# Greedy-Based Feature Selection for Efficient LiDAR SLAM

Jianhao Jiao[1], Yilong Zhu[1], Haoyang Ye[1], Huaiyang Huang[1], Peng Yun[1], Linxin Jiang[2],
Lujia Wang[2], Ming Liu[1]

*Abstract*— Modern LiDAR-SLAM (L-SLAM) systems have shown excellent results in large-scale, real-world scenarios. However, they commonly have a high latency due to the expensive data association and nonlinear optimization. This paper demonstrates that actively selecting a subset of features significantly improves both the accuracy and efficiency of an L-SLAM system. We formulate the feature selection as a combinatorial optimization problem under a cardinality constraint to preserve the information matrix's spectral attributes. The stochastic-greedy algorithm is applied to approximate the optimal results in real-time. To avoid ill-conditioned estimation, we also propose a general strategy to evaluate the environment's degeneracy and modify the feature number online. The proposed feature selector is integrated into a multi-LiDAR SLAM system. We validate this enhanced system with extensive experiments covering various scenarios on two sensor setups and computation platforms. We show that our approach exhibits low localization error and speedup compared to the state-of-the-art L-SLAM systems. To benefit the community, we have released the source code: `https://ram-lab.com/file/site/m-loam`.

## I. INTRODUCTION

### A. Motivation

State estimation is a classic and fundamental problem in robotics [1]. Over the past decades, LiDARs have attracted much attention from the simultaneous localization and mapping (SLAM) community due to their accuracy and reliability in range measurements. Recent work [2]–[6] has pushed LiDAR-SLAM (L-SLAM) systems that are accurate and robust. However, L-SLAM systems commonly present a high latency on a variety of on-board processors with limited computation resources. This issue is critical if the scale of SLAM becomes large, or modules such as high-level decision making are integrated. Thus, towards real-time SLAM for diverse applications, such systems must exhibit low latency (time delay between input and output) along with the preservation of their accuracy and robustness.

L-SLAM comprises two major computational tasks in L-SLAM: data association and optimization. Data association indicates feature matching between the current frames to the reference frames, while optimization solves the pose parameters by maximizing a likelihood function given a set of constraints. Compared to visual features such as SIFT [7] and ORB [8], matching 3D features is known to be less accurate [9], thus producing much higher outlier rates. To enforce accuracy, most L-SLAM systems exploit thousands of features to solve a large nonlinear least-squares (NLS) problem. However, this scheme presents significant drawbacks. The data association has to perform numerous nearest-neighbor queries to match correspondences, which is commonly time-consuming. Given plentiful measurements, the computational complexity of the optimization based on gradient descent also grows quadratically.

A prevalent solution to bound the complexity is to perform data sampling. For instance, many LiDAR-based object detectors [10]–[12] leverage the farthest point sampling (FPS) [13] to process input points. However, classic sampling methods do not consider downstream tasks specifically. Taking FPS as an example, it selects a subset of points with the objective to achieve a maximal coverage of the input set [14]. But for SLAM, it would be better if the process of point selection conforms to the optimization objective. Ideally, exploiting the set of selected features in optimization should lead to low latency and performance improvements.

### B. Contributions

This paper proposes a general and straightforward feature selection algorithm for L-SLAM systems. We have a crucial observation behind our approach: that not all geometric constraints contribute equally to the localization accuracy. Intuitively, well-conditioned constraints should distribute different directions, constraining the pose from different angles [15]. For instance, orthogonal constraints commonly outweigh their parallel counterparts. The selected features, which are the most valuable/informative to the pose estimation, are defined as **good features** [16], and both the data association and state optimization utilize them only.

This paper extends our previous work on multi-LiDAR SLAM [17]. Multiple LiDARs enable a robot to maximize its perceptual awareness of environments and obtain sufficient measurements, but inevitably increase the processing time. In this paper, we investigate the latency issue. From the traditional perspective, there is a trade-off between the latency and accuracy [18]. But in Section VI, we demonstrate that the proposed feature selection method boosts the accuracy ($> 22\%$ error reduction) and efficiency ($> 30\%$ time reduction) of an L-SLAM system. Furthermore, by evaluating the environment's degeneracy and adaptively setting the number of good features, our method also works well in non-ideal cases. Overall, our work presents the following contributions:

1) We transform the good feature section in LiDAR-based pose estimation into a problem that preserves the spectral property of information matrices.
2) We propose and integrate the feature selection method into a multi-LiDAR SLAM system. We also propose to evaluate the environment's degeneracy and adaptively change the number of good features online.
3) We evaluate our approach under extensive experiments with two sensor setups, and computation platforms in terms of accuracy, robustness, and latency.

## II. RELATED WORK

Scholarly works on SLAM are extensive. Since we focus on the optimal feature selection to improve L-SLAM's efficiency, we review related works on two research topics: feature extraction and selection.

### A. Feature Extraction

Feature extraction is a process to build an informative, compact, and interpretable representation of raw measurements [19]. It has played a crucial role in the front end of many L-SLAM systems to facilitate subsequent tasks. SuMa and its variants [4], [20], [21] convert point clouds into range images and generated surfel-based maps. In contrast, LOAM [2] was proposed to extract features on both edge lines and planar surfaces, and LEGO-LOAM [3] leverages ground features to constrain poses in the vertical direction. The following approaches apply visual detection [22] or directly used dense scanners [5], [23] to enhance the feature extraction in noisy or structureless environments. To decrease the number of features, Zhao et al. [24] presented a probabilistic framework to extract important region of interest by calculating features' densities and distributions. This solution enables the removal of dynamic objects and redundant points in busy urban environments.

