**IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France**

Estimation with Fast Feature Selection in Robot Visual Navigation

Hossein K. Mousavi and Nader Motee[1]

Abstract— **We consider the robot localization problem with sparse visual feature selection. The underlying key property is that contributions of trackable features (landmarks) appear linearly in the information matrix of the corresponding estimation problem. We utilize standard models for motion and vision system using a camera to formulate the feature selection problem over moving finite-time horizons. We propose a scalable randomized sampling algorithm to select more informative features to obtain a certain estimation quality. We provide probabilistic performance guarantees for our method. The time-complexity of our feature selection algorithm is linear in the number of candidate features, which is practically plausible and outperforms existing greedy methods that scale quadratically with the number of candidate features. Our numerical simulations confirm that not only the execution time of our proposed method is comparably less than that of the greedy method, but also the resulting estimation quality is very close to the greedy method.**

I. Introduction

Safe Safe and robust navigation in uncertain environments is one of the fundamental problems in robotics. The recent technological advances in the computing devices have opened up new opportunities and made several breakthroughs possible in this research area [1], where estimation and planning problems can be solved close to real-time in some applications. However, robot navigation in rapidly changing environments still suffers from computational complexities. Even if one uses high-performance computational units, the demand for agility and higher levels of autonomy always mandates us to execute onboard procedures in shorter periods.

One of the essential subproblems during robot navigation is to solve the localization, mapping, and visual odometry at an acceptable level of accuracy while spending a minimal amount of computational resources [2]. To achieve this goal, many researchers have investigated visual feature selection problems [3]–[10]. The underlying idea is that depending on the current state of the robot and planned motion in the near future (i.e., the task), tracking certain features across a time horizon can be more informative than tracking other features. In other words, certain visual features may deserve more attention compared to the rest. In this regard, [11] uses a greedy method to select a subset of pre-identified visual landmarks, which facilitate the pose estimation of the robot. In [7], the authors combine solving the simultaneous localization and mapping (SLAM) using the unscented Kalman filtering with reinforcement learning. Their approach generates policies that govern the feature selection. In [10], a two-stage methodology for measurement planning is discussed. The first stage is the selection of the subset of landmarks for observation, which is followed by the design of observation times for each feature. In [5], the authors consider the task-aware design of a subset of features such that an uncertainty metric is minimized. In [12], the authors consider a visual-inertial navigation problem and analyze the problem of feature-selection, where the design variable is the features that will be tracked during a fixed time-horizon. They use convex relaxations as well as the greedy method for feature selection and quantify performance guarantees for the quality of the resulting estimations.

In this paper, we propose a method to reduce the time-complexity of the feature selection subproblem during the navigation. The navigation setup consists of a robot that moves based on generated position estimates. The robot is assumed to use an onboard camera to (passively) track selected features over a fixed time horizon to improve the quality of the estimation. Our utilized model for vision system is similar to that of [12], while instead of using the greedy method and convex relaxations, we propose a randomized sampling algorithm for feature selection. In our approach, a sampling probability (a number between 0 and 1) is assigned to each available feature, where our randomized algorithm interprets these numbers as a measure of informativeness during sampling process (a feature is more informative if its sampling probability is closer to 1). Several theoretical guarantees on the quality of the estimation derived. It turns out that the time-complexity of our randomized sampling algorithm scales linearly with the number of available features, while time complexity of the greedy method of [12] scales quadratically for the same problem. It should be emphasized that our algorithm is more suitable for problems with hundreds or thousands of features, where greedy methods become practically inefficient. Numerical simulations confirm that the estimation quality using features provided by our randomized sampling is very close to the quality of estimation provided by the greedy method, while the required time to run our randomized sampling algorithm is significantly less than the greedy method.

After stating the problem in Section II, we discuss the details of the motion and vision models in Section III. Then, three estimation measures are introduced to quantify

TThis paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in parts by the NSF CAREER ECCS-1454022, and ONR N00014-19-1-2478, AFOSR FA9550-19-1-0004.

[1] The Authors are with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015, USA {mousavi, motee}@lehigh.edu

the quality of estimation based on the selected features. In Section V, we propose a randomized algorithm for feature selection and conduct performance and time-complexity analysis for our approach. In Section VI, a numerical case is explained, wherein we compare the performance of our (weighted) randomized feature selection method against: (i) a randomized method that selected features uniformly, and (ii) the greedy method. The proofs of all theoretical results can be found in [13]. This paper has not been submitted to any other conference or journal.

Notations: The sets of nonnegative integer and real numbers are denoted by $\mathbb{Z}_+$ and $\mathbb{R}_+$, respectively. The vectors and matrices are denoted by lower-case and upper-case letters, respectively (e.g. x and X). The identity matrix of size n is denoted by I_n. The set of positive definite matrices of size n is denoted by $\mathcal{S}^n_{++}$. The partial ordering on the cone of positive-semidefinite matrices is denoted by $\succ$, $\succcurlyeq$, $\prec$, and $\preccurlyeq$ operators. The block-diagonal matrix with diagonal elements $X_1, \dots, X_N$ is denoted by $\mathrm{diag}(X_1, \dots, X_N)$. For a set S, $|S|$ denotes its cardinality. For a map $\mathbf{g}$, $\nabla_x \mathbf{g}$ denotes the corresponding partial derivative. A Gaussian random variable with mean vector μ and covariance matrix Σ is denoted by $\mathcal{N}(\mu, \Sigma)$. $X \otimes Y$ denotes the Kronecker product of matrices X and Y. The special orthogonal group in $\mathbb{R}^3$ is denoted by $\mathrm{SO}(3)$.

