Research Article

Balancing Accuracy and Efficiency for Large-Scale SLAM: A Minimal Subset Approach for Scalable Loop Closures

Nikolaos Stathoulopoulos[1], Christoforos Kanellakis[1], George Nikolakopoulos[1]

1. Robotics and AI Group, Department of Computer, Electrical and Space Engineering, Lulea University of Technology, Sweden

Typical LiDAR SLAM architectures feature a front-end for odometry estimation and a back-end for refining and optimizing the trajectory and map, commonly through loop closures. However, loop closure detection in large-scale missions presents significant computational challenges due to the need to identify, verify, and process numerous candidate pairs for pose graph optimization. Keyframe sampling bridges the front-end and back-end by selecting frames for storing and processing during global optimization. This article proposes an online keyframe sampling approach that constructs the pose graph using the most impactful keyframes for loop closure. We introduce the Minimal Subset Approach (MSA), which optimizes two key objectives: redundancy minimization and information preservation, implemented within a sliding window framework. By operating in the feature space rather than 3-D space, MSA efficiently reduces redundant keyframes while retaining essential information. In sum, evaluations on diverse public datasets show that the proposed approach outperforms naive methods in reducing false positive rates in place recognition, while delivering superior ATE and RPE in metric localization, without the need for manual parameter tuning. Additionally, MSA demonstrates efficiency and scalability by reducing memory usage and computational overhead during loop closure detection and pose graph optimization.

Corresponding author: Nikolaos Stathoulopoulos, niksta@ltu.se

I. Introduction

Simultaneous Localization and Mapping (SLAM) in large-scale missions presents significant challenges, particularly in maintaining accuracy and efficiency as the environment grows over time. A key component of back-end SLAM is Pose Graph Optimization (PGO), which ensures consistent mapping by optimizing the global trajectory. However, this process relies heavily on loop closure detection, to reduce accumulated drift, which is computationally expensive, especially in large-scale environments, due to the vast number of candidate pairs that need to be identified, verified, and processed to establish node connections in the pose graph. As the mission progresses, the number of loop closure candidates grows rapidly, leading to significant delays in processing times, with some datasets generating over 150,000 candidates, requiring four and a half hours to process, exceeding that of the mission time[1]. This complexity is further compounded by the computational intensity of data association and sequence matching during map optimization, where the

processing time can range from seconds to several minutes, depending on the size of the data[2][3]. Due to these computational challenges, many SLAM systems in large-scale environments have opted to either avoid or minimize loop closure detection to reduce the burden on limited computational resources. Some approaches, particularly in expansive environments, forego loop closures entirely, as the growing number of candidates becomes increasingly difficult to handle in real-time applications, as highlighted recently in[4]. While loop closure detection is critical for correcting drift and ensuring map consistency, the computational complexity can overwhelm systems if not managed effectively, often leading to false positives that degrade the overall quality of the map[5].
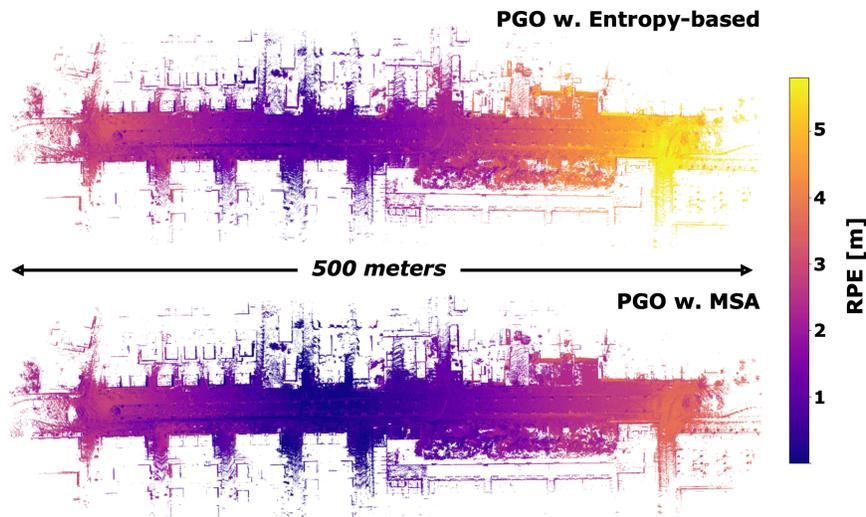
Figure 1. An example on `KITTI 06`. Comparison of two trajectories from `KISS-ICP`, after Pose Graph Optimization (PGO). In the top figure, keyframes are sampled using an entropy-based approach, while in the bottom figure, keyframes are sampled with the proposed Minimal Subset Approach (MSA)[6] which achieves a lower Relative Pose Error (RPE).

A. Related Work

Efficient implementations of pose graph optimization have significantly enhanced its scalability by leveraging factor graphs and incremental optimization techniques. In more detail, `GTSAM`[7] models probabilistic relationships using factor graphs and solves large-scale problems efficiently with solvers like `iSAM2`[8], which incrementally updates the SLAM solution in real-time through a Bayes tree representation. Similarly, `g2o`[9] provides a versatile graph optimization framework, leveraging sparse pose graph connectivity for efficiency while supporting various parameterizations and loss functions. Furthermore, sparsification methods[10][11][12] and hierarchical approaches[13][14] have further improved efficiency by reducing the number of active nodes in the optimization process. However, despite these advances, loop closure detection remains a major bottleneck due to the large number of candidate pairs that must be checked and processed to add meaningful constraints to the pose graph.

Keyframe sampling helps reduce the computational demands of loop closure detection by limiting the number of frames stored and processed in SLAM pipelines. In LiDAR-(Inertial) Odometry, systems such as LIO-SAM[15] and DiSCo-SLAM[16] or back-end global optimization systems like LAMP[17], use fixed Euclidean distance intervals (1–2 meters) for keyframe generation to maintain local maps and optimize global poses. Although effective, these methods lack adaptability in dynamic environments with varying densities of loop closure candidates. Recent advancements have introduced adaptive keyframe sampling techniques, adjusting intervals based on environmental spaciousness[18][19], although manual threshold adjustment is still required. Entropy-based methods, like the one proposed by Zeng et al.[20], use information theory to select keyframes, but face similar adaptability issues. Other approaches, such as the one proposed by Ou et al.[21], focus on displacement vector similarity to balance computational cost and map completeness while managing keyframes in a sliding window.

In contrast to the above that focus solely on back-end optimization, reducing the number of nodes in pose graph optimization, or generating keyframes for submap creation, our work centers on sampling keyframes online to reduce memory allocation and loop closure query times while preserving retrieval performance during place recognition and maintaining localization accuracy after pose graph optimization. In this work, we evaluate the impact of our previously introduced optimized keyframe sampling method[6] on critical back-end tasks, including loop closure detection and pose graph optimization, with a glimpse of the results illustrated in Fig. 1. The proposed Minimal Subset Approach (MSA) employs two key criteria, *redundancy minimization* and *information preservation*, within a sliding window optimization framework to streamline keyframe selection. By reducing redundant information while preserving essential data, MSA enhances scalability and computational efficiency for both place recognition and pose graph optimization. Unlike traditional methods that rely on spatial heuristics and 3-D space, MSA utilizes the feature space, making it inherently environment-agnostic. Its reliance on the descriptor extraction framework used for loop closure detection eliminates the need for manual parameter tuning, ensuring consistent performance across diverse scenarios.