All of these methods extract features depending on local geometric structures, but they have not considered the explicit relationship between the pose estimation to select the most useful features. Our system is built on LOAM's framework in feature selection. As a complement to the above appearance-based approaches, our solution identifies a set of good features by utilizing motion information.

### B. Feature Selection

Motion-based feature selection methods have been widely applied in visual SLAM [16], [25]–[27], and many of them are based on the covariance or information matrices that capture uncertainties of poses [28]. These methods formulate the feature selection as a submatrix selection problem and aim to find a subset of features with the objective to preserve the information matrix's spectral attributes. Recent works by Carlone et al. [26] and Zhao et al. [16] have investigated greedy-based algorithms [29] to solve this NP-hard feature selection problem at a polynomial-time complexity.

A common limitation of Carlone's and Zhao's works is that they assure sufficient features to be available. Under this assumption, the pose optimization with a set of good features remains well-conditioned. However, robots sometimes need to work in degraded environments such as textureless regions for cameras and narrow corridors for LiDARs. Therefore, only utilizing good features with a fixed size becomes degenerate. Based on Zhao's feature selection approach, our method additionally evaluates environments' degeneracy online, which enables us to adaptively change the number of good features to avoid the risk of ill conditions.

## III. NONLINEAR LEAST-SQUARES POSE ESTIMATION

We formulate the pose estimation of a L-SLAM system as an maximum likelihood estimation (MLE) [30]:

$$\hat{\mathbf{x}}_K = \arg\max_{\mathbf{x}_K} p(\mathcal{F}_K|\mathbf{x}_K) = \arg\min_{\mathbf{x}_K} f(\mathbf{x}_K, \mathcal{F}_K), \quad (1)$$

where $\mathcal{F}_K$ represents the available features at the $K^{th}$ frame, $\mathbf{x}_K$ is the robot's pose to be optimized, and $f(\cdot)$ is the objective function. Assuming the measurement model to be Gaussian, problem (1) is solved as an NLS problem:

$$\hat{\mathbf{x}}_K = \arg\min_{\mathbf{x}_K} \sum_{i=1}^{N} \rho\big(||\mathbf{r}(\mathbf{x}_K, \mathbf{p}_{Ki})||^2_{\mathbf{W}_i}\big), \quad (2)$$

where $\rho(\cdot)$ is the robust loss [31] and $\mathbf{W}_i$ is the covariance matrix. Problem (2) is equivalently rewritten as [30]

$$\hat{\mathbf{x}}_K = \arg\min_{\mathbf{x}_K} \sum_{i=1}^{N} ||\mathbf{r}(\mathbf{x}_K, \mathbf{p}_{Ki})||^2_{\mathbf{\Sigma}_i}, \quad (3)$$

where $\mathbf{\Sigma}_i^{-1} = \rho'(||\mathbf{r}(\mathbf{x}_{op,K}, \mathbf{p}_{Ki})||^2_{\mathbf{W}_i})\mathbf{W}_i^{-1}$ is the alternative covariance matrix and $\rho'(\cdot)$ is the derivative of $\rho(\cdot)$. (2) is simplified as an iterative reweighed least-squares problem. Iterative methods such as Gauss-Newton or Levenberg-Marquardt can offen be used to solve this problem. These methods locally linearize the objective function by computing the Jacobian w.r.t. $\mathbf{x}_K$ as $\mathbf{J} = \partial f / \partial \mathbf{x}_K$. Given an initial guess, $\mathbf{x}_K$ is iteratively optimized by usage of $\mathbf{J}$ until convergence to find an optimum.

At the final iteration, the least-squares covariance of the state is calculated as $\mathbf{\Xi} = \mathbf{\Lambda}^{-1}$ [32], where $\mathbf{\Lambda} = \mathbf{J}^{\top}\mathbf{J}$ is called the *information matrix*. This equation reveals the relationship between the pose covariance and information matrix. Generally, exploiting plentiful measurements in optimization should minimize poses' uncertainties. This explains why SLAM systems commonly use all available features.

This paper focuses on low-latency applications in which the speed is highly prioritized. It requires us to utilize only a subset of features to accelerate the algorithm. As suggested in [16], we can check whether a feature is selected or not by comparing the gains in spectrum of $\mathbf{\Lambda}$. The word: "spectrum" denotes the set of eigenvalues of a matrix [33].

## IV. METHODOLOGY

This section first formulates the good feature selection problem and introduces a metric to guide the selection. We apply the stochastic-greedy method, which combines the random sampling procedure, to solve this problem in real time. We then extend this algorithm to achieve efficient feature selection. Finally, we propose to evaluate the environment's degeneracy online to avoid ill-conditioning estimation.

