Nonsubmodular Visual Attention for Robot Navigation

Reza Vafaee, *Member, IEEE*, Kian Behzad, *Graduate Student Member, IEEE*, Milad Siami, *Senior Member, IEEE*, Luca Carlone, *Senior Member, IEEE*, and Ali Jadbabaie, *Fellow, IEEE*

***Abstract*—This article presents a task-oriented computational framework to enhance visual-inertial navigation (VIN) in robots, addressing challenges such as limited time and energy resources. The framework strategically selects visual features using a mean squared error (MSE)-based, nonsubmodular objective function and a simplified dynamic anticipation model. To address the NP-hardness of this problem, we introduce four polynomial-time approximation algorithms: a classic greedy method with constant-factor guarantees; a low-rank greedy variant that significantly reduces computational complexity; a randomized greedy sampler that balances efficiency and solution quality; and a linearization-based selector based on a first-order Taylor expansion for near-constant-time execution. We establish rigorous performance bounds by leveraging submodularity ratios, curvature, and elementwise curvature analyses. Extensive experiments on both standardized benchmarks and a custom control-aware platform validate our theoretical results, demonstrating that these methods achieve strong approximation guarantees while enabling real-time deployment.**

***Index Terms*—Autonomous agents, greedy-based algorithms, localization, nonsubmodularity, visual-based navigation.**

I. Introduction

Achieving efficient navigation in unpredictable environments remains a significant challenge in robotics. Recent advancements in computing devices have opened up new possibilities, propelling substantial progress in this research area [1]. These advancements enable near-real-time resolution of estimation and planning tasks in specific applications. However, navigating robots in rapidly changing environments still faces computational complexities. Despite the availability of high-performance computational units, the increasing demand for agility and autonomy necessitates more efficient onboard processing to ensure timely decision-making and execution in dynamic conditions.

One of the key challenges in robot navigation is achieving accurate visual odometry while minimizing computational resource usage. Many researchers have focused on the issue of selecting visual features (e.g., [2], [3], [4], [5], [6], [7]). The core idea is that, based on the robot's current state and its planned future motion (i.e., the task at hand), tracking specific features over a time horizon can provide more valuable information than tracking others. Essentially, some visual features may require greater attention than others.

In this context, Davison [8] employed a greedy approach to select a subset of preidentified visual features, thereby streamlining the robot's pose estimation. Strasdat et al. [9] combined simultaneous localization and mapping (SLAM) using unscented Kalman filtering with reinforcement learning to create policies for feature selection. Mu et al. [4] proposed a two-stage methodology for measurement planning, where the initial stage involves selecting a subset of landmarks for observation, followed by determining observation times for each feature. Additionally, in [3], a task-aware approach is investigated to select a subset of features with the goal of minimizing an uncertainty metric.

Related sparsification problems are explored in other contexts. For example, the authors in [10] and [11] addressed optimizing feedback interconnections in linear consensus networks, while the authors in [12] and [13] focused on selecting sparse sets of sensors or actuators to preserve system observability and controllability. However, these frameworks are not directly applicable to the motion-aware feature selection problem discussed here, as they focus on dyads rather than selecting a limited number of positive semidefinite (PSD) matrices.

Prior works such as [5] and [6] showed that feature selection for robot navigation can be framed as maximizing a nonnegative monotone submodular function under a matroid constraint, with the greedy algorithm achieving an optimal approximation of $1 - 1/e \approx 0.632$ for cardinality constraints [14]. This process accelerates with lazy evaluations [15] and further optimizes with randomization for linear time complexity [16]. Recently, Buchbinder and Feldman [17] proposed a deterministic nonoblivious local search algorithm with a $1 - 1/e - \varepsilon$ (for any $\epsilon > 0$) guarantee. However, these methods are not applicable to our problem, since mean squared error, our primary performance metric, is not submodular.

The most closely related prior work to the present study is conducted by [5] and [6]. In [5], the focus is on visual-inertial navigation (VIN), in which the design variable is the selection

Received 13 June 2025; revised 7 August 2025; accepted 22 September 2025. Date of publication 8 October 2025; date of current version 28 October 2025. This work was supported in part by the U.S. Office of Naval Research under Grant Award N00014-21-1-2431, in part by the U.S. National Science Foundation under Grant Award 2121121 and Grant Award 2208182, in part by the U.S. Department of Homeland Security under Grant Award 22STESE00001-03-02, and in part by the DEVCOM Analysis Center and was accomplished under Contract W911QX-23-D0002 (RV, KB, MS). The work of Ali Jadbabaie was supported by the DARPA Artificial Intelligence Quantified (AIQ) Program. This article was recommended for publication by Guest Associate Editor T.A. Vidal-Calleja and Editor J. Civera upon evaluation of the reviewers' comments. *(Corresponding author: Milad Siami.)*

Reza Vafaee, Kian Behzad, and Milad Siami are with the Department of Electrical & Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: vafaee.r@northeastern.edu; behzad.k@northeastern.edu; m.siami@northeastern.edu).

Luca Carlone and Ali Jadbabaie are with the Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: lcarlone@mit.edu; jadbabai@mit.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TRO.2025.3619067, provided by the authors.

Digital Object Identifier 10.1109/TRO.2025.3619067

of features to track over a fixed time horizon. The authors use convex relaxations and a greedy method for feature selection, providing a quantitative assessment of performance guarantees for the quality of the resulting state estimations. They evaluate the effectiveness of the greedy heuristic using the submodularity ratio from [18]. However, calculating the submodularity ratio for the proposed set functions is computationally challenging due to the combinatorial complexity involved in its definition [19]. Additionally, the complexity of a simple semidefinite programming (SDP) convex relaxation scales cubically with the number of detected features, making it infeasible for real-time implementation. Moreover, their measures used to quantify the confidence ellipsoid of the forward predictor are not explicitly related to MSE, which is often the performance metric of interest for estimation problems [20].

A recent study detailed in [21] also tackles the challenge of reducing latency in visual odometry (VO)/Visual SLAM (VSLAM) systems by identifying and matching a subset of features deemed most valuable for pose estimation. While their focus lies in optimizing feature matching latency using the Max-logDet metric, our work concentrates on selecting informative features that minimize MSE under motion-aware dynamics, providing theoretical guarantees under nonsubmodular objectives.

Mousavi and Motee [6] proposed a randomized sampling algorithm for feature selection instead of using the greedy method or convex relaxations. In their approach, a sampling probability (a number between zero and one) is assigned to each available feature, with these probabilities interpreted as measures of informativeness during the sampling process. This procedure provides a performance guarantee compared to using all visual features, rather than the optimal set, for a range of VIN measures, including MSE. This is achieved by approximating the complete spectrum of the information matrix. However, for some measures, such as the worst-case error, this approach imposes unnecessary computational burden because we only need to approximate the minimum eigenvalue of the information matrix, rather than the full spectrum. Additionally, the proposed approach requires sampling $\mathcal{O}((3T+3)\log(3T+3)/\epsilon^2)$ features, where T is the forward time horizon and $\epsilon \in (0,1)$ is an approximation parameter, to guarantee the proposed performance bounds. We observe that this requirement for the number of sampled features is significant and, in practical scenarios, necessitates sampling all features extracted in the estimation problems.

Pandey et al. [7] later addressed the problem of sparse feature selection for localizing a team of agents, where they exchange relative measurements leading to a graphical network. Compared to [6], they improve the probabilistic bound of the randomized feature selection algorithm, although the related problems mentioned in the previous paragraph still exist.

VIN methods are broadly categorized into filtering-based approaches, fixed-lag smoothing, and full smoothing. Filtering methods, such as the extended Kalman filter-based and its variants analyzed in [22], offer real-time performance by sequentially processing data. However, they typically underutilize cross-time correlations, which may limit estimation accuracy in complex scenarios.

Fixed-lag smoothers [23], [24] enhance estimation accuracy via sliding-window optimization. Specifically, [23] integrates visual and inertial data in a keyframe-based bundle adjustment framework for visual-inertial SLAM. In contrast, Sibley et al. [24] employed a constant-time sliding-window filter using stereo imagery for real-time relative pose estimation during planetary landing. While both methods retain recent poses and marginalize older ones, their objectives and sensing modalities differ.

Full smoothing methods [25], [26] jointly optimize over all past states using nonlinear optimization. The first achieves efficient maximum-a-posteriori estimation through IMU preintegration on manifolds, while the second handles high-dynamic motion without requiring prior state initialization. These methods offer increased accuracy but incur higher computational complexity due to global optimization.

More recently, ORB-SLAM3 [27] has emerged as a leading keyframe-based SLAM system that integrates visual and inertial data via local bundle adjustment and global pose graph optimization. While it does not implement full smoothing over all states, its flexibility and support for multiple sensor configurations make it a strong practical baseline. This article focuses on the complementary task of selecting informative visual measurements, which can enhance the front-end performance of VIN systems across all categories.

Our contributions to address the existing issues, in this work are as follows.

1) Using a similar simplified model for forward simulation of robot dynamics as [5], we formulate the task of feature selection as maximizing a monotone nonsubmodular objective function directly related to the mean-square state estimation error.
2) We propose constant factor approximation bounds for the greedy algorithm based on recent concepts of submodularity ratio and curvature [28]. We derive bounds on these values according to the spectrum of the information matrices, eliminating the need for combinatorial search. These bounds suggest an easy-to-obtain performance guarantee for the greedy approach.
3) We exploit the low-rank structure of the landmark information matrices to significantly reduce the computational cost of matrix inversion in each greedy iteration. This leads to a fast version of the greedy algorithm that maintains the same approximation guarantees as the standard approach, while offering substantial practical speedups.
4) Inspired by [29], we include a randomization step in a simple greedy procedure to increase computational efficiency and practicality for scenarios with numerous detected features or longer prediction horizons. This randomized greedy framework is supported by a performance bound based on the elementwise curvature concept.
5) We transform the problem into an efficient modular maximization by using a first-order Taylor series approximation of the VIN performance measure. In this form, the greedy selection method guarantees an optimal solution.
6) We validate the proposed selectors in two complementary experiments. First, multiple EuRoC sequences [30] allow direct comparison with prior work. Because EuRoC lacks control inputs for horizon prediction, we also run a full visual–inertial experiment on the QCar platform [31], where stereo images, IMU data, and control commands are all available. This two-pronged evaluation demonstrates both benchmark accuracy and real-world applicability.

Unlike [6], all proposed performance bounds characterize proximity to the optimal visual feature set. We provide some of the proofs in Appendix B, to keep the main text focused on the discussion of the problem and proposed solutions.

Fig. 1 provides a comprehensive overview of the visual attention mechanism investigated in this article. This mechanism functions as a critical subblock in the front-end, where

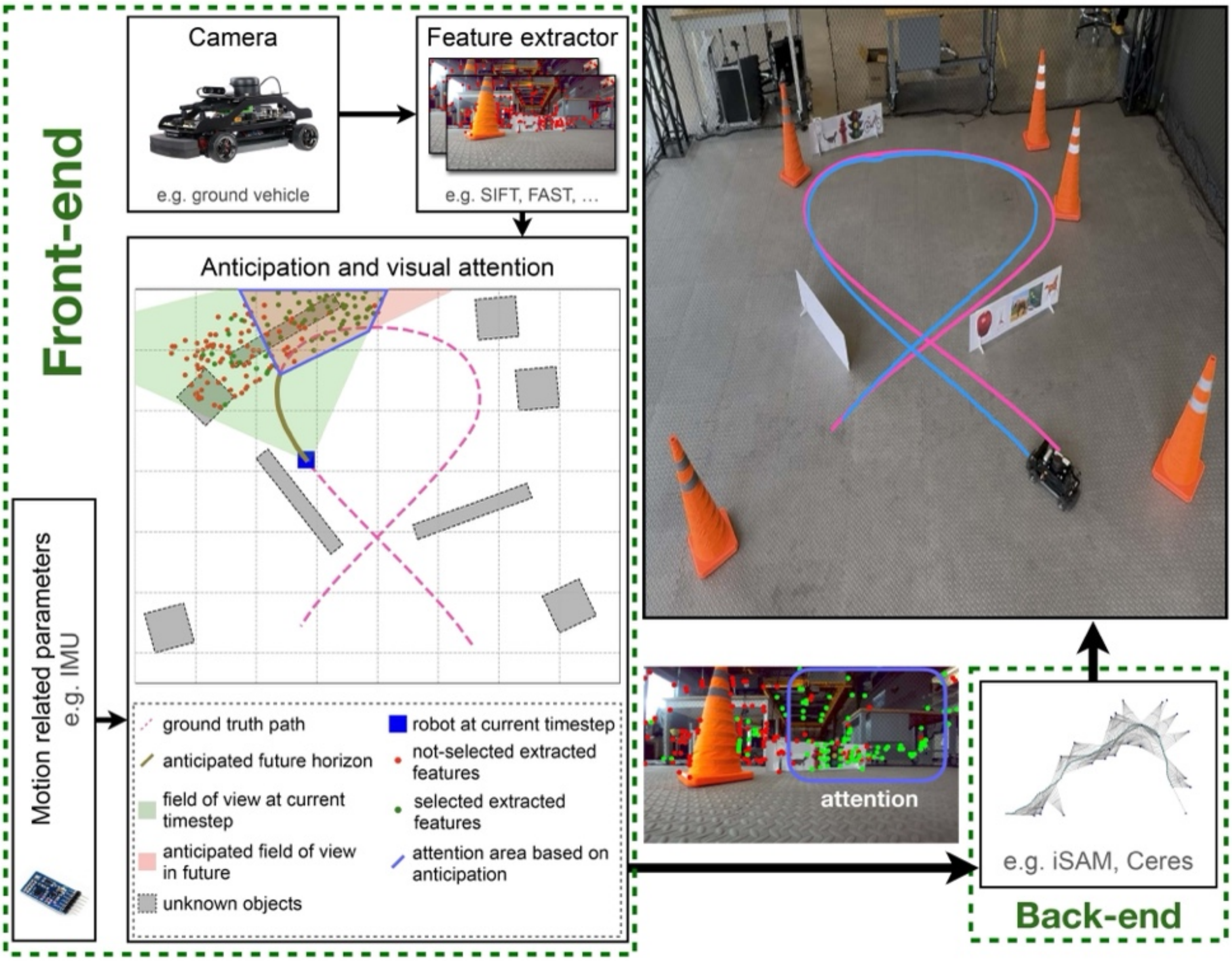

Front-end
Camera
e.g. ground vehicle
Feature extractor
e.g. SIFT, FAST, ...
Anticipation and visual attention
Motion related parameters
e.g. IMU
ground truth path
anticipated future horizon
field of view at current timestep
anticipated field of view in future
unknown objects
robot at current timestep
not-selected extracted features
selected extracted features
attention area based on anticipation
attention
e.g. iSAM, Ceres
Back-end

Fig. 1. Detailed illustration of the visual attention mechanism as the front-end's filtering block. Image features extracted from each keyframe are combined with motion priors provided by the IMU to form a baseline information estimate. For every candidate feature, the mechanism projects the feature into predicted future poses to simulate its visibility and then computes its expected reduction in estimation error. Features are then ranked by this information-gain metric, and only the highest-value features are forwarded to the back-end optimizer for state estimation. This selective process reduces computational load while maintaining high-accuracy pose estimates.

it processes inputs such as extracted features from keyframes and motion-related parameters obtained, for example, from an IMU. The primary objective of the visual attention mechanism is to determine which features should be selected for back-end optimization. This selection process is guided by evaluating both the quality of the features and their potential impact in decreasing uncertainties based on robot dynamics. By effectively filtering and prioritizing features, the visual attention mechanism ensures that the back-end optimization process operates with the most relevant and high-quality data, thereby enhancing the overall performance and accuracy of the system.

II. Preliminaries and Problem Definition

A. Notations

Integers are denoted by $\mathbb{Z}$ and real numbers by $\mathbb{R}$. The set of integers (respectively, real numbers) greater than or equal to $a \in \mathbb{R}$ is denoted by $\mathbb{Z}_{\geq a}$ (respectively $\mathbb{R}_{\geq a}$). Given any integer $n \in \mathbb{Z}_{\geq 1}$, we define $[n] = \{1, \ldots, n\}$. Finite sets are denoted by sans-serif fonts (e.g., A). The cardinality of a finite set A is denoted by $|\mathsf{A}|$. I denotes an identity matrix, with its dimension inferred from context. The transpose and rank of a matrix X are denoted by $X^\top$ and $\operatorname{rank}(X)$, respectively. The set of positive definite (PD) (respectively, PSD) matrices of size n is denoted by $\mathbb{S}^n_{++}$ (respectively $\mathbb{S}^n_{+}$). The symbol $\|\cdot\|_2$ denotes the Euclidean norm when applied to vectors and the spectral norm when applied to matrices. The symbol $\|\cdot\|_F$ denotes the Frobenius norm for matrices. For vector x, $\operatorname{diag} x$ is a diagonal matrix with the elements of x orderly sitting on its diagonal. $\mathcal{N}(\mu, \Sigma)$ denotes a normal (Gaussian) distribution with mean (average) μ and covariance Σ.