II. Feature Selection Problem

Let us denote spatial location of a robot at time $t \in \mathbb{Z}_+$ by $x_t \in \mathbb{R}^3$. For a given positive integer T, the vector of future states over the discrete time horizon $[t, t+T] = t, t+1, \dots, t+T$ is represented by

$$\mathbf{x}_{t,T} := \Big[x_t^T,\ x_{t+1}^T,\ \dots\ ,\ x_{t+T}^T \Big]^T \in \mathbb{R}^{3(T+1)}.$$

Since robot motion creates uncertainty, having access to the statistics of $\mathbf{x}_{t,T}$ will help us measure quality of our prediction of robot whereabouts over the time horizon [12]. As it is shown in Subsection III-A, one can obtain mean vector $\bar{\boldsymbol{\mu}}_{t,T} \in \mathbb{R}^{3(T+1)}$ and covariance matrix $\bar{\mathbf{\Sigma}}_{t,T} \in \mathcal{S}^{3(T+1)}_{++}$ of $\mathbf{x}_{t,T}$ under popular Gaussianity assumption. These quantities can be equivalently transformed into more relevant forms for the feature selection problem, namely, information vector and matrix, which are given by [2]

$$\bar{\mathbf{b}}_{t,T} = \bar{\boldsymbol{\mu}}_{t,T}^T \, \bar{\mathbf{\Sigma}}_{t,T}^{-1} \tag{1}$$

$$\bar{\mathbf{H}}_{t,T} = \bar{\mathbf{\Sigma}}_{t,T}^{-1}. \tag{2}$$

There is a one-to-one correspondence between mean vector and covariance matrix and their counterparts information vector and information matrix. A striking property of the latter representation is that the contribution of each feature (or landmark) to the information vector and matrix is fused linearly [14]. Having the prior estimation parameters (1)-(2), as it is shown in Subsection III-B, the quality of estimation for $\mathbf{x}_{t,T}$ can be improved by fusing information of newly observed visual features using an onboard camera. The updated information matrix and vector are

$$\mathbf{H}_{t,T}(\Theta_t) = \bar{\mathbf{H}}_{t,T} + \sum_{f \in \Theta_t} \mathbf{H}^f_{t,T} \tag{3}$$

$$\mathbf{b}_{t,T}(\Theta_t) = \bar{\mathbf{b}}_{t,T} + \sum_{f \in \Theta_t} \mathbf{b}^f_{t,T} \tag{4}$$

in which $\mathbf{b}^f_{t,T}$ and $\mathbf{H}^f_{t,T}$ are contributions of feature f to the overall information matrices of the estimation problem. The set of all identifiable features (landmarks) at time t, which can be triangulated using multiple frames over the time horizon $[t, t+T]$, is denoted by Θ_t. Suppose that $|\Theta_t| = N_t$ is assumed to be large.

Tracking a large number of features (landmarks) for accurate navigation usually requires substantial onboard computational power [4]. As a result, a desirable navigation objective is to select and track a small subset of more informative features, while providing an acceptable estimation quality. Suppose that robot is only capable of tracking at most q, which is comparably less than N_t, features during the horizon.

Definition 1: A map $\rho : \mathcal{S}^n_{++} \to \mathbb{R}$ is called monotone decreasing if $X \preceq Y$ implies $\rho(X) \geq \rho(Y)$.

Then, the feature selection problem can be formulated as

$$\underset{\Phi_t \subset \Theta_t}{\text{minimize}} \quad \rho\big(\mathbf{H}_{t,T}(\Phi_t)\big) \tag{5}$$

$$\text{subject to}: \quad |\Phi_t| \leq q \tag{6}$$

where $\rho : \mathcal{S}^{3(T+1)}_{++} \to \mathbb{R}$ is a monotone decreasing map that measures the estimation quality.

The optimization problem (5)-(6) is combinatorial and usually NP-hard. The *research problem* is to propose a scalable algorithm that provides solutions for (5)-(6) with performance guarantees.

III. Models for Robot Motion and Vision System

To calculate faithful estimates for the robot's position, one needs to properly fuse estimates resulting from models of robot motion with the ones resulting from the camera models. The contents of this Section are similar in spirit with a similar work [15][1].

A. Statistics of Robot Position

The goal is to calculate information vector and matrix of $\mathbf{x}_{t,T}$ when dynamics of robot evolves over time horizon $[t, t+T]$. To achieve this, we utilize a model that is inspired by the dynamic model analyzed in [14]. Suppose that dynamics of the robot's position is governed by

$$x_\tau = \mathbf{g}(x_{\tau-1}, u_\tau) + \delta_\tau \tag{7}$$

for all $\tau \in [t+1, t+T]$, where $\mathbf{g} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ is a (possibly nonlinear) known map, where u_τ in the control

[1] This was brought to our attention by an anonymous reviewer.

command at time τ, and $\delta_\tau \sim \mathcal{N}(0, \Lambda_\tau)$ is a temporally independent random process that captures the aggregate effect of all uncertainties induced by robot motion. We assume that a feedback control law with the following structure is given

$$u_\tau = \mathbf{h}(u_\tau^{\text{ref}}, x_{\tau-1}) \tag{8}$$

that ensures the robot with dynamics (7) tracks a reference path, at least in the absence of uncertainties, with some desired accuracy. The command u_τ^{ref} may have to be prefiltered to enhance the tracking performance.

Let us represent the actual robot's position by $x_{\tau-1}$, which is a random variable and its true value is unknown. We use its mean value $\bar{\mu}_{\tau-1}$ as a faithful estimate of its value in (8) to obtain

$$\bar{u}_\tau = \mathbf{h}\left(u_\tau^{\text{ref}}, \bar{\mu}_{\tau-1}\right). \tag{9}$$

In presence of uncertainties, the trajectory of the closed-loop system (7)-(9) will fluctuate around the reference path and the tracking quality will depend on the quality of estimation $\bar{\mu}_{\tau-1}$. The control mechanism (9) is merely using the initial statistics of position of the robot. In the next subsection, we show that incorporating new information obtained from observing features (landmarks) will help us improve the estimation quality, which in turn will improve the path tracking quality.