## A. Problem Formulation

We denote $N = |\mathcal{F}_K|$ as the number of all available features, $M$ as the maximum number of selected features, and $\mathcal{S}_K$ as the good feature set. We denote $f(\cdot)$ as the metric to quantify the spectral attribute of a matrix. We formute the feature selection problem under a cardinality constraint as

$$\underset{\mathcal{S}_K \subset \mathcal{F}_K}{\arg\max} f\left[\boldsymbol{\Lambda}(\mathcal{S}_K)\right] \quad \text{subject to } |\mathcal{S}_K| \leq M, \quad (4)$$

where $\boldsymbol{\Lambda}(\mathcal{S}_K)$ is the information matrix on the good feature set. There are several options to define $f(\boldsymbol{\Lambda})$: $\text{tr}(\boldsymbol{\Lambda})$ the trace [34], $\lambda_{\min}(\boldsymbol{\Lambda})$ the minimum eigenvalue [26], and $\log \det(\boldsymbol{\Lambda})$ the log determinant [16].

Since problem (4) is NP-hard, we cannot find efficient algorithms to obtain the optimal subset for real-time applications. Fortunately, all these metrics are submodular and monotone increasing [29], allowing the solution to be approximate via greedy methods with a performance guarantee. Zhao et al. [16] experimented with these metrics in bundle adjustment, where the log determinant was demonstrated to have the lowest pose error and computation time. We thus choose the $\log \det(\boldsymbol{\Lambda})$ option as our metric.

## B. Stochastic Greedy Algorithm

The class of greedy methods has been studied to solve problem (4). Here, we introduce the stochastic-greedy algorithm [29], which applies randomized acceleration to avoid brute-force search. The idea is simple: at each round, the current best feature is picked up by examining all elements from a random subset. This is different from the simple greedy approach, which has to search the whole set. We define the size of the random subset as $\frac{N}{M} \log(\frac{1}{\epsilon})$, where $\epsilon$ is the *decay factor*. The time complexity is $O(N \log(\frac{1}{\epsilon}))$, which is independent of $M$. The stochastic-greedy algorithm has been proved to have near-optimal performance in [29]:

**Theorem 1:** Let $f(\cdot)$ be the non-negative, monotone, and submodular function. Setting the size of the random subset as $\frac{N}{M} \log(\frac{1}{\epsilon})$. Denote $\mathcal{S}_K^*$ as the optimal set and $\mathcal{S}_K^\#$ the result found by the stochastic-greedy algorithm. $\mathcal{S}_K^\#$ enjoys the approximation guarantees in expectation:

$$E\left(f\left[\boldsymbol{\Lambda}(\mathcal{S}_K^\#)\right]\right) \geq \underbrace{(1 - 1/e - \epsilon)}_{\text{expected ratio}} f\left[\boldsymbol{\Lambda}(\mathcal{S}_K^*)\right]. \quad (5)$$

## C. Good Feature Selection for Pose Estimation

Based on the theoretical results, we employ the stochastic-greedy to achieve the feature selection for pose estimation. The detailed pipeline is summarized in Algorithm 1.

1) Overview: The algorithm starts with the $\log \det(\cdot)$ metric, the number of good features $M$, the *decay factor* $\epsilon$, the map in the reference frame $\mathcal{M}_K$, the set of all available features in the current frame $\mathcal{F}_K$, and the initial pose $\mathbf{x}_K$. It produces the good feature set $\mathcal{S}_K^\#$.

2) Line 2: The loop is exited if one of the following conditions is satisfied: $M$ good features are found or the computation time exceeds $t_{max}$. The second condition limits the cost of finding good features. Since

---

**Algorithm 1:** Stochastic Greedy-Based Good Feature Selection for NLS Pose Estimation

**Input:** $f(\cdot) \triangleq \log \det(\cdot)$, $M$, $\epsilon$, $\mathcal{M}_K$, $\mathcal{F}_K$, $\mathbf{x}_K$;
**Output:** good feature set $\mathcal{S}_K^\#$;

1 Initialize the set $\mathcal{S}_K^\# = \emptyset$;
2 **while** $|\mathcal{S}_K^\#| < M$ and $t_{comp} < t_{max}$ **do**
3    $\mathcal{R} \leftarrow$ the random subset is obtained by sampling $\frac{N}{M} \log(\frac{1}{\epsilon})$ random elements from $\mathcal{F}_K \backslash \mathcal{S}_K^\#$;
4    **foreach** $\mathbf{p}_i \in \mathcal{R}$ **do**
5       Search the correspondence from $\mathcal{M}_K$;
6       **if** the correspondence is found **then**
7          Compute the residual $\mathbf{r}_i(\check{\mathbf{x}}_K)$;
8          Compute $\boldsymbol{\Lambda}_i = \mathbf{J}_i^\top \boldsymbol{\Sigma}_i^{-1} \mathbf{J}_i$ w.r.t. $\mathbf{p}_i$ where $\mathbf{J}_i$ is the Jacobian of $\mathbf{r}_i(\cdot)$;
9       **else**
10          $\mathcal{R} \leftarrow \mathcal{R} \backslash \{\mathbf{p}_i\}$;   $\mathcal{F}_K \leftarrow \mathcal{F}_K \backslash \{\mathbf{p}_i\}$;
11    $i^* \leftarrow \arg\max_{\mathbf{p}_i \in \mathcal{R}} \log \det\left[\boldsymbol{\Lambda}(\mathcal{S}_K^\#) + \boldsymbol{\Lambda}_i\right]$;
12    $\boldsymbol{\Lambda}(\mathcal{S}_K^\#) \leftarrow \boldsymbol{\Lambda}(\mathcal{S}_K^\#) + \boldsymbol{\Lambda}_{i^*}$;
13    $\mathcal{S}_K^\# \leftarrow \mathcal{S}_K^\# \cup \{\mathbf{p}_{i^*}\}$;   $\mathcal{F}_K \leftarrow \mathcal{F}_K \backslash \{\mathbf{p}_{i^*}\}$;

---

$\log \det(\cdot)$ is submodular with diminishing returns, early termination does not induce much information loss.