B. Forward-Simulation Optimization

The forward-simulation model considered in this article depends on the anticipated future motion of the robot, as we will later show that the inertial measurement unit (IMU) and vision models are functions of the robot's predicted future poses. Anticipation is a key element of our approach: the feature selection mechanism is aware of the robot's immediate future intended motion and selects features that remain in the field of view longer given these intentions [5].

In typical VIN pipelines, the primary objective is to estimate the state of the robot at each frame. Let k denote the time frame for estimation, where our focus is on selecting a limited number of features. Let $x(k)$ represent the state of the robot at time k. For now, readers can consider $x(k)$ to include the pose and velocity of the robot at time instant k, as well as the IMU biases. We denote

$$\hat{x}(k:k+T) \triangleq \begin{bmatrix}\hat{x}^\top(k) & \hat{x}^\top(k+1) & \cdots & \hat{x}^\top(k+T)\end{bmatrix}^\top$$

as the future state estimates within the horizon T. Moreover, we denote P as the covariance matrix of our estimate $\hat{x}(k:k+T)$, and $\Omega \triangleq P^{-1}$ as the corresponding Fisher information matrix. The error variance for the MSE estimator $\hat{x}(k:k+T)$ can then be obtained by summing the variances of its individual scalar components, i.e.,

$$\mathbb{E}\left\|x(k:k+T) - \hat{x}(k:k+T)\right\|_2^2 = \mathrm{Trace}\Omega^{-1} \quad (1)$$

where $x(k:k+T)$ is the stacked vector of actual values of the state vectors over the horizon T.

Let U, with $|\mathsf{U}| = N$, denote the set of all available features at time frame k. Our goal is to select a limited number of $\kappa \ll N$ features that maximize

$$\max_{\mathsf{S}\subset\mathsf{U}} f(\mathsf{S}) \quad \text{subject to} \quad |\mathsf{S}| \leq \kappa \quad (2)$$

with $f(\mathsf{S}) \triangleq \mathrm{Trace}\Omega_\emptyset^{-1} - \mathrm{Trace}\Omega_\mathsf{S}^{-1}$, $\Omega_\mathsf{S} \triangleq \Omega_\emptyset + \sum_{l\in\mathsf{S}} \Delta_l$, where $\Omega_\emptyset$ is the information matrix of the estimate when no features are selected, and Δ_l is the information matrix associated with the selection of the lth visual feature.

It is evident that the maximization problem (2) aims to minimize the MSE of the estimate as defined in (1). Furthermore, we incorporate the term $\mathrm{Trace}\Omega_\emptyset^{-1}$ in the objective of (2) to ensure the normalization of the objective, such that it returns zero for empty sets. It is important to note that the MSE-based objective f is not necessarily submodular, as demonstrated by counterexamples in [32]. We note that similar criteria have been developed in [29], [33], and [34], for sparse sensing purposes.

At first glance, the problem definition in (2) may appear similar to the often-studied sensor selection problem for Kalman filtering, such as in[29], [34], or [35]. However, the contribution of each visual feature to the information matrix, i.e., the Δ_l matrices, as we will see, is PSD and typically of higher rank. In contrast, each sensor's contribution to the information matrix of Kalman state estimation is generally a rank-one matrix, often in the form of dyads. This key difference in the structure of the information matrices imposes challenges that preclude directly applying similar algorithms from sensor selection contexts to the visual feature selection problem.

Remark 1: Buchbinder and Feldman [17] have recently developed a significant advancement in submodular maximization under a matroid constraint, introducing a deterministic nonoblivious local search algorithm with an approximation guarantee of $1 - 1/e - \varepsilon$ (for any $\varepsilon > 0$). This approach effectively bridges the gap between deterministic and randomized algorithms for such problems. However, their method is specifically designed for submodular functions. Since our problem (2) involves the MSE measure, which is nonsubmodular, their deterministic algorithm cannot be directly applied to achieve the desired optimization in our context.

In the subsequent two sections, we discuss how the integration of IMU and camera measurements can be utilized to determine the information matrices $\Omega_\emptyset$ and Δ_l at each time frame.

C. IMU Model

There is often a significant discrepancy in measurement frequencies, with the IMU operating at a much higher frequency than the camera. Consequently, following the methodology outlined in [5], we integrate a set of IMU measurements between two consecutive camera frames k and j, treating this integrated measurement as a single measurement that constrains the states $x(k)$ and $x(j)$ using a linear measurement model.

Our simplified IMU model assumes that the accumulation of rotation error through gyroscope integration over time is negligible. Given this assumption of accurate rotation estimates from the gyroscope, we reduce the state variables to the robot's position, linear velocity, and accelerometer bias.

The onboard accelerometer measures the acceleration $a(k)$ of the sensor with respect to the inertial frame. This measurement is affected by additive white noise $\xi(k)$ and a slowly varying sensor bias $b(k)$. Therefore, the measurement $\tilde{a}(k) \in \mathbb{R}^3$ acquired by the accelerometer at time k is given as

$$\tilde{a}(k) = R^\top(k)\left(a(k) - g\right) + b(k) + \xi(k) \quad (3)$$

where g is the gravity vector expressed in the inertial frame, and $R(k)$ denotes the attitude of the robot at time k, assumed to be known from gyroscope integration over the horizon.

Using (3), we can show that the vector of IMU measurements between frame k and the future frame j exhibits linear dependence on the stack of state vectors $x(k:k+T)$. By applying these linear models to all consecutive frames k and j within the horizon, we can deduce the information matrix for the estimate of $x(k:k+T)$ based on the IMU data.

Note that the information matrix obtained only from relative measurements lacks a constraint anchoring the information to a global reference, making it rank-deficient. To address this deficiency, we integrate a prior on the state at time k from the VIN back-end into the information matrix. This incorporation yields a positive definite (PD) information matrix $\Omega_\emptyset \in \mathbb{S}_{++}^{9T+9}$. This PD matrix is essential for our problem definition in (2).

The derivation of the information matrix $\Omega_\emptyset$ follows the same process as in [5, Sec. III-B1]. To avoid repetition, detailed derivations are omitted in this article. For a comprehensive derivation, please refer to the relevant section of the cited article.

D. Vision Model

We use a linearized version of the nonlinear perspective projection model as the vision model. In this approach, we represent a pixel measurement as a linear function of the unknown state we aim to estimate. A calibrated pixel measurement of an external 3-D point (or feature) l provides the 3-D bearing of the visual feature in the camera frame.

Mathematically, let $u_l(k)$ be the unit vector corresponding to the (calibrated) pixel observation of feature l from the robot's pose at time k. This unit vector $u_l(k)$ satisfies the following relation:

$$u_l(k) \times \left(\left(R_{\mathrm{cam}}^{\mathrm{W}}(k)\right)^\top \left(p_l - t_{\mathrm{cam}}^{\mathrm{W}}(k)\right)\right) = \mathbf{0}_3 \quad (4)$$

where × denotes the cross product between two vectors, p_l is the 3-D position of visual feature l in the world frame, and $R_{\mathrm{cam}}^{\mathrm{W}}(k)$

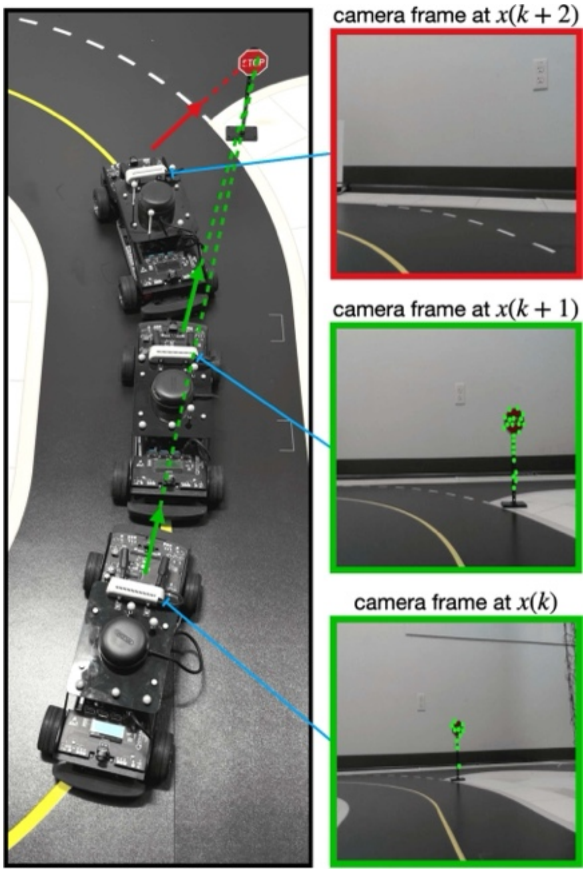


Fig. 2. Forward propagation of a feature's bearing vector from a 3-D object (a stop sign) detected at frame k. Left: The robot's positions and bearing vectors to the stop sign features at time steps $k+1$ and $k+2$. Right: The corresponding camera image planes at those time steps. Colored rectangles around each image indicate whether the feature is visible (green) or not (red). In this example, the predicted bearing at $k+2$ falls outside the camera's field of view, causing the visibility check to fail. See Section IV-B for details.

and $t^{\mathrm{W}}_{\mathrm{cam}}(k)$ are the rotation and translation describing the camera pose at time k with respect to the world frame.

Since our state at each time step k includes the position, velocity of the robot in the IMU frame, and accelerometer bias, we reparametrize (4) to incorporate these variables. This reparametrization uses the known extrinsic transformation between the camera and IMU obtained from calibration. It also enables us to derive a linear model for the calibrated feature pixel as a function of the state vector $x(k:k+T)$ and p_l, the unknown position of feature l.

In the context of the obtained linear measurement model, our objective is to determine the visibility of each feature l at future time points over the horizon T. The key idea is that if the lth feature is expected to remain in view throughout the future horizon, it will provide more information compared to a feature that is anticipated to be quickly obscured. To achieve this, we project the lth feature onto the image plane of the robot's camera at future time instances. Ultimately, our task is to verify whether the feature l will be visible in the forward predictions. Forward simulation and visibility checking are illustrated in Fig. 2.

We then stack the linear measurement model for each observation pose from which feature l is visible. It is important to note that the stacked linear model cannot be directly used to estimate our state vector $x(k:k+T)$ because it includes the unknown position of feature, p_l. To address this challenge, we employ the Schur complement trick [36] to analytically eliminate the 3-D point from the estimation process. To achieve this, we first construct the information matrix for the joint state vector $\left[x(k:k+T)^\top \quad p_l^\top\right]^\top$ using the stacked linear measurement model

$$\Omega^{(l)} = \begin{bmatrix} F_l^\top F_l & F_l^\top E_l \\ E_l^\top F_l & E_l^\top E_l \end{bmatrix} \in \mathbb{R}^{(9(T+1)+3)\times(9(T+1)+3)}. \quad (5)$$

For detailed definitions of the matrices F_l and E_l, we refer interested readers to [5]. By applying the Schur complement trick, we marginalize out the feature l and obtain the information matrix for our state $x(k:k+T)$ given the measurements

$$\Delta_l \triangleq F_l^\top F_l - F_l^\top E_l \left(E_l^\top E_l\right)^{-1} E_l^\top F_l \in \mathbb{R}^{9(T+1)\times 9(T+1)}. \quad (6)$$

This matrix Δ_l represents the additive contribution to the information matrix of our state estimate resulting from the measurements of a single visual landmark l. It is a PSD matrix.

III. Approximation Methods

The combinatorial problem in (2) is NP-hard by reduction to the classical set cover problem; see, e.g., [29], [37], requiring an exhaustive search over all selections of κ features to find the optimal solution. Therefore, in this section, we discuss various polynomial-time greedy-based algorithms to approximate its solution. We start with the more computationally expensive algorithms and then move on to the less demanding ones.

Remark 2: The approximation algorithms proposed in this article, along with their theoretical analyses and performance guarantees, can be extended to alternative set function objectives, such as the minimum eigenvalue or the log-determinant criterion discussed in [5]. In this work, however, we focus on the MSE objective due to its direct and interpretable connection to estimation accuracy, making it particularly suitable for the scenarios considered.

A. Greedy

In this section, we analytically demonstrate why a simple greedy algorithm provides a reliable approximate solution to problem (2). Our findings are based on recent literature on submodularity and submodular maximization [28].

As previously defined, U represents the ground set containing all extracted features, and $f(\cdot)$ denotes our set function as in (2). We define the marginal gain of a feature set $\mathsf{R} \subset \mathsf{U}$ in the context of a set $\mathsf{S} \subset \mathsf{U}$ as

$$f_{\mathsf{R}}(\mathsf{S}) \triangleq f(\mathsf{R} \cup \mathsf{S}) - f(\mathsf{S}). \quad (7)$$

For future analysis, we may use the shorthand notation l and $f_l(\cdot)$ for $\{l\}$ and $f_{\{l\}}(\cdot)$, respectively.

Definition 1: A set function

1) $f : 2^{\mathsf{U}} \to \mathbb{R}$ is called *submodular* if for all subsets $\mathsf{R} \subseteq \mathsf{S} \subseteq \mathsf{U}$ and all elements $l \notin \mathsf{S}$, it holds that

$$f(\mathsf{R} \cup \{l\}) - f(\mathsf{R}) \geq f(\mathsf{S} \cup \{l\}) - f(\mathsf{S}). \quad (8)$$

A set function is *supermodular* if the reversed inequality in (8) holds, and is *modular* if (8) holds with equality;

2) is monotone nondecreasing if for all $\mathsf{R} \subseteq \mathsf{U}$ and $l \in \mathsf{U}$, $f(\mathsf{R} \cup \{l\}) \geq f(\mathsf{R})$;

3) is normalized if $f(\emptyset) = 0$.

Algorithm 1 Simple Greedy Algorithm.

```
Input: $\Omega_\emptyset$, $\Delta_l$ for all $l \in \mathsf{U}$, and $\kappa$
Output: subset $\mathsf{S} \subset \mathsf{U}$ with $|\mathsf{S}| = \kappa$
$\mathsf{S} \leftarrow \emptyset$
while $|\mathsf{S}| < \kappa$ do
    $\mathsf{S} \leftarrow \mathsf{S} \cup \{\arg\max_{l \in \mathsf{U} \setminus \mathsf{S}} [f(\mathsf{S} \cup \{l\}) - f(\mathsf{S})]\}$
end while
```

Submodularity intuitively reflects the diminishing returns property: adding an element to a smaller set yields a greater benefit than adding it to a larger set. This principle is used to derive constant-factor approximation guarantees, specifically $(1 - 1/e)$, for the greedy algorithm when applied to submodular maximization problems with cardinality constraints [14].

Algorithm 1 operates by iteratively selecting the feature that provides the maximum marginal gain in reducing the estimation error, as defined by the objective function $f(\mathsf{S})$. At each iteration, the feature that contributes most to decreasing the MSE is added to the current subset, until the cardinality constraint κ is met. However, since the set function $f(\cdot)$ is not submodular [28], a quantitative assessment of solutions using the simple greedy procedure (Algorithm 1) requires further elaboration. To address this, we need the following two technical definitions from [28].

Definition 2. (Curvature, α): The curvature of any nonnegative set function $f(\cdot)$ is the smallest scalar $\alpha \in \mathbb{R}_{\geq 0}$ that satisfies

$$f_l\left(\mathsf{S}\setminus\{l\} \cup \mathsf{R}\right) \geq (1 - \alpha) \cdot f_l(\mathsf{S}\setminus\{l\}) \tag{9}$$

for all subsets R and S of U, and for all $l \in \mathsf{S}\setminus\mathsf{R}$.

For a nondecreasing set function $f(\cdot)$, the curvature α satisfies $\alpha \in [0, 1]$. The function $f(\cdot)$ is supermodular if and only if $\alpha = 0$.

Definition 3. (Submodularity ratio, γ): The submodularity ratio of any nonnegative set function $f(\cdot)$ is the largest scalar $\gamma \in \mathbb{R}_{\geq 0}$ such that

$$\sum_{l \in \mathsf{R}\setminus\mathsf{S}} f_l(\mathsf{S}) \geq \gamma\, f_{\mathsf{R}}(\mathsf{S}) \tag{10}$$

for all $\mathsf{R}, \mathsf{S} \subseteq \mathsf{U}$.

