. With the increasing level of reduction, the efficiency of the algorithm decreases due to having less constrained nodes in the onboard factor graph.

| Method | RMSE (m) | Time (ms) | Server (m) |
|--------|----------|-----------|------------|
| ANYmal 1 | 2.22 m | 5.2 ms | 0.21 m |
| Proposed | 0.30 m | 5.9 ms | |

| Method | RMSE (m) | Time (ms) | Server (m) |
|--------|----------|-----------|------------|
| ANYmal 2 | 0.25 m | 3.4 ms | 0.14 m |
| Proposed | 0.16 m | 4.1 ms | |

Table 4. Comparison of the RMSE of the onboard estimation before and after the supplying additional constraints.
The submap constraints are particularly useful for correcting large-scale drifts, while the smaller constraint types performed a local refinement. Consequently, the submap constraints play a most significant role in the correction and recovery of the onboard drift. Hence, in the following, we will investigate the constraint generation process as well as the interrelations between the construction of the graph and the type of constraints.

**Analysis of the Constraint Generation** In this experiment, we investigate which and where constraints are generated to fix the onboard estimation. We additionally compare the performance of different distance metrics and construction approaches of the graphs in comparison to our previous work (Bernreiter et al. 2022).

In more detail, we configure our framework to perform the signal and edge weight calculations in different manifolds. Specifically, we investigate three different instances of our proposed framework where we use: (i) $L_2$-Norm in $\mathbb{R}^3$, (ii) using Eq. 7 in $SO(3)$, and (iii) using Eq. (6) in $SE(3)$. The specific manifold is not swapped for the graph and signal, i.e., the same manifold is applied for computing the distance for both the edge weights and the signal comparisons.

The resulting constraints for each selection strategy are presented in Figure 5. It is evident that selecting only the top constraint candidates leads to primarily adjacent constraints for $\mathbb{R}^3$ and n-hop and submap constraints for $SO(3)$. For the $SE(3)$ strategy, however, it can be seen as a combination of the other strategies, leading to a lower error than for the other approaches.

We present the results in Table 5 for all the different signal and weight computation strategies. Each strategy is separated into top 10, 30, and 50 constraints to get a deeper insight into the constraint generation process. In this experiment, the onboard estimation is only updated once with the respective top $n$ constraints at the end of the run when the server has received and processed all the information to allow for a fair comparison of the individual strategies. By providing constraints only once with the highest differences, it can be shown how the manifold influences the identification of the drifting nodes.

| Indoor/Outdoor Dataset - Constraint Generation | $\mathbb{R}^3$ | $SO(3)$ | $SE(3)$ |
|-----------------------------------------------|------------|---------|---------|
| Top 10 constraints                            | 2.35 m     | 1.65 m  | 0.33 m  |
| Top 30 constraints                            | 2.05 m     | 0.35 m  | 0.32 m  |
| Top 50 constraints                            | 1.01 m     | 0.33 m  | 0.31 m  |

Table 5. Evaluation of different graph construction and signal generation strategies for the ANYmal 1 indoor/outdoor dataset.

Notably, the top 10 constraints for $\mathbb{R}^3$ even increase the error of the trajectory. This is due to the fact that solely adjacent constraints were applied to local regions, resulting in a larger error as the drift worsened when assessed globally. Subsequent evaluations of graphs would then result in more effective constraints.

Furthermore, the results also indicate superior performance of the $SE(3)$ computations. Even by only evaluating the top ten constraints, the approach attains a significant improvement to the baseline estimation. Most importantly, adding more constraints does not account for substantial improvements anymore when using $SE(3)$ computations since the initial result using ten constraints has already identified and corrected the most prominent discrepancies.

**Figure 5.** Illustration of the different types of constraints added by using (A) $\mathbb{R}^3$, (B) $SO(3)$ and (C) $SE(3)$. At each comparison, only the top 30 constraints were propagated to the onboard correction node. The top right shows a top-down view of the new constraints.

**Large-Scale Multi-Robot Subterranean Exploration**

We demonstrate the suitability of our approach for complex real-world applications by utilizing it during an autonomous multi-robot exploration (Dang et al. 2020) and mapping mission conducted at the Hagerbach underground facility in Switzerland. In this experiment, three ANYmal quadrupedal robots were deployed during an hour-long mission and autonomously navigated distances of roughly 1.2 km, 1.1 km, and 600 m, respectively. Each robot is equipped with the same sensor payload as described in the previous experiment.