Suppose that the current, i.e., before accounting for the dynamics of the robot, pose estimates for x_t is described by mean vector μ_t and covariance matrix Σ_t. For the time step starting at $\tau = t$ in the horizon, let us set $\bar{\mu}_t = \mu_t$ and $\bar{\Sigma}_t = \Sigma_t$. For the next steps, we define the composed map

$$\mathbf{f}(x, \mu, u) := \mathbf{g}\big(x, \mathbf{h}(u, \mu)\big). \tag{10}$$

Then, upon linearizing the dynamics of the system at working point $(x, \mu, u) = (\bar{\mu}_{\tau-1}, \bar{\mu}_\tau, u_\tau^{\text{ref}})$ with respect to x, we get

$$x_\tau \;\approx\; \bar{\Delta}_\tau + A_\tau \left(x_{\tau-1} - \bar{\mu}_{\tau-1}\right) + \delta_\tau \tag{11}$$

in which vector $\bar{\Delta}_\tau$ and A_τ are given by

$$\begin{aligned} \bar{\Delta}_\tau &:= \mathbf{g}\big(\bar{\mu}_{\tau-1}, \mathbf{h}(u_\tau^{\text{ref}}, \bar{\mu}_{\tau-1})\big) && (12) \\ A_\tau &:= \nabla_x \mathbf{f}(\bar{\mu}_{\tau-1}, \bar{\mu}_{\tau-1}, u_\tau^{\text{ref}}) && (13) \end{aligned}$$

for all $\tau \in [t+1, t+T]$.

Lemma 1: By setting $\bar{\mu}_t = \mu_t$ and $\bar{\Sigma}_t = \Sigma_t$, the mean and covariance of $\mathbf{x}_{t,T}$ is given by

$$\bar{\boldsymbol{\Sigma}}_{t,T} = \begin{bmatrix} \bar{\Sigma}_t & \bar{\Sigma}_{t,t+1} & \cdots & \bar{\Sigma}_{t,t+T} \\ \bar{\Sigma}_{t,t+1}^T & \bar{\Sigma}_{t+1} & \cdots & \bar{\Sigma}_{t+1,t+T} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\Sigma}_{t,t+T}^T & \bar{\Sigma}_{t+1,t+T}^T & \cdots & \bar{\Sigma}_{t+T} \end{bmatrix} \tag{14}$$

$$\bar{\boldsymbol{\mu}}_{t,T} = \begin{bmatrix} \bar{\mu}_t^T & \bar{\mu}_{t+1}^T & \cdots & \bar{\mu}_{t+T}^T \end{bmatrix}^T \tag{15}$$

where

$$\begin{aligned} \bar{\Sigma}_\tau &= A_\tau \bar{\Sigma}_{\tau-1} A_\tau^T + \Lambda_\tau \\ \bar{\mu}_\tau &= \bar{\Delta}_\tau \end{aligned}$$

for every instant $\tau \in [t+1, t+T]$ and

$$\bar{\Sigma}_{\tau_1,\tau_2} = \left(\prod_{i=1}^{\tau_2-\tau_1} A_{\tau_2-i-1} \right) \bar{\Sigma}_{\tau_1}$$

for all $\tau_1, \tau_2 \in [t, t+T]$ with $\tau_1 < \tau_2$.

We can substitute (14) and (15) into (1) and (2) to calculate information vector $\bar{\mathbf{b}}_{t,T}$ and matrix $\bar{\mathbf{H}}_{t,T}$. In the next subsection, it is shown that these vectors and matrices will be updated upon receipt of certain information about the observed features over the time horizon $[t, t+T]$.

B. Camera Model for Feature Tracking and Estimation

We employ the observation model proposed by in [12] for an onboard camera. For every $\tau \in [t, t+T]$, let us denote orientation of the robot by rotation matrix $R_\tau \in \mathrm{SO}(3)$, orientation of the camera with respect to the robot by rotation matrix $R_{\mathrm{c}} \in \mathrm{SO}(3)$, translation of the camera with respect to the robot pose by $x_{\mathrm{c}} \in \mathbb{R}^3$, the unit vector corresponding to pixel measurement of feature $f \in \Theta_t$ at time τ by $u_{\tau,T}^f \in \mathbb{R}^3$, and the position vector of the feature by $y_f \in \mathbb{R}^3$. We recall that the corresponding skew-symmetric matrix induced by $u_{\tau,T}^f$ satisfies

$$U_{\tau,T}^f v = u_{\tau,T}^f \times v$$

for every vector $v \in \mathbb{R}^3$. As it is discussed in [12], one may reasonably assume that the observation vector in the image and its counterpart in real world are collinear (i.e., parallel). However, due to existence of noise in the process, one may consider a disrupted version of this assumption by considering the following noisy observation model

$$U_{\tau,T}^f \left((R_\tau R_{\mathrm{c}})^T \left(y_f - (x_\tau + R_\tau x_{\mathrm{c}}) \right) \right) = \eta_{\tau,T}^f, \tag{16}$$

where $\eta_{\tau,T}^f \sim \mathcal{N}(0, \sigma^2 I_3)$. The observation model (16) can be rewritten as

$$z_{\tau,T}^f = U_{\tau,T}^f \left(R_\tau R_{\mathrm{c}}\right)^T \left(x_\tau - y_f\right) + \eta_{\tau,T}^f \tag{17}$$

with $z_{f,\tau} := -U_{f,\tau} R_c^T x_c = U_{f,\tau}^T R_c^T x_c$. The camera takes one frame at every time instant over time horizon $[t, t+T]$. With knowledge of planned motion (i.e., location and orientation) for robot over the time horizon, suppose that robot is capable of running forward simulations to determine a feature will be visible in n_f frames out of all $T+1$ frames over the time horizon. By considering relation (17) for such visible features, one can stack all these equations and write them in more compact form

$$\mathbf{z}_{t,T}^f = \mathbf{F}_{t,T}^f \mathbf{x}_{t,T} + \mathbf{E}_{t,T}^f y_f + \boldsymbol{\eta}_{t,T}^f, \tag{18}$$

for some appropriate matrices $\mathbf{F}_{t,T}^f$ and $\mathbf{E}_{t,T}^f$. A given apriori information matrix $\bar{\mathbf{H}}_{t,T}$, which is obtained from (14), can