B. Contributions

Based on the aforementioned, the main contributions of this article can be summarized as follows: (a) To the best of our knowledge, this is the first paper to present the pose graph optimization (PGO) problem with an emphasis on the interplay between place recognition, keyframe sampling and loop closure detection. We highlight the importance of addressing these interconnected problems jointly to achieve a more comprehensive solution and provide further insights on the inherent challenges. (b) This paper significantly extends our previous work[6] by demonstrating the impact of keyframe sampling not only in place recognition but on pose graph optimization and metric localization as well. This provides valuable insights into the interconnection between the front-end and back-end SLAM, showcasing how sampling intervals influence the performance of the latter. (c) Through our results we demonstrate that MSA enhances scalability and efficiency by reducing computational demands. It consistently achieves lower false positive rates in place recognition and superior ATE and RPE in metric localization across diverse environments, without manual parameter tuning. By minimizing loop closure query times, memory usage, and the overall time required for pose graph optimization, MSA is well-suited for large-scale missions.

II. Problem Formulation

A. Pose Graph Optimization

The objective of PGO is to find the most likely configuration of a robot's trajectory by minimizing the error in a graph-based representation of the poses. The pose graph is represented as a set of nodes and edges, where each node represents a robot pose and each edge encodes a spatial constraint between two poses[22]. Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ be the set of N robot poses, where, $\mathbf{x} \in SE(3)$, $\mathbf{z}_{ij} = h(\mathbf{x}_i, \mathbf{x}_j)$ be the observed spatial constraint between a pair of poses $\langle i, j \rangle$ and $\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ be the predicted value of that constraint. The error $\mathbf{e}_{ij} \in \mathbb{R}^6$ quantifies the difference between the observed relative transformation and the predicted transformation between poses $\mathbf{x}_i$ and $\mathbf{x}_j$ and is denoted as:

$$\mathbf{e}_{ij} = \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

We can then define the set of edges $\mathcal{E} = \left\{ \langle i, j \rangle \mid \exists \mathbf{z}_{ij} \right\}$, that contains all the pairs in which a relative constraint exists. The goal of pose graph optimization is to find the optimal configuration of poses $\mathbf{X}^*$ that minimizes the overall error function $E(\mathbf{X})$, denoted as:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{\langle i,j \rangle \in \mathcal{E}} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j)^\top \, \mathbf{\Omega}_{ij} \, \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j), \tag{2}$$

where $\mathbf{\Omega}_{ij} \in \mathbb{R}^{6 \times 6}$ is the information matrix that reflects the certainty of the relative pose measurement for each pair.

B. Odometry and Loop Closure Edges

Pose graph optimization is widely used as a global optimization method to mitigate accumulated pose drift during large-scale and long-term missions. This would not be possible without loop closures. Relying solely on pose-to-pose odometry constraints is insufficient to maintain global consistency, hence, loop closures are essential. Given this, it is reasonable to assume that the set of edges $\mathcal{E}$ in the pose graph can be partitioned into two distinct subsets. The first subset, $\mathcal{E}_o \subseteq \mathcal{E}$, consists of *odometry edges*, which represent consecutive movements of the robot based on local motion estimates, while the second subset, $\mathcal{E}_l \subseteq \mathcal{E}$, consists of *loop closure edges*, which correspond to observations of previously visited locations. Thus, the overall error function can be decomposed into two components:

$$E(\mathbf{X}) = \sum_{\langle i,j \rangle \in \mathcal{E}_o} \mathbf{e}_{ij}^\top \mathbf{\Omega}_{ij} \mathbf{e}_{ij} + \sum_{\langle i,j \rangle \in \mathcal{E}_l} \mathbf{e}_{ij}^\top \mathbf{\Omega}_{ij} \mathbf{e}_{ij}, \tag{3}$$

where the edge sets satisfy the following two relationships: $\mathcal{E} = \mathcal{E}o \cup \mathcal{E}_l$ and $\mathcal{E}_o \cap \mathcal{E}_l = \varnothing$.

C. Loop Closure Detection

Detecting loop closures is challenging, particularly in large-scale missions, where accumulated drift can cause traditional radius-based searches to fail by missing nearby candidates. A common solution is *place recognition*, which compares current sensory data (e.g., images or LiDAR scans) against a database of past observations to identify matches. To formalize this, we define a keyframe set $\mathbf{K} = \{(\mathbf{x}_i, \mathbf{d}_i, \mathbf{z}_i) \mid i = 1, \dots, N\}$, where $\mathbf{x}_i \in SE(3)$ is the pose of keyframe i,

$\mathbf{d}_i \in \mathbb{R}^M$ represents its descriptor which encodes distinct features[23][24], and $\mathbf{z}_i$ is the set of associated measurements. The goal is to find loop closure candidates $\mathcal{L}$ by comparing descriptors:

$$\mathcal{L} = \big\{ \langle i,j \rangle \mid f(\mathbf{d}_i, \mathbf{d}_j) > \tau, \ \forall \langle \mathbf{k}_i, \mathbf{k}_j \rangle \in \mathbf{K} \times \mathbf{K}, \ \mathbf{k}_i \neq \mathbf{k}_j \big\}, \tag{4}$$

where $f(\mathbf{d}_i, \mathbf{d}_j)$ is a similarity function (e.g., cosine similarity, Euclidean distance) and τ is a threshold for considering two keyframes representing the same place. This approach generalizes to various scenarios, such as comparing a current keyframe with the k nearest past keyframes in single-robot detection or comparing keyframes across multiple robots.

D. Reducing Search Space in Place Recognition

The detection and verification of loop closure edges $\mathcal{E}_l \triangleq \mathcal{L}$ is computationally demanding due to the large number of potential keyframe pairs, which scales quadratically as $O(n^2)$, where $n = |\mathbf{K}|$ is the number of keyframes. While using descriptors and methods like k-nearest neighbor (k-NN) search[25] can accelerate the process, additional steps are still required to remove outliers, mitigate false positives, or prioritize certain nodes[1], while extracting the relative pose transformation between loop closure nodes also involves computationally intensive techniques such as General Iterative Closest Point (GICP)[26]. This pipeline does not scale well for extensive missions, often resulting in processing times of 3–4 hours or more[4][17], and the challenge becomes even greater in multi-robot systems[3].