For a nondecreasing set function $f(\cdot)$, the submodularity condition holds if and only if $\gamma = 1$. However, in general, γ lies within the interval $[0, 1]$.

Proposition 1. (Approximate nonsubmodular maximization [28]): Let f be a nonnegative, nondecreasing, normalized set function with submodularity ratio $\gamma \in [0, 1]$ and curvature $\alpha \in [0, 1]$. Then, Algorithm 1, when applied to problem (2), provides the following approximation guarantee:

$$f(\mathsf{S}_{\text{greedy}}) \geq \frac{1}{\alpha}\left(1 - e^{-\alpha\gamma}\right) f(\mathsf{S}_{\text{OPT}}) \tag{11}$$

where $\mathsf{S}_{\text{greedy}}$ is the subset returned by Algorithm 1 and S_{OPT} is the optimal solution to problem (2).

To apply the results from Proposition 1 to our feature selection problem, we must demonstrate that the objective function in (2) satisfies the conditions specified in the lemma. This verification is provided in Lemma 1.

Let $n = 9T + 9$ denote the dimension of the information matrices in the VIN problem. It is easy to see that $\Omega_{\mathsf{S}} = \Omega_\emptyset + \sum_{l \in \mathsf{S}} \Delta_l$ is a symmetric PD matrix. This allows for factorization as $V\ \operatorname{diag}\left[\lambda_1(\Omega_{\mathsf{S}}) \quad \cdots \quad \lambda_n(\Omega_{\mathsf{S}})\right]^\top V^\top$, according to the eigendecomposition. Consequently, one can derive $\operatorname{Trace}\Omega_{\mathsf{S}}^{-1} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{1}{\lambda_{\mathrm{i}}(\Omega_{\mathsf{S}})}$.

Lemma 1: The objective of (2) is nonnegative, monotone increasing, and normalized.

Proof: The nonnegativity follows from $\Omega_\emptyset$ being PD and Δ_l for all l being PSD. It is also straightforward to confirm that $f(\emptyset) = 0$, thus ensuring normalization. To demonstrate that the objective in (2) is monotone nondecreasing, we need to show that for all $l \in \mathsf{U} \setminus \mathsf{S}$, the inequality $f(\mathsf{S} \cup \{l\}) - f(\mathsf{S}) \geq 0$ holds. This is established as follows: $f(\mathsf{S} \cup \{l\}) - f(\mathsf{S}) = \sum_{i=1}^{n}\left[\frac{1}{\lambda_i(\Omega_{\mathsf{S}})} - \frac{1}{\lambda_i(\Omega_{\mathsf{S}\cup\{l\}})}\right] > 0$, due to the *Cauchy interlacing inequality* for singular values. □

Lemma 1 confirms that the constant-factor approximation bound of Proposition 1 applies to our VIN problem. The challenge, however, lies in the intractability of computing the submodularity ratio and curvature for a given set function due to the combinatorial number of constraints (of the order of $2^{2|\mathsf{U}|}$) in (9) and (10), respectively. This difficulty is akin to the exhaustive search required for solving problem (2). Nonetheless, establishing a positive lower bound on the submodularity ratio and an upper bound on the curvature for f justifies the use of the greedy algorithm for problem (2) via Proposition 1. Our objective is to derive such bounds for f corresponding to the nonsubmodular objective in (2).

Theorem 1: The Greedy Algorithm 1 provides the following approximation guarantee for the feature selection problem (2):

$$f(\mathsf{S}_{\text{greedy}}) \geq \frac{1}{\alpha}(1 - e^{-\alpha\gamma}) f(\mathsf{S}_{\text{OPT}}) \geq \frac{1}{\overline{\alpha}}\left(1 - e^{-\overline{\alpha}\underline{\gamma}}\right) f(\mathsf{S}_{\text{OPT}}) \tag{12}$$

where $\overline{\alpha}$ and $\underline{\gamma}$ are the upper bound on the curvature and the lower bound on the submodularity ratio of the objective in (2), respectively. These bounds can be obtained using the spectrum of the information matrices at each time frame of the VIN pipeline as follows:

$$\overline{\alpha} = 1 - \underline{\gamma} = 1 - \frac{\delta \cdot \lambda_{\min}(\Omega_\emptyset)}{\lambda_{\max}(\Omega_{\mathsf{U}}) \cdot (\lambda_{\max}(\Omega_{\mathsf{U}}) - \lambda_{\min}(\Omega_\emptyset))} \tag{13}$$

where $\Omega_\emptyset$ is the information matrix of the estimate when no features are selected, Ω_{U} is the information matrix when all features are utilized, and δ is a positive number such that $\operatorname{Trace}\Delta_l \geq \delta$ for all $l \in \mathsf{U}$.

Proof: See Appendix B. □

Carlone and Karaman [5] utilized the concept of the submodularity ratio, as proposed by [18], to evaluate the effectiveness of a simple greedy heuristic in addressing a problem similar to problem (2). However, determining the submodularity ratio for a given set function remains intractable due to the combinatorial number of constraints inherent in the definition presented by [18]. In our results, as shown in Theorem 1, we also incorporate the notion of curvature to derive an improved constant-factor approximation. To avoid the combinatorial search required for obtaining these quantities, we present a set of computable bounds on the submodularity ratio and curvature.

In our analysis, we establish the relationship $\overline{\alpha} = 1 - \underline{\gamma}$. This correlation arises from the conservative approach taken in formulating these bounds. However, it is important to emphasize that this connection is specific to the derived values and does not generalize to the broader context of submodularity ratio and curvature values. For instance, for a submodular function, the submodularity ratio (γ) is equal to 1. However, this does not imply that the curvature (α) is necessarily $\alpha \leq 1 - 1 = 0$. Such an implication would hold true only if the function were also supermodular and, consequently, modular. Despite the conservative nature of these bounds, their existence is promising and aligns with empirical observations regarding the effectiveness of the greedy algorithm in optimizing nonsubmodular set functions.

It is worth emphasizing that the values of curvature and submodularity ratio are not intrinsic properties of the set function alone, as they also depend on the choice of ground set and the budget parameter κ. In other words, even for a fixed function $f(\cdot)$, the computed values of α and γ may vary depending on the context in which they are evaluated. This implies that the approximation guarantee given in Proposition 1 is meaningful only after verifying that $\gamma > 0$ for the particular setup at hand. In practice, this can often be assessed numerically, as we demonstrate in our empirical results [e.g., Fig. 5(a)], where a nonzero submodularity ratio validates the use of the greedy algorithm in our VIN pipeline.

In greedy Algorithm 1, we iteratively examine all feature candidates and find the one whose addition will enhance the estimation quality the most. This method requires $\mathcal{O}(\kappa N T^3)$ operations, implying that the complexity of computation grows quadratically with the number of available features if, in the worst-case, $\kappa = \mathcal{O}(N)$, and cubically with the length of the time horizon. Therefore, the greedy algorithm may be less practical in scenarios where the number of detected features or the length of the considered horizon is substantial.

After marginalizing out the 3-D position of landmark l, the expression in (6) can be rewritten in a structured, low-rank form

$$\Delta_l = F_l^\top Q_l F_l, \qquad Q_l \triangleq I - E_l \left(E_l^\top E_l\right)^{-1} E_l^\top \tag{14}$$

where $F_l \in \mathbb{R}^{3n_\ell \times 9(T+1)}$ and $E_l \in \mathbb{R}^{3n_\ell \times 3}$. Here, n_ℓ denotes the *number of frames in which landmark l is observed*, which is readily available via visibility checks (see [5] for details). This decomposition enables a computational shortcut: instead of recomputing the matrix inverse $(\Omega_{\mathsf{S}} + \Delta_l)^{-1}$ at each iteration, we maintain Ω_{S}^{-1} and apply the Kailath variant of the Woodbury identity. Since the update cost scales with $\mathrm{rank}(\Delta_l)^2 \cdot n$ rather than n^3, this yields a practical "fast greedy" algorithm with a per-iteration complexity of approximately $\mathcal{O}(\kappa N n_\ell^2 T)$, compared to the original $\mathcal{O}(\kappa N T^3)$. The next section details this efficient implementation.

B. Fast Low-Rank Greedy

Selecting the top κ landmarks from a set of N candidates is appealing in theory, but quickly becomes computationally infeasible in practice. Even for a modest time horizon $T = 20$, the matrix dimension becomes $n = 9(T+1) = 189$, and inverting a dense 189×189 matrix for each candidate landmark at every greedy step entails a cost of $\mathcal{O}(n^3) \approx 7 \times 10^6$ flops. This makes real-time decision-making impractical when N is large.

To overcome this bottleneck, we leverage the low-rank structure of each landmark update Δ_l derived in (14). By applying the Kailath variant of the Woodbury identity and maintaining the inverse Ω_{S}^{-1} throughout the algorithm, each matrix inverse can be updated efficiently with complexity $\mathcal{O}(n \cdot n_\ell^2)$, where n_ℓ is the number of frames in which landmark l is observed.

Algorithm 2 Fast Low-Rank Greedy.

```
Input: $\Omega_\emptyset$, $\{F_l, Q_l\}$ for all $l \in \mathsf{U}$, and $\kappa$
Output: subset $\mathsf{S} \subset \mathsf{U}$ with $|\mathsf{S}| = \kappa$, and $\Omega_{\mathsf{S}}^{-1}$
$\mathsf{S} \leftarrow \emptyset$,   $\Omega_{\mathsf{S}}^{-1} \leftarrow \Omega_\emptyset^{-1}$
while $|\mathsf{S}| < \kappa$ do
  for each $l \in \mathsf{U} \setminus \mathsf{S}$ do
    $W_l \leftarrow I + Q_l F_l \Omega_{\mathsf{S}}^{-1} F_l^\top$
    $f_l(\mathsf{S}) \leftarrow \mathrm{Trace}\left(\Omega_{\mathsf{S}}^{-1} F_l^\top W_l^{-1} Q_l F_l \Omega_{\mathsf{S}}^{-1}\right)$
  end for
  $j \leftarrow \arg\max_{l \in \mathsf{U} \setminus \mathsf{S}} f_l(\mathsf{S})$
  $\mathsf{S} \leftarrow \mathsf{S} \cup \{j\}$
  $\Omega_{\mathsf{S}}^{-1} \leftarrow \Omega_{\mathsf{S}}^{-1} - \Omega_{\mathsf{S}}^{-1} F_j^\top W_j^{-1} Q_j F_j \Omega_{\mathsf{S}}^{-1}$
end while
```

The efficient update for a candidate $l \notin \mathsf{S}$ proceeds by computing

$$(\Omega_{\mathsf{S}} + \Delta_l)^{-1} = \Omega_{\mathsf{S}}^{-1} - \Omega_{\mathsf{S}}^{-1} F_l^\top (I + Q_l F_l \Omega_{\mathsf{S}}^{-1} F_l^\top)^{-1} Q_l F_l \Omega_{\mathsf{S}}^{-1} \tag{15}$$

where $Q_l = I - E_l (E_l^\top E_l)^{-1} E_l^\top$. Substituting (15) into the objective function, the marginal information gain $f(\mathsf{S} \cup \{l\}) - f(\mathsf{S})$ becomes

$$\mathrm{Trace}\left(\Omega_{\mathsf{S}}^{-1} F_l^\top (I + Q_l F_l \Omega_{\mathsf{S}}^{-1} F_l^\top)^{-1} Q_l F_l \Omega_{\mathsf{S}}^{-1}\right). \tag{16}$$

offering a substantial reduction in runtime compared to full inversion.

Algorithm 2 presents the full greedy selection method based on the Kailath/Woodbury-based update (15) and the marginal gain computation (16). This version of the greedy algorithm maintains the same theoretical guarantees as the classic method but reduces each iteration's computational cost from $\mathcal{O}(Nn^3)$ to $\mathcal{O}(Nnn_\ell^2)$.

Although this method significantly reduces the cubic dependence on T, the overall complexity still scales linearly with κ, the number of selected landmarks. To further enhance efficiency, we draw inspiration from the randomized greedy strategy introduced in [29]. This approach samples a random subset of candidates at each iteration, which reduces the evaluation cost without severely degrading performance. While it is possible to apply the low-rank Kailath/Woodbury update within this randomized framework, we opt not to combine the two here in order to keep the conceptual contributions clearly delineated.

C. Randomized Greedy

The complexity associated with the greedy algorithm for feature selection can become prohibitive. To address this, we have devised a computationally efficient randomized greedy algorithm, inspired by the techniques introduced in[16] and [29]. This algorithm, detailed in Algorithm 3, is accompanied by a comprehensive explanation of its performance guarantees in this section.

Algorithm 3 performs the task of feature selection as follows. At each iteration, a subset V of size r is sampled uniformly at random and without replacement from the set of available features at the given time frame. The marginal gain provided by each of these r features to the objective function $f(\cdot)$ in (2) is computed, and the feature yielding the highest marginal gain is added to the set of selected features. This procedure is repeated κ times.

Algorithm 3 Randomized Greedy Algorithm

```
1: **Input:** $\Omega_\emptyset$, $\Delta_l$ for all $l \in \mathsf{U}$, $\kappa$, and $\epsilon \in [e^{-\kappa}, 1]$
2: **Output:** subset $\mathsf{S} \subset \mathsf{U}$ with $|\mathsf{S}| = \kappa$
3: $\mathsf{S} \leftarrow \emptyset$
4: **for** $j = 1$ to $\kappa$ **do**
5:     sample subset $\mathsf{V}$ with $r = \frac{N}{\kappa}\log(\frac{1}{\epsilon})$ features
       uniformly from $\mathsf{U} \setminus \mathsf{S}$
6:     $\mathsf{S} \leftarrow \mathsf{S} \cup \{\arg\max_{l \in \mathsf{V}} [f(\mathsf{S} \cup \{l\}) - f(\mathsf{S})]\}$
7: **end for**
```

The parameter $\epsilon \in [e^{-\kappa}, 1]$ in Algorithm 3, which determines the size of the uniformly sampled features at each iteration, represents a predefined constant selected to achieve a desired tradeoff between performance and complexity. When $\epsilon = e^{-\kappa}$, each iteration involves all nonselected features in U, reducing Algorithm 3 to the simple greedy scheme of Algorithm 1. Conversely, as ϵ approaches 1, the size of V and, consequently, the overall computational complexity decrease. However, as we will see, this reduction in complexity comes at the cost of decreased performance.

Definition 4: The elementwise curvature α_i of a monotone nondecreasing function f is defined as $\alpha_i \triangleq \max_{(\mathsf{S},\mathsf{R},l)\in\Pi_i} \frac{f_l(\mathsf{R})}{f_l(\mathsf{S})}$, where

$$\Pi_i = \{(\mathsf{S},\mathsf{R},l)\,|\,\mathsf{S} \subset \mathsf{R} \subset \mathsf{U}, l \in \mathsf{U} \setminus \mathsf{R}, |\mathsf{R} \setminus \mathsf{S}| = i, |\mathsf{U}| = N\}.$$

The maximum elementwise curvature is denoted by $\alpha_{\max} = \max_i \alpha_i$.

A set function is submodular if and only if $\alpha_{\max} \leq 1$. In the following theoretical analysis, we evaluate Algorithm 3 and establish a bound on its performance when used to approximate a solution to problem (2).

Theorem 2: Let $\alpha_{\max}$ be the maximum elementwise curvature of the VIN performance measure (2), S_{rand} denote the subset of features selected by Algorithm 3, and S_{OPT} be the optimal solution to Problem (2). Then, in expectation, $f(\mathsf{S}_{\text{rand}})$ is a constant multiplicative factor away from $f(\mathsf{S}_{\text{OPT}})$, specifically

$$\mathbb{E}[f(\mathsf{S}_{\text{rand}})] \geq \left(1 - e^{-1/c} - \frac{\epsilon^\eta}{c}\right) f(\mathsf{S}_{\text{OPT}}) \tag{17}$$

where $c = \max\{\alpha_{\max}, 1\}$, $\epsilon \in [e^{-\kappa}, 1]$, and $\eta = 1 + \max\{0, \frac{r}{2N} - \frac{1}{2(N-r)}\}$.

Proof: The proof is a straightforward variation of the proof of [29, Th. 2] and is not repeated here. □

Theorem 2 demonstrates that Algorithm 3 identifies a subset of visual features that, on average, achieves a VIN objective within a constant multiplicative factor of the objective obtained by the optimal set.