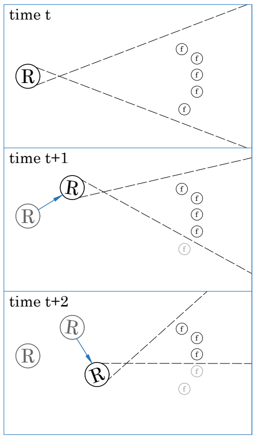

Fig. 1: The schematic of the motion and vision model adopted in this paper at three consecutive snapshots. The location of the robot is denoted by letter R, which moves and changes its orientation across three frames. The features are denoted by the letter f. As a result of this movement, the visible features (those in between the dashed lines) in each frame will vary. Set Θ_t will consist of all features that can be triangulated during this horizon.

be updated by fusing information of a visible feature $\{f\}$ according to the following rule [12]

$$\mathbf{H}_{t,T}(\{f\}) = \bar{\mathbf{H}}_{t,T} + \mathbf{H}^f_{t,T}, \tag{19}$$

where the linearly added information matrix is given by

$$\mathbf{H}^f_{t,T} = \sigma^{-2}\Big(\left(\mathbf{F}^f_{t,T}\right)^T \mathbf{F}^f_{t,T} - \left(\mathbf{F}^f_{t,T}\right)^T \mathbf{E}^f_{t,T}\left(\left(\mathbf{E}^f_{t,T}\right)^T \mathbf{E}^f_{t,T}\right)^{-1}\left(\mathbf{E}^f_{t,T}\right)^T \mathbf{F}^f_{t,T}\Big).$$

This additive property of the information matrix can be verified by application of the Bayes law [14] together with the Schur complement [12]. A similar treatment allows us to derive the following update rule for the information vector.

Proposition 1: The information vector of $\mathbf{x}_{t,T}$ upon tracking feature $f \in \Theta_t$ is updated according to

$$\mathbf{b}_{t,T}(\{f\}) = \bar{\mathbf{b}}_{t,T} + \left(\mathbf{B}^f_{t,T}\,\mathbf{z}^f_{t,T}\right)^T, \tag{20}$$

where matrix $\mathbf{B}^f_{t,T}$ is given by

$$\mathbf{B}^f_{t,T} := \sigma^{-2}\left(\left(\mathbf{F}^f_{t,T}\right)^T - \left(\mathbf{F}^f_{t,T}\right)^T \mathbf{E}^f_{t,T}\left(\left(\mathbf{E}^f_{t,T}\right)^T \mathbf{E}^f_{t,T}\right)^{-1}\left(\mathbf{E}^f_{t,T}\right)^T\right).$$

Since contributions of different features are independent of each other, for a selected subset of features $\Phi_t \subset \Theta_t$, one can verify that the updates to the information matrix and vector upon the choice of these features are given by

$$\mathbf{H}_{t,T}(\Phi_t) = \bar{\mathbf{H}}_{t,T} + \sum_{f\in\Phi_t} \mathbf{H}^f_{t,T}, \tag{21}$$

$$\mathbf{b}_{t,T}(\Phi_t) = \bar{\mathbf{b}}_{t,T} + \sum_{f\in\Phi_t} \left(\mathbf{B}^f_{t,T}\mathbf{z}^f_{t,T}\right)^T. \tag{22}$$

The corresponding mean vector and covariance matrix for the localization problem can be calculated through

$$\mathbf{\Sigma}_{t,T}(\Phi_t) = \mathbf{H}_{t,T}(\Phi_t)^{-1}, \tag{23}$$

$$\boldsymbol{\mu}_{t,T}(\Phi_t)^T = \mathbf{b}_{t,T}(\Phi_t)\,\mathbf{H}_{t,T}(\Phi_t)^{-1}. \tag{24}$$

In Fig. 1, we illustrate the essence of the motion and vision model described in this section. To specify the set of available features Θ_t for tracking, the robot needs to determine which features can be triangulated and which ones will result in invertible information matrices $(\mathbf{E}^f_{t,T})^T\mathbf{E}_f$ (cf. [12]).

Remark 1: Studying the applicability of the current approach could be translated to a more fundamental question: under what conditions, the formulations of extended Kalman filtering (EKF) is applicable and reliable? The answer to this question is beyond the scope of this paper, we refer to [16] for further discussions.

IV. Estimation Measures

Given a subset of trackable features Φ_t, we can quantify the quality of resulting estimation in various meaningful ways: *(i) Variance of the Error:* Given the covariance matrix, the variance of error equals the sum of the variances of all scalar components of vector $\mathbf{x}_{t,T}$. This measure can be characterized as

$$\rho_v(\mathbf{H}_{t,T}(\Phi_t)) := \mathrm{Tr}(\mathbf{H}(\Phi_t)^{-1}) = \mathrm{Tr}(\mathbf{\Sigma}(\Phi_t)). \tag{25}$$

(ii) Differential Entropy of the Estimation Error: It is known that the differential entropy of a multivariate Gaussian random variable with covariance $\mathbf{\Sigma}$ is

$$h = \frac{1}{2}\log(\det(\mathbf{\Sigma})) + \frac{n}{2}\big(1 + \log(2\pi)\big).$$