To address this computational complexity, we propose a *sampling strategy* to reduce the number of keyframes considered for matching, while maintaining minimal impact on the loop closure set $\mathcal{L}$. Let $\mathbf{K}_\mathrm{S} \subset \mathbf{K}$ represent a subset of keyframes selected through a sampling process $S : \mathbf{K} \to \mathbf{K}_\mathrm{S}$ such that $|\mathbf{K}_\mathrm{S}| < |\mathbf{K}|$. Our objective is to reduce the search space by confining the loop closure detection process to only the selected keyframes in $\mathbf{K}_\mathrm{S}$. The goal is to identify a reduced set $\mathbf{K}_\mathrm{S}^*$ that minimizes redundant keyframes while preserving the set of loop closures $\mathcal{L}$, a process denoted as:

$$\mathbf{K}_\mathrm{S}^* = \underset{\mathbf{K}_\mathrm{S} \subseteq \mathbf{K}}{\arg\min}(|\mathbf{K}_\mathrm{S}|), \ \text{subject to } \mathcal{L}^* = \mathcal{L}, \ \text{where} \tag{5}$$

$$\mathcal{L}^* = \big\{ \langle i,j \rangle \mid f(\mathbf{d}_i, \mathbf{d}_j) > \tau, \ \forall \langle \mathbf{k}_i, \mathbf{k}_j \rangle \in \mathbf{K}_\mathrm{S}^* \times \mathbf{K}_\mathrm{S}^* \big\}, \tag{6}$$

In this approach, the overall pose graph optimization is reformulated as follows:

$$\mathbf{X}_\mathrm{S}^* = \underset{\mathbf{X}_\mathrm{S}}{\arg\min} \sum_{\langle i,j \rangle \in \mathcal{E}_o^*} \mathbf{e}_{ij}^\top \, \mathbf{\Omega}_{ij} \, \mathbf{e}_{ij} + \sum_{\langle i,j \rangle \in \mathcal{E}_l^*} \mathbf{e}_{ij}^\top \, \mathbf{\Omega}_{ij} \, \mathbf{e}_{ij}, \tag{7}$$

where $\mathbf{X}_\mathrm{S}$ represents the poses corresponding to the optimally sampled keyframe set $\mathbf{K}_\mathrm{S}^*$, and $\mathcal{E}_o^*$, $\mathcal{E}_l^* \triangleq \mathcal{L}^*$ denote the odometry and loop closure edge pairs between the sampled keyframe poses $\mathbf{X}_\mathrm{S}$, respectively.

E. Underlying Challenges

Addressing the combined problem of pose graph optimization and loop closure detection through place recognition presents several underlying challenges that affect both the efficiency and accuracy of the solution.

1. Impact of Sampling

While the sampling strategy S aims to preserve key loop closure candidates, in practice it must retain a subset $\bar{\mathcal{L}} \subseteq \mathcal{L}$ that still ensures an effective solution for pose graph optimization. Redundant loop closure edges can lead to issues, such as overfitting, increased computational burden, and numerical instability in the optimization process. Moreover, because the sampling process reduces the set of poses, the odometry edge set $\mathcal{E}_o^* \subset \mathcal{E}_o$ is also reduced, which can affect the overall accuracy of the optimization[27], with an example illustrated on Fig. 2.

2. Combined Problem

Combining loop closure detection through place recognition with pose graph optimization presents challenges due to their differing characteristics. Loop closure detection relies on matching high-dimensional feature descriptors, which is computationally expensive and sensitive to environmental factors. Conversely, pose graph optimization relies on geometric consistency and is affected by noise and outliers in loop closures. Balancing these require careful consideration of both feature reliability and constraint robustness to prevent false positives and reduced efficiency from degrading optimization performance.

3. Dynamic Keyframe Selection

To minimize the keyframe set size in real-time, we must consider the entire set $\mathbf{K}$; however, this is impractical for maintaining computational feasibility in loop closure detection. The challenge is to determine the contribution of each keyframe, which requires anticipating future keyframes and introduces a lack of causality. Moreover, finding the optimal subset $\mathbf{K}_S^*$ is an NP-hard problem, involving an exhaustive search of all keyframe combinations, which becomes computationally infeasible for sets larger than 15–20 keyframes.

The goal of this research is to show that reducing the number of keyframes through an informed sampling strategy can significantly decrease the search space for loop closure detection, improving computational efficiency by reducing memory usage and query time without compromising SLAM accuracy. The proposed approach seeks to balance computational complexity and optimization precision, ensuring scalability and robustness.

III. Minimal Subset Approach

To address these challenges, we propose the Minimal Subset Approach (MSA) to approximate a solution to the problems in Eq.(5)-(6). Unlike our previous work[6], which focuses solely on place recognition retrieval performance, in this article we extend the analysis to assess its impact on the subsequent step described in Eq.(7) and its comparison to the original Eq.(2). The proposed MSA identifies and eliminates redundancy within a keyframe set while preserving essential data. Designed for real-time use, it employs a sliding window combinatorial optimization with dual objectives: balancing *information preservation* and *redundancy minimization*.

A. Redundancy in Keyframes

The frequency of a sensor, the platform's speed, and the environmental characteristics can cause keyframe samples to capture redundant information if they are too closely spaced. To address this, we define redundancy within a keyframe set and propose a metric to quantify it based on the descriptor space. A keyframe $\mathbf{k}$ is redundant within a set $\mathbf{K}$ if its removal does not affect the loop closure edge set $\mathcal{L}$ and does not create discontinuities in the map representation:

$$\bar{\mathcal{L}} \equiv \mathcal{L} = \left\{ \langle i, j \rangle \mid f(\mathbf{d}_i, \mathbf{d}_j) > \tau, \ \forall \langle \mathbf{k}_i, \mathbf{k}_j \rangle \in \bar{\mathbf{K}} \times \bar{\mathbf{K}} \right\},$$

$$\text{subject to} \quad \delta_l \leqslant \|\mathbf{x}_i - \mathbf{x}_{i+1}\|_2 \leqslant \delta_u, \ \forall \mathbf{x} \in \mathbf{X}, \tag{8}$$

where $\bar{\mathbf{K}} = \mathbf{K} \setminus \{\mathbf{k}\}$. The spatial constraint in Eq. (8) ensures comprehensive map coverage and that consecutive odometry edges can be computed safely. The lower and upper pose distances, δ_l and δ_u, usually range from 1 to 5 meters[23]. We quantify redundancy in a keyframe set using a metric that captures the similarity between consecutive keyframes, utilizing any similarity function f_σ, provided by the corresponding descriptor extraction framework:

$$\rho_\tau(\mathbf{K}) = \frac{1}{N} \sum_{i=0}^{N-1} f_\sigma(\mathbf{k}_i, \mathbf{k}_{i+1}), \tag{9}$$

where $0 < \rho_\tau(\mathbf{K}) \leqslant 1$ and N is the number of keyframes in $\mathbf{K}$. Higher values indicate greater redundancy.

B. Information Preservation in Keyframes

Let $F : \mathbb{R}^3 \to \mathbb{R}^M$, represent the descriptor extraction function, either learning-based[23][28] or handcrafted[24][29], which maps each observation from 3-D space to an M-dimensional representation. This function does not directly depend on the pose $\mathbf{x}_t$, but rather on the input observation $\mathbf{z}_t$ (e.g. LiDAR scan), which itself depends on the pose through the sensor model $\mathbf{z}_t \propto h(\mathbf{x}_t, \mathbf{W})$, where $\mathbf{W}$ represents the world. To understand how the descriptors are sensitive to pose changes, we can compute the Jacobian $\mathbf{J}$ of F with respect to poses $\mathbf{x}$ through the chain rule $(\partial \mathbf{F}\, \partial \mathbf{z})/(\partial \mathbf{z}\, \partial \mathbf{x})$. Explicitly deriving the Jacobian can be impractical because of the intricate nature of functions like deep neural networks. Consequently, we employ numerical approximations to estimate the rate of change. Moreover, since poses are encompassed within $SE(3)$, we reduce the dimensional complexity by using the Euclidean norm for distance measurement instead of individual axis derivatives. It is essential to use yaw–invariant descriptors, as suggested in[23][24], to ensure the Jacobian calculations are meaningful by preventing orientation changes. For a comprehensive discussion of these assumptions and derivations, we refer the reader to our previous work[6], which provides detailed explanations. Considering the descriptors $\mathbf{d} \in \mathbb{R}^M$ as random variables and poses $\mathbf{x} \in SE(3)$ as samples, the product $\mathbf{J}_{\mathbf{F}}^{\top}\mathbf{J}_{\mathbf{F}}$ estimates the covariance matrix:

$$\mathbf{J}_{\mathbf{F}}^{\top}\mathbf{J}_{\mathbf{F}} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right)^{\top} \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}, \tag{10}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues and $\mathbf{V}$ is the matrix of eigenvectors from the decomposition. The eigenvectors represent the principal directions of variation, and the eigenvalues represent their magnitudes. The descriptors are transformed using the eigenvectors and scaled by the square root of the eigenvalues, $\mathbf{D}' = \sqrt{\mathbf{\Lambda}} \cdot \mathbf{V} \cdot \mathbf{D}$,

aligning them with the main directions of variability. Therefore, we can define the *information preservation* term for a keyframe set as:

$$\pi_\tau(\mathbf{K}) = -\frac{1}{N} \sum_{i=0}^{N-1} f_\delta(\mathbf{d}'_i, \mathbf{d}'_{i+1}), \tag{11}$$

where $-1 \leqslant \pi_\tau(\mathbf{K}) < 0$, f_δ is the distance function between descriptors, and $\mathbf{d}' \in \mathbf{D}'$ are the transformed descriptors. Higher values of π_τ indicate that the poses better preserve the variability in the descriptor space. Eigenvectors $\mathbf{v}_k$ denote directions of maximal variability, and the k-th eigenvalue λ_k quantifies the variance described by each eigenvector. Larger eigenvalues indicate more significant patterns of variability.

In summary, while both ρ_τ and π_τ utilize a similarity or distance function, they differ in objectives: the redundancy term focuses on local redundancy within the keyframe set, while the information preservation term evaluates the preservation of information structure with respect to pose changes. The distinct goals lead to different interpretations of relationships between descriptive vectors within the keyframe set.
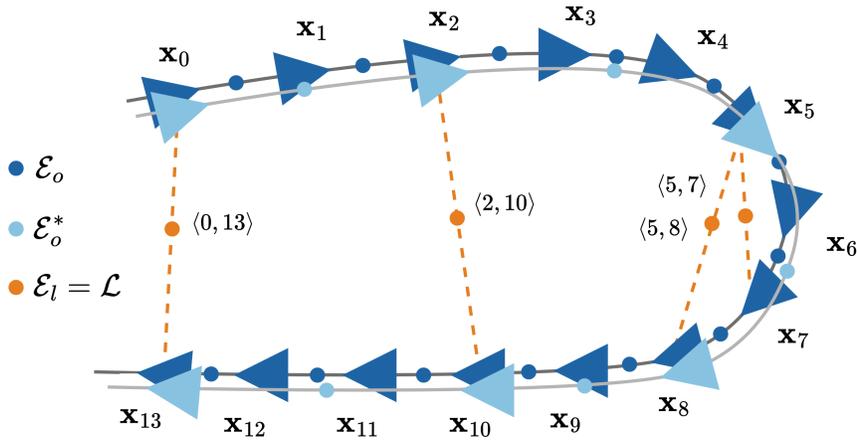
Figure 2. Redundant keyframes effect. Example of removing redundant keyframes (light blue). The updated odometry edge set, $\mathcal{E}_o^*$, influences the pose graph optimization, while the sparsified loop closure edges, $\bar{\mathcal{L}} = \mathcal{E}_l \backslash \langle 5, 7 \rangle$, can further reduce the computational complexity.

C. Sliding Window Optimization

To manage the computational complexity of optimizing a large keyframe set and proactively selecting keyframes for future queries, we introduce a sliding window optimization method. This approach continuously optimizes keyframe sets $\mathbf{K}_t$ over the mission duration T, approximating the optimal keyframe set as:

$$\mathbf{K}_S^* \cong \bigcup_{t \in [0,T]} \mathbf{K}_t^*, \tag{12}$$

where $\mathbf{K}_t^*$ is the minimal subset of $\mathbf{K}_t$ that retains maximum information. To compute this subset, we use the redundancy and information preservation terms defined earlier.

The process begins by initializing a window keyframe set $\mathbf{K}_t$ containing N keyframes. The time step t advances once N new keyframes are available, and the optimization converges to the optimal window keyframe set $\mathbf{K}_t^*$. To find this optimum, we generate all possible subsets of $\mathbf{K}_t$, forming the power set $\mathbb{P}(\mathbf{K}_t)$ with cardinality $|\mathbb{P}(\mathbf{K}_t)| = 2^N$. To reduce computational complexity, we apply constraints on the power set, retaining only subsets that satisfy minimum and maximum distances between consecutive poses, as per Eq. (8), reducing the power set by 5–10 times. The constrained power set can be denoted as:

$$\bar{\mathbb{P}}(\mathbf{K}_t) = \left\{ \mathbf{K}_t^{\mathbb{S}} \in \mathbb{P}(\mathbf{K}_t) | \delta_l \leqslant \|\mathbf{x}_i - \mathbf{x}_{i+1}\|_2 \leqslant \delta_u, \forall \mathbf{k} \in \mathbf{K}_t^{\mathbb{S}} \right\}, \tag{13}$$

where δ_l and δ_u are the lower and upper limits for the distance between poses. The window optimization problem can be formulated as:

$$\mathbf{K}_t^* = \underset{\mathbf{K}_t^{\mathbb{S}}}{\arg\min} \; \left(\rho_\tau \left(\mathbf{K}_t^{\mathbb{S}} \right) + \alpha \right) \big/ \left(\pi_\tau (\mathbf{K}_t^{\mathbb{S}}) - \beta \right) \tag{13}$$

$$\text{where} \;\; \alpha, \beta > 0 \text{ and } \mathbf{K}_t^{\mathbb{S}} \in \bar{\mathbb{P}}(\mathbf{K}_t). \tag{14}$$

The minimization problem is solved through an exhaustive search, evaluating each subset's information matrix $\mathbf{J}_\mathbf{F}^\top \mathbf{J}_\mathbf{F}$ and quantifying its effectiveness using the information preservation and redundancy terms, selecting the subset with the best combined score.

IV. Experimental Setup and Results

To support the claims of our second contribution, on how the sampling interval affects performance, we conducted a series of experiments with the following setup. Descriptor extraction was performed using two approaches: the learning-based `OverlapTransformer (OT)`[23], which generates a 1×256 feature vector, and the hand-crafted `Scan Context (SC)` [24], which produces a 20×60 feature representation. These choices demonstrate the descriptor-agnostic nature of our method and its compatibility with both learned and hand-crafted descriptors. We benchmark our approach against constant sampling intervals of 1, 2, and 3 meters[15][16][17][30], as well as adaptive sampling strategies based on LiDAR spaciousness[18] and entropy[20]. All experiments were conducted on a 14th Gen Intel Core i9 system with 128GB of RAM, where the sampling window optimization ($N = 10$) is solved in approximately 5–15 milliseconds, making it suitable for real-time sampling.
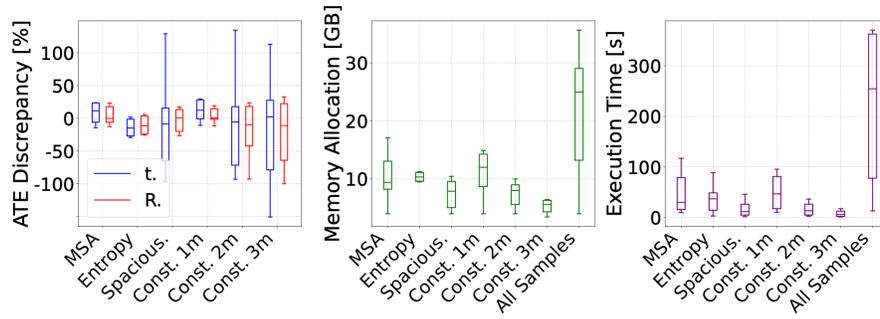
Figure 3. Box plot comparisons. Translational (t) and Rotational (R) Absolute Trajectory Error (ATE) difference of every method compared to the baseline of All Samples, as well as the Memory Allocation in Gigabytes (GB) and the total Execution Time in seconds (s).