3) Lines 4–10: The correspondence of each feature in the random subset $\mathcal{R}$ is found from $\mathcal{M}_K$. The residual is then computed. If a feature has already been visited at previous iterations, we skip these steps.

4) Lines 11–13: The feature which leads to the maximum enhancement of the objective is selected. After that, the information matrix $\boldsymbol{\Lambda}(\mathcal{S}_K^\#)$, $\mathcal{S}_K^\#$, and $\mathcal{F}_K$ are updated.

Furthermore, the process of feature selection implicitly performs outlier rejection: outliers are penalized by the robust loss in (3) with relatively small weights. They contribute less to $\boldsymbol{\Lambda}$ than standard features and will be selected with a low probability. Therefore, selecting good features might reduce the biases between estimates and the ground truth.

## D. Setting the Number of Good Features

Setting a proper size for the good feature set is essential to the system. Previous methods manually set $M$ as a constant value ($M = 100$ [16]) or a fixed ratio of all features ($M = 0.5N$ [26]). These schemes are feasible if sufficient features are always available. But if a robot has to work in non-ideal scenarios such as textureless walls or narrow corridors, utilizing a small set of features is not reliable. On the other hand, if we change the hard-coded number $M$ in a specific situation, it will inevitably increase the cost of deploying and maintaining a SLAM system on real platforms.

It would be better if $M$ were adaptively changed by evaluating the degeneracy online. Inspired by [15], the magnitude of the degeneracy can be quantified by a *factor* $\lambda$. Differently, we define the factor using the log determinant metric as $\lambda = \log \det \boldsymbol{\Lambda}(\mathcal{F}_K)$. Computing the information matrix on the full feature set is time-consuming. Since the
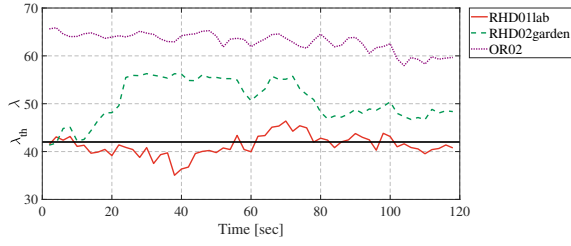
Fig. 1. The value of the *degeneracy factor* $\lambda$ on different sequences.



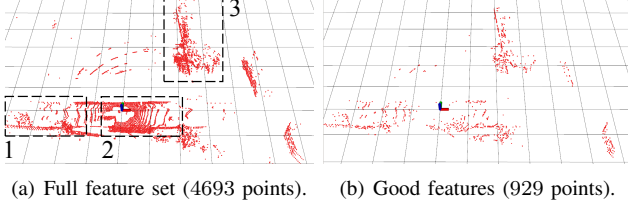(a) Full feature set (4693 points).   (b) Good features (929 points).

Fig. 2. A qualitative example of good features selected by our greedy-based feature selection algorithm. This method pick up points on objects which provide strong geometric constraints, as indicated in the region 1 and 3. Points on the ground, which occupy around 50% of the full feature set, are indicated by the region 2. Since they only constrain poses on the $z-$axis, our method only selected them with a small number. This is the main difference from the fully random sampling method (see Section VI-B).

robot performs a continuous movement in an environment, it is enough to compute $\lambda$ at every time interval ($1s \sim 2s$).

Fig. 1 plots the values of $\lambda$ on different sequences. *RHD01lab* contains several degenerate scenarios, while other sequences do not (see Section VI-C). Therefore, we empirically set $\lambda_{th} = 42$. If $\lambda \geq \lambda_{th}$, we select 20% of features from the full set as the good features (i.e., $M = 0.2N$). Otherwise, we use 80% of features from the set.

## V. GF-ENHANCED MULTI-LiDAR SLAM SYSTEM

The proposed feature selection method has been verified in an L-SLAM system called M-LOAM [17]. To distinguish it from the original system, the enhanced system is denoted by **M-LOAM-gf**. M-LOAM-gf solves SLAM with multiple LiDARs by two algorithms: *odometry* and *mapping*. Generally, these algorithms are designed to estimate poses in a coarse-to-fine fashion. Since they similarly formulate the NLS problem for pose estimation, the feature selection can be applied to both. Fig. 3 illustrates the overall structure of M-LOAM-gf. Note that loop closure is not included.