This measure quantifies the uncertainty volume of the estimation error, which is given by

$$\rho_e(\mathbf{H}_{t,T}(\Phi_t)) = \log(\det(\mathbf{\Sigma}(\Phi_t))) = -\log(\det(\mathbf{H}_{t,T}(\Phi_t))),$$

(iii) Spectral Variance: Let us consider the eigen-space of the largest eigenvalue of the covariance matrix of the estimator. This is the subspace across which the estimation is less accurate than the rest of the directions. Thus, we can use the following estimation measure

$$\rho_\lambda(\mathbf{H}_{t,T}(\Phi_t)) = \lambda_{\max}(\mathbf{\Sigma}(\Phi_t)) = \lambda_{\min}(\mathbf{H}_{t,T}(\Phi_t))^{-1}.$$

All these measures are monotonically decreasing. They are also spectral functions, i.e., they only depend on the eigenvalues of the information or covariance matrices. Therefore, having lower and upper bounds for the covariance matrix can be potentially useful to obtain similar bounds for these estimation measures. Measures *(ii)* and *(iii)* have been also discussed in [12].

V. Feature Selection via Randomized Sampling

We propose a scalable algorithm that provides feasible solutions for (5)-(6) with provable performance bounds.

A. Leverage Scores and Induced Probabilities

Each available feature $f \in \Theta_t$ is assigned some nonnegative numbers, which are so-called leverage scores, that are closely related to the notion of effective resistances in graph sparsification problem [17]. The maximal information matrix of $\mathbf{x}_{t,T}$, over the cone of positive-definite matrices, corresponds to the case where all features are employed in the estimation process. This matrix is given by

$$\mathbf{H}_{t,T}(\Theta_t) = \bar{\mathbf{H}}_{t,T} + \sum_{f \in \Theta_t} \mathbf{H}^f_{t,T}. \tag{26}$$

Therefore, for any $\Phi_t \subset \Theta_t$, it holds that

$$\mathbf{H}_{t,T}(\Phi_t) \preceq \mathbf{H}_{t,T}(\Theta_t). \tag{27}$$

Hence, for every monotone decreasing map $\rho : \mathcal{S}^{3(T+1)}_{++} \to \mathbb{R}$, it follows that

$$\rho(\mathbf{H}_{t,T}(\Theta_t)) \leq \rho(\mathbf{H}_{t,T}(\Phi_t)).$$

Using the maximal matrix, we define

$$\bar{\mathbf{H}}^f_{t,T} := \frac{1}{N_t} \bar{\mathbf{H}}_{t,T} + \mathbf{H}^f_{t,T}. \tag{28}$$

for every $f \in \Theta_t$ with $|\Theta_t| = N_t$.

Definition 2: For a given set of features Θ_t, the leverage scores are nonnegative numbers that are defined by

$$r_f := \operatorname{Tr}\left(\mathbf{H}_{t,T}(\Theta_t)^{-1} \bar{\mathbf{H}}^f_{t,T}\right) \tag{29}$$

for every feature $f \in \Theta_t$.

One can associate a probability mass function denoted by $\pi : \Theta_t \to [0, 1]$ to elements of Θ_t by setting

$$\pi(f) = \pi_f = \frac{r_f}{n}, \tag{30}$$

where $n = 3(T+1)$ is the dimension of $\bar{\mathbf{H}}_{t,T}$. The resulting function is a well-defined probability mass function since

$$\sum_{f \in \Theta_t} \pi(f) \; = \; \frac{1}{n} \sum_{f \in \Theta_t} \operatorname{Tr}\left(\mathbf{H}_{t,T}(\Theta_t)^{-1} \bar{\mathbf{H}}^f_{t,T}\right) = 1.$$

B. Sampling Algorithm

The steps of our method are given in Algorithm 1. First, we iteratively and independently sample a feature from Θ_t with replacement for q iterations. This sampling takes place according to probability mass function π, which is defined by (30). The sampled feature f is added to Φ_t provided that it has not been sampled before. At the end of the procedure, set of selected features Φ_t will have at most q elements.

Remark 2: Our approach is inspired by graph sparsification methods using effective resistances [17], [18]. A special form of our method in this paper appears in [19] for selecting rank-one matrices when the constant term $\bar{\mathbf{H}}_{t,T}$ is zero.

Remark 3: Our randomized feature selection algorithm does not depend on robot motion; see Section III-A for more details. For instance, our feature selection approach can be used instead of the feature selection routines in the inertial-visual navigation setup described in [12] that are based on the greedy method and convex relaxations.

Algorithm 1 Randomized Feature Selection

```
input: initial information matrix $\bar{\mathbf{H}}_{t,T}$
       set of available features $\Theta_t$, number of samples $q$
output: selected features $\Phi_t$, information matrix $\mathbf{H}_{t,T}$
initialize: $\Phi_t = \varnothing$, $\mathbf{H}_{t,T} = \bar{\mathbf{H}}_{t,T}$
  for $k = 1$ to $q$ do
      sample a feature from $\Theta_t$ using distribution $\pi \to f$
      select the corresponding matrix
          $\mathbf{H} \leftarrow \mathbf{H}^f_{t,T}$
      if $f \notin \Phi_t$, then
          add $f$ to $\Phi_t$
          update the information matrix:
              $\mathbf{H}_{t,T} \leftarrow \mathbf{H}_{t,T} + \mathbf{H}$
      end if
  end for
```

C. Performance Guarantee

Algorithm 1 provides us with an information matrix that is a constant-factor approximation to the maximal information matrix $\mathbf{H}_{t,T}(\Theta_t)$ given by (26).