A. Quantitative and Qualitative Analysis

For the batch experiments in Table I, we evaluate our approach on three datasets: `KITTI Odometry`[31], `MulRan`[32], and `Apollo-SouthBay`[33], representing diverse environments, including urban areas, rural landscapes, and complex structures like bridges and tunnels. Keyframes are sampled for all methods, and the pose graph is constructed using `GTSAM`[7] with odometry provided from `KISS-ICP`[34]. Loop closure candidates are identified using `OT` or `SC` descriptors with a similarity threshold of 0.8. Candidates within a 3-meter radius of the ground truth are classified as true positives and processed with `small_gicp`[35] to estimate relative transformations. Matches with registration residuals below 0.3 meters are verified and added as edges to the pose graph, which is optimized using the Levenberg–Marquardt algorithm.

	Datasets	KITTI Odometry (Seq. 05/06)			MulRan (DCC/KAIST)			Apollo–SB (SanJose/Columbia)		
	Metrics	t. ATE [%]	R. ATE [%]	FPR [%]	t. ATE [%]	R. ATE [%]	FPR [%]	t. ATE [%]	R. ATE [%]	FPR [%]
ScanContext	All Samples	72.7/76.1	94.4/49.0	06.6/06.4	30.2/69.4	24.1/56.3	08.8/20.8	92.0/93.7	93.7/92.3	29.4/06.8
	Const. 1.0m	90.5/76.2	95.9/52.6	04.3/04.1	63.2/88.6	55.2/70.7	11.6/24.0	84.3/92.5	87.4/92.0	26.1/05.5
	Const. 2.0m	86.2/62.3	94.0/46.4	06.0/05.9	72.1/83.3	69.4/72.0	12.1/13.3	66.7/79.2	59.5/78.6	25.9/03.3
	Const. 3.0m	79.8/60.1	77.4/42.3	10.5/09.9	58.7/79.5	60.6/69.2	19.6/11.7	54.4/77.1	40.9/69.8	24.0/03.1
	Entropy-based	66.9/46.7	93.4/48.3	09.1/05.5	68.7/71.8	69.0/66.8	14.2/15.6	90.1/80.3	89.8/82.9	24.5/03.2
	Spaciousness	77.8/55.6	87.6/46.7	09.3/05.9	67.9/70.7	68.5/66.1	10.0/16.5	79.7/79.6	84.2/82.1	25.2/06.0
	MSA (Ours)	90.3/76.4	96.1/53.0	03.3/03.4	71.9/91.9	70.1/76.6	06.8/12.3	89.9/87.5	91.4/80.3	23.3/02.7
OverlapTrans	All Samples	75.7/78.6	95.9/51.1	05.1/05.6	29.8/67.4	24.9/59.5	08.0/21.2	96.2/98.1	95.3/97.5	30.7/07.5
	Const. 1.0m	94.6/78.7	97.3/55.3	06.2/07.0	59.3/87.1	56.4/71.0	20.2/23.8	85.8/97.1	84.1/97.2	25.1/04.1
	Const. 2.0m	84.4/44.9	97.2/23.7	04.3/03.1	70.0/80.9	68.4/73.7	07.6/11.6	06.8/75.9	06.8/77.2	25.3/02.3
	Const. 3.0m	90.5/22.6	87.9/13.8	06.0/09.5	63.6/87.7	64.6/79.1	04.4/07.7	00.0/83.6	00.0/84.4	16.6/02.4
	Entropy-based	67.9/24.1	93.0/09.1	06.1/05.4	69.2/68.7	69.5/63.1	16.4/14.6	77.6/69.4	76.2/72.6	21.0/03.1
	Spaciousness	85.8/35.5	97.2/39.1	06.8/06.9	68.4/68.6	68.1/63.1	08.5/16.1	03.5/67.9	03.5/71.7	24.1/04.9
	MSA (Ours)	93.2/78.8	97.5/55.5	04.0/04.9	70.1/91.4	69.7/77.8	08.2/11.4	88.7/84.0	88.3/85.1	21.5/01.6

Table I. Translational (t) and Rotational (R) Absolute Trajectory Error Improvement after PGO and the False Positive Rate (FPR)*

The underlined results correspond to the best performing constant sampling interval, while the bold results correspond to the best adaptive sampling method.

Table I summarizes the results for all datasets, descriptors, and methods. The metrics include the Absolute Trajectory Error (ATE) for both translation and rotation, presented as the percent improvement in the `KISS-ICP` trajectory after pose graph optimization with each method. Additionally, the False Positive Ratio (FPR) for descriptor matching is reported, as sampling impacts place recognition performance[6]. The results demonstrate that the proposed MSA consistently delivers the best performance, while constant intervals yield varying results across different environments, highlighting the absence of a universal fixed sampling interval. In Fig. 3, the data from Table I are presented as box plots,

illustrating the deviation of each method's ATE performance compared to using all samples, alongside variations in memory usage and total pipeline execution time. The proposed approach sustains overall improvements with minimal performance loss in both translation and rotation, while significantly reducing memory usage and execution time. Although other methods also reduce memory and processing overhead, their performance lacks robustness and varies considerably. Finally, Figs. 1 and 4 compare ground truth poses, raw `KISS-ICP` poses, and corrected trajectories for MSA and the entropy-based method post optimization. Across all illustrated trajectories, the proposed approach achieves consistently lower translation and rotation RPE than other methods, with loop closure segments closely aligning with the ground truth, demonstrating its ability to retain essential edges.
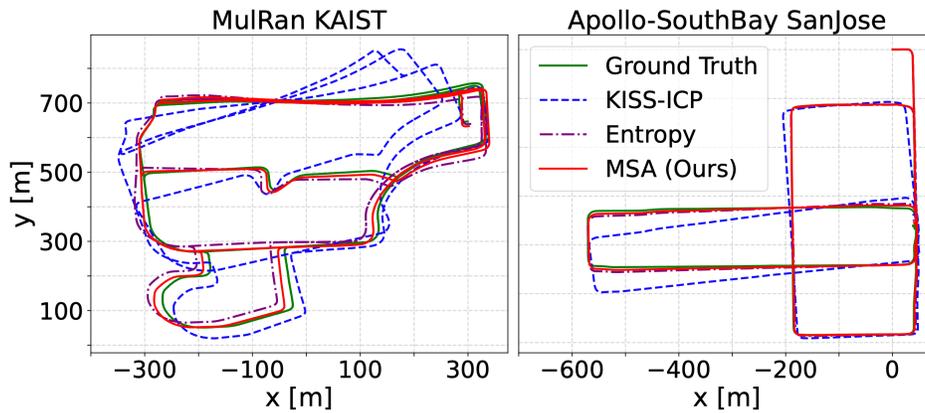
Figure 4. Trajectory comparisons. Comparison of the ground truth, raw `KISS-ICP`, and the sampled poses after the pose graph optimization for the proposed Minimal Subset Approach (MSA) and the entropy-based approach on the `KAIST` sequence of the `MulRan` dataset and the `SanJoseDowntown` sequence of the `Apollo-SouthBay` dataset.