Drawing from the classical analysis in [14], we can establish that the approximation factor for the greedy algorithm is $1 - e^{-1/c}$. Thus, the term ϵ^η/c in the multiplicative factor in (17) quantifies the gap between the performance of the proposed randomized greedy algorithm and the traditional greedy approach of Algorithm 1.

Notice that the multiplicative factor in (17) decreases with both c and ϵ. While c is entirely dependent on the objective, reducing ϵ increases the multiplicative factor, leading to a better approximation. However, reducing ϵ also means approaching the conventional greedy algorithm, thereby increasing complexity.

Algorithm 4 Linearization-Based Algorithm.

```
**Input:** $\Omega_\emptyset$, $\Delta_l$ for all $l \in \mathsf{U}$, and $\kappa$
**Output:** subset $\mathsf{S} \subset \mathsf{U}$ with $|\mathsf{S}| = \kappa$
$\mathsf{S} \leftarrow \emptyset$
compute $\text{Trace}(\Omega_\emptyset^{-2}\Delta_\text{l})$ for all $l \in \mathsf{U}$
select the indices of the $\kappa$ largest elements to form $\mathsf{S}$
```

Observe that line 6 in Algorithm 3 has a computational cost of $\mathcal{O}(\frac{N}{\kappa}T^3\log(\frac{1}{\epsilon}))$, due to the requirement to evaluate $\frac{N}{\kappa}\log(\frac{1}{\epsilon})$ marginal gains, each necessitating $\mathcal{O}(T^3)$ operations. With κ such iterations, the overall computational complexity of Algorithm 3 is $\mathcal{O}(NT^3\log(1/\epsilon))$. Here, N represents the total number of extracted features at the current time frame, and T denotes the forward horizon estimation. Consequently, this approach results in a significant complexity reduction, by a factor of $\kappa/\log(1/\epsilon)$. This efficiency gain is particularly beneficial in practical applications with a large number of detected features and extended forward horizons.

D. Linearization-Based Greedy

This approach relies on a first-order (linear) approximation of the MSE objective (1). The approximation is valid when the Frobenius norms of the candidate information increments Δ_l ($l \in \mathsf{U}$) are sufficiently small relative to the baseline matrix $\Omega_\emptyset$. The next result derives the Taylor expansion of ρ by evaluating its directional derivative.

Lemma 2. (First-order Taylor expansion of ρ): Let $\rho : \mathbb{S}^n_{++} \to \mathbb{R}$ be defined by $\rho(A) = \text{Trace}(\text{A}^{-1})$, i.e., the mean-squared-error objective in (1). Fix $\Omega_\emptyset \in \mathbb{S}^n_{++}$ and define $\Delta_\mathsf{S} \triangleq \sum_{l\in\mathsf{S}} \Delta_l$ with $\Delta_l \in \mathbb{S}^n_+$.

If ρ is differentiable at $\Omega_\emptyset$, then for any sufficiently small $\epsilon > 0$

$$\rho\big(\Omega_\emptyset + \epsilon\,\Delta_\mathsf{S}\big) = \rho\big(\Omega_\emptyset\big) + \epsilon\,\text{Trace}\big(\nabla\rho(\Omega_\emptyset)\,\Delta_\mathsf{S}\big) + \mathcal{O}\big(\epsilon^2\big). \tag{18}$$

Proof: The differentiability of ρ at $\Omega_\emptyset$ (established later) allows us to write its first-order Taylor expansion in the direction Δ_S. For sufficiently small $\epsilon > 0$

$$\rho\big(\Omega_\emptyset + \epsilon\Delta_\mathsf{S}\big) = \rho\big(\Omega_\emptyset\big) + \epsilon\,D\rho(\Omega_\emptyset)\,[\Delta_\mathsf{S}] + \mathcal{O}\big(\epsilon^2\big)$$

where $D\rho(\Omega_\emptyset)[\cdot]$ denotes the directional derivative of ρ at $\Omega_\emptyset$.

On the space $\mathbb{S}^n$ of symmetric matrices equipped with the trace inner product $\langle X, Y\rangle \triangleq \text{Trace}(\text{XY})$, this directional derivative can be expressed through the gradient matrix $\nabla\rho(\Omega_\emptyset)$ as

$$D\rho(\Omega_\emptyset)\,[\Delta_\mathsf{S}] = \big\langle\nabla\rho(\Omega_\emptyset), \Delta_\mathsf{S}\big\rangle = \text{Trace}\big(\nabla\rho(\Omega_\emptyset)\,\Delta_\mathsf{S}\big).$$

Substituting this representation into the expansion yields (18), completing the proof. □

Given (18) and the monotonicity of the performance measure ρ, if the norms of the information matrices associated with the candidate features are sufficiently small, then the original problem (2) can be approximated by

$$\max_{\mathsf{S}\subset\mathsf{U}} -\sum_{l\in\mathsf{S}} \text{Trace}\big(\nabla\rho(\Omega_\emptyset)\,\Delta_\text{l}\big) \quad \text{subject to} \quad |\mathsf{S}| \leq \kappa. \tag{19}$$

We now show that the MSE performance measure ρ is differentiable and derive its directional derivative at $\Omega_\emptyset$ in the direction of Δ_l.

Let $\mathrm{sort} : \mathbb{R}^n \to \mathbb{R}^n$ denote the operator that rearranges a vector's components in nonincreasing order. A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *symmetric* if $f(x) = f\big(\mathrm{sort}(x)\big)$ for every $x \in \mathbb{R}^n$; that is, permuting the entries of x leaves the value of f unchanged. When a symmetric function is applied to the eigenvalues of a symmetric matrix, the resulting mapping is referred to as a *spectral function*.

The MSE-based objective $\rho : \mathbb{S}^n_{++} \to \mathbb{R}_{>0}$, $\rho(\Omega) = \mathrm{Trace}\big(\Omega^{-1}\big)$ with $n = 9T + 9$ is a spectral function whose value depends solely on the eigenvalues of Ω. In particular, we can write ρ as the composition of the eigenvalue map λ with the scalar function ϕ: $\rho(\Omega) = (\phi \circ \lambda)(\Omega) = \phi\big(\lambda(\Omega)\big)$, with $\Omega \in \mathbb{S}^n_{++}$. Here $\lambda(\Omega) = \big[\lambda_1(\Omega), \lambda_2(\Omega), \ldots, \lambda_n(\Omega)\big]^{\mathsf{T}}$ is a vector in $\mathbb{R}^n_{>0}$ that collects the eigenvalues of Ω, and $\phi(\lambda) = \sum_{i=1}^{n} \frac{1}{\lambda_i}$ with $\lambda \in \mathbb{R}^n_{>0}$. The next lemma is a restatement of [38, Corollary 5.2.5].

Lemma 3. (Spectral differentiability): Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be symmetric, closed, and convex. Then $\phi \circ \lambda$ is differentiable at a matrix $X \in \mathbb{S}^n_{++}$ if and only if ϕ is differentiable at $\lambda(X)$.

Proposition 2: The MSE objective $\rho(\Omega) = \mathrm{Trace}\big(\Omega^{-1}\big) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{1}{\lambda_{\mathrm{i}}(\Omega)}$, is differentiable at every $\Omega \in \mathbb{S}^n_{++}$.

In the reminder of this section, we provide a closed-form expression for the directional derivative of the MSE objective ρ. We also establish an explicit bound on the quadratic remainder term in its Taylor expansion, thereby identifying the regime in which the linear approximation remains accurate.

Proposition 3. (Quadratic-order error bound): Let $\Omega_\emptyset \in \mathbb{S}^n_{++}$ be the base information matrix, and define $\Delta_{\mathsf{S}} = \sum_{l \in \mathsf{S}} \Delta_l$ for any $\mathsf{S} \subseteq \mathsf{U}$ with $|\mathsf{S}| \leq \kappa$. Assume each $\Delta_l \succeq 0$ satisfies $\|\Delta_l\|_F \leq \zeta$, and choose $\epsilon > 0$ so that $\epsilon \, \|\Omega_\emptyset^{-1} \, \Delta_{\mathsf{S}}\|_2 < 1$. Then

$$\rho\big(\Omega_\emptyset + \epsilon \, \Delta_{\mathsf{S}}\big) = \rho(\Omega_\emptyset) - \epsilon \sum_{l \in \mathsf{S}} r_l\big(\Omega_\emptyset^2\big) + \underbrace{\epsilon^2 \, \mathrm{Trace}\Big(\Omega_\emptyset^{-1} \, \Delta_{\mathsf{S}} \, \Omega_\emptyset^{-1} \, \Delta_{\mathsf{S}} \, \Omega_\emptyset^{-1}\Big)}_{\triangleq R_{\mathsf{S}}(\epsilon)} + \mathcal{O}(\epsilon^3) \tag{20}$$

where $r_l\big(\Omega_\emptyset^2\big) = \mathrm{Trace}\big(\Omega_\emptyset^{-2} \, \Delta_{\mathrm{l}}\big)$, is the leverage score of the lth feature with respect to $\Omega_\emptyset^2$ (see [6] and the references therein for a complete definition of leverage score). Moreover, the quadratic remainder $R_{\mathsf{S}}(\epsilon)$ is nonnegative and satisfies

$$0 \leq R_{\mathsf{S}}(\epsilon) \leq \epsilon^2 \, \|\Omega_\emptyset^{-1}\|_2^3 \, \|\Delta_{\mathsf{S}}\|_F^2 \leq \epsilon^2 \, \|\Omega_\emptyset^{-1}\|_2^3 \, (\kappa \, \zeta)^2. \tag{21}$$

Implications for feature selection: Proposition 3 shows that the leading-order term in the Taylor expansion of the MSE objective is the (negated) sum of leverage scores, $-\,\epsilon \sum_{l \in \mathsf{S}} r_l\big(\Omega_\emptyset^2\big)$, which is precisely the modular objective maximized in Algorithm 4. Because modular objectives admit an optimal greedy solution, Algorithm 4 returns the *exact* maximiser of the linear surrogate (19). Using a heap to keep the current top κ scores, the procedure runs in $\mathcal{O}\big(N\,T^3 \log \kappa\big)$ time, a reduction of roughly $\kappa / \log \kappa$ compared with the classical greedy algorithm.

Accuracy of the linear surrogate: The quadratic remainder $R_{\mathsf{S}}(\epsilon) = \epsilon^2 \, \mathrm{Trace}\Big(\Omega_\emptyset^{-1} \Delta_{\mathsf{S}} \, \Omega_\emptyset^{-1} \Delta_{\mathsf{S}} \, \Omega_\emptyset^{-1}\Big)$, is always nonnegative and satisfies the bound $0 \leq R_{\mathsf{S}}(\epsilon) \leq \epsilon^2 \, \|\Omega_\emptyset^{-1}\|_2^3 \, \|\Delta_{\mathsf{S}}\|_F^2 \leq \epsilon^2 \, \|\Omega_\emptyset^{-1}\|_2^3 \, (\kappa\zeta)^2$. Hence, when each individual information increment obeys $\|\Delta_l\|_F \leq \zeta \ll \lambda_{\min}(\Omega_\emptyset)$ and at most κ features are chosen, the total perturbation remains small ($\|\Delta_{\mathsf{S}}\|_F \leq \kappa\zeta$). In this *small-increment regime* the second-order term is $\mathcal{O}\big((\kappa\zeta)^2\big)$ and is dominated by the linear term for sufficiently small ϵ.

We adopt the small–increment condition $\zeta \ll \lambda_{\min}(\Omega_\emptyset)$ for three reasons: 1) it guarantees $\epsilon\|\Omega_\emptyset^{-1}\Delta_{\mathsf{S}}\|_2 < 1$, so the Neumann expansion in Proposition 3 is valid; 2) it bounds the quadratic remainder by $R_{\mathsf{S}}(\epsilon) \leq \epsilon^2\|\Omega_\emptyset^{-1}\|_2^3(\kappa\zeta)^2 = \epsilon^2(\kappa\zeta)^2\lambda_{\min}^{-3}(\Omega_\emptyset) \ll \epsilon^2(\kappa\zeta)\lambda_{\min}^{-2}(\Omega_\emptyset)$, ensuring this term is negligible compared with the linear one that scales as $\mathcal{O}(\kappa\zeta)$; and 3) it keeps $\Omega_\emptyset + \epsilon\Delta_{\mathsf{S}}$ well conditioned and PD, so $\Omega_\emptyset^{-1}$ remains a stable reference for leverage scores. These points collectively justify the accuracy of the linearized surrogate in the small–increment regime.

Guarantee: Consequently, the greedy subset returned by Algorithm 4 incurs a true MSE cost that differs from the optimal by at most $\epsilon^2 \, \|\Omega_\emptyset^{-1}\|_2^3 \, (\kappa\zeta)^2$, a quantity that can be made arbitrarily small by choosing ϵ proportional to the typical size of the increments Δ_l. Proposition 3 thus provides a rigorous justification for replacing the original nonsubmodular objective with its linearized surrogate and for employing Algorithm 4 as an efficient, near-exact feature-selection strategy.

IV. Experiments

The attention pipeline discussed in this article introduces three main challenges for real-world deployment. First, the system must integrate a feature extraction module to detect salient features from each image frame. Second, it must incorporate real onboard IMU measurements. Third, it must perform visibility checks over a future time horizon, which requires predicting landmark observability based on the robot's dynamical model and control inputs.

To evaluate the proposed selection methods under these challenges, we design two complementary sets of experiments. The first set uses sequences from the EuRoC dataset [30] and compares our methods against baseline approaches. This dataset provides feature measurements and IMU data, enabling a fair benchmarking against existing methods in the literature. However, since control inputs are not available in EuRoC, the visibility check is performed using ground-truth future poses rather than predicted ones based on the robot dynamical model.

To address this limitation and evaluate the full attention pipeline proposed in the article, we conduct a second set of experiments using a 1/10-scale RC car equipped with onboard stereo cameras and an IMU. In this setup, control inputs are available, allowing us to implement visibility checks based on predicted future horizons using the robot's dynamics and control commands. Various selection strategies are tested on this platform to assess their practical performance.

The remainder of this section details the experimental setup and evaluation results for both datasets.

A. Benchmark Evaluation and Comparison

For this study, we use multiple sequences from the EuRoC benchmark dataset [30]. The visual–inertial state estimation pipeline is based on VINS-Mono [39], excluding loop closure (i.e., using full odometry), which provides both the front-end and back-end processing modules. We modify the front-end to incorporate the different feature selection strategies proposed in this article. The EuRoC benchmarks were executed on a laptop equipped with an Intel Core i9-12900H CPU (20 cores, 5.0 GHz), 32 GB of RAM, and an NVIDIA GeForce RTX 3080 Ti GPU.

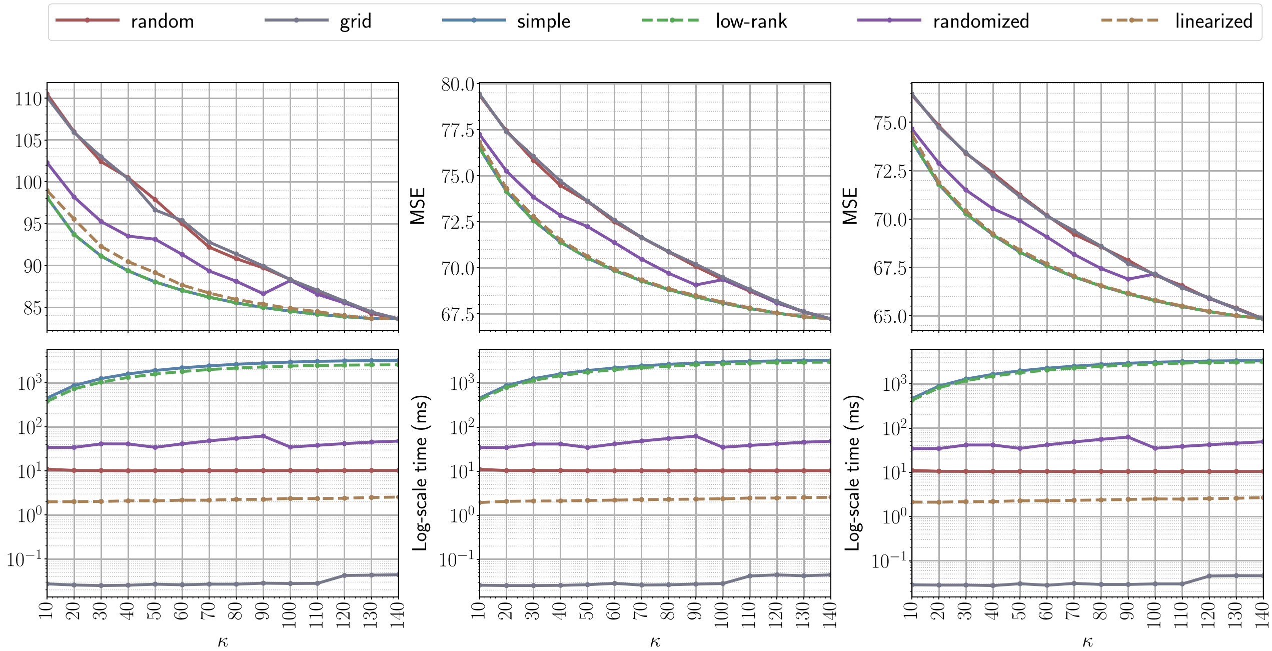


Fig. 3. Performance comparison of different feature selection methods on the MH_01_easy sequence from the EuRoC dataset. Each column corresponds to a single randomly selected video frame. The top plot in each column shows the MSE values versus the number of selected features κ, while the lower plot presents the computation time on a logarithmic scale to enhance visibility and highlight discrepancies. Methods compared include uniform random selection ("random"), grid-based selection ("grid"), simple greedy ("simple"), fast low-rank greedy ("low-rank"), randomized greedy ("randomized"), and linearization-based greedy ("linearized"). For randomized methods, each experiment was repeated 20 times, and the mean values are reported. The prediction horizon T is set to 13 for the information-aware selection methods, and the hyperparameter ϵ in the randomized greedy algorithm is set to 0.5. Note that MSE values and feature counts vary across frames, so results are presented for three representative frames without averaging across the sequence.