The collaborative multi-robot map and individual robot maps are shown in Figure 6, with quantitative results presented in Table 6, demonstrating that the collaborative mapping approach greatly benefits from the feedback constraints by significantly reducing the onboard pose estimation error.

In this experiment, we configured our approach to provide as many constraints as possible to achieve the lowest possible onboard trajectory error. It is important to note that there is a tradeoff between the incorporation of the constraints into the onboard graph and the resulting improvement it yields. Table 6 also lists the number of components in the optimization problems along with a number of non-zero (NNZ) components.

Specifically, when incorporating more constraints in the onboard graph, the optimization has to deal with more factors in it, i.e., the problem becomes less sparse as there are more interrelations between the individual factors. Hence, when using an incremental solver (iSAM2), as it is running on each
Figure 6. Illustration of the multi-robot map built by our approach. The left image (A) shows the global multi-robot map at the mapping server comprising three individual robot missions. The right image (B) shows a top-down view of the individual maps.

| Underground Tunnel - Ground Truth Evaluation |
|--------------------------------------------|
| Method | ANYmal 1 | ANYmal 2 | ANYmal 3 |
|--------|----------|----------|----------|
|        | RMSE     | N   | NNZ | RMSE     | N   | NNZ | RMSE     | N   | NNZ       |
| Onboard| 1.15 m   | 2098368 | 79.2% | 0.97 m   | 1831320 | 79.2% | 0.43 m   | 1070280 | 79.2% |
| Proposed| 0.14 m  | 5335128 | 50.0% | 0.10 m   | 7194816 | 34.4% | 0.07 m   | 6140088 | 23.3% |

Table 6. RMSE comparison for the original onboard, server, and corrected onboard graph. We additionally provide insights into the sparsity of the problem by showing the total number of components (N) and the number of non-zero components (NNZ) per dataset.

robot, it might happen that a large portion of the newly added factors are factors between nodes that are significantly far away in the state matrix, invalidating the locality assumption of the solver along with all its performance optimizations. As a consequence, the optimization needs to perform more frequently expensive operations and can take a longer time to incorporate the constraints of our approach into the optimization problem.

Degeneracy Detection of the State Estimation

Next, we investigate the application of our approach in the case of a degenerate condition in the onboard state estimation. In particular, each robot employs a LiDAR-based state estimation (Khattak et al. 2020).

The experiment took place in a subterranean cave in Switzerland and comprised two legged ANYmal robots and one flying tricopter (Tranzatto et al. 2022). Similar to the other experiments, all robots deploy the same sensor payload. The flying tricopter as well as one of the ANYmal robots, explore the cave environment without any issues. However, one of the ANYmal robots enters a long narrow tunnel where its onboard estimation system degenerates for a short period, causing shifted maps in the onboard mapping system. This is particularly critical as a safe exploration of the cave cannot be guaranteed any longer for this robot, and it needs to be grounded.

Since our approach is not directly applicable to the detection of degenerate states, we investigate whether, by providing additional constraints, the onboard estimation can recover from it. Our approach employed a constraint generation strategy of the top 15 constraints and can seamlessly recover from the incorrect state by only using a few constraints. The global multi-robot map comprises all three robots and, along with individually corrected robot trajectories, is shown in Figure 7 together with the degenerate and corrected onboard estimation.

The degeneracy affects the relative position estimation in the direction of the tunnel, i.e., the robot is stuck until the environment becomes unique enough again. Hence, the degeneracy is recognizable by the fact that the map is estimated shorter in length on the way back than on the way in. In contrast, a drift in the estimation would have been roughly the same length but still shifted.

Although the multi-modal mapping server requires at least a few optimization cycles to repair the broken map, our approach can still recover the onboard map reasonably. Hence, enabling the robot to continue the exploration of the underground environment.

In addition, Figure 8 visualizes in detail which constraints our proposed approach sent to the onboard graph. Most interestingly, the degeneracy in the state estimation of the robot leads to that the prevalence of constructed constraints being between adjacent nodes in the affected area. Since the degeneracy does not properly estimate the transformation between multiple consecutive steps, the small scales are more predominant than the other scales.