We give real definitions to the feature set $\mathcal{F}_K$. We extract features located on local edges and planar surfaces from the LiDARs' raw measurements. According to the points' curvatures, we select a set of edge and planar features to form $\mathcal{F}_K$. The next step is to match features between the reference frame and the robot's current frame. In both *odometry* and *mapping*, we use the feature map in the reference frame to associate data with $\mathcal{F}_K$. The only difference is the scale. The local map in *odometry* is built within a small time interval ($< 0.5s$), while the global map in *mapping* is constructed using all features in keyframes. For convenience, we use $\mathcal{M}_K$ to denote both the local and global maps.
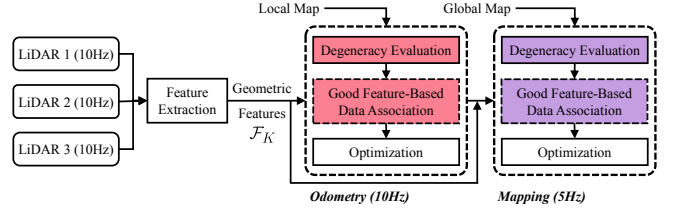


Fig. 3. Block diagram of the pipeline of M-LOAM-gf.

TABLE I
MEAN AND STD OF $\log \det \mathbf{\Lambda}(\mathcal{S}_K^{\#})$ ON RHD AND OR SEQUENCES.

| Sequence | Greedy | Rnd | Full |
|---|---|---|---|
| *RHD01lab* | **35.1 ± 2.3** | 30.3 ± 2.7 | 39.3 ± 2.5 |
| *RHD02garden* | **41.4 ± 4.0** | 39.7 ± 5.0 | 48.2 ± 5.1 |
| *OR01* | 52.6 ± 1.0 | **52.7 ± 1.6** | 61.8 ± 1.7 |
| *OR02* | 50.8 ± 0.9 | **51.8 ± 2.2** | 60.1 ± 2.1 |
| *OR03* | **51.7 ± 1.4** | 49.9 ± 2.6 | 58.3 ± 2.4 |
| *OR04* | **51.6 ± 1.2** | 50.3 ± 2.2 | 58.7 ± 2.1 |
| *OR05* | **52.5 ± 1.7** | 51.9 ± 2.7 | 59.3 ± 2.4 |
| Average | **47.96** | 46.66 | 55.10 |

With the found correspondences, we can optimize the relative transformation by minimizing the sum of all errors. The good feature selection algorithm enables M-LOAM-gf to select only a set of features in optimization while preserving the spectrum of the information matrix. An example of good features are shown in Fig. 2. After obtaining the good feature set $\mathcal{S}_K^{\#}$ in Section IV-C, the objective function is written as

$$\hat{\mathbf{x}}_K = \arg\min_{\mathbf{x}_K} \sum_{\mathbf{p} \in \mathcal{S}_K^{\#}} \left|\left|\mathbf{r}(\mathbf{x}_K, \mathbf{p})\right|\right|_{\mathbf{\Sigma}_{\mathbf{p}}}^2, \quad (6)$$

for which the expression of the residuals and Jacobians is detailed in our supplementary material [35].

## VI. EXPERIMENT

We evaluate the performance of M-LOAM-gf on real-world experiments. First, we validate the stochastic-greedy-based feature selection. Second, we demonstrate the localization accuracy of M-LOAM-gf in various scenarios covering indoor environments and outdoor urban roads with two multi-LiDAR setups. Three SOTA L-SLAM systems are compared. We also study M-LOAM-gf's latency on an onboard processor with limited computation resources.

### A. Implementation Details

We use the PCL library [37] to process point clouds and the Ceres Solver [38] to solve the NLS problems. Our method is tested on sequences collected with two platforms:

- **Real Handheld Device (RHD)** is made for indoor tests and shown in Fig. 4(a). It is installed with two VLP-16[1]. We held this device to collect two sequences (*RHD01lab* and *RHD02garden*) with an average speed of $2m/s$.

[1]`https://velodynelidar.com/products/puck`

(a) The real handheld device (RHD).



(b) Results on the *RHD01lab*.



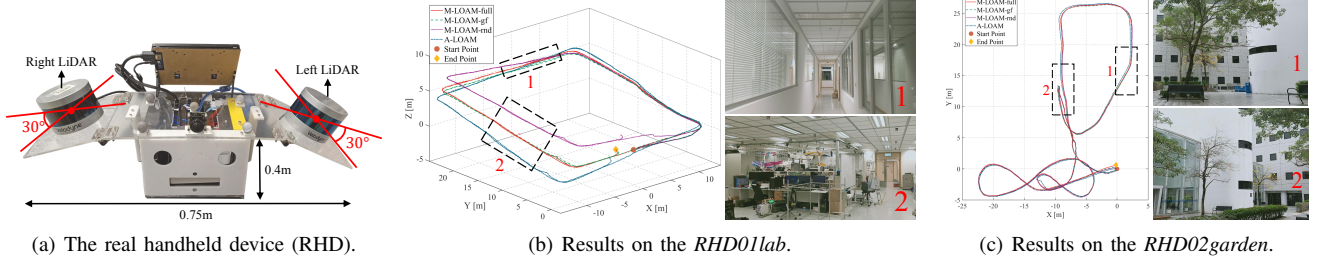(c) Results on the *RHD02garden*.