The EuRoC dataset was collected using an AscTec Firefly hex-rotor UAV equipped with a stereo visual–inertial sensor suite. The stereo camera captures images at a resolution of 752×480 pixels and a frame rate of 20 Hz, while the IMU measurements are recorded at 200 Hz. In our experiments, the measurement noise parameters, along with the intrinsic and extrinsic calibrations, are set to exactly match those provided in the dataset.

In the front-end, we employ OpenCV's implementation of the Shi–Tomasi method [40] for feature detection and the Lucas–Kanade method [41] for feature tracking. The feature extractor detects up to $N = 150$ features per frame. From this set, our selection algorithms retain at most $\kappa \ll N$ features, which are then passed to the back-end for state estimation.

In practice, future pose estimates along the prediction horizon can be obtained by integrating the vehicle's dynamics using control inputs. However, since the EuRoC dataset does not provide control inputs, we approximate future poses by applying ground-truth motion increments to the current pose estimate. The feature selector operates over a prediction horizon of length T, set to $T = 13$ in all plots in this section. This value reflects the longest horizon that allows efficient simulation without incurring significant computational overhead. In general, larger values of T increase the computational cost for all selection methods. Later in this section, we provide further analysis and plots illustrating the impact of T on both performance and runtime.

We compare various VIN approaches designed to minimize the MSE and alternative objectives over the forward prediction horizon. When the algorithm is limited by a selection budget κ, and r features are successfully tracked from the previous frame, only $\kappa - r$ new features are selected in the current frame. However, this preselection mechanism is disabled for per-frame experiments, such as the one shown in Fig. 3. For completeness, we initially considered including the lazy greedy algorithm [5, Algorithm 1]), known for its computational efficiency. However, it is excluded from our evaluation because it requires a lower bound (in the case of minimization) on the cost function $f_l(\mathsf{S})$ for each candidate feature l at every iteration. Existing bounds in the literature for MSE-based objectives are known to be loose, causing the lazy greedy algorithm to converge to the same performance as standard greedy, thus offering no practical advantage in this setting.

We evaluate four greedy algorithms proposed in this article: simple greedy (Algorithm 1), fast low-rank greedy (Algorithm 2), randomized greedy (Algorithm 3), and linearization-based greedy (Algorithm 4), which are labeled in the figures as "simple," "low-rank," "randomized," and "linearized," respectively. For comparison, we also include two baseline methods. The first is a purely random selection algorithm, labeled "random," which uniformly samples features without regard to spatial or informational structure. The second, labeled "grid," divides each frame into a 15×12 grid and selects an equal number of features from each cell, promoting spatially uniform coverage across the image. For easier interpretation, we use the same color for each method consistently across all plots in this section.

Fig. 3 presents the performance of the six selection methods on the MH_01_easy sequence from the EuRoC dataset. Each column corresponds to a single video frame. In each column, the top subplot shows the MSE values (vertical axis), and the bottom subplot shows the computation time on a logarithmic scale to enhance visibility (vertical axis), both plotted against the number of selected features κ (horizontal axis). The MSE values vary significantly across frames due to changes in the robot's location along the trajectory, making it difficult to meaningfully report average values across all frames (note the differing MSE scales in each column). Additionally, the total number of features detected by the extractor varies from frame to frame, which prevents consistent plotting of mean performance curves across methods for a fixed range of κ. Therefore, we report results for three randomly selected frames to provide representative comparisons without aggregating over inconsistent conditions.

In each curve shown in the figure, every point on the horizontal axis corresponds to a selection budget κ enforced for the given frame. For methods involving randomization, each experiment is repeated 20 times, and the average performance is reported. The hyperparameter ϵ in the randomized greedy algorithm is set to 0.5 for this figure.

The performance of all algorithms improves as κ increases. The uniform random and spatially gridded methods, while computationally efficient, underperform in terms of estimation accuracy. The randomized greedy algorithm performs better than these two, particularly at small κ, but still lags behind the three greedy-based strategies proposed in this article. Although randomized greedy is relatively fast, its runtime remains higher than that of the linearization-based greedy method.

Notably, the performance of the randomized greedy method begins to converge to that of uniform random as κ increases. This is a result of using a fixed ϵ: as κ grows, the number of candidates sampled in line 5 of Algorithm 3 becomes a smaller fraction of the total, causing the method to behave increasingly like a random selector. Similarly, the observed fluctuations in the runtime of randomized greedy stem from this fixed ϵ: although increasing κ generally raises cost, smaller sample sizes at larger κ can offset this, producing nonmonotonic trends. Nevertheless, the overall time complexity remains stable and approximately constant.

As expected, the fast low-rank greedy algorithm (Algorithm 2) closely tracks the performance of the simple greedy algorithm (Algorithm 1) while executing significantly faster. Interestingly, the linearization-based greedy method (Algorithm 4) also achieves nearly identical estimation performance, but at a fraction of the computational cost, with a runtime that is effectively constant across κ, outperforming all other methods except for the spatially gridded baseline.

These results highlight that the choice of feature selection method should be guided by the specific requirements of the application. When computational resources are limited and fast decision-making is crucial, for instance in onboard processing on micro aerial vehicles, the linearization-based greedy algorithm emerges as the most practical option. It provides near-optimal accuracy at a fraction of the cost, with virtually constant runtime. In contrast, if achieving the lowest possible estimation error is the primary goal and computational constraints are relaxed, the fast low-rank greedy algorithm offers the best tradeoff: it maintains the performance of full greedy selection while significantly reducing runtime. For offline evaluation or small-scale problems where cost is not a limiting factor, the simple greedy method remains a reliable, though expensive, benchmark. When approximate results are acceptable and further speedup is desired, randomized greedy presents a balanced middle ground that is especially effective at smaller selection budgets, though it may degrade as κ grows unless ϵ is adaptively tuned. Finally, in extreme low-power scenarios, such as onboard deployment with severe runtime limits, uniform random or spatially gridded methods are the cheapest alternatives, though they should be avoided if accuracy is a priority. Overall, these findings enable practitioners to make informed choices based on the performance–efficiency tradeoffs relevant to their robotics pipeline.

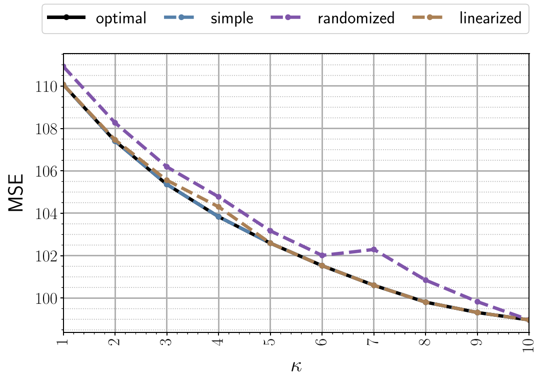
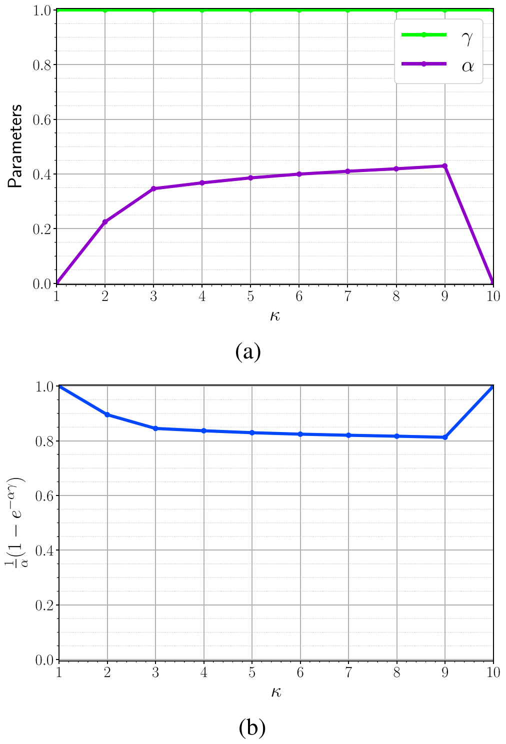


Fig. 4. Performance comparison of the proposed feature selection methods on the MH_01_easy sequence from the EuRoC dataset. The evaluation considers selecting $\kappa = 1, \ldots, 10$ features from a pool of 10 candidates, allowing comparison against the optimal solution obtained via exhaustive search. The results show that the greedy algorithm achieves identical performance to the optimal method, despite the MSE-based objective not being submodular. The linearized method closely matches this performance with minimal deviation, while the randomized approach exhibits slight fluctuations due to the fixed ϵ, yet follows a similar overall trend.

Fig. 4 presents a numerical evaluation comparing various selection methods proposed in this article against the optimal solution, thereby highlighting their relative performance bounds. To make exhaustive combinatorial search tractable, the experiment is limited to selecting from the top 10 candidate features, screened through a quality check, for a single frame of the MH_01_easy sequence. All other hyperparameters used in this experiment are identical to those in Fig. 3. Interestingly, the greedy and optimal results are identical. This is notable because the MSE-based VIN objective in (1) is not generally submodular, and the greedy algorithm typically offers weaker theoretical guarantees, as characterized in (11). Nonetheless, this observation is consistent with findings from prior work [5] and related studies in other domains [34], [42].

With only minor fluctuations, the linearized algorithm closely tracks the performance of both the greedy and optimal solutions. This supports our conclusion that, although linearization provides a surrogate for the MSE objective, it yields nearly identical performance while being significantly faster. As expected, the randomized greedy algorithm, despite some variability due to the fixed ϵ, follows the same general trend as the optimal, albeit with slightly reduced accuracy.

The curvature (α) and submodularity ratio (γ) for the greedy selection shown in Fig. 4 were calculated via exhaustive search using their respective definitions provided in (9) and (10). These values are then utilized to obtain the performance bound suggested by Theorem 1. The values of α and γ as functions of κ are presented in Fig. 5(a), while the corresponding calculated performance bound using these parameters is displayed in Fig. 5(b). Notably, for our nonsubmodular MSE-based VIN performance function (1), the submodularity ratio γ for the greedy selection



(a)

(b)

Fig. 5. (a) Curvature (α) and submodularity ratio (γ) as functions of κ. (b) Performance bound calculated using α and γ values.

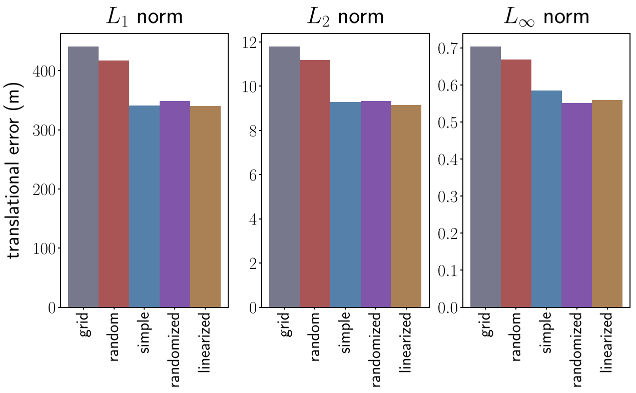


Fig. 6. Translational error (in meters) between the ground-truth and estimated UAV trajectories on the MH_01_easy sequence from the EuRoC dataset. At each time step, the Euclidean distance between corresponding positions, computed after applying a rigid transformation for alignment, is calculated. The overall error is then summarized using the L_1, L_2, and L_∞ norms over the sequence of these distances. The feature selection budget is fixed at $\kappa = 70$ for this experiment, and the selection methods correspond to those shown in Fig. 3.

over this specific set remains consistently equal to one for all values of κ, indicating submodularlike behavior. However, nonsubmodular functions can exhibit submodular behavior in certain regions or under specific conditions. For instance, the marginal gains for some particular subsets may exhibit submodular characteristics, even if the function globally violates submodularity.

The calculated bound for this greedy feature selection shows that the performance objective of the subset obtained by the greedy algorithm exceeds 80% of the optimal performance objective. This result perfectly explains why, in Fig. 4, the objective of the greedy selection closely matches that of the optimal action.

Fig. 6 presents the L_1, L_2, and L_∞ norms of the translational error (in meters) between the ground-truth and estimated UAV trajectories, computed over the entire flight path. To this end, the estimated and ground-truth trajectories are first aligned using the procedure described in [43]. Then, at each time step, the Euclidean distance between corresponding aligned positions is calculated. The overall error is summarized using the L_1, L_2, and L_∞ norms over the resulting sequence of distances. Each bar in the figure represents one of the selection methods from Fig. 3; the simple greedy method includes both the standard and low-rank variants, which yield identical results.

A closer look at the three panels yields several key observations. The linearization-based greedy algorithm achieves nearly identical performance to simple greedy in terms of both cumulative error (L_1) and root-mean-square error (L_2) across the full trajectory. Interestingly, the L_∞ plot shows that linearized greedy achieves slightly lower worst-case deviation from ground-truth compared to simple greedy. While this does not imply improved robustness in a strict sense, it suggests that the linear surrogate may better regulate error accumulation in certain trajectories. In addition, randomized greedy shows comparable performance to both simple and linearization-based greedy in L_1 and L_2, and even outperforms them in terms of L_∞. This indicates that a modest degree of random sampling can, at times, match or even exceed the performance of more computationally intensive methods. Finally, both uniform random and grid-based baselines remain the least accurate across all error metrics, reinforcing their limited suitability for precision-sensitive applications.

These insights refine our practical guidelines: when accuracy is the top priority and computational resources allow, simple (or fast low-rank) greedy remains the most reliable choice. For scenarios with tight runtime constraints, linearization-based greedy offers an attractive balance between accuracy and efficiency. Meanwhile, randomized greedy may be favored when both computational efficiency and robustness to worst-case errors are important, as it achieves competitive performance in L_1 and L_2, and attains the lowest L_∞ error among all methods, making it suitable for scenarios where large deviations must be minimized. Simpler heuristics such as uniform random or grid-based selection should be reserved for the most constrained applications, where runtime dominates and estimation precision is less critical.

Fig. 7 compares the performance of greedy feature selection when using two different objective functions: MSE and log-det. This comparison highlights how the choice of objective influences the quality of the solution in practice. The left subfigure presents the relative translational error (RTE) across various values of κ for both objectives, while the right subfigure shows the absolute translational error (ATE) for the same settings. Both RTE and ATE are computed following the definitions provided in [43]. In most cases, the MSE-based objective yields lower errors, consistently outperforming the log-det approach. This is expected, as MSE directly quantifies estimation error, whereas the log-det objective serves as a surrogate for overall uncertainty, capturing the volume of the estimation covariance ellipsoid but not directly minimizing specific error metrics like MSE.