DARPA Subterranean Challenge

Finally, to show our approach’s pertinence to a real-world autonomous multi-robot search and rescue mission, we...
Figure 7. Illustration of the global multi-robot map with corrected robot trajectories (A). The onboard estimation of ANYmal 2 became degenerate during the exploration of a long tunnel (B). The multi-modal mapping server was able to overcome the degeneracy and, by providing additional constraints, fixed the onboard estimation (C).

Table 7. Comparison of the RMSE of the onboard estimation before and after the supplying additional constraints.

| Robot     | Distance | Onboard RMSE          | Server RMSE          | Corrected RMSE       |
|-----------|----------|-----------------------|----------------------|----------------------|
| ANYmal 1  | 240 m    | 0.72 m (±0.41 m)      | 0.25 m (±0.13 m)     | 0.27 m (±0.14 m)     |
| ANYmal 2  | 687 m    | 1.29 m (±0.90 m)      | 0.36 m (±0.28 m)     | 0.37 m (±0.41 m)     |
| ANYmal 3  | 311 m    | 0.23 m (±0.43 m)      | 0.20 m (±0.34 m)     | 0.22 m (±0.22 m)     |
| ANYmal 4  | 500 m    | 1.00 m (±0.71 m)      | 0.24 m (±0.17 m)     | 0.29 m (±0.14 m)     |

Figure 8. Illustration of the employed constraints constructed by our proposed approach at the degenerate region.

employ it on the DARPA Subterranean (SubT) dataset of Team CERBERUS (Tranzatto et al. 2022). The SubT Challenge was an international robotics competition for fast and autonomous exploration in complex underground environments, where team CERBERUS won the final event. Each participating team had to deploy a robotic team that explores an unknown environment reporting the location of specific artifacts in it.

Team CERBERUS’ dataset comprises four ANYmal legged robots covering roughly 2 km of semi-autonomous exploration within a 1-hour mission. During the exploration, multiple robots communicated with a central mapping server, providing visual feedback to a human supervisor. The environment of the final SubT Challenge consisted of three individual regions: i) a tunnel, ii) an urban iii) and a cave environment making it particularly challenging for onboard localization and mapping modules.

Although the proposed approach was not deployed during the competition, we show qualitative, post-processed results of the corrected onboard estimations in Figure 9 with respect to the ground truth map.

Figure 9. Corrected robot trajectories using our approach with respect to the ground truth map.

Moreover, Table 7 shows quantified results for each individual robot and that our approach can significantly improve the onboard estimation modules. Since none of the robots employed an onboard loop closure detection, each robot accumulated an error over time that could not have been corrected. Only the mapping server at the base station runs a multi-robot loop closure detection. Hence, by using our approach to construct additional onboard constraints, each robot implicitly incorporates the loop closures from the mapping server and decreases its drift.

It is important to note that Team CERBERUS scored 23 points along with the second-best team and won due to a tie-breaker rule. During the exploration of ANYmal 4, it reported an incorrect location of a cell phone (artifact ID: L22) artifact.
for which no score was accounted then. Cell phone detection was implemented by reporting the current position of the robot when the Bluetooth RSSI values exceeded a specific threshold. Thus, the location of the artifact primarily depends on how close the robot walks to the artifact and how large its current drift is.

During the final run, the reported position of artifact L22 was slightly above the 5 m threshold of the ground truth location of the artifact, therefore, was not scored. The drift of the robot was not significant but just enough to exceed the threshold. However, by employing our approach to the post-processed run, the onboard estimation can be greatly improved. Figure 10 illustrates the difference in the reported location of artifact L22 between the onboard estimation during the final run and the corrected trajectory using our approach. As a result, the reported position of the artifact would have been within the 5 m threshold and, thus, would have scored.

Conclusions

This paper proposed a novel collaborative multi-robot framework for updating the pose graph of individual robots with constraints from a centralized mapping server. In this context, we presented a graph-based spectral analysis of the robot and server graphs to identify the underlying structural differences in the onboard estimation. In particular, our approach computes signals in $SE(3)$ and, along with graph Wavelets, finds nodes in the onboard graph that contradict the globally optimized graph of the mapping server. Most importantly, by efficiently adding 6-DoF constraints on drifting nodes, minimal additional computation and communication resources are needed, while significantly improving the onboard estimation.

The presented results using large-scale multi-robot field deployments in challenging environments demonstrate the real-world potential of the proposed approach for both research and industry.