Theorem 1: For a given parameter $\epsilon \in (0, 1)$, suppose that Algorithm 1 is executed with a fixed $q = O(n \log n/\epsilon^2) < N_t$. Then, the resulting information matrix, see (21), based on the resulting set of features Φ_t, satisfies

$$\mathbf{H}_{t,T}(\Phi_t) \succeq \frac{1-\epsilon}{4\bar{\chi}} \mathbf{H}_{t,T}(\Theta_t). \tag{31}$$

with probability at least $1/4$ for a number $\bar{\chi}$.

The proof of this theorem and definition of $\bar{\chi}$ is rather involved and inspired by [17]. The spectral bound (31) can be used to obtain performance bounds for the estimation measures.

Theorem 2: Under the settings of Theorem 1, the estimation quality losses compared to the case where all features in Θ_t are used satisfy

$$\frac{\rho_v(\mathbf{H}_{t,T}(\Theta_t)) - \rho_v(\mathbf{H}_{t,T}(\Phi_t))}{\rho_v(\mathbf{H}_{t,T}(\Phi_t))} \;\leq\; \frac{4\bar{\chi}}{1-\epsilon} - 1 \tag{32}$$

$$\rho_e(\mathbf{H}_{t,T}(\Theta_t)) - \rho_e(\mathbf{H}_{t,T}(\Phi_t)) \;\leq\; n \log\left(\frac{4\bar{\chi}}{1-\epsilon}\right) \tag{33}$$

$$\frac{\rho_\lambda(\mathbf{H}_{t,T}(\Theta_t)) - \rho_\lambda(\mathbf{H}_{t,T}(\Phi_t))}{\rho_\lambda(\mathbf{H}_{t,T}(\Phi_t))} \;\leq\; \frac{4\bar{\chi}}{1-\epsilon} - 1, \tag{34}$$

with probability at least $1/4$.

D. Implementation of Algorithm

The nature of the performance guarantees provided in Theorem 1 and 2 motivates us to run the algorithm with multiple random seeds, i.e., by conducting Monte-Carlo simulations. To this end, we choose a design that corresponds to the minimal value of the estimation measure of interest. We inspect that there are (at least) two steps during the feature selection process that are amenable to parallel implementation: (i) evaluation of sampling probabilities π_f for different features, and (ii) independent executions of Algorithm 1 to find different designs.

E. Time-Complexity Analysis

To find the sampling probabilities, we need $O(N_t T^3)$ operations, where $N_t = |F_t|$ is the number of available features at time t. One execution of Algorithm 1 requires $O(qT^2)$ operations. Evaluation of any of these estimation measures requires $O(T^3)$ operations. Therefore, if we run p independent samples of this algorithm, we will need $O(pqT^2 + pT^3)$ operations. Hence, the overall feature selection will require $O(N_t T^3 + pqT^2 + pT^3)$ operations.

For comparison purposes, we also analyze the time complexity of feature selection using the greedy method of [12]. For this method, iteratively, we should examine all candidates and find the feature whose addition will enhance the estimation quality more than the remaining features. This method requires $O(qN_t T^3)$ operations. In the worst-case, $q = O(N_t)$. Thus, in the worst case, its time complexity is $O(N_t^2 T^3)$, i.e., quadratic in the number of available features. This suggests that the random sampling using the leverage scores can potentially be faster than the greedy method (see next section for a numerical example).

Remark 4: The runtime of our algorithm is not probabilistic. Instead, it has a performance with a probabilistic nature. In practice, it implies that given a fixed computational budget (i.e., running the random sampling by a constant number of times), each sample outcome of our algorithm is highly likely to have a different outcome (i.e., performance). The performance guarantee of our approach describes the distribution of this outcome.

VI. Simulation Results

We explain the details of a numerical experiment, which is conducted to demonstrate the effectiveness of Algorithm 1.

A. Model and Environment Description

We consider a robot that is translating and rotating. Let us denote its position vector by $x_\tau^T = [p_\tau, y_\tau, z_\tau]$[2]. We suppose that the high-level dynamics of robot follow[3]

$$\begin{cases} p_{\tau+1} = p_\tau + u_\tau^p + \delta_\tau^p \\ y_{\tau+1} = y_\tau + u_\tau^y + \delta_\tau^y \\ z_{\tau+1} = z_\tau + u_\tau^z + \delta_\tau^z \end{cases}, \tag{35}$$

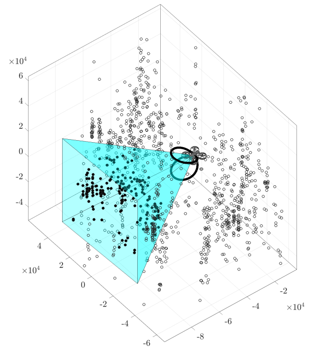

Fig. 2: A snapshot of the environment. The blue pyramid demonstrates the camera's field of view. The features that are inside the frame at this time are highlighted. The deformed 8*-shaped curve is the reference path as parametrized in* (36)*. The robot is also rotating according to* (38)*.*

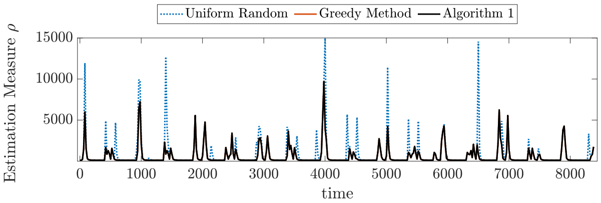

Fig. 3: The estimation measure values resulting from three methods. The curves corresponding to the to the greedy method and the proposed method may not be distinguished in this plot.