Importantly, the log-det function is known to be submodular [5], which allows for the use of greedy algorithms with

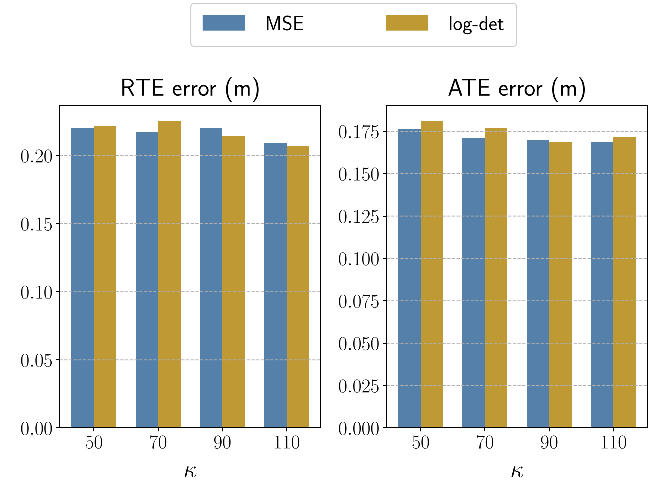
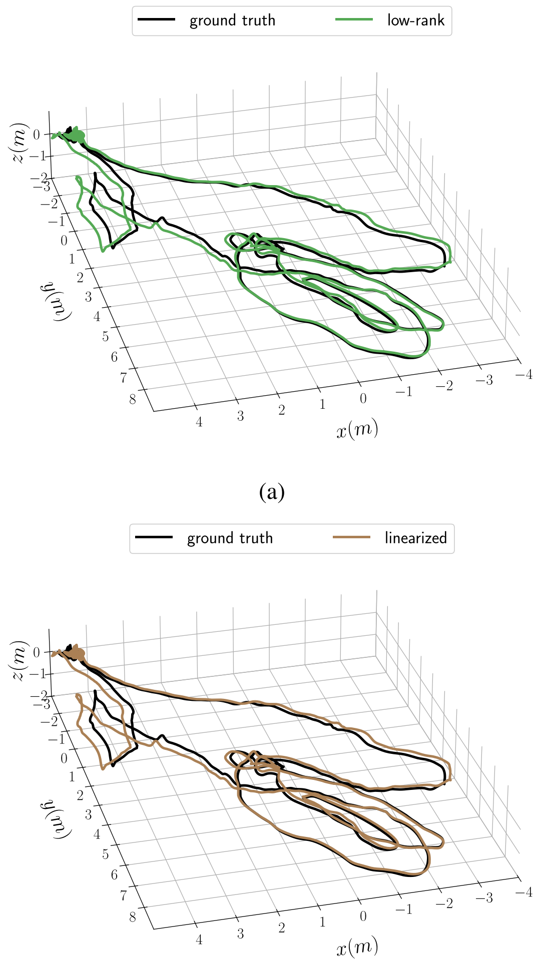


Fig. 7. Comparison between greedy feature selection using MSE and log-det objectives across different values of κ. Left: (RTE. Right: ATE. The MH_01_easy sequence from the EuRoC dataset is used for evaluation.

a worst-case approximation guarantee of $(1 - 1/e) \approx 63\%$ of the optimal. While this provides a strong theoretical guarantee, it does not always result in better empirical performance. In contrast, the MSE objective is not submodular. However, this article provides a constant-factor approximation guarantee for greedy selection under the MSE objective, despite its nonsubmodularity. As shown in Fig. 5, our proposed approximation bound is tighter in practice, demonstrating that the greedy solution under the MSE objective performs within 80% of the optimal solution even in the worst case. Notably, the bound developed in this article is general and can also be applied to the log-det objective, potentially yielding tighter guarantees than the standard $(1 - 1/e)$ bound.

To further analyze and visually assess the performance of the greedy algorithm, Fig. 8 presents the estimated UAV trajectories alongside the ground-truth over the entire flight path. For computational efficiency, the fast low-rank variant of greedy is used. The top subplot shows the estimated trajectory from the low-rank greedy method compared to the ground-truth, while the bottom subplot includes the trajectory obtained from the linearization-based greedy method. This method was included due to its comparable performance and significantly lower runtime, as demonstrated in earlier experiments.

The results shown in this figure support the earlier observations: both the fast low-rank greedy and linearization-based greedy methods follow the ground-truth trajectory closely, and the distance between their estimates and the ground-truth remains consistently small throughout the entire sequence. This explains why their translational error metrics (L_1, L_2, and L_∞) in Fig. 6 are nearly identical. For this experiment, the selection budget κ is set to 70.To exploit more deeply the EuRoC dataset for more meaningful results, Table I presents the performance of different feature selection strategies on three sequences from this benchmark, covering a range of difficulty levels. Challenging datasets such as MH_05_difficult involve fast motion and rapid viewpoint changes. All results are reported under a strict feature budget of $\kappa = 70$ features per frame. This constraint ensures fair comparison across methods and highlights the ability of each selection strategy to operate effectively under limited visual information.



(a)

(b)

Fig. 8. Visual comparison of the trajectories obtained using different selection techniques on the MH_01_easy sequence from the EuRoC dataset. The top subplot shows the estimated trajectory of the UAV using the fast low-rank greedy algorithm (in green), and the bottom subplot shows the trajectory estimated by the linearization-based greedy algorithm (Algorithm 4), in brown. The ground-truth trajectory is shown in black in both plots for baseline comparison. Both methods follow the ground-truth trajectory closely and maintain a similar distance from it over the entire flight path, which explains their nearly identical translational error norms in Fig. 6. For this experiment, the selection budget κ is set to 70 and the prediction horizon T is set to 13. (a) Estimated trajectory using fast low-rank greedy (green) vs. ground truth (black). (b) Estimated trajectory using linearization-based greedy (brown) vs. ground truth (black).

We include a simple baseline method, denoted “quality,” which selects the κ features with the highest Shi-Tomasi scores using the `goodFeaturesToTrack` function in OpenCV. This method is commonly used in VIN pipelines and only accounts for the appearance quality of the visual feature. Table I reports the RTE for all approaches, with the ATE also included for completeness. As before, both RTE and ATE are computed following the definitions provided in [43].

The results confirm the trends observed earlier. On the easier sequence (MH_01_easy), the simple greedy method achieves the lowest error, with randomized and linearization-based greedy methods performing nearly as well. The “quality” baseline performs competitively on this sequence, which is expected

TABLE I
EuRoC Results With 70 Features and MSE

Sequence	Method	RTE [m]	ATE [m]
MH_01_easy	quality	0.3843 ± 0.076	0.2323 ± 0.114
	random	0.4437 ± 0.085	0.2621 ± 0.127
	grid	0.4491 ± 0.088	0.2767 ± 0.133
	simple	0.3561 ± 0.080	0.2176 ± 0.110
	randomized	0.3613 ± 0.064	0.2187 ± 0.105
	linearized	0.3664 ± 0.073	0.2142 ± 0.10
MH_03_medium	quality	0.4753 ± 0.197	0.2923 ± 0.137
	random	0.8184 ± 0.284	0.3300 ± 0.143
	grid	0.5366 ± 0.195	0.3302 ± 0.142
	simple	0.5392 ± 0.203	0.3197 ± 0.154
	randomized	0.4971 ± 0.187	0.3066 ± 0.135
	linearized	0.4825 ± 0.192	0.2947 ± 0.132
MH_05_difficult	quality	0.7011 ± 0.214	0.3150 ± 0.075
	random	0.6655 ± 0.172	0.3005 ± 0.091
	grid	0.6482 ± 0.166	0.3091 ± 0.084
	simple	0.7204 ± 0.220	0.3189 ± 0.091
	randomized	0.6623 ± 0.168	0.3173 ± 0.078
	linearized	0.6606 ± 0.200	0.3039 ± 0.074

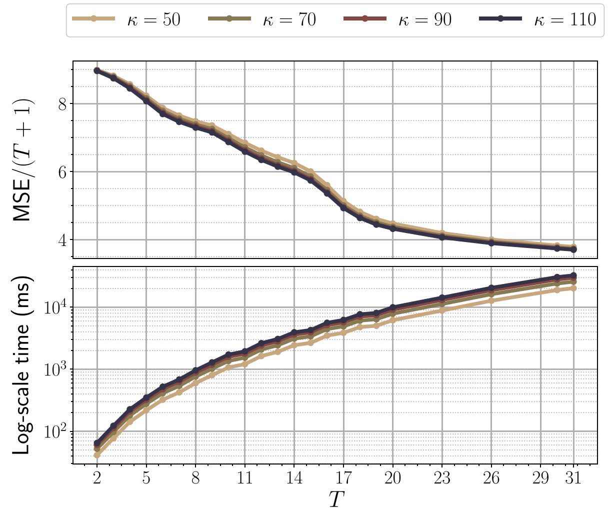


Fig. 9. Effect of prediction horizon T on estimation performance and computational cost for a single frame from the MH_01_easy sequence in the EuRoC dataset. The simple greedy algorithm is applied to the MSE-based objective. Top: Scaled MSE (trace of the inverse information matrix) as a function of horizon length T; performance improves with longer horizons until saturation. Bottom: Computational time (log scale) versus T; runtime increases with horizon length. Each curve represents a different feature budget κ.

given its moderate motion and stable feature tracks. However, on more dynamic sequences such as MH_03_medium and MH_05_difficult, methods that incorporate motion-aware criteria (such as greedy, randomized, and linearized) consistently outperform the quality-based, random, and grid-based baselines.

In MH_05_difficult, where motion is more aggressive, the linearization-based method achieves the lowest ATE and RTE among all strategies, except for one baseline in each case. This result suggests that its surrogate objective, despite being an approximation of MSE, remains effective even in challenging settings. Randomized greedy also performs well across all sequences, striking a good balance between computational cost and accuracy, especially in medium-difficulty settings.

These results further emphasize that appearance-based methods alone (e.g., "quality") may suffice for simple scenes but quickly degrade in more dynamic environments. In contrast, strategies that account for motion dynamics, such as the proposed greedy-based methods, offer more consistent and robust performance across a range of operating conditions.

Several existing feature selection approaches, such as [21] and [44], employ a similar idea but are limited to a one-step prediction horizon ($T = 1$). To examine the effect of increasing the prediction horizon T on both performance and computational cost, we conducted an experiment using the MH_01_easy sequence from the EuRoC dataset. In this experiment, we applied the simple greedy algorithm to an MSE-based objective function.

Fig. 9 summarizes the results. The top plot shows the scaled MSE performance, measured as the trace of the inverse information matrix, as a function of the horizon T. Since the size of the information matrix increases with $(T + 1)$, we normalize the MSE by $1/(T + 1)$ to enable a fair comparison across different horizon lengths. The bottom plot presents the corresponding computational time (on a logarithmic scale) versus T. Each curve corresponds to a different feature budget κ.

Two main observations can be made from the Fig. 9. First, for a fixed value of T, larger feature budgets κ lead to better MSE performance but also higher computational cost. Second, increasing the horizon T consistently improves performance up to a point. This improvement occurs because a longer horizon allows the algorithm to account for features that remain visible over extended periods, which tend to be more informative and reduce estimation error. However, once the horizon exceeds the visibility duration of these features, the performance gain saturates. Based on the figure, this saturation becomes evident when T reaches around 23 or higher. Given that the EuRoC camera operates at 20 Hz, a horizon of $T = 23$ corresponds to approximately one second into the future.

We verified that the same qualitative scaling trends with respect to κ and T hold for all algorithmic variants (low-rank, randomized, and linearized greedy). Including every corresponding plot would require multiple subfigures beyond the page limit; therefore, we make the complete set of results available in the accompanying project repository[1].

The MSE in Fig. 9 is computed for a single frame based on predicted feature visibility and the corresponding information matrix. Therefore, the MSE continues to improve with increasing T until this additional visibility information is fully exploited. In contrast, computational cost increases steadily with T, as shown in the bottom plot.

The MSE and computational cost reported in Fig. 9 are computed for a single frame with the highest number of detected features in the sequence, using precomputed feature visibility and the corresponding information matrix. As the prediction horizon T increases, the MSE improves until all available visibility information is fully exploited, while the computational cost steadily increases, as shown in the bottom plot. The purpose of this figure is to illustrate the trend in MSE and computational time as a function of the prediction horizon T, rather than to evaluate the real-time applicability of the selection method.

[1] https://github.com/SiamiLab/NonsubmodularVisualAttention The GitHub repository provides access to the dataset, implementation code, and video demonstrations associated with this work.

The next subsection details the real-time implementation of the proposed methods on a realistic QCar trajectory and presents corresponding runtime results.

B. Experimental Case Study

While the EuRoC dataset serves as a valuable benchmark for evaluating feature selection algorithms under diverse visual and motion conditions, it does not include control input data. This limitation prevents deployment of the full anticipation-based feature selection pipeline proposed in this work, which relies on forward simulation of robot dynamics for predicting future feature visibility.

To overcome this limitation and evaluate the complete pipeline, we conducted a complementary experiment using the QCar platform, a 1/10-scale autonomous vehicle designed for academic research in robotics and self-driving technologies [31]. This setup allows us to integrate control-aware future visibility prediction into our feature selection strategy.

The QCar is equipped with a 9-axis IMU and a ZED stereo camera. The IMU exhibits accelerometer and gyroscope noise standard deviations of $\sigma_a = 0.2$ and $\sigma_g = 0.2$, respectively, with corresponding bias random walks $\sigma_{a_b} = 0.001$ and $\sigma_{g_b} = 0.001$. IMU data is collected at 50 Hz, while the stereo camera captures synchronized images at a resolution of 1280×720 pixels at 30 Hz.

In the experiment, the QCar follows a cancer-ribbon shaped trajectory designed to introduce a range of headings. All sensor data, including IMU measurements, stereo images, and control commands, are recorded and formatted to align with the EuRoC dataset structure[1]. Ground-truth trajectories are obtained using a network of eight motion capture cameras (OptiTrack Prime X) installed in a $6 \times 6\tilde{}\mathrm{m}^2$ indoor lab environment. The experimental setup, comprising the QCar, the executed trajectory, and a representative set of real-world objects, was presented earlier in Fig. 1.

We extended the open-source VINS-Mono framework [39], excluding loop closure (i.e., using full odometry), to incorporate our feature selection techniques. In the anticipation-based approach, we simulate future motion using a discrete-time kinematic bicycle model. This model propagates future states based on the current pose estimate from the VIO backend and recent control commands (linear velocity and steering angle). Specifically, at each prediction step h, the heading ψ_h and position $t_h \in \mathbb{R}^3$ are updated according to

$$t_{h+1} = t_h + u_h \begin{bmatrix} \cos\psi_h \\ \sin\psi_h \\ 0 \end{bmatrix} \cdot \Delta t$$

$$\psi_{h+1} = \psi_h + \frac{u_h}{L} \tan\delta_h \cdot \Delta t \tag{22}$$

where u_h and δ_h are the linear speed and steering angle at step h, L is the vehicle's wheelbase, and Δt is the integration time step. The predicted trajectory $\{(t_h, \psi_h)\}$ allows forward visibility prediction and enables more informed feature selection. Although a constant velocity is assumed during propagation, small disturbances are naturally present in practice.

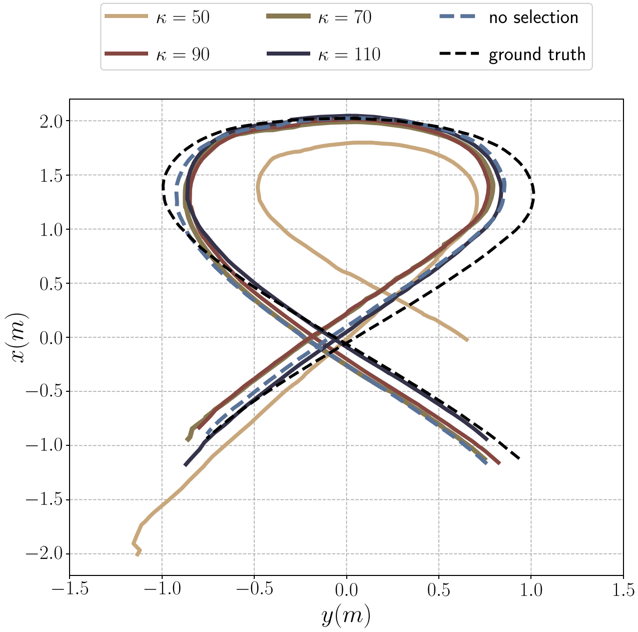


Fig. 10. Comparison of estimated trajectories under different feature budgets, using MSE-based selection with the simple greedy algorithm. The ground-truth trajectory from motion capture and the trajectory obtained without feature selection (i.e., using all features) are shown for reference.

We adapted the open-source VINS-Mono framework [39] to incorporate our feature selection methods. As in the EuRoC experiment, we use OpenCV's Shi–Tomasi [40] for feature detection and Lucas–Kanade [41] for tracking, extracting up to $N = 150$ features per frame. Our selection algorithms retain at most $\kappa \ll N$ features, which are then used by the back-end for state estimation.