We intend to continue our research in two directions. First, we will investigate the construction of graph hierarchies to circumvent the accuracy trade-off for the reduction levels. Second, we plan to explore the possibility of keeping the onboard graph sparse using a minimal spanning tree that keeps only a limited set of the most prominent constraints in the onboard graph.

Acknowledgements

This work was supported as a part of NCCR Robotics, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 51NF40_185543).

The authors are thankful to Marco Tranzatto, Patrick Pfundschuh, Samuel Zimmermann and Timon Homberger, Gabriel Waibel for their assistance with field experiments.

References

Akai N, Morales LY, Hirayama T and Murase H (2019) Misalignment Recognition Using Markov Random Fields With Fully Connected Latent Variables for Detecting Localization Failures. *IEEE Robotics and Automation Letters* 4(4): 3955–3962. DOI: 10.1109/LRA.2019.2929999.

Barfoot TD (2017) *State Estimation for Robotics*. Cambridge: Cambridge University Press. ISBN 9781316671528. DOI: 10.1017/9781316671528.

Bernreiter L, Khattak S, Ott L, Siegwart R, Hutter M and Cadena C (2022) Collaborative Robot Mapping using Spectral Graph Analysis. In: 2022 International Conference on Robotics and Automation (ICRA). IEEE. ISBN 978-1-7281-9681-7, pp. 3662–3668. DOI: 10.1109/ICRA46639.2022.9812102.

Bloesch M, Burri M, Omari S, Hutter M and Siegwart R (2017) Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research* 36(10): 1053–1072. DOI: 10.1177/0278364917728574.

Brunskill E, Kollar T and Roy N (2007) Topological mapping using spectral clustering and classification. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. ISBN 978-1-4244-0911-2, pp. 3491–3496. DOI: 10.1109/IROS.2007.4399611.

Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW and Siegwart R (2016) The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research* 35(10): 1157–1163. DOI: 10.1177/0278364915620033.

Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I and Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics* 32(6): 1309–1332. DOI: 10.1109/TRO.2016.2624754.

Campos C, Elvira R, Rodriguez JIG, M Montiel JM and D Tardos J (2021) ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics* : 1–17DOI: 10.1109/TRO.2021.3075644.

Chandra R, Guan T, Panuganti S, Mittal T, Bhattacharya U, Bera A and Manocha D (2020) Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs. *IEEE Robotics and Automation Letters* 5(3): 4882–4890. DOI: 10.1109/LRA.2020.3004794.

Chang Y, Ebadi K, Denniston CE, Ginting MF, Rosinol A, Reinke A, Palieri M, Shi J, Chatterjee A, Morrell B, Agha-mohammed Aa and Carlone L (2022) LAMP 2.0: A Robust Multi-Robot SLAM
System for Operation in Challenging Large-Scale Underground Environments. *IEEE Robotics and Automation Letters*.

Cunningham A, Indelman V and Dellaert F (2013) DDF-SAM 2.0: Consistent distributed smoothing and mapping. In: 2013 *IEEE International Conference on Robotics and Automation*. IEEE. ISBN 978-1-4673-5643-5, pp. 5220–5227. DOI: 10.1109/ICRA.2013.6631323.

Dang T, Tranzatto M, Khattak S, Mascarich F, Alexis K and Hutter M (2020) Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics* 37(8): 1363–1388.

Deutsch I, Liu M and Siegwart R (2016) A framework for multi-robot pose graph SLAM. 2016 *IEEE International Conference on Real-Time Computing and Robotics*. RCAR 2016: 567–572. DOI: 10.1109/RCAR.2016.7784092.

Donnat C, Zitnik M, Hallac D and Leskovec J (2018) Learning Structural Node Embeddings via Diffusion Wavelets. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: ACM. ISBN 978-1-4503-55520, pp. 1320–1329. DOI: 10.1145/3219819.3220025.

Dorfler F and Bullo F (2013) Kron Reduction of Graphs With Applications to Electrical Networks. *IEEE Transactions on Circuits and Systems I: Regular Papers* 60(1): 150–163. DOI: 10.1109/TCSI.2012.2215780.