where $u_\tau^T := [u_\tau^p, u_\tau^y, u_\tau^z]$ is the input command signal and $\delta_\tau^T := [\delta_\tau^p, \delta_\tau^y, \delta_\tau^z]$ is the random process describing the uncertainty propagation due to the motion of the robot (compare to (7)). We consider the robot that is planning to move in the following reference path

$$\begin{cases} p_\tau^{\text{ref}} = p_0 + R\cos(\omega\tau) \\ y_\tau^{\text{ref}} = R\sin(\omega\tau) \\ z_\tau^{\text{ref}} = R\sin(\omega\tau/2) \end{cases}, \tag{36}$$

which looks like a deformed 3-D number 8 (see Fig. 2). To set the control inputs, we set

$$\begin{cases} u_\tau^p = p_{\tau+1}^{\text{ref}} - \bar{\mu}_\tau^p \\ u_\tau^y = y_{\tau+1}^{\text{ref}} - \bar{\mu}_\tau^y \\ u_\tau^z = z_{\tau+1}^{\text{ref}} - \bar{\mu}_\tau^z \end{cases}. \tag{37}$$

Moreover, we suppose that the Euler angles describing the absolute orientation of the camera at time τ are given by

$$\begin{cases} \alpha_\tau = 2\pi \sin(\omega_r \tau) \\ \beta_\tau = -\dfrac{\pi}{2} + \dfrac{\pi}{20}\sin(\omega_r \tau) \\ \gamma_\tau = 0 \end{cases}, \tag{38}$$

where the sequence of rotations is z-y-p. The visible landmarks in the environment consists of 1752 points in the space, which is constructed by by putting a circular array of randomly sampled points in a 3-D model of a room[4]. Two views from a snapshot of the environment have been

[2] We use letter p for the first coordinate to prevent conflict with use of position vector x.

[3] The implicit assumption here is that robot is already controlled by an internal feedback control mechanism and (35) represents the dynamics of the robot from reference (or tracking) signal to the state variables.

[4] The STL graphical file is adapted from `https://grabcad.com/library/room-blender-test-1`

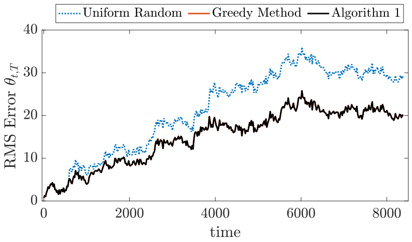

Fig. 4: The RMS error in the positions of the robot resulting from feature selection using different methods. The curces corresponding to the greedy method and Algorithm 1 may not be distinguished in this plot.

illustrated in Fig. 2. We set parameters $R = 7500$, $\omega = 0.08$, $\omega_r = 0.0064$, $\sigma = 0.1$ and $\Lambda_t = \mathrm{diag}(4, 4, 16)$ and initial covariance to be $\Sigma_0 = I$.

B. Different Approaches for Feature Selection

We consider the navigation setup for 400 time horizons each of length $T = 20$. The overall simulation consists of doing almost 107 full turns around the 3-D path of interest. For each horizon, after finding the eligible features to track (i.e., features with full rank information matrix $(\mathbf{E}^f_{t,T})^T\mathbf{E}^f_{t,T}$), we select at most half of the features. For each horizon, we do this task via three different methods:
(i) randomized sampling by leverage scores: we run Algorithm 1 for $p = 50$ independent experiments and choose the set Φ_t which induces the minimal value of the estimation measure. We denote the CPU time spent on this task by τ_t.
(ii) randomized sampling using uniform probabilities: we run Algorithm 1 for $p = 50$ independent experiments, except that instead of evaluating the probabilities, we assume an equal probability for each feature to be sampled. Similar to the previous case, we choose the design that induces minimal value of the estimation measure. We denote this design by Φ^u_t, where u stands for uniform probabilities. Similarly, we denote the corresponding CPU time for this task by τ^u_t.
(ii) greedy method: the features are added one-by-one, where at each iteration the feature which enhances the estimation quality the most is selected [12]. This method produces a single design for the feature selection denoted by Φ^g_t, where g stands for the greedy method.

In this example, we consider the estimation measure ρ_v as the monotone function governing the feature selection.

C. Metrics for Comparison of Methods

Many researchers have observed that the greedy method over-performs other approaches in several similar combinatorial problems [12], [20]. In general, the brute-force method in these settings is computationally infeasible[5], we select the greedy method as the base approach. Moreover, to have a clear understanding of the error in the position, we define the root mean squared error (RMSE) as we define

$$\theta_{t,T} := \frac{1}{3(T+1)}\sqrt{\sum_{\tau=t}^{t+T}\|x_\tau - \mu_t\|_2^2}, \tag{39}$$

for $t \in \{0, T, 2T, \dots\}$. We define similar error indices for the uniform random (method *(ii)*) and the greedy method (method *(iii)*) as well and denote them by $\theta^u_{t,T}$ and $\theta^g_{t,T}$, respectively. To compare the relative difference of these values, we use

$$\phi_{t,T} := \frac{\theta_{t,T} - \theta^g_{t,T}}{\theta^g_{t,T}} \times 100. \tag{40}$$

Similarly, we define $\phi^u_{t,T}$, which compares the value of RMSE resulting from the totally random choice of features with greedy selection. Finally, to compare the CPU times, we look at

$$\kappa_t := \frac{\tau_t}{\tau^g_t}, \tag{41}$$

which represent the ratio of the CPU time spent in methods *(i)* to the one spent by the greedy method. Similarly, we use κ^u_t to compare the time spent by method *(ii)* with method *(iii)*.

D. Numerical Results

In Fig 3, we show the resulting values of the estimation measure versus time, which demonstrate that the estimation measure resulting from Algorithm 1 is almost identical to the estimation measure resulting from the greedy method. This is not the case for the totally random choice of features, in which larger spikes can be observed.

In Fig. 4, we illustrate the values of the RMS error versus times for these methods. The errors corresponding to the feature selection using Algorithm 1 is very close to those values for the greedy method, while the totally random method may result in larger errors. This shows that in this example, not only the choice of features is a non-trivial computational task, but also our algorithm functions with quality and reliability that are very close to these factors in the case of the greedy method. The metric $\phi_{t,T}$ is also illustrated in Fig. 5, which quantifies these deviations.