Fig. 4.   Estimated trajectories of different methods and the scene image on two *RHD* sequences.
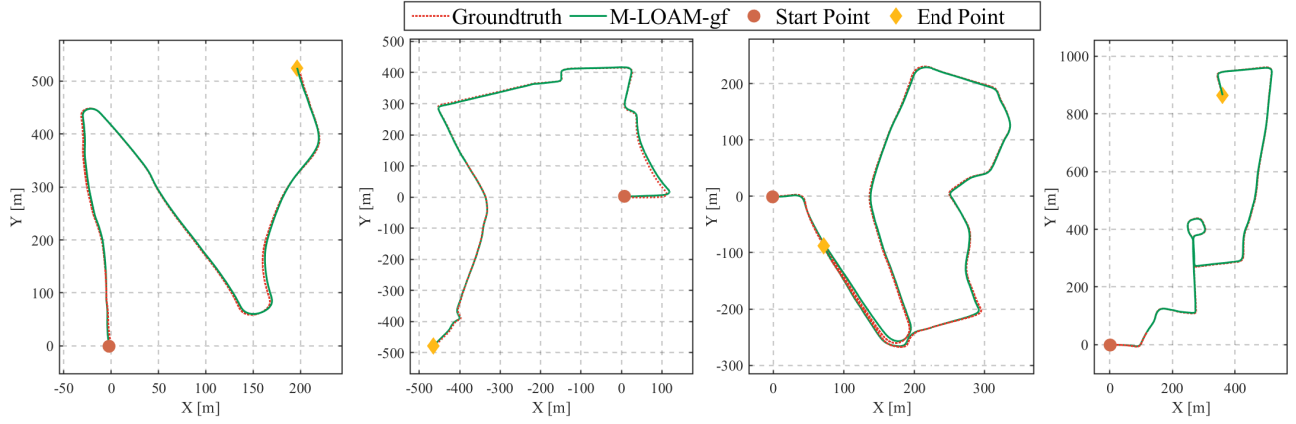


Fig. 5.   M-LOAM-gf's trajectories on *OR01*, *OR03*, *OR04*, and *OR05* from the Oxford Robocar dataset [36] are aligned with the ground truth .
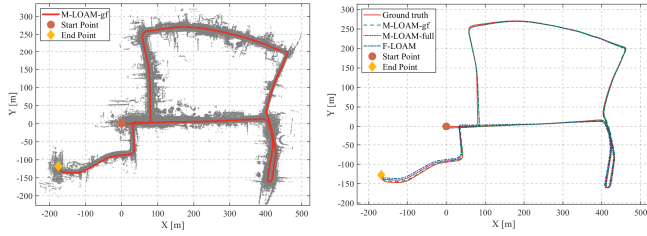


Fig. 6.   Results on the *OR02*. (left) Map and M-LOAM-gf's path. (right) Estimated trajectories of different methods are aligned with the ground truth.

- **Oxford Robocar (OR)** [36] is a vehicle equipped with two HDL-32E[2]. Datasets were recorded by driving the car on urban roads at an average speed of $10m/s$. 32 repeated traversals of a $9\ km$ route were collected. Ground-truth poses in $SE(2)$ are available. We select one sequence lasting 34 minutes and split it into 5 sequences named *OR01–OR05* for evaluation.

### B. Validation on Good Feature Selection

This section validates that the greedy algorithm selects a set of valuable features with a large $\log\det\mathbf{\Lambda}(\mathcal{S}_K^\#)$. Our stochastic-greedy method (label: **greedy**) is compared with the fully randomized selection method (label: **rnd**). Table I reports the means and standard deviations (std) of $\log\det\mathbf{\Lambda}(\mathcal{S}_K^\#)$. The values with the full feature set (label: **full**) are provided for reference. On *OR01–OR02*, the greedy method gains a smaller objective than the rnd method. This is

---

[2]https://velodynelidar.com/products/hdl-32e

reasonable since the greedy algorithm cannot always achieve the best performance according to (5). By considering the std and the larger means on most sequences, we conclude that the greedy method outperforms the rnd method.

### C. Performance of SLAM

We compare the accuracy, robustness, and latency of M-LOAM-gf with several baseline methods:

- **M-LOAM-rnd** is the variant of M-LOAM with the rnd feature selection module in mapping.
- **M-LOAM-full** is the original M-LOAM that uses the full feature set in mapping.
- **A-LOAM**[3], **F-LOAM**[4], and **LEGO-LOAM** [3] are three SOTA, open-source L-SLAM systems. All of them are the improved versions of LOAM [2].

The *odometry* and *mapping* of all methods run at 10 Hz and 5 Hz. For a fair comparison, the loop closure modules in some baselines are deactivated. The resolutions of the voxel filter [37] on the edge and planar features are $0.2m$ and $0.4m$.