Ebadi K, Chang Y, Palieri M, Stephens A, Hatteland A, Heiden E, Thakur A, Funabiki N, Morrell B, Wood S, Carlone L and Agha-Mohammadi AA (2020) LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments. *Proceedings - IEEE International Conference on Robotics and Automation*: 80–86. DOI: 10.1109/ICRA40945.2020.9197082.

Egilmez HE and Ortega A (2014) Spectral anomaly detection using graph-based filtering for wireless sensor networks. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ii. IEEE. ISBN 978-1-4799-2893-4, pp. 1085–1089. DOI: 10.1109/ICASSP.2014.6853764.

Hammond DK, Vanderhegnsn P and Girovbaln R (2011) Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* 30(2): 129–150. DOI: 10.1016/j.acha.2010.04.005.

Hammond DK, Vanderhegnsn P and Girovbaln R (2019) The Spectral Graph Wavelet Transform: Fundamental Theory and Fast Computation. In: *Signals and Communication Technology*. Springer. ISBN 9783030035747, pp. 141–175. DOI: 10.1007/978-3-030-03574-7\_3.

Hutter M, Gehring C, Lauber A, Gunther F, Bellicosod C, Tsouinis V, Fankhauser P, Diethelm R, Bachmann S, Blösch M et al. (2017) Anymal-toward legged robots for harsh environments. *Advanced Robotics* 31(17): 918–931.

Indelman V (2018) Towards Cooperative Multi-robot Belief Space Planning in Unknown Environments. In: *Springer Proceedings in Advanced Robotics*, volume 2. IEEE. ISBN 9781538630815, pp. 441–457. DOI: 10.1007/978-3-319-51532-8\_27.

Kaes M, Johannsson H, Roberts R, Ila V, Leonard JJ and Dellaert F (2012) ISAM2: Incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research* 31(2): 216–235. DOI: 10.1177/0278784911430419.

Karrer M, Schmuck P and Chili M (2018) CVI-SLAM—Collaborative Visual-Inertial SLAM. *IEEE Robotics and Automation Letters* 3(4): 2762–2769. DOI: 10.1109/LRA.2018.2837226.

Khattak S, Nguyen H, Mascarich F, Dang T and Alexis K (2020) Complementary Multi–Modal Sensor Fusion for Resilient Robot Pose Estimation in Subterranean Environments. In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. ISBN 978-1-7281-4278-4, pp. 1024–1029. DOI: 10.1109/ICUAS48674.2020.9213865.

Kim B, Kaess M, Fletcher L, Leonard J, Bachrach A, Roy N and Teller S (2010) Multiple relative pose graphs for robust cooperative mapping. In: *2010 IEEE International Conference on Robotics and Automation*. IEEE. ISBN 978-1-4244-5038-1, pp. 3185–3192. DOI: 10.1109/ROBOT.2010.5509154.

Lajoie PY, Ramtoula B, Chang Y, Carlone L and Beltrame G (2020) DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams. *IEEE Robotics and Automation Letters* 5(2): 1656–1663. DOI: 10.1109/LRA.2020.2967681.

Mangelson JG, Dominic D, Eustice RM and Vasudevan R (2018) Pairwise Consistent Measurement Set Maximization for Robust Multi-Robot Map Merging. *Proceedings - IEEE International Conference on Robotics and Automation*: 2916–2923. DOI: 10.1109/ICRA.2018.8460217.

Mangelson JG, Ghaffari M, Vasudevan R and Eustice RM (2019) Characterizing the Uncertainty of Jointly Distributed Poses in the Lie Algebra. DOI: 10.1109/tro.2020.2994457.

Mateos G, Segarra S, Marques AG and Ribeiro A (2019) Connecting the Dots: Identifying Network Structure via Graph Signal Processing. *IEEE Signal Processing Magazine* 36(3): 16–43. DOI: 10.1109/MSP.2018.2890143.

Moon J and Lee BH (2020) Object-oriented Semantic Graph Based Natural Question Generation. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. ISBN 978-1-7281-7395-5, pp. 4892–4898. DOI: 10.1109/ICRA40945.2020.9196563.

Nobili S, Tinchev G and Fallon M (2018) Predicting Alignment Risk to Prevent Localization Failure. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. ISBN 978-1-5386-3081-5, pp. 1003–1010. DOI: 10.1109/ICRA.2018.8462890.