Finally, we compare the CPU times spent on each method. In Fig. 6, we demonstrate the experimental CDF's of the parameters κ_t and κ^u_t, which show that the randomized methods are considerably faster than the greedy method in most cases. For instance, these data suggest that Algorithm 1 has been more than 20 times faster than the greedy method in about 85% of the assigned tasks, while in most cases it has been least 10 times faster. Note that the times include running the random sampling algorithms for 50 independent experiments.

[5] For instance, for a choice of 25 features out of 50 candidates we have to examine more than 10^{14} possible combinations. In fact, finding the brute-force solution becomes rapidly computationally prohibitive as the size of the candidate set grows.

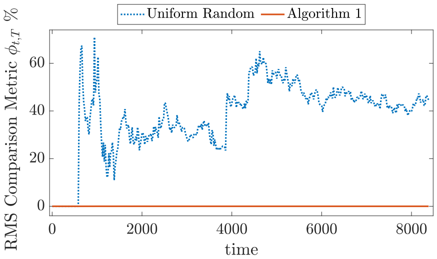

Fig. 5: Comparison of the RMS error resulting from the uniform random method and our proposed approach with the greedy method. This plot shows Algorithm 1 and the greedy method have almost identical performances for this example.

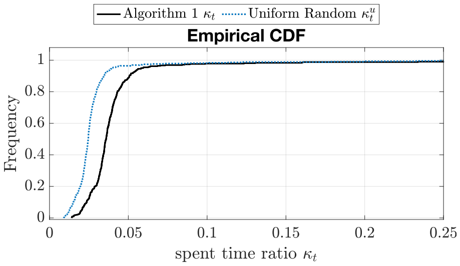

Fig. 6: The empirical CDF's for parameters κ_t and κ_t^u, which show the ratio of the spent CPU time for the random sampling (including all independent 50 experiments) to the greedy method. The plot show that in this example the random sampling is faster than the greedy method by almost an order of magnitude.

VII. Discussion and Conclusion

We propose a randomized algorithm for visual feature selection over a fixed-length moving time-horizon. The idea is to associate a sampling probability to each candidate feature, randomly sample a subset of features according to these probabilities for many independent experiments, and select the outcome with the best estimation quality. The most important property of our algorithm is that its time complexity scales linearly with the number of features, which makes it suitable for applications with hundreds or thousands of features.

If the estimation measure enjoys submodularity, then the greedy method provides a performance guarantee compared to the optimal solution [12], [21]. However, it is known that certain measures, for instance, ρ_v, are not submodular [22]. Moreover, in our work, we offer a different type of performance guarantee. Theorem 2 compares the estimation quality to the case that we leverage all features for tracking. Nevertheless, our extensive numerical simulations assert that the resulting estimation quality from our algorithm and that of the greedy method are often close to each other(e.g., see Fig. 3). Further research is required to uncover the practical and theoretical differences of this randomized algorithm and the greedy method.

According to the time complexity analysis in Section V-E, if the number of selected features is small, the computational cost of our randomized method will be comparable to those of greedy methods. However, in the worst case, the greedy-method scales quadratically with the number of candidate features (i.e., scaling with $N_t^2 = |\Theta_t|^2$). This justifies the significant speed-up in the feature-selection that is observed in our numerical simulations (see Fig. 6). The low time complexity of our method opens up new opportunities for real-time implementation of this algorithm and utilizing it for agile robot navigation.

References

[1] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield *et al.*, "The grand challenges of science robotics," *Science Robotics*, vol. 3, no. 14, p. eaar7650, 2018.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[3] R. Sim and G. Dudek, "Learning and evaluating visual features for pose estimation," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1217–1222.

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

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

[6] S. Chen, Y. F. Li, W. Wang, and J. Zhang, *Active sensor planning for multiview vision tasks*. Springer, 2008, vol. 1.

[7] H. Strasdat, C. Stachniss, and W. Burgard, "Which landmark is useful? learning selection policies for navigation in unknown environments," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 1410–1415.

[8] A. Gorbenko and V. Popov, "The problem of selection of a minimal set of visual landmarks," *Applied Mathematical Sciences*, vol. 6, no. 95, pp. 4729–4732, 2012.

[9] M. Beinhofer, J. Müller, A. Krause, and W. Burgard, "Robust landmark selection for mobile robot navigation." in *IROS*, 2013, pp. 3637–2643.

[10] 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 Transactions on Robotics*, vol. 33, no. 1, pp. 124–140, 2017.

[11] A. J. Davison, "Active search for real-time vision," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 66–73.

[12] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Transactions on Robotics DOI 10.1109/TRO.2018.2872402*, 2018.

[13] H. K. Mousavi and N. Motee, "Estimation with fast landmark selection in robot visual navigation," *arXiv preprint arXiv:1902.01026*, 2019.

[14] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[15] J. Van Den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[16] B. F. La Scala, R. R. Bitmead, and M. R. James, "Conditions for stability of the extended kalman filter and their application to the frequency tracking problem," *Mathematics of Control, Signals and Systems*, vol. 8, no. 1, pp. 1–26, 1995.

[17] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011.

[18] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng, "Spectral sparsification of graphs: theory and algorithms," *Communications of the ACM*, vol. 56, no. 8, pp. 87–94, 2013.

[19] H. K. Mousavi, Q. Sun, and N. Motee, "Space-time sampling for network observability," *arXiv preprint arXiv:1811.01303*, 2018.

[20] F. Lin, M. Fardad, and M. R. Jovanović, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1789–1802, 2014.

[21] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Sensor placement for optimal kalman filtering: Fundamental limits, submodularity, and algorithms," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 191–196.

[22] A. Olshevsky, "On (non) supermodularity of average control energy," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1177–1181, 2018.