As in the EuRoC experiments, the prediction horizon is set to $T = 13$. To evaluate the effect of the feature budget on estimation accuracy, Fig. 10 shows estimated trajectories under different values of κ, the number of features selected for back-end optimization. The simple greedy algorithm is applied to our MSE-based objective for feature selection. For comparison, we also show the ground-truth trajectory from the motion capture system and the trajectory obtained without any feature selection (i.e., using all extracted features at all times), labeled as "no selection". As expected, increasing the number of selected features leads to more accurate estimates, with trajectories that deviate less from the ground truth. Notably, the trajectory obtained under a moderate selection budget of $\kappa = 70$ achieves visual accuracy comparable to the no selection case, demonstrating the efficiency of the proposed selection method.

Quantitatively, the ATE computed for each trajectory following [43] further supports this discussion. The "no selection" case achieves $0.1681 \pm 0.069\,\mathrm{m}$ while the greedy selection with $\kappa = 50$, 70, 90, and 110 yields 0.4966 ± 0.248, 0.2394 ± 0.094, 0.2400 ± 0.091, and $0.1684 \pm 0.060\,\mathrm{m}$, respectively, confirming that budgets of $\kappa \geq 70$ attain accuracy statistically comparable to using all features.

The top plot in Fig. 11 shows the translational error (vertical axis) between the estimated and ground-truth trajectories from Fig. 10, plotted over time (horizontal axis). While the error fluctuates along the trajectory due to changes in robot motion and feature visibility, increasing the feature budget consistently

[2] https://github.com/SiamiLab/NonsubmodularVisualAttention The GitHub repository provides access to the dataset, implementation code, and video demonstrations associated with this work.

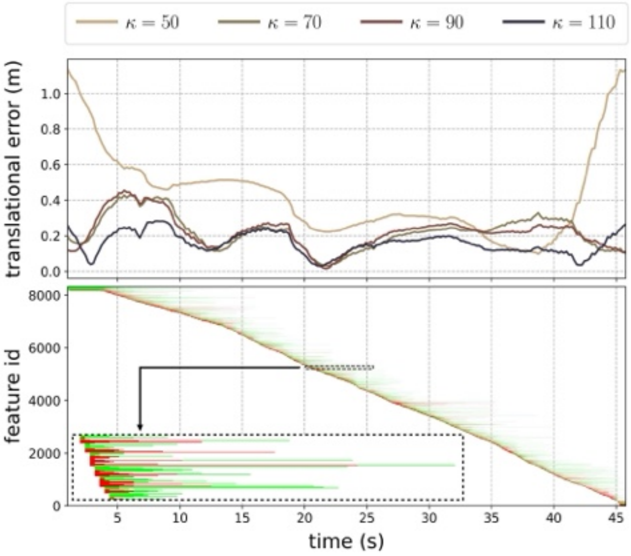


Fig. 11. Top: Translational error over time between the estimated and ground-truth trajectories for different feature budgets, corresponding to Fig. 10. Feature selection is performed using the simple greedy algorithm applied to the MSE-based objective. As expected, increasing the number of selected features reduces the estimation error. Bottom: Feature selection history over time for $\kappa = 90$. Each row represents a feature ID; green indicates selected features, red indicates observed but unselected features, and white indicates features not visible. The diagonal pattern reflects the short visibility window of each feature, and the consistent reselection of visible features demonstrates the expected behavior of the algorithm.

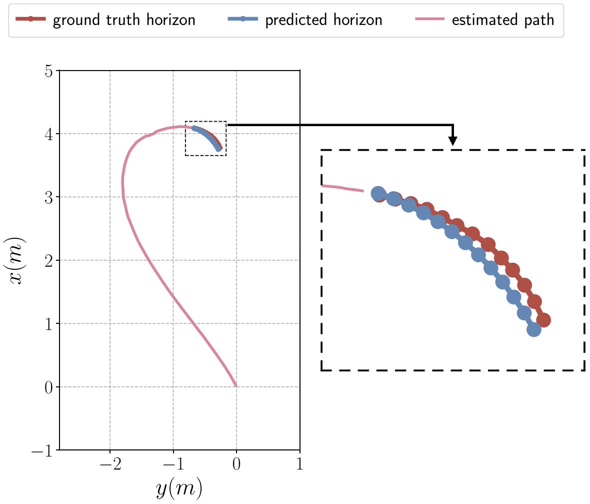


Fig. 12. Comparison between the predicted future horizon generated by the kinematic bicycle model and the ground-truth trajectory. The right panel shows a zoomed-in view of the predicted horizon to highlight the alignment between the simulated and actual motion.

TABLE II
Runtime Comparison for QCar Experiment Under $\kappa = 70$ Feature Budget

Selector	Selection [ms]	Ceres [ms]	Total [ms]
No Selection	–	24.75	24.75
Simple	82.01	13.14	95.14
Low-rank	72.25	13.85	86.10
Randomized	11.18	13.18	24.36
Linearized	7.75	13.61	21.36

Times are averaged per frame over the full trajectory .

reduces the overall estimation error, as expected. The bottom plot depicts the selection history of visual features over time when selecting up to $\kappa = 90$ features per frame (from a maximum of 150). The vertical axis corresponds to feature IDs. White indicates the feature was not observed at that time, red indicates it was observed but not selected, and green indicates it was selected. The diagonal banding pattern reflects that features are typically visible only for short durations as the robot moves. Moreover, once a feature is selected, it tends to be consistently reselected in subsequent frames until it leaves the field of view, demonstrating the expected temporal consistency of the selection algorithm.

In addition, Fig. 12 presents a snapshot of the algorithm at a specific point along the trajectory. At this moment, the predicted future positions using the bicycle model (discussed earlier) over a horizon of $T = 13$ are shown alongside the actual observed feature horizon. As illustrated, the predicted and actual horizons closely align at this randomly selected point. This figure serves as a demonstration of the accuracy of the motion prediction used in the anticipation-based feature selection.

Table II presents the average per-frame runtime for different feature selection strategies in the QCar experiment under a fixed feature budget of $\kappa = 70$. The "No Selection" baseline uses all available features (up to 150 per frame) without any filtering, requiring no runtime for selection but resulting in a significantly higher optimization time in Ceres. In contrast, all other methods apply selection before back-end optimization, reducing the number of features passed to Ceres and therefore improving its runtime. Notably, the linearized and randomized methods achieve the lowest total runtime, with the linearized approach being the most efficient overall. The simple and low-rank greedy methods, while more computationally intensive during selection, still significantly reduce the cost of the optimization stage.

It is important to note that the runtimes reported here differ from those in Figs. 3 and 9, where the selection algorithms were evaluated intentionally on frames containing the maximum number of features (i.e., 150). In the QCar experiment, however, the actual number of candidate features per frame is often lower due to limited visibility and occlusion. Moreover, features selected in previous frames are directly carried over to the next frame if they remain visible. As a result, the number of new features that need to be selected is reduced, contributing to the lower selection times observed in this table.

In addition, the runtimes reported under the "Selection" column include the total computation required per frame, including generating the forward prediction horizon, performing visibility checks, constructing the information matrices, and applying the selection algorithms. In contrast, the runtimes shown in Figs. 3 and 9 reflect only the time required to apply the selection algorithms, assuming all required inputs are already available. The lower computation times in the table, especially for the linearized method, highlight its suitability for real-time, on-the-fly applications.

To further support this experiment, we conducted an additional analysis to provide insight into the per-frame runtime of two contrasting selection methods: the computationally intensive yet effective *simple greedy* method, and the efficient

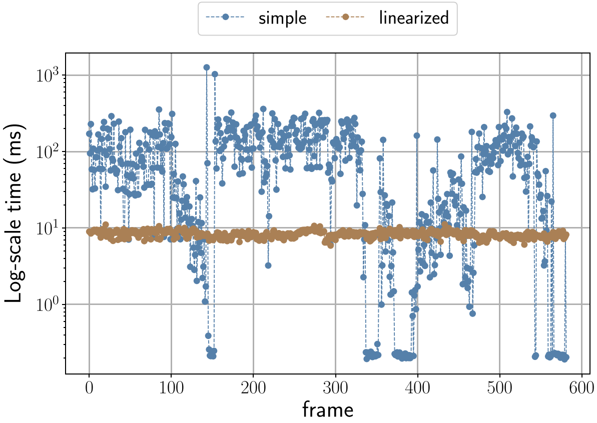


Fig. 13. Per-frame runtime comparison between the simple greedy and linearized selection methods in the QCar experiment with time horizon $T = 13$ and feature budget $\kappa = 70$. The linearized method exhibits consistently lower and more stable runtimes, highlighting its suitability for real-time applications.

and fast *linearized* method. Fig. 13 presents the per-frame runtime (vertical axis) of the two methods over a sequence of frames (horizontal axis) in the QCar experiment. As shown, the linearized method consistently achieves significantly lower runtime compared to the simple greedy method.

The observed variability in the runtime of the simple greedy method is expected. This variability arises due to fluctuations in the total number of visible features and, more importantly, the number of features that need to be newly selected at each frame, since some features are carried over from previous frames if they remain visible. These factors directly impact the selection time in greedy approaches. In contrast, the linearized method avoids iterative selection and instead computes a single score per feature based on the linear part of the objective, which is the main computational step, regardless of how many features are to be selected. It then selects the top-κ features, making its runtime largely insensitive to the number of features selected in each frame. These results highlight the suitability of the linearized method for real-time use. The experiment uses the same time horizon and feature budget as Table II, with $T = 13$ and $\kappa = 70$.

The QCar experiments demonstrate the feasibility of deploying our full anticipation-based feature selection pipeline in a control-aware setting. Leveraging control inputs and a motion prediction model enables forward visibility checks, which inform the selection process. Experimental results validate that increasing the feature budget consistently enhances state estimation accuracy, and runtime evaluations reveal practical tradeoffs among feature selection strategies. Overall, the proposed anticipation-based selection methods offer a promising mechanism for balancing feature usage, accuracy, and computational cost in visual-inertial estimation tasks, and may generalize to other multistep estimation frameworks.

V. Conclusion

We have presented and rigorously evaluated four practical algorithms for task-aware feature selection in VIN across two experimental settings. On the EuRoC MAV benchmark, chosen for its standardized indoor sequences and wide adoption for fair comparison, we showed that our fast low-rank greedy variant matches the classic greedy method in accuracy while reducing computation time, and that our linearization-based approach operates in real time with performance close to the greedy baseline. On our custom QCar platform, designed to capture control-aware navigation dynamics, we evaluated how each method adapts under realistic motion and sensing conditions, demonstrating reliable performance without relying on idealized assumptions.

Despite these strengths, our study has two notable limitations. First, the theoretical approximation bounds based on submodularity ratio and curvature, while providing worst-case guarantees, can be conservative and may not fully predict empirical performance in all scenarios. Second, the randomized greedy variant introduces sampling variability and may require careful tuning of the randomization parameter to balance consistency and computational load.

These results confirm that, depending on application constraints, one can choose *maximum accuracy* (fast low-rank greedy), *extreme speed* (linearization-based greedy), or *balanced performance* (randomized greedy), all with rigorous approximation guarantees. In particular, the real-time capability of the linearization-based method makes it especially well suited for onboard deployment on resource-constrained platforms without sacrificing estimation quality.

Our work underlines both the theoretical depth and the practical breadth of task-aware sensor selection, demonstrating its effectiveness from controlled benchmark scenarios to real-world robotic systems. Looking forward, several promising directions remain.

1) *Multiagent extensions:* Exploit cross-robot feature sharing and cooperative information gains in teams operating in dynamic environments.
2) *New sensor modalities:* Generalize the modular Taylor-approximation framework to LiDAR, depth, or event-camera measurements under richer noise and motion models.
3) *Adaptive budgeting:* Develop online schemes to adjust the selection budget in response to scene complexity and control objectives.
4) *Tighter theoretical bounds:* Incorporate higher-order approximations or stochastic batch selection to refine performance guarantees.
5) *Matrix-free strategies:* Investigate ultralightweight selection schemes that eliminate matrix operations entirely, enabling deployment on resource-constrained platforms requiring extreme-speed decisions.

Addressing these open questions will further enhance the applicability and impact of task-aware sensor selection across diverse robotic applications.

Appendix

A. Background and Definitions

This section presents background material and definitions that support the theoretical developments in the article.

Definition 5. (Closed function): Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be an extended-real-valued function, and define its (effective) domain by $\operatorname{dom} f = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$. The function f is said to be *closed* if, for every $\beta \in \mathbb{R}$, the sublevel set $\{x \in \operatorname{dom} f \mid f(x) \leq \beta\}$, is a closed subset of $\mathbb{R}^n$.

Remark 3: If f is continuous on a closed domain, then f is closed. When $\mathrm{dom}\, f$ is open, f is closed if and only if $f(x_k) \to +\infty$ for every sequence $\{x_k\}$ that approaches a boundary point of $\mathrm{dom}\, f$ [45].

B. Missing Proofs

This section presents the proofs that complement the theoretical findings discussed in the main text of the article.

1) Proof of Theorem 1: Recall that $\mathrm{Trace}\Omega_{\mathsf{S}}^{-1} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{1}{\lambda_{\mathrm{i}}(\Omega_{\mathsf{S}})}$, where $n = 9T + 9$. Let $\Omega_{\mathsf{U}} = \Omega_{\emptyset} + \sum_{l \in \mathsf{U}} \Delta_l$ represent the information matrix considering all extracted features at the given time frame.

In the first part of the proof, we establish a lower bound for the submodularity ratio γ. Referring to Definition 3, this can be achieved by finding a lower bound for the fraction

$$\frac{\sum_{l \in \mathsf{R} \setminus \mathsf{S}} f_l(\mathsf{S})}{f_{\mathsf{R}}(\mathsf{S})} = \frac{\sum_{l \in \mathsf{R} \setminus \mathsf{S}} [f(\mathsf{S} \cup \{l\}) - f(\mathsf{S})]}{f(\mathsf{R} \cup \mathsf{S}) - f(\mathsf{S})}. \tag{23}$$

For the numerator of this fraction, we have

$$\begin{aligned}
&\sum_{l \in \mathsf{R} \setminus \mathsf{S}} [f(\mathsf{S} \cup \{l\}) - f(\mathsf{S})] \\
&= \sum_{l \in \mathsf{R} \setminus \mathsf{S}} \left[\sum_{i=1}^{n} \frac{1}{\lambda_i(\Omega_{\mathsf{S}})} - \frac{1}{\lambda_i(\Omega_{\mathsf{S} \cup \{l\}})} \right] \\
&\geq \frac{\sum_{l \in \mathsf{R} \setminus \mathsf{S}} \sum_{i=1}^{n} \lambda_i(\Omega_{\mathsf{S} \cup \{l\}}) - \lambda_i(\Omega_{\mathsf{S}})}{\lambda_{\max}^2(\Omega_{\mathsf{U}})} \\
&= \frac{\sum_{l \in \mathsf{R} \setminus \mathsf{S}} \left[\mathrm{Trace}\, \Omega_{\mathsf{S} \cup \{\mathrm{l}\}} - \mathrm{Trace}\, \Omega_{\mathsf{S}} \right]}{\lambda_{\max}^2(\Omega_{\mathsf{U}})} \\
&= \lambda_{\max}^{-2}(\Omega_{\mathsf{U}}) \sum_{l \in \mathsf{R} \setminus \mathsf{S}} \mathrm{Trace}\, \Delta_{\mathrm{l}} \geq \delta \cdot |\mathsf{R} \setminus \mathsf{S}| \cdot \lambda_{\max}^{-2}(\Omega_{\mathsf{U}})
\end{aligned} \tag{24}$$

where $\delta \triangleq \min_{j \in \mathsf{U}} \mathrm{Trace}\Delta_{\mathrm{j}}$. Analogously, for the denominator, we have

$$\begin{aligned}
f(\mathsf{R} \cup \mathsf{S}) - f(\mathsf{S}) &= \sum_{i=1}^{n} \frac{1}{\lambda_i(\Omega_{\mathsf{S}})} - \sum_{j=1}^{n} \frac{1}{\lambda_j(\Omega_{\mathsf{S} \cup \mathsf{R}})} \\
&\overset{\text{(i)}}{\leq} \sum_{i=n-|\mathsf{R} \setminus \mathsf{S}|+1}^{n} \frac{1}{\lambda_i(\Omega_{\mathsf{S}})} - \sum_{j=1}^{|\mathsf{R} \setminus \mathsf{S}|} \frac{1}{\lambda_j(\Omega_{\mathsf{S} \cup \mathsf{R}})} \\
&\leq |\mathsf{R} \setminus \mathsf{S}| \left(\frac{1}{\lambda_{\min}(\Omega_{\emptyset})} - \frac{1}{\lambda_{\max}(\Omega_{\mathsf{U}})} \right) \\
&= |\mathsf{R} \setminus \mathsf{S}| \frac{\lambda_{\max}(\Omega_{\mathsf{U}}) - \lambda_{\min}(\Omega_{\emptyset})}{\lambda_{\min}(\Omega_{\emptyset}) \cdot \lambda_{\max}(\Omega_{\mathsf{U}})}
\end{aligned} \tag{25}$$

where inequality (i) holds due to the interlacing inequality of eigenvalues. The combination of (24) and (25) provides the first part of the proof as

$$\gamma \geq \frac{\delta \cdot \lambda_{\min}(\Omega_{\emptyset})}{\lambda_{\max}(\Omega_{\mathsf{U}}) \cdot (\lambda_{\max}(\Omega_{\mathsf{U}}) - \lambda_{\min}(\Omega_{\emptyset}))} = \underline{\gamma}.$$