Ortega A, Frossard P, Kovacevic J, Moura JMF and Vanderhegnsn P (2018) Graph Signal Processing: Overview, Challenges, and Applications. *Proceedings of the IEEE* 106(5): 808–828. DOI: 10.1109/JPROC.2018.2820126.

Qin T, Li P and Shen S (2018) VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics* 34(4): 1004–1020. DOI: 10.1109/TRO.2018.2835729.

Ramezani M, Wang Y, Camurri M, Wisth D, Mattamala M and Fallon M (2020) The newer college dataset: Handheld lidar, inertial and vision with ground truth. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4353–4360.
Rao K, Kim D and Hwang JJ (2010) Fast Fourier Transform - Algorithms and Applications. Signals and Communication Technology, Dordrecht: Springer Netherlands. ISBN 978-1-4020-6628-3. DOI:10.1007/978-1-4020-6629-0.

Ricaud B, Borgnat P, Tremblay N, Gonçalves P and Vandergheynst P (2019) Fourier could be a data scientist: From graph Fourier transform to signal processing on graphs. Comptes Rendus Physique 20(5): 474–488. DOI:10.1016/j.crhy.2019.08.003.

Sandryhaila A and Moura JM (2014) Discrete signal processing on graphs: Frequency analysis. IEEE Transactions on Signal Processing 62(12): 3042–3054. DOI:10.1109/TSP.2014.2321121.

Schmuck P and Chi M (2019) CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. Journal of Field Robotics 36(4): 763–781. DOI:10.1002/rob.21854.

Schmuck P, Ziegler T, Karrer M, Perraudin J and Chi M (2021) COVINS: Visual-Inertial SLAM for Centralized Collaboration. In: 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). IEEE. ISBN 978-1-6654-1298-8, pp. 171–176. DOI:10.1109/ISMAR-Adjunct54149.2021.00043.

Schwertfeger S and Birk A (2013) Evaluation of map quality by matching and scoring high-level, topological map structures. In: 2013 IEEE International Conference on Robotics and Automation. IEEE. ISBN 978-1-4673-5643-5, pp. 2221–2226. DOI:10.1109/ICRA.2013.6630876.

Shuman DI, Ricaud B and Vandergheynst P (2016) Vertex-frequency analysis on graphs. Applied and Computational Harmonic Analysis 40(2): 260–291. DOI:10.1016/j.acha.2015.02.005.

Sundvall P and Jensfelt P (2006) Fault detection for mobile robots using redundant positioning systems. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE. ISBN 0-7803-9505-0, pp. 3781–3786. DOI:10.1109/ROBOT.2006.1642280.

Tian Y, Chang Y, Herrera Arias F, Nieto-Granda C, How J and Carlone L (2022) Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems. IEEE Transactions on Robotics : 1–17DOI:10.1109/TRO.2021.3137751.

Tranzatto M, Miki T, Dharmadhikari M, Bernreiter L, Kulkarni M, Mascarich F, Andersson O, Khattak S, Hutter M, Siegwart R and Alexis K (2022) CERBERUS in the DARPA Subterranean Challenge. Science Robotics 7(66): 9742. DOI:10.1126/scirobotics.abp9742.

Trefethen LN and Bau III D (1997) Numerical Linear Algebra, volume 50. Siam. ISBN 978-0-898713-61-9.

Tremblay N and Borgnat P (2014) Graph wavelets for multiscale community mining. IEEE Transactions on Signal Processing 62(20): 5227–5239. DOI:10.1109/TSP.2014.2345355.

Van Opdenbosch D and Steinbach E (2019) Collaborative Visual SLAM Using Compressed Feature Exchange. IEEE Robotics and Automation Letters 4(1): 57–64. DOI:10.1109/LRA.2018.2878920.

Zhang J, Kaess M and Singh S (2016) On degeneracy of optimization-based state estimation problems. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), volume 2016-June. IEEE. ISBN 978-1-4673-8026-3, pp. 809–816. DOI:10.1109/ICRA.2016.7487211.

Ziegler T, Karrer M, Schmuck P and Chi M (2021) Distributed Formation Estimation Via Pairwise Distance Measurements. IEEE Robotics and Automation Letters 6(2): 3017–3024. DOI:10.1109/LRA.2021.3062347.

Zou D and Tan P (2013) CoSLAM: Collaborative Visual SLAM in Dynamic Environments. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(2): 354–366. DOI:10.1109/TPAMI.2012.104.