B. Larger-Scale Evaluations

To support our last contribution, we conduct more realistic, larger-scale, online experiments. Poses are processed sequentially, simulating real-time operation. Keyframes are sampled online using the previously mentioned methods, appended to the pose graph, and optimized at each step with `iSAM2`[8]. Simultaneously, `OT` descriptors are queried against past keyframes to detect loop closures, which are verified with `small_gicp` and integrated into the graph.

1. *Urban Driving Scenario*

For this scenario we use one of the datasets from the batch experiments, `Apollo-SouthBay`, focusing on the `SunnyvaleBig-loop` sequence, which spans over 100 kilometers. Table II reports the translation and rotation ATE, FPR, total allocated memory, and processing time. Notably, when using all samples, the system exhausts memory, approximately 70% into the sequence, preventing the calculation of the metrics. The other sampling methods reduce memory usage while maintaining excellent performance. This is likely due to the sequence revisiting the same locations

multiple times, enabling frequent loop closure edges that mitigate accumulated drift. In Fig. 5, the optimized trajectory of the entropy–based method is compared to MSA, with a color gradient representing translational RPE. The visualization highlights MSA's superior performance, achieved with lower memory allocation and a reduced false positive rate. Lastly, Fig. 6 shows the memory usage over time, illustrating how sampling methods scale more efficiently compared to retaining all samples or using a constant 1–meter interval.
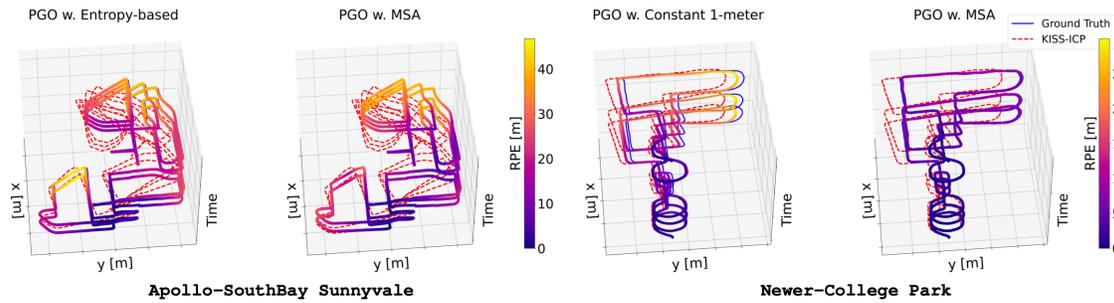
Figure 5. Trajectory comparisons. The ground truth and raw `KISS-ICP` poses compared against the sampled and optimized poses using the proposed Minimal Subset Approach (MSA) and the entropy–based method for the `Sunnyvale` and the constant 1–meter approach for the `Park`. The color gradient represents the Relative Pose Error (RPE). Grid spacing corresponds to 0.5x0.5 km^2 for the left figure and 50x50 m^2 for the right figure.

	Metrics	t. ATE	R. ATE	FPR	MEM.	TIME
Apollo-SB Sunnyvale	All Samples	nan	nan	nan	>128GB	nan
	Constant 1.0m	98.5%	98.4%	12%	90.9GB	7340s
	Constant 2.0m	98.1%	97.3%	17%	47.2GB	2264s
	Constant 3.0m	97.9%	97.0%	28%	35.4GB	1515s
	Entropy-based	95.5%	93.4%	15%	51.0GB	4945s
	Spaciousness	94.9%	97.5%	24%	38.3GB	1752s
	MSA (default)	**98.3%**	**94.1%**	**12%**	41.2GB	1951s
	MSA (w/o π_τ)	94.7%	93.2%	13%	32.4GB	1438s
	MSA (w/o ρ_τ)	98.9%	96.4%	12%	43.7GB	2022s
	MSA ($\alpha = 10$)	96.5%	93.9%	12%	37.6GB	1764s
	MSA ($\beta = 10$)	98.6%	95.8%	12%	42.0GB	2004s
Newer-College Park	All Samples	93.9%	78.9%	02%	65.8GB	6742s
	Constant 0.1m	89.7%	71.4%	03%	63.2GB	4987s
	Constant 0.3m	79.9%	58.5%	08%	28.9GB	1262s
	Constant 1.0m	60.7%	43.1%	23%	09.2GB	165s
	Entropy-based	87.3%	65.8%	05%	26.3GB	1401s
	Spaciousness	72.5%	52.5%	14%	27.2GB	1096s
	MSA (default)	**90.1%**	**67.0%**	**02%**	38.7GB	1884s
	MSA (w/o π_τ)	81.7%	41.4%	03%	12.8GB	346s
	MSA (w/o ρ_τ)	97.7%	81.4%	02%	64.7GB	6226s
	MSA ($\alpha = 10$)	86.9%	64.4%	02%	29.2GB	1498s
	MSA ($\beta = 10$)	93.2%	69.4%	02%	42.5GB	2791s

Table II. Results for the Larger-Scale Evaluations and the Ablation Study for the proposed approach

2. Campus Scenario

To provide additional insights, we evaluate a different dataset, the `Newer-College`[36], and more specifically the `Park` sequence, a 30-minute mapping session with multiple loops. This scenario, resembling a mobile ground robot, differs

significantly from `Apollo-SouthBay` due to its more confined environment and denser frame sampling, approximately 0.1 meters compared to 1 meter. These differences highlight how distinct missions impose unique requirements.