In the final part of the proof, we obtain the bound for the curvature. To do this, we determine the bound for $1 - \alpha$. Referring to (9), this task is equivalent to establishing a lower bound for

$$\frac{f(\mathsf{S} \cup \mathsf{R}) - f(\mathsf{S} \setminus \{l\} \cup \mathsf{R})}{f(\mathsf{S}) - f(\mathsf{S} \setminus \{l\})}. \tag{26}$$

For the numerator of this fraction, we have

$$\begin{aligned}
&f(\mathsf{S} \cup \mathsf{R}) - f(\mathsf{S} \setminus \{l\} \cup \mathsf{R}) \\
&= \sum_{i=1}^{n} \frac{1}{\lambda_i(\Omega_{\mathsf{S} \setminus \{l\} \cup \mathsf{R}})} - \sum_{j=1}^{n} \frac{1}{\lambda_j(\Omega_{\mathsf{S} \cup \mathsf{R}})} \geq \delta\, \lambda_{\max}^{-2}(\Omega_{\mathsf{U}})
\end{aligned} \tag{27}$$

where we use a similar derivation as in (24). Let us now consider the denominator

$$f(\mathsf{S}) - f(\mathsf{S} \setminus \{l\}) = \sum_{i=1}^{n} \frac{1}{\lambda_i(\Omega_{\mathsf{S} \setminus \{l\}})} - \sum_{j=1}^{n} \frac{1}{\lambda_j(\Omega_{\mathsf{S}})}.$$

This can be upper-bounded using the *Cauchy interlacing inequality* as follows:

$$\frac{1}{\lambda_{\min}(\Omega_{\mathsf{S} \setminus \{l\}})} - \frac{1}{\lambda_{\max}(\Omega_{\mathsf{S}})} \leq \frac{\lambda_{\max}(\Omega_{\mathsf{U}}) - \lambda_{\min}(\Omega_{\emptyset})}{\lambda_{\min}(\Omega_{\emptyset}) \cdot \lambda_{\max}(\Omega_{\mathsf{U}})}. \tag{28}$$

Combining (27) and (28) yields a lower bound on $1 - \alpha$

$$\begin{aligned}
1 - \alpha &= \frac{f(\mathsf{S} \cup \mathsf{R}) - f(\mathsf{S} \setminus \{l\} \cup \mathsf{R})}{f(\mathsf{S}) - f(\mathsf{S} \setminus \{l\})} \\
&\geq \frac{\delta\, \lambda_{\min}(\Omega_{\emptyset})}{\lambda_{\max}(\Omega_{\mathsf{U}}) \left(\lambda_{\max}(\Omega_{\mathsf{U}}) - \lambda_{\min}(\Omega_{\emptyset})\right)}.
\end{aligned}$$

Consequently

$$\alpha \;\leq\; \overline{\alpha} \;=\; 1 - \frac{\delta\, \lambda_{\min}(\Omega_{\emptyset})}{\lambda_{\max}(\Omega_{\mathsf{U}}) \left(\lambda_{\max}(\Omega_{\mathsf{U}}) - \lambda_{\min}(\Omega_{\emptyset})\right)} \tag{29}$$

which completes the proof.

2) Proof of Proposition 2: Define $\phi(\lambda) = \sum_{i=1}^{n} 1/\lambda_i$ for all $\lambda \in \mathbb{R}_{>0}^{n}$. The function ϕ is continuous, symmetric, and convex. Its domain is open, and $\phi(\lambda) \to +\infty$ as $\min_i \lambda_i \to 0^+$; by Remark 3, ϕ is therefore closed. Because ϕ is smooth on $\mathbb{R}_{>0}^{n}$, it is differentiable at every $\lambda \in \mathbb{R}_{>0}^{n}$. Lemma 3 then implies that $\rho = \phi \circ \lambda$ is differentiable at every $\Omega \in \mathbb{S}_{++}^{n}$.

3) Proof of Proposition 3: Set $A \triangleq \Omega_{\emptyset}$, and $B \triangleq \Delta_{\mathsf{S}}$, so that $A \succ 0$ and $B \succeq 0$. Under the assumption $\epsilon \, \|A^{-1}B\|_2 < 1$, we expand $A + \epsilon\, B = A\big(I + \epsilon\, A^{-1}B\big)$, and apply the Neumann-series to $(I + \epsilon\, A^{-1}B)^{-1}$

$$\big(I + \epsilon\, A^{-1}B\big)^{-1} \;=\; I \;-\; \epsilon\, A^{-1}B \;+\; \epsilon^2 (A^{-1}B)^2 \;+\; \mathcal{O}(\epsilon^3).$$

Multiplying on the right by A^{-1} gives

$$\begin{aligned}
(A + \epsilon\, B)^{-1} \;&=\; A^{-1} \;-\; \epsilon\, A^{-1} B\, A^{-1} \\
&\quad+\; \epsilon^2\, A^{-1} B\, A^{-1} B\, A^{-1} \;+\; \mathcal{O}(\epsilon^3).
\end{aligned}$$

Taking the trace yields

$$\begin{aligned}
\mathrm{Trace}\big((\mathrm{A} + \epsilon\, \mathrm{B})^{-1}\big) \;&=\; \mathrm{Trace}(\mathrm{A}^{-1}) \;-\; \epsilon\, \mathrm{Trace}(\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}) \\
&\quad+\; \epsilon^2\, \mathrm{Trace}\big(\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\big) \;+\; \mathcal{O}(\epsilon^3).
\end{aligned}$$

Since $\rho(X) = \mathrm{Trace}(\mathrm{X}^{-1})$, we obtain

$$\begin{aligned}
\rho(A + \epsilon\, B) \;&=\; \rho(A) \;-\; \epsilon\, \mathrm{Trace}\big(\mathrm{A}^{-2}\mathrm{B}\big) \\
&\quad+\; \epsilon^2\, \mathrm{Trace}\big(\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\big) \;+\; \mathcal{O}(\epsilon^3).
\end{aligned}$$

Notice that

$$\mathrm{Trace}\big(\mathrm{A}^{-2}\mathrm{B}\big) = \mathrm{Trace}\Big(\Omega_{\emptyset}^{-2}\sum_{\mathrm{l}\in\mathsf{S}}\Delta_{\mathrm{l}}\Big) = \sum_{\mathrm{l}\in\mathsf{S}}\mathrm{Trace}\big(\Omega_{\emptyset}^{-2}\Delta_{\mathrm{l}}\big)$$

$$= \sum_{l\in\mathsf{S}} r_l\big(\Omega_{\emptyset}^{2}\big).$$

This proves (20).

To bound the second-order term, define $M \triangleq A^{-1/2}\,B\,A^{-1/2}$, so that $M \succeq 0$ and $B = A^{1/2}\,M\,A^{1/2}$. Then

$$\mathrm{Trace}\big(\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\big)$$

$$= \mathrm{Trace}\big(\mathrm{A}^{-1/2}\,(\mathrm{A}^{-1/2}\mathrm{B}\mathrm{A}^{-1/2})\,(\mathrm{A}^{-1/2}\mathrm{B}\mathrm{A}^{-1/2})\,\mathrm{A}^{-1/2}\big)$$

$$= \mathrm{Trace}\big(\mathrm{M}^2\,\mathrm{A}^{-1}\big).$$

Because $A^{-1} \succeq 0$, we have

$$\mathrm{Trace}\big(\mathrm{M}^2\mathrm{A}^{-1}\big) \;\leq\; \|\mathrm{A}^{-1}\|_2\,\mathrm{Trace}\big(\mathrm{M}^2\big) \;=\; \|\mathrm{A}^{-1}\|_2\,\|\mathrm{M}\|_{\mathrm{F}}^2.$$

Moreover

$$\|M\|_F = \big\|A^{-1/2}\,B\,A^{-1/2}\big\|_F$$

$$\leq \;\big\|A^{-1/2}\big\|_2^2\,\|B\|_F \;=\; \|A^{-1}\|_2\,\|\Delta_{\mathsf{S}}\|_F.$$

Combining gives

$$\mathrm{Trace}\big(\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}\big) \;\leq\; \|A^{-1}\|_2\,\big(\|A^{-1}\|_2\,\|\Delta_{\mathsf{S}}\|_F\big)^2$$

$$=\; \|A^{-1}\|_2^3\,\|\Delta_{\mathsf{S}}\|_F^2.$$

Since $\|\Delta_{\mathsf{S}}\|_F \leq \sum_{l\in\mathsf{S}}\|\Delta_l\|_F \leq \kappa\,\zeta$, we obtain the bound in (21). This completes the proof.

Acknowledgment

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the DEVCOM Analysis Center, the U.S. Department of Homeland Security, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] G.-Z. Yang et al., "The grand challenges of science robotics," *Sci. Robot.*, vol. 3, no. 14, 2018, Art. no. eaar7650.

[2] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 334–349, Apr. 2006.

[3] R. Lerner, E. Rivlin, and I. Shimshoni, "Landmark selection for task-oriented navigation," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 494–505, Jun. 2007.

[4] B. Mu, L. Paull, A.-A. Agha-Mohammadi, J. J. Leonard, and J. P. How, "Two-stage focused inference for resource-constrained minimal collision navigation," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 124–140, Feb. 2017.

[5] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 1–20, Feb. 2019.

[6] H. K. Mousavi and N. Motee, "Estimation with fast feature selection in robot visual navigation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3572–3579, Apr. 2020.

[7] V. Pandey et al., "Scalable networked feature selection with randomized algorithm for robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 5704–5709.

[8] A. J. Davison, "Active search for real-time vision," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 66–73.

[9] H. Strasdat, C. Stachniss, and W. Burgard, "Which landmark is useful? learning selection policies for navigation in unknown environments," in *Proc. 2009 IEEE Int. Conf. Robot. Autom.*, 2009, pp. 1410–1415.

[10] M. Siami and N. Motee, "Growing linear dynamical networks endowed by spectral systemic performance measures," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 2091–2106, Jul. 2018.

[11] M. Siami and N. Motee, "Network abstraction with guaranteed performance bounds," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3301–3316, Oct. 2018.

[12] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1686–1701, Apr. 2021.

[13] M. Siami and A. Jadbabaie, "A separation theorem for joint sensor and actuator scheduling with guaranteed performance bounds," *Automatica*, vol. 119, 2020, Art. no. 109054.

[14] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, pp. 265–294, 1978.

[15] A. Badanidiyuru and J. Vondrák, "Fast algorithms for maximizing submodular functions," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2014, pp. 1497–1514.

[16] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 1812–1818.

[17] N. Buchbinder and M. Feldman, "Deterministic algorithm and faster algorithm for submodular maximization subject to a matroid constraint," in *Proc. IEEE 65th Annu. Symp. Found. Comput. Sci.*, 2024, pp. 700–712.

[18] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1057–1064.

[19] T. Summers and M. Kamgarpour, "Performance guarantees for greedy maximization of non-submodular controllability metrics," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 2796–2801.

[20] A. Hashemi, M. Ghasemi, H. Vikalo, and U. Topcu, "A randomized greedy algorithm for near-optimal sensor scheduling in large-scale sensor networks," in *Proc. Amer. Control Conf.*, 2018, pp. 1027–1032.

[21] Y. Zhao and P. A. Vela, "Good feature matching: Toward accurate, robust VO/VSLAM with low latency," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 657–675, Jun. 2020.

[22] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based localization: Observability analysis and consistency improvement," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 182–201, 2014.

[23] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[24] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *J. Field Robot.*, vol. 27, no. 5, pp. 587–608, 2010.

[25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proc. Robot.: Sci. Syst.*, Rome, Italy, 2015.

[26] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[27] C. Campos, R. Elvira, J. J. G. Rodrıǵuez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[28] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 498–507.

[29] A. Hashemi, M. Ghasemi, H. Vikalo, and U. Topcu, "Randomized greedy sensor selection: Leveraging weak submodularity," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 199–212, Jan. 2021.

[30] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[31] Q. Inc., "Qcar: Sensor-rich autonomous vehicle for self-driving applications," 2024. [Online]. Available: https://www.quanser.com/products/qcar/#details

[32] A. Olshevsky, "On (non) supermodularity of average control energy," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1177–1181, Sep. 2018.

[33] R. Vafaee and M. Siami, "Learning-based sparse sensing with performance guarantees," *IEEE Trans. Autom. Control*, vol. 70, no. 1, pp. 387–402, Jan. 2025.

[34] R. Vafaee and M. Siami, "Exploring non-submodular scheduling for large-scale sensor networks," *IEEE Contr. Syst. Lett.*, vol. 7, pp. 3968–3973, 2023.

[35] L. F. O. Chamon, G. J. Pappas, and A. Ribeiro, "Approximate supermodularity of Kalman filter sensor selection," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 49–63, Jan. 2021.

[36] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert, "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 4290–4297.

[37] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[38] J. Borwein and A. Lewis, *Convex Analysis*. Berlin, Germany: Springer, 2006.

[39] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[40] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[41] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, vol. 2, pp. 674–679.

[42] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for optimal filtering of linear dynamical systems: Complexity and approximation," in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 5002–5007.

[43] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.

[44] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1182–1189, Apr. 2022.

[45] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Reza Vafaee (Member, IEEE) received the second master's degree in pure and applied mathematics (with highest distinction) from Montclair State University, Montclair, NJ, USA, in 2020, and the Ph.D. degree in electrical engineering from Northeastern University, Boston, MA, USA, in 2024.

He is currently a Postdoctoral Fellow with the Department of Computer Science at Boston College, Chestnut Hill, MA, USA. His research interests include the mathematical and algorithmic foundations of network optimization and control with applications in robotics and multiagent systems, estimation theory, and spectral graph theory.

Kian Behzad (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the Amirkabir University of Technology, Tehran, Iran. He is currently working toward the Ph.D. degree in electrical engineering with Northeastern University (NEU), Boston, MA, USA.

He is a Research Assistant with the Siami Lab., NEU. He is conducting research focusing on robotics, Wi-Fi sensing, and control systems, with applications in autonomous systems and wireless sensing. His research interests include visual–inertial navigation, optimization, and active sensing.

Milad Siami (Senior Member, IEEE) received the B.Sc. degrees in electrical engineering and pure mathematics, and the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2009 and 2011, respectively, and the M.Sc. and Ph.D. degrees in mechanical engineering from Lehigh University, Bethlehem, PA, USA, in 2014 and 2017, respectively.

He was a Postdoctoral Associate with the Massachusetts Institute of Technology, Cambridge, MA, USA, from 2017 to 2019. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. His research interests include distributed control, distributed optimization, and sparse sensing in robotics and cyber–physical systems.

Luca Carlone (Senior Member, IEEE) received the Ph.D. degree in robotics from the Polytechnic University of Turin, Turin, Italy, in 2012.

He is currently an Associate Professor with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, where he is also a Principal Investigator with the Laboratory for Information and Decision Systems (LIDS). His research interests include estimation, optimization, and learning, with applications to robot perception and decision-making.

Dr. Carlone has received several honors, including the NSF CAREER Award, the RSS Early Career Award, a Sloan Research Fellowship, and multiple Best Paper Awards at leading robotics conferences.

Ali Jadbabaie (Fellow, IEEE) received the Ph.D. degree in control and dynamical systems from the California Institute of Technology, Pasadena, CA, USA, in 2001.

He is currently the JR East Professor and Head of the Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, where he is also a core faculty member with the Institute for Data, Systems, and Society and a Principal Investigator with the Laboratory for Information and Decision Systems (LIDS). His research interests include networked dynamical systems, distributed optimization, and multiagent coordination and control.

Dr. Jadbabaie was the recipient of the Vannevar Bush Faculty Fellowship, NSF CAREER Award, and the George S. Axelby Best Paper Award.