*1) Qualitative Comparison:* We first test our method on RHD sequences. *RHD01lab* is recorded by moving around an office space, in which several scenes provide only poor geometric constraints. Fig. 4(b) indicates two examples. Scene 1 is a long and narrow corridor, which is a typical degenerate environment [40]. Scene 2 is an indoor office, providing well-conditioned constraints. M-LOAM-gf successfully tracks robot poses due to its capability in evaluating

---

[3]https://github.com/HKUST-Aerial-Robotics/A-LOAM
[4]https://github.com/wh200720041/floam

## TABLE II
TRANSLATIONAL ATE [39] ON OR SEQUENCES (THE TWO BEST RESULTS ARE MARKED AS BOLD TEXT).

| | Seq. | Dimension | M-LOAM-gf | M-LOAM-rnd | M-LOAM-full | A-LOAM | F-LOAM | LEGO-LOAM |
|---|---|---|---|---|---|---|---|---|
| RMSE$_t$ [m] | OR01 | $525m \times 252m$ | **1.986** ± 0.013 | 2.324 ± 0.028 | 2.449 ± 0.014 | 4.504 ± 0.055 | **2.174** ± 0.007 | 4.334 ± 0.000 |
| | OR02 | $629m \times 431m$ | **2.473** ± 0.151 | **3.107** ± 0.104 | 3.120 ± 0.076 | 7.535 ± 0.410 | 3.709 ± 0.044 | 4.369 ± 0.083 |
| | OR03 | $573m \times 896m$ | **1.955** ± 0.105 | 2.829 ± 0.158 | 3.063 ± 0.062 | 12.995 ± 0.006 | **2.019** ± 0.024 | 5.440 ± 0.102 |
| | OR04 | $337m \times 498m$ | **1.720** ± 0.047 | 2.242 ± 0.084 | 2.239 ± 0.024 | **1.913** ± 0.057 | 2.238 ± 0.062 | 6.224 ± 0.021 |
| | OR05 | $517m \times 968m$ | **1.719** ± 0.027 | **2.387** ± 0.432 | 2.567 ± 0.487 | 6.446 ± 0.015 | 2.469 ± 0.061 | 7.248 ± 0.260 |
| | Average | | **1.971** | 2.578 | 2.688 | 6.679 | **2.522** | 5.523 |

the environment's degeneracy. M-LOAM-rnd has a sudden drift in scene 1 since using only 20% of features cannot constrain the poses. A-LOAM also fails because it cannot model the uncertainty in mapping, which is detailed in [17]. *RHD02garden* is collected in a garden. Estimated trajectories and scene images are shown in Fig. 4(c). Since the environment is well-conditioned, all trajectories are comparative.

*2) Localization Accuracy:* We then perform a large-scale outdoor test on OR sequences under 10 repetitions. Environments on *OR* sequences commonly provide sufficient features. We visualize M-LOAM-gf's trajectory against the ground truth and the built map on *OR02* in Fig. 6, and plot trajectories the other sequences in Fig. 5. Each method is evaluated the absolute trajectory error (ATE) and the relative pose error (RPE) [39]. Due to limited space, we report the translation ATE in Table II, and show complete results in our supplementary material [35]. M-LOAM-gf does not just preserve the accuracy of M-LOAM-full, but it also reduces the ATE on all sequences. The average translation ATE of M-LOAM-gf is 22% lower than that of F-LOAM (the second-best method). The feature selection implicitly rejects outliers (see Section IV-C), which is essential to such accuracy gains. Thus, M-LOAM-rnd also improves M-LOAM-full. But its drift is larger than that of M-LOAM-gf. The performance of the fully randomized operation in M-LOAM-rnd is not guaranteed, which occasionally lead to inconsistent results.

*3) Latency:* Experiments in the above sections are conducted on a desktop with an i7 CPU@4.2 GHz and 32 GB memory. The average latency of *mapping* of M-LOAM-gf over 1725 and 10356 frames on the RHD and OR sequences is $62.69ms$ and $106.82ms$ respectively. To demonstrate that our feature selection method boosts an L-SLAM system on processors with limited resources, M-LOAM-gf is also tested on an Intel NUC[5] with an i7 CPU@3.1 GHz and 8 GB memory. The average latency is reported in Table III. We run the rosbag at a low frequency to ensure no data loss.

First of all, we observe that M-LOAM-rnd has lower latency than M-LOAM-gf in the GF-based data association. This is because the rnd is an $O(M)$ algorithm, while the stochastic-greedy algorithm is $O(N \log(1/\epsilon))$. Second, compared with M-LOAM-full, M-LOAM-gf may need more time for feature matching but significantly save time in nonlinear optimization. Finally, M-LOAM-gf, M-LOAM-rnd, and LEGO-LOAM are three real-time systems ($< 200ms$ at

[5]zh.wikipedia.org/wiki/Next_Unit_of_Computing

## TABLE III
AVERAGE LATENCY [MS] OF MAPPING ON AN INTEL NUC.

| Seq. | Method | Mapping | | |
|---|---|---|---|---|
| | | Data association | Optimization | Total |
| RHD | M-LOAM-gf | 17.90 | 3.09 | **115.48** |
| | M-LOAM-rnd | 6.33 | 3.65 | **92.57** |
| | M-LOAM-full | 16.17 | 7.01 | 128.32 |
| | A-LOAM | – | – | 131.08 |
| OR | M-LOAM-gf | 46.18 | 4.64 | **149.25** |
| | M-LOAM-rnd | 21.70 | 6.86 | **108.10** |
| | M-LOAM-full | 54.35 | 22.85 | 230.35 |
| | A-LOAM | – | – | 313.69 |
| | F-LOAM | – | – | 271.30 |
| | LEGO-LOAM | – | – | 158.23 |

Latency: Time delay between the input and output of a function.

each mapping frame) for the Intel NUC. Both M-LOAM-gf and M-LOAM-rnd outperform LEGO-LOAM in terms of accuracy. LEGO-LOAM implicitly performs feature selection since it filters out points if the distances to their correspondences are larger than a threshold. But this naive and hard-coded solution leads to large accuracy loss.