From Table II, constant intervals show that, a 1-meter sampling, severely underperforms in this scenario, retaining only 7% of the frames. This leads to significant performance losses in both ATE and FPR. To address this, we include lower constant intervals of 0.1 and 0.3 meters for comparison. While these improve performance, our proposed MSA outperforms them by prioritizing informative sampling, providing more valuable nodes to the pose graph. The entropy-based method closely follows, emphasizing the importance of spatial features in sampling. Here, we note that both the spaciousness- and entropy-based methods required additional tuning for this scenario. As in the previous experiment, Fig. 5 compares the optimized trajectory of MSA with the 1-meter approach, using a color gradient to depict translational RPE, which highlights MSA's superior metric localization performance. Figs 6 and 7 further compare memory allocation over time and processing times. Fig. 7 presents box plots for incremental PGO (left) and Loop Closure Detection (right) processing times per step. Notably, both tasks exhibit significantly high processing times, potentially causing delays in real-world missions. While lower fixed intervals (0.1 and 0.3 meters) reduce processing time, 0.3-meter and 1-meter intervals show subpar performance, and the 0.1-meter interval demands substantial memory allocation since it is very close to the native sampling of the dataset. Interestingly, methods achieving good ATE performance, such as the entropy-based and MSA, demonstrate higher outliers in the incremental PGO, suggesting that valuable edges are added to the pose graph, contributing to their better performance.
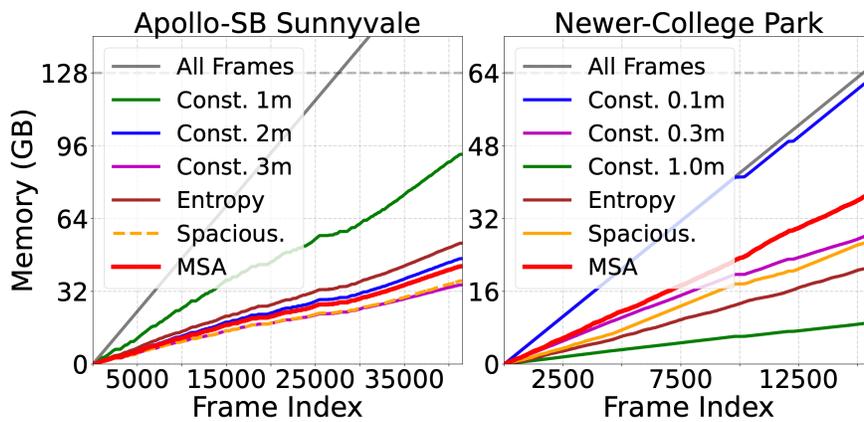
Figure 6. Memory allocation. Memory against each acquired frame for the different sampling methods. Retaining all samples exceeds system memory of 128GB, demonstrating the need for efficient and scalable algorithms.
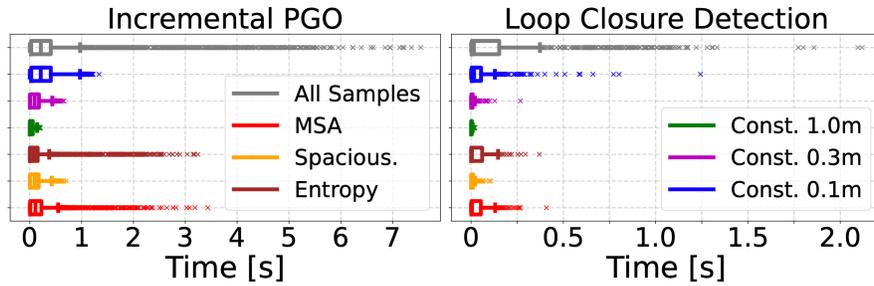
Figure 7. Processing times. The distributions of the incremental pose graph optimization convergence (left), and the loop closure detection (right) time, per new frame, for the `Park` sequence of the `Newer-College` dataset.

C. Ablation Study

In all experiments, we maintained constant hyperparameters ($\alpha = \beta = 1$), demonstrating that the proposed method adapts seamlessly to different environments and descriptors without requiring manual tuning. To further analyze the impact of each optimization term, we conducted an ablation study, with results shown in Table II. Specifically, we tested scenarios where only one term, ρ_τ or π_τ, was used, and scenarios where the weight of one term was increased to 10 while keeping the other at 1. The results indicate that emphasizing the *information preservation* term or using it exclusively improves performance but leading to increased number of keyframes sampled. Conversely, prioritizing *redundancy minimization* significantly reduces memory usage but negatively impacts performance. Based on these, practitioners can decide whether to prioritize memory or performance, depending on their specific application requirements.

V. Discussion

The results highlight that no universal constant sampling interval is suitable for all scenarios. While a constant 1-meter interval performs well in datasets similar to the `Urban Driving Scenario`, showing minimal deviations in Fig. 3, it performed poorly in the mobile robotics `Campus Scenario`. This reinforces our core argument: adaptive methods that account for both place recognition performance and pose graph optimization are essential. Naive methods that rely on spatial heuristics, such as spaciousness- and entropy-based approaches, require extensive tuning and prior environmental knowledge, limiting their applicability. Our proposed MSA consistently excels across all scenarios without the need for parameter tuning, demonstrating robustness in this traditional loop closure setup. It is effective with both widely used handcrafted and learning-based descriptors, showcasing adaptability to different datasets and descriptors. As future work, we aim to test our method with full global localization pipelines, such as `RING++`[37] or `LCDNet`[38].

VI. Conclusions

In conclusion, as the robotics community advances towards larger-scale and long-term missions, addressing system limitations with efficient and scalable solutions becomes crucial. This article introduced the Minimal Subset Approach

(MSA) to address the combined challenges of loop closure detection and pose graph optimization by reducing redundant samples while maintaining robust performance. Experimental evaluations across multiple datasets show that MSA excels in terms of FPR, as well as in achieving superior ATE and RPE after pose graph optimization. Unlike other adaptive sampling methods, MSA requires minimal to no parameter tuning, while using a fraction of the system's memory and maintaining comparable computational times.

Data Availability

This paper is supported by code and demonstration files, available at https://github.com/LTU-RAI/opt-key.git

References

1. [a, b]*Denniston C, Chang Y, Reinke A, Ebadi K, Sukhatme GS, et al. (2022). "Loop Closure Prioritization for Efficient and Scalable Multi-Robot SLAM". IEEE Robotics and Automation Letters. 7: 9651–9658.*

2. [^]*Yin P, Zhao S, Lai H, Ge R, Zhang J, et al. (2023). "AutoMerge: A Framework for Map Assembling and Smoothing in City-Scale Environments". IEEE Transactions on Robotics. 39 (5): 3686–3704. doi:10.1109/TRO.2023.3290448.*

3. [a, b]*Stathoulopoulos N, Lindqvist B, Koval A, Agha-Mohammadi AA, Nikolakopoulos G. (2024). "FRAME: A Modular Framework for Autonomous Map Merging: Advancements in the Field". IEEE Transactions on Field Robotics. 1: 1–26. doi:10.1109/TFR.2024.3419439.*

4. [a, b]*Ebadi K, Bernreiter L, Biggie H, Catt G, Chang Y, et al. (2024). "Present and Future of SLAM in Extreme Environments: The DARPA SubT Challenge". IEEE Transactions on Robotics. 40: 936–959. doi:10.1109/TRO.2023.3323938.*

5. [^]*Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, et al. (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.*

6. [a, b, c, d, e, f]*Stathoulopoulos N, Sumathy V, Kanellakis C, Nikolakopoulos G. Why Sample Space Matters: Keyframe Sampling Optimization for LiDAR-based Place Recognition. 2024. Available from: https://arxiv.org/abs/2410.02643*

7. [a, b]*Dellaert F, GTSAM Contributors. Borglab/gtsam. Georgia Tech Borg Lab 2022. doi:10.5281/zenodo.5794541*

8. [a, b]*Kaess M, Johannsson H, Roberts R, Ila V, Leonard J, et al. (2011). "iSAM2: Incremental Smoothing and Mapping with Fluid Relinearization and Incremental Variable Reordering". In: 2011 IEEE international conference on robotics and automation (ICRA). pp. 3281–3288.*

9. [^]*Kummerle R, Grisetti G, Strasdat H, Konolige K, Burgard W. (2011). "g2o: A General Framework for Graph Optimization". In: 2011 IEEE international conference on robotics and automation (ICRA).: IEEE pp. 3607–3613.*

10. [^]*Huang G, Kaess M, Leonard JJ. (2013). "Consistent Sparsification for Graph Optimization". In: 2013 european conference on mobile robots. pp. 150–157.*