## VII. CONCLUSION

In this paper, we propose a greedy-based feature selection method for NLS pose estimation using LiDARs. The feature selector retains the most valuable LiDAR features with the objective of preserving the information matrix's spectrum. The stochastic-greedy algorithm is applied for the real-time selection. Moreover, we also investigate the degeneracy issue of utilizing good features for pose estimation in structureless environments. We propose a strategy to adaptively change the number of good features to avoid ill-conditioned estimation. The feature selection is integrated into a multi-LiDAR SLAM system, followed by evaluation on sequences with two sensor setups and computation platforms. The enhanced system is shown to have great efficiency and higher localization accuracy than SOTA methods. The idea of feature selection is general and can be applied to many NLS problems.

Future work will concern two possible directions. The first direction is to utilize data-driven methods [41] to online tune parameters which were mannually set. Another direction is to apply the proposed feature selection to other tasks, such as bundle adjustment [42] and cross-model localization [43].

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in Robotics: Science and Systems, vol. 2, no. 9, 2014.

[3] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 4758–4765.

[4] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in Robotics: Science and Systems, 2018.

[5] J. Lin, X. Liu, and F. Zhang, "A decentralized framework for simultaneous calibration, localization and mapping with multiple lidars," arXiv preprint arXiv:2007.01483, 2020.

[6] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," arXiv preprint arXiv:2010.08215, 2020.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in 2011 International Conference on Computer Vision, 2011, pp. 2564–2571.

[9] H. Yang and L. Carlone, "A polynomial-time solution for robust registration with extreme outlier rates," arXiv preprint arXiv:1903.08588, 2019.

[10] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85.

[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in Advances in Neural Information Processing Systems, 2017, pp. 5099–5108.

[12] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 770–779.

[13] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," IEEE Transactions on Image Processing, vol. 6, no. 9, pp. 1305–1315, 1997.

[14] I. Lang, A. Manor, and S. Avidan, "Samplenet: Differentiable point cloud sampling," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7578–7588.

[15] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 809–816.

[16] Y. Zhao and P. A. Vela, "Good feature matching: Toward accurate, robust vo/vslam with low latency," IEEE Transactions on Robotics, vol. 36, no. 3, pp. 657–675, 2020.

[17] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," arXiv preprint arXiv:2010.14294, 2020.

[18] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Luján, S. Furber, A. J. Davison et al., "Slambench2: Multi-objective head-to-head benchmarking for visual slam," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 1–8.

[19] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, Feature extraction: foundations and applications. Springer, 2008, vol. 207.

[20] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019, pp. 4530–4537.

[21] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, and F. Fraunhofer, "Overlapnet: Loop closing for lidar-based slam."

[22] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, "Sloam: Semantic lidar odometry and mapping for forest inventory," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 612–619, 2020.

[23] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1104–1119, 2012.

[24] Z. Zhou, M. Yang, C. Wang, and B. Wang, "Roi-cloud: A key region extraction method for lidar odometry and localization," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 3312–3318.

[25] S. Choudhary, V. Indelman, H. I. Christensen, and F. Dellaert, "Information-based reduced landmark slam," in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 4620–4627.

[26] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," IEEE Transactions on Robotics, vol. 35, no. 1, pp. 1–20, 2018.

[27] A. Fontan, J. Civera, and R. Triebel, "Information-driven direct rgb-d odometry," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4929–4937.

[28] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," Robotics and Autonomous Systems, vol. 57, no. 12, pp. 1198–1210, 2009.

[29] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[30] T. D. Barfoot, State estimation for robotics. Cambridge University Press, 2017.

[31] M. Bosse, G. Agamennoni, I. Gilitschenski et al., Robust estimation and applications in robotics. Now Publishers, 2016.

[32] A. Censi, "An accurate closed-form estimate of icp's covariance," in Proceedings 2007 IEEE international conference on Robotics and automation. IEEE, 2007, pp. 3167–3172.

[33] G. H. Golub and C. F. Van Loan, Matrix computations. JHU press, 2013, vol. 3.

[34] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," IEEE Transactions on Control of Network Systems, vol. 3, no. 1, pp. 91–101, 2015.

[35] J. Jiao, Y. Zhu, H. Ye, H. Huang, P. Yun, L. Jiang, L. Wang, and M. Liu, "Supplementary material to:," Tech. Rep. [Online]. Available: https://ram-lab.com/file/site/m-loam/m-loam-gf/supp.pdf

[36] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," arXiv preprint arXiv:1909.01300, 2019.

[37] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 1–4.

[38] S. Agarwal and K. Mierle, "Others,ceres solver,," 2015.

[39] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 7244–7251.

[40] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 3144–3150.

[41] J. N. Wong, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot, "A data-driven motion prior for continuous-time trajectory estimation on se (3)," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1429–1436, 2020.

[42] Y. Zhao, J. S. Smith, and P. A. Vela, "Good graph to optimize: Cost-effective, budget-aware bundle adjustment in visual slam," arXiv preprint arXiv:2008.10123, 2020.

[43] H. Huang, H. Ye, Y. Sun, and M. Liu, "Gmmloc: Structure consistent visual localization with gaussian mixture models," IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 5043–5050, 2020.