11. [^]*Bai F, Vidal-Calleja T, Grisetti G. (2021). "Sparse Pose Graph Optimization in Cycle Space". IEEE Transactions on Robotics. 37 (5): 1381–1400. doi:10.1109/TRO.2021.3050328.*

12. △*Nam J, Hyeon S, Joo Y, Noh D, Shim H. (2024). "Spectral Trade-Off for Measurement Sparsification of Pose-Graph SLAM". IEEE Robotics and Automation Letters. 9 (1): 723–730. doi:10.1109/LRA.2023.3337590.*

13. △*Guadagnino T, Di Giammarino L, Grisetti G. (2022). "HiPE: Hierarchical Initialization for Pose Graphs". IEEE Robotics and Automation Letters. 7: 287–294. doi:10.1109/LRA.2021.3125046.*

14. △*Tazaki Y. (2022). "A Spanning Tree-Based Multi-Resolution Approach for Pose-Graph Optimization". IEEE Robotics and Automation Letters. 7 (4): 10033–10040. doi:10.1109/LRA.2022.3185779.*

15. a, b*Shan T, Englot B, Meyers D, Wang W, Ratti C, et al. (2020). "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping". In: IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 5135–5142.*

16. a, b*Huang Y, Shan T, Chen F, Englot B. (2022). "DiSCo-SLAM: Distributed Scan Context-Enabled Multi-Robot LiDAR SLAM with Two-Stage Global-Local Graph Optimization". IEEE Robotics and Automation Letters. 7 (2): 1150–1157. doi:10.1109/LRA.2021.3138156.*

17. a, b, c*Chang Y, Ebadi K, Denniston CE, Ginting MF, Rosinol A, et al. (2022). "LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments". IEEE Robotics and Automation Letters. 7 (4): 9175–9182. doi:10.1109/LRA.2022.3191204.*

18. a, b*Chen K, Lopez BT, Agha-mohammadi AA, Mehta A. (2022). "Direct LiDAR Odometry: Fast Localization With Dense Point Clouds". IEEE Robotics and Automation Letters. 7 (2): 2000–2007. doi:10.1109/LRA.2022.3142739.*

19. △*Kim B, Jung C, Shim DH, Agha−mohammadi AA. (2023). "Adaptive Keyframe Generation based LiDAR Inertial Odometry for Complex Underground Environments". In: 2023 IEEE international conference on robotics and automation (ICRA). pp. 3332−3338. doi:10.1109/ICRA48891.2023.10161207.*

20. a, b*Zeng Q, Liu D, Zhou Y, Peng Y. (2023). "Entropy-based Keyframe Established and Accelerated Fast LiDAR Odometry and Mapping". ITOEC 2023 - IEEE 7th Information Technology and Mechatronics Engineering Conference. 7: 347–354. doi:10.1109/ITOEC57671.2023.10291983.*

21. △*Ou F, Li Y, Miao Z, Zhou J. (2021). "Lidar Odometry Key Frame Selection Based on Displacement Vector Similarity". Chinese Control Conference, CCC. 2021-July: 3588–3593. doi:10.23919/CCC52363.2021.9549411.*

22. △*Grisetti G, Kümmerle R, Stachniss C, Burgard W. (2010). "A Tutorial on Graph-Based SLAM". IEEE Intelligent Transportation Systems Magazine. 2 (4): 31–43. doi:10.1109/MITS.2010.939925.*

23. a, b, c, d, e*Ma J, Zhang J, Xu J, Ai R, Gu W, et al. (2022). "OverlapTransformer: An Efficient and Yaw-Angle-Invariant Transformer Network for LiDAR-Based Place Recognition". IEEE Robotics and Automation Letters. 7 (3): 6958–6965. doi:10.1109/LRA.2022.3178797.*

24. a, b, c, d*Kim G, Choi S, Kim A. (2022). "Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments". IEEE Transactions on Robotics. 38 (3): 1856–1874. doi:10.1109/TRO.2021.3116424.*

25. △*Yin H, Xu X, Lu S, Chen X, Xiong R, et al. (2024). "A Survey on Global LiDAR Localization: Challenges, Advances and Open Problems". International Journal of Computer Vision. 132 (8): 3139–3171. doi:10.1007/s11263-024-02019-5.*

26. △*Segal A, Haehnel D, Thrun S. (2009). "Generalized-ICP". In: Robotics: Science and systems.: Seattle, WA p. 435.*

27. ^Placed JA, Strader J, Carrillo H, Atanasov N, Indelman V, et al. (2023). "A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers". IEEE Transactions on Robotics. 39 (3): 1686–1705. doi:10.1109/TRO.2023.3248510.

28. ^Stathoulopoulos N, Saucedo MAV, Koval A, Nikolakopoulos G. (2024). "RecNet: An Invertible Point Cloud Encoding through Range Image Embeddings for Multi-Robot Map Sharing and Reconstruction". In: 2024 IEEE international conference on robotics and automation (ICRA). pp. 4883–4889. doi:10.1109/ICRA57147.2024.10611602.

29. ^Yuan C, Lin J, Liu Z, Wei H, Hong X, et al. (2024). "BTC: A Binary and Triangle Combined Descriptor for 3-D Place Recognition". IEEE Transactions on Robotics. 40: 1580–1599. doi:10.1109/TRO.2024.3353076.

30. ^Zhao S, Zhang H, Wang P, Nogueira L, Scherer S. (2021). "Super Odometry: IMU-centric LiDAR-Visual-Inertial Estimator for Challenging Environments". In: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 8729–8736. doi:10.1109/IROS51168.2021.9635862.

31. ^Geiger A, Lenz P, Urtasun R. (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361.

32. ^Kim G, Park YS, Cho Y, Jeong J, Kim A. (2020). "MulRan: Multimodal Range Dataset for Urban Place Recognition". In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 6246–6253. doi:10.1109/icra40945.2020.9197298.

33. ^Lu W, Zhou Y, Wan G, Hou S, Song S. (2019). "L3-Net: Towards Learning Based LiDAR Localization for Autonomous Driving". In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6382–6391. doi:10.1109/cvpr.2019.00655.

34. ^Vizzo I, Guadagnino T, Mersch B, Wiesmann L, Behley J, et al. (2023). "KISS-ICP: In Defense of Point-to-Point ICP − Simple, Accurate, and Robust Registration If Done the Right Way". IEEE Robotics and Automation Letters. 8 (2): 1029–1036. doi:10.1109/lra.2023.3236571.

35. ^Koide K. (2024). "small_gicp: Efficient and parallel algorithms for point cloud registration". Journal of Open Source Software. 9 (100): 6948. doi:10.21105/joss.06948.

36. ^Ramezani M, Wang Y, Camurri M, Wisth D, Mattamala M, et al. (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). pp. 4353–4360. doi:10.1109/iros45743.2020.9340849.

37. ^Xu X, Lu S, Wu J, Lu H, Zhu Q, et al. (2023). "RING++: Roto-Translation Invariant Gram for Global Localization on a Sparse Scan Map". IEEE Transactions on Robotics. 39 (6): 4616–4635. doi:10.1109/tro.2023.3303035.

38. ^Cattaneo D, Vaghi M, Valada A. (2022). "LCDNet: Deep Loop Closure Detection and Point Cloud Registration for LiDAR SLAM". IEEE Transactions on Robotics. 38 (4): 2074–2093. doi:10.1109/tro.2022.3150683.

